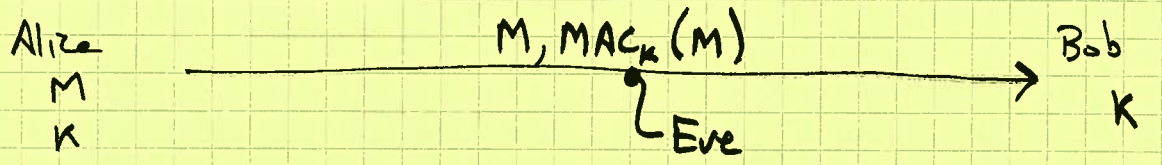
Administrivia: PS#3 out later today
Thanks to Eran!

Outline:
- ☐ SHA-3 conference review
- ☐ MAC's: review def
  review CBC-MAC
  one-time MAC (problem stmt)
- ☐ Finite fields review
- ☐ One-time MAC solution

SHA-3 conference review:

- 3 days
- 36 candidate presentations (5 missing); 18 minutes each
- 3 NIST sessions, 1 overview talks session, 1 rump session
- AES: contest similar
  many AES-based or inspired designs
  announcement of AES instructions on Intel CPU's
  discussion of side-channel attacks - are they important?
- strong emphasis on speed; NIST wants SHA-3 speed to be at least as good as SHA-2 (≅ 20 cpB)
- round 2 candidates to be announced ≈ August '09 (15 or so candidates)
- thanks for PS#1 evaluations! (put on class web site??)
  no/anon/straight

MAC (Message Authentication code)



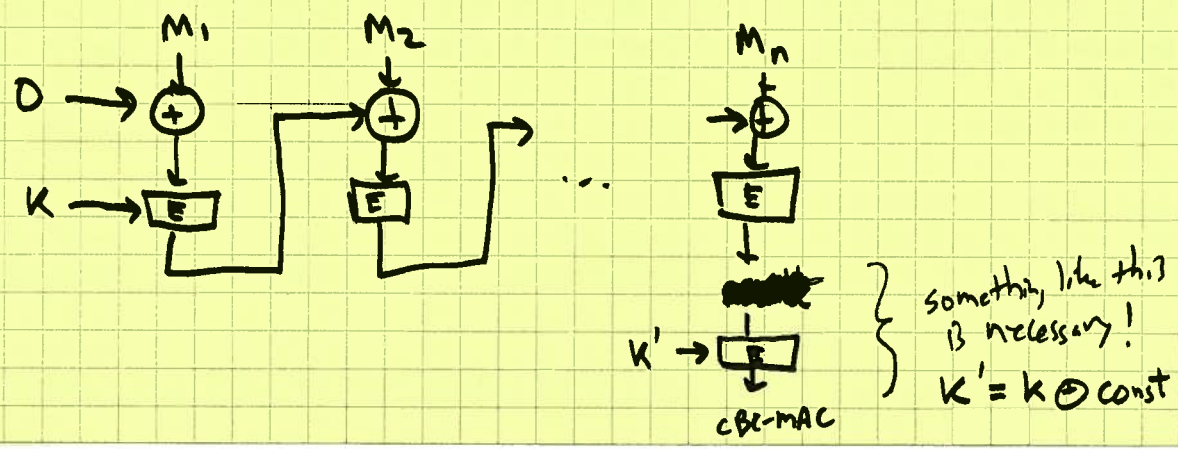
Note: M could be ciphertext, e.g. $M = E_K(P)$ for "plaintext" P.

- Bob recomputes $MAC_K(M)$ & verifies correctness before accepting M as authentic
- Eve wants to forge $M', MAC_K(M')$ without knowing K
  - she may hear a number of $(M, MAC_K(M))$ pairs first possibly even with M's other choice (chosen msg attacks)
  - she wants to forge for M' for which she hasn't seen $(M', MAC_K(M))$. (possibly modification of one...)
- HMAC: $HMAC(K,M) = h(K_1 \| h(K_2 \| M))$

  $K_1 = K \oplus opad$ ipad, opad fixed constants

  $K_2 = K \oplus ipad$
- CBC-MAC:

  CBC-MAC(K,M) = last block of CBC encryption of M



One-time MAC (problem stmt)

- like OTP, except for authentication (integrity) instead of confidentiality
- Alice & Bob share secret key K key K is "use-once"

(note: M may be plaintext or could be ciphertext)

- Eve can learn M, T, then try to replace M, T with M', T' that Bob accepts
- Eve is computationally unbounded: want unconditional security

		Confidentiality	Authentication
Unconditional		OTP ✓	One-time MAC
Computational crypto	Conventional	block ciphers AES	MAC (HMAC)
	Public-key	PK enc.	Dig sig

Finite fields

system $(S, +, \cdot)$ s.t.

- S is a finite set containing "0" & "1"
- $(S, +)$ is abelian (commutative) group with identity 0

group: $\begin{cases} (a+b)+c = a+(b+c) & \text{associative} \\ a+0 = 0+a = a & \text{identity 0} \\ (\forall a)(\exists b)\ a+b = 0 & \text{(additive) inverses} \quad (b = -a) \end{cases}$

$a+b = b+a$ commutative

- $(S^*, \cdot)$ is an abelian group with identity 1

S^* = nonzero elements of S

group: $\begin{cases} (a\cdot b)\cdot c = a\cdot(b\cdot c) & \text{associative} \\ a\cdot 1 = 1\cdot a = a & \text{identity 1} \\ (\forall a \in S^*)(\exists b \in S^*)\ a\cdot b = 1 & \text{(multiplicative inverses)} \quad b = a^{-1} \end{cases}$

$a\cdot b = b\cdot a$ commutative

- Distributive laws hold: $a\cdot(b+c) = a\cdot b + a\cdot c$

$(b+c)\cdot a = b\cdot a + c\cdot a$ (follows)

Familiar fields: $\mathbb{R}$ (reals) are infinite
$\mathbb{C}$ (complex)

For crypto, we're interested in finite fields (Z_p = integers mod prime p)

Over field, usual algorithms work (mostly):

e.g. solving linear eqns:

$$ax + b = 0 \pmod p$$

$$\Rightarrow \quad x = a^{-1}\cdot(-b) \pmod p \text{ is soln.}$$

$$3x + 5 = 6 \pmod 7$$

$$3x = 1 \pmod 7$$

$$x = 5 \pmod 7$$

- Notation: $GF(q)$ is finite field ("Galois field") with q elements
- **Theorem:** There is a finite field $GF(q)$ whenever

$$q = p^k \quad , \; p \text{ prime}, \; k \geq 1$$

Two cases $GF(p)$: work modulo p

Z_p integers mod p $= \{0, 1, \ldots, p-1\}$

$Z_p^* = \{1, 2, \ldots, p-1\}$

usual mod p arithmetic

$GF(p^k)$: $k > 1$

work with polynomials of degree $< k$ whose coefficients are elements of $GF(p)$

(modulo fixed irreducible polynomial of degree k)

Common case $GF(2^k)$

All operations can be performed efficiently. (inverses?)

- "Repeated squaring" to compute a^b in field: (b integer ≥ 0)

$$a^b = \begin{cases} 1 & \text{if } b = 0 \\ (a^{b/2})^2 & \text{if } b \text{ even}, b > 0 \\ a \cdot a^{b-1} & \text{if } b \text{ odd} \end{cases}$$

requires $\leq 2 \lg(b)$ multiplications in field **efficient**

$\approx$ a few milliseconds for $a^b \pmod p$ 1024-bit integers

$\approx \Theta(k^3)$ time for k-bit inputs

- **Computing inverses:** (Multiplicative; additive inverses are easy.)

Thm: In $GF(q)$ "Fermat's Little theorem" for $GF(p)$

$$(\forall a \in GF(q)^*) \quad a^{q-1} = 1$$

Cor: $(\forall a \in GF(q)^*) \quad a^{-1} = a^{q-2}$ ⇐ e.g. $3^{-1} \pmod 7$

$= 3^5 \pmod 7$

$= 5 \pmod 7$

Cor: $(\forall a \in GF(q)) \quad a^q = a$

- How to find large (k-bit) prime #?

  do p ← random k-bit integer
  until p is prime

  } "generate & test"

- Primes are "dense", so this works:
  - about $2^k / \ln(2^k)$ k-bit primes (Prime Number Theorem)
  - One out of every $k \cdot \ln(2) = 0.69\,k$ k-bit integers is prime.
- To test if a large randomly-chosen k-bit integer p is prime:

  suffices to test only:

  $$2^{p-1} \overset{?}{\equiv} 1 \pmod{p}$$ works w.h.p.

  - doesn't work for adversarially chosen p
  - see CLRS for Miller-Rabin primality test (randomized)
  - technically, you get "base-2 pseudoprime", but these are almost always prime.
  - There is deterministic poly-time primality test (Agrawal, Kayal, Saxena) (2002)

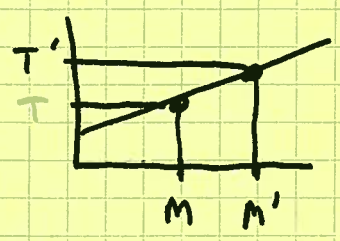
    test $(x-a)^p \equiv x^p - a \pmod{p}$ x variable tests as polynomials

    which is true iff p is prime

    test mod p & mod $x^r - 1$ for small r & small set of a.

Want MAC(M) to say nothing about MAC(M'), even for unbounded Eve.

Idea:



$T = ax + b \pmod p$

need two points to determine line

(M,T) is only one.

Assume: p large prime (e.g. $p = 2^{128} + 51$)

$0 \le M < p$

MAC key $K = (a,b)$ $0 \le a < p$ $0 \le b < p$ } p^2 possible keys (uniformly randomly chosen)

Use (a,b) to authenticate M (use-once!)

$$\boxed{T = aM + b \pmod p}$$

Security: Suppose adversary replaces (M,T) with (M',T') $M' \neq M$

then (if Bob accepts):

$$\left.\begin{aligned} aM + b &= T \pmod p \\ aM' + b &= T' \pmod p \end{aligned}\right\} (*)$$

$$a = (T - T')/(M - M') \pmod p$$

$$b = T - a \cdot M \pmod p$$

For any given value of $M' \neq M$:

For each choice of T', $\exists$ exactly one pair (a,b) consistent with (*). All equally likely

Knowing (M,T) reduces # possible keys from p^2 to just p

but each of the p possible keys gives different value for T'.

∴ adversary has no information on $MAC_K(M')$

Information-theoretically secure.

[true even if eve chose M.

true even if eve saw other msgs (M'',T'') before (different keys)]