Admin:

Pset #3 due today
Project team & multi-psyce write-up due Friday

Today:

- Message Authentication Codes (MAC's)
- HMAC, CBC-MAC, PRF-MAC
- One-time MAC (problem stmt)
- Finite fields review
- One-time MAC (soln)
MAC (Message Authentication Code)

- Not confidentiality, but integrity (recall “CIA”)
- Alice wants to send messages to Bob, such that Bob can verify that messages originated with Alice & arrive unmodified.
- Alice & Bob share a secret key $K$
- Orthogonal to confidentiality; typically do both (e.g. encrypt, then append MAC for integrity)
- Need additional methods (e.g. counters) to protect against replay attacks

Alice $\xrightarrow{M, MAC_k(M)}$ Bob $^K$

[Here $M$ is message to be authenticated, which could be ciphertext resulting from encryption.]

- Alice computes $MAC_k(M)$ & appends it to $M$.
- Bob recomputes $MAC_k(M)$ & verifies it agrees with what is received. If $\neq$, reject message.
Adversary (Eve) wants to forge $M', \text{MAC}_k(M')$ pair that Bob accepts, without Eve knowing $K$.

- She may hear a number of valid $(M, \text{MAC}_k(M))$ pairs first, possibly even with $M'$s of her choice (chosen msg attacks).
- She wants to forge for $M'$ for which she hasn't seen $(M', \text{MAC}_k(M'))$ valid pair.

**Two common methods:**

$$\text{HMAC}(K, M) = h(k_1 || h(k_2 || M))$$

where $k_1 = K \oplus \text{opad}$ \{opad, ipad are fixed constants\}

$k_2 = K \oplus \text{ipad}$

$$\text{CBC-MAC}(K, M) \equiv \text{last block of CBC enc. of } M$$

Something like this is necessary...
**MAC using random oracle (PRF):**

\[ MAC_k(M) = h(K || M) \]

(OK if \( h \) is indistinguishable from RO, which means, as we saw, for sequential hash fns, that last block may need special treatment.)

**One-Time MAC (problem stmt):**

Can we achieve security against unbounded Eve, as we did for confidentiality with OTP, except here for integrity?

Here key \( K \) may be "use-once" [as it was for OTP].

\[ A \xrightarrow{K} B \]

\[ T = MAC_k(M) \] ("tag")

- Eve can learn \( M, T \) then try to replace \( M, T \) with \( M', T' \) (where \( M' \neq M \)) that Bob accepts.
- Eve is computationally unbounded.
<table>
<thead>
<tr>
<th></th>
<th>Confidentiality</th>
<th>Integrity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unconditional</td>
<td>OTP ✓</td>
<td>One-time MAC?</td>
</tr>
<tr>
<td>Conventional</td>
<td>Block ciphers (AES) ✓</td>
<td>MAC (HMAC) ✓</td>
</tr>
<tr>
<td>(symmetric key)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Public-key</td>
<td>PK enc.</td>
<td>Digital signature</td>
</tr>
<tr>
<td>(asymmetric)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Finite fields:

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

• $S$ is a finite set containing "0" & "1"

• $(S, +)$ is an abelian (commutative) group with identity 0

\[
\begin{align*}
(a + b) + c &= (a + (b + c)) & \text{associative} \\
0 + a &= a & \text{identity 0} \\
(\forall a \exists b) a + b &= 0 & \text{(additive) inverses } b = -a \\
q + b &= b + q & \text{commutative}
\end{align*}
\]

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

\[
\begin{align*}
S^* &= \text{nonzero elements of } S \\
(a \cdot b) \cdot c &= a \cdot (b \cdot c) & \text{associative} \\
1 \cdot a &= a & \text{identity 1} \\
(\forall a \in S^*)(\exists b \in S^*) a \cdot b &= 1 & \text{(multiplicative inverses)} b = a^{-1} \\
a \cdot b &= b \cdot a & \text{commutative}
\end{align*}
\]

• Distributive laws:

\[
\begin{align*}
a \cdot (b + c) &= a \cdot b + a \cdot c \\
(b + c) \cdot a &= b \cdot a + c \cdot a & \text{(follows)}
\end{align*}
\]

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

For crypto, we're usually interested in finite fields, such as $\mathbb{Z}_p$ (integers mod prime $p$).
Over a field, usual algorithms work (mostly).

E.g. solving linear eqns:

\[ ax + b = 0 \pmod{\rho} \]

\[ \Rightarrow x = a^{-1} \cdot (-b) \pmod{\rho} \] is soln.

\[ 3x + 5 = 6 \pmod{7} \]

\[ 3x = 1 \pmod{7} \]

\[ x = 5 \pmod{7} \]
Notation: \( \text{GF}(q) \) is the finite field ("Galois field") with \( q \) elements.

Theorem: \( \text{GF}(q) \) exists whenever \( q = p^k \), \( p \) prime, \( k > 1 \)

Two cases:

1. \( \text{GF}(p) \) - work modulo prime \( p \)
   \[ \mathbb{Z}_p = \text{integers mod } p = \{0, 1, \ldots, p-1\} \]
   \[ \mathbb{Z}_p^* = \mathbb{Z}_p - \{0\} = \{1, 2, \ldots, p-1\} \]

2. \( \text{GF}(p^k) \) : \( k > 1 \)
   work with polynomials of degree < \( k \) with coefficients from \( \text{GF}(p) \)
   modulo fixed irreducible polynomial of degree \( k \)

Common case is \( \text{GF}(2^k) \)

Note: all operations can be performed efficiently

(inverses to be demonstrated)
"Repeated squaring" to compute $a^b$ in field

(Here $b$ is a non-negative integer)

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

Requires $\leq 2 \cdot \lg(b)$ 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 (multiplicative) inverses:

**Theorem:** (For $GF(p)$ called "Fermat's Little Theorem")

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

**Corollary:** \((\forall a \in GF(q)) a^q = a\)

**Corollary:** \((\forall a \in GF(q^k)) a^{-1} = a^{q-2}\)

**Example:** \(3^{-1} \pmod{7}\)

\[ = 3^5 \pmod{7} \]

\[ = 5 \pmod{7} \]
- How to find large (k-bit) random prime #?

  **Generate & test:**
  
  \[
  \text{do } p \leftarrow \text{random k-bit integer} \\
  \text{until } p \text{ is prime}
  \]

- Works because primes are "dense":

  \[
  \frac{2^k}{\ln(2^k)} \quad \text{k-bit primes (Prime Number Theorem)}
  \]

  \[
  \Rightarrow \text{one of every } \approx 0.69k \quad \text{k-bit integers is prime.}
  \]

- To test if a large randomly-chosen k-bit integer is prime, it suffices to test

  \[
  2^{p-1} \equiv 1 \pmod{p}
  \]

  - This works with high probability (w.h.p.) for random \( p \);
  
    doesn't work for adversarially chosen \( p \).

- See CLRS for Miller-Rabin primality test (randomized)

- Technically, above gives "base-2 pseudoprime", but this

  is almost always prime

- \exists deterministic poly-time primality test (Agrawal, Kayal, Saxena 2002):

  \[
  \text{Test } (x-a)^p = x^p - a \pmod{p} \quad x \text{ variable}
  \]

  which is true iff \( p \) is prime

  Test \( \mod{p} \) \& \( \mod{x^r-1} \) for small \( r \) \& small \( a \)'s.
One-time MAC (soln): 

Idea:

\[ T' = T + (T - T') \]

\[ k = (a, b) \]

\( p \) public

\( K \) is use-once

\[ T = \text{MAC}_k(M) = ax + b \pmod{p} \]

\[ [x=M] \] \( \ast \)

Need two points to determine line; Eve hears just one: \((M, T)\)

\( p \) large prime (e.g. \( 2^{128} + 51 \))

Key \( K = (a, b) \) \( 0 \leq a < p, 0 \leq b < p \) \( (p^2 \text{ keys}) \)

Security:

If adversary hears \((M, T)\) on the line, and replaces it with \((M', T')\) \([M' \neq M] \), then Bob accepts with probability \( \frac{1}{p} \).

PF: Hearing \((M, T)\) reduces set of possible keys to those satisfying \( \ast \). Nonetheless, for each possible \( T' \), there is an \((a, b)\) satisfying both \( \ast \) and

\[ T = aM' + b \pmod{p} \] \( \ast \ast \)

all such keys are equally likely; Eve has no way to pick correct \( T' \).
Details:

For fixed $M, M' [M \neq M']$, fixed $T$ s.t.

$$a \cdot M + b = T \quad (\text{mod } p) \quad (\ast)$$

For each $T'$, $\exists$ exactly one key $(a, b)$ s.t. $(\ast)$ and

$$a \cdot M' + b = T' \quad (\text{mod } p) \quad (\ast\ast)$$

holds:

$$a = (T - T')/(M - M') \quad (\text{mod } p)$$

$$b = T - a \cdot M \quad (\text{mod } p)$$

Thus Eve gains no information on $T' = MAC_e(M')$

by hearing $(M, T)$. Method is information-theoretically

secure.

- True even if Eve can control $M$.

- Note that key $K$ is twice as large as message $M$. 