Admin:
- Problem Set #2 due
- Problem Set #3 out

Project Ideas:
- Where did Mt. Gox bitcoins go?
- Attack reduced round “Simon” or “Speck” with SAT solver?

Today:
- Message Authentication Codes
  - HMAC
  - CBC-MAC
  - PRF-MAC
  - One-time MAC
- Combined mode
  - AEAD (Authenticated encryption with associated data)
  - EAX mode (ref. pages 1-10 of paper only)
- Finite fields and number theory
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 \[ \rightarrow \] \[ M, \text{MAC}_K(M) \] \[ \rightarrow \] Bob

Note: if MAC has $t$ bits, then Adv can forge with prob $2^{-t}$ just by guessing, $t=32$ might be ok in practice...

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

- Alice computes $\text{MAC}_K(M)$ & appends it to $M$.
- Bob recomputes $\text{MAC}_K(M)$ & verifies it agrees which 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 \( k_1 = k \oplus \text{opad} \) \( \text{opad, ipad are fixed constants} \)

\[
\text{CBC-MAC}(k, M) \equiv \text{last block of CBC enc. of } M
\]

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

\[ \text{MAC}_K(M) = h(K \| M) \]

(OK if \( h \) is indistinguishable from RO, which means, as we saw, for sequential hash funs, 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].

\[ A \xrightarrow[K]{M,T} B \]

\[ T = \text{MAC}_K(M) \text{ ("tag")} \]

- 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>
EAX mode

See pgs 1-10 of
The EAX Mode of Operation
by Bellare, Rogaway, & Wagner
Finite fields:

System $(S, +, \cdot)$ s.t.
- $S$ is a finite set containing "0" & "1"
- $(S, +)$ is an abelian (commutative) group with
  identity $0$
  
  \[
  \begin{align*}
  \forall a, b, c \in S : (a + b) + c &= a + (b + c) & \textit{associative} \\
  a + 0 &= 0 + a = a & \textit{identity 0} \\
  (\exists b) : a + b = 0 & \textit{additive inverses} \ b = -a \\
  a + b &= b + a & \textit{commutative}
  \end{align*}
  \]

- $(S^*, \cdot)$ is an abelian group with identity 1
  
  $S^* = \text{nonzero elements of } S$

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

- Distributive laws:
  
  \[
  \begin{align*}
  a \cdot (b + c) &= a \cdot b + a \cdot c & \textit{(follows)} \\
  (b + c) \cdot a &= b \cdot a + c \cdot a & \textit{(follows)}
  \end{align*}
  \]

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 field, usual algorithms work (mostly).

E.g., solving linear eqns:

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

\[ \Rightarrow x = a^{-1} \cdot (-b) \pmod{p} \]

is soln.

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

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

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

Theorem: \( GF(q) \) exists whenever
\[
q = p^k, \quad p \text{ prime, } k \geq 1
\]

Two cases:
1. \( GF(p) \) - work modulo prime \( p \)
   \[
   \mathbb{Z}_p = \text{integers mod } p = \{0, 1, \ldots, p-1\}
   \]
   \[
   \mathbb{Z}_p^* = \mathbb{Z}_p - \{0\} = \{1, 2, \ldots, p-2\}
   \]
2. \( GF(p^k) : k > 1 \)
   work with polynomials of degree < \( k \)
   with coefficients from \( GF(p) \)
   modulo fixed irreducible polynomial of degree \( k \)

Common case is \( GF(2^k) \)

Note: all operations can be performed efficiently
(inverses to be demonstrated)
Construction of $GF(2^3) = GF(8)$

Has 4 elements.

Is not arithmetic mod 4, (where $2$ has no mult inverse)

elements are polynomials of degree 2 with coefficients
mod 2 (i.e. in $GF(2)$):

$$0, 1, x, x+1$$

Addition is component-wise according to powers, as usual

$$(x) + (x+1) = (2x+1) = 1 \quad \text{(coeff. mod 2)}$$

Multiplication is modulo $x^2 + x + 1$ which is irreducible (doesn’t factor)

$$\begin{array}{cccc}
0 & 0 & 1 & x & x+1 \\
0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & x & x+1 \\
x & 0 & x & x+1 & 1 \\
x+1 & 0 & x+1 & 1 & x
\end{array}$$

$x^2 \text{ mod } (x^2 + x + 1)$ is $x+1$ (note that $x \equiv -x$ coeff mod 2)
"Repeated squaring" to compute $a^b$ in field

Here $b$ is a non-negative integer

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

Requires $\leq 2 \cdot \log(b)$ multiplications in field (efficient)

$\propto$ a few milliseconds for $a^b \pmod{p}$ 1024-bit integers

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

Computing (multiplicative) inverses:

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

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

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

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

**Example:**

$3^{-2} \pmod{7}$

$= 3^5 \pmod{7}$

$= 5 \pmod{7}$