TOPIC: 6.857 - Computer & Network Security

DATE: 3/12/14

FILE UNDER:

PAGE: L11.1

Admin:

pset #1 — back
#2 — in (see passwords)
#3 — out, due Mon 3/24 (late OK) (groups!)

project ideas/writeup 3/21 ⇐ next Friday

Project idea:

- evaluate encrypted email options (esp. ease-of use)
- overview of open-source crypto libraries
- Note: final projects to be posted online! (unless security vulnerability disclosed...)

Today:

More "crypto math":

- finding large primes
- one-time mac
- divisors, gcd, extended gcd, mult. inverses
- orders of elements
- finding generators

TOPIC:

DATE:

FILE UNDER:

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

  Generate & test: do $p \leftarrow$ random k-bit integer
  until p is prime

- Works because primes are "dense":

  about $2^k / \ln(2^k)$ k-bit primes (Prime Number Theorem)

  $\Rightarrow$ one of every $\approx 0.69k$ k-bit integers is prime.

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

$$2^{p-1} \stackrel{?}{=} 1 \pmod{p}$$

- 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):

  Test $(x-a)^p = x^p - a \pmod{p}$ x variable

  which is true iff p is prime

  Test mod p & mod $x^r - 1$ for small r & small a's.

  storage requirements ?! (see handout)

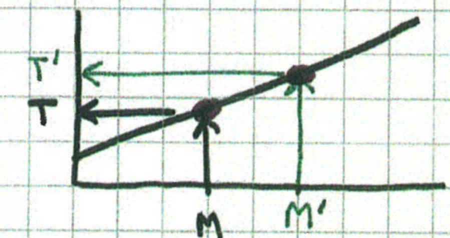
TOPIC:

DATE:

FILE UNDER:

One-time MAC (soln):

Idea:



$K = (a,b)$
p public
K is use-once

$$\boxed{T = MAC_K(M) = ax + b \pmod p} \qquad [x = M] \quad (*)$$

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

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

Key $K = (a,b)$ $\quad 0 \le a < p,\ 0 \le b < p \quad$ (p^2 keys)

Security:

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

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

$$T = aM' + b \pmod p \qquad (**)$$

all such keys are equally likely; Eve has no way to pick correct T'.

TOPIC:

DATE:

FILE UNDER:

Details:

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

$$aM + b = T \pmod{p} \qquad (*)$$

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

$$aM' + b = T' \pmod{p} \qquad (**)$$

holds:

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

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

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

TOPIC:

DATE:

FILE UNDER:

Divisors

- $d \mid a \equiv$ "d divides a" (evenly)

  $\equiv (\exists k)\ a = d \cdot k$
- d is a divisor of a if $d \geq 0$ & $d \mid a$
- $(\forall d)\ d \mid 0$
- $(\forall a)\ 1 \mid a$
- If d is a divisor of a & a divisor of b, then d is a common divisor of a & b.
- The greatest common divisor of a & b is the largest of their common divisors.

  [But $\gcd(0,0) = 0$ by definition.]
- Examples: $\gcd(24,30) = 6$

  $\gcd(5,0) = 5$

  $\gcd(33,12) = 3$
- Def: a & b are relatively prime if $\gcd(a,b) = 1$

TOPIC:

DATE:

FILE UNDER:

- **Euclid's algorithm for computing gcd(a,b) [a,b ≥ 0]:**

$$\gcd(a,b) = \begin{cases} a & \text{if } b=0 \\ \gcd(b, a \bmod b) & \text{else} \end{cases}$$

- **Example:** $\gcd(7,5)$
  $= \gcd(5,2)$
  $= \gcd(2,1)$
  $= \gcd(1,0)$
  $= 1$

- Running time is $\approx \lg(a) \cdot \lg(b)$ bit operations
  (Polynomial running time, like multiplying.)

TOPIC:

DATE:

FILE UNDER:

PAGE: L13.4

Theorem $(\forall a,b)(\exists x,y)\ ax+by=\gcd(a,b)$

Proof "by example" $a=7,\ b=5$

$$\left.\begin{aligned} 7 &= 7\cdot 1 + 5\cdot 0 \\ 5 &= 7\cdot 0 + 5\cdot 1 \end{aligned}\right\} \text{initial values}$$

$$2 = 7\cdot 1 + 5\cdot(-1) \quad [\text{subtract 2 eqns}]$$

$$1 = 7\cdot(-2) + 5\cdot 3$$

$$= a\,x + b\,y$$

This is the "extended version of Euclid's algorithm".

Computing modular multiplicative inverses with Euclid's extended alg:

Suppose $a \in Z_p^*$ (so $1 \le a < p$ & $\gcd(a,p)=1$, p prime(?))

How to compute $a^{-1} \pmod p$?

If p prime: $a^{-1} = a^{p-2} \pmod p$

Otherwise:

Find x,y s.t. $ax+py=1$

so $ax = 1 \pmod p$

and $x = a^{-1} \pmod p$

Example: $5^{-1} = 3 \pmod 7$

TOPIC:

DATE:

FILE UNDER:

Order of elements (in Z_p^* or Z_n^*):

Define: $\text{order}_n(a)$ = "order of a, modulo n"

$= \text{least } t>0 \text{ s.t. } a^t = 1 \pmod{n}$

Recall Fermat's Little Theorem:

If p prime, then $(\forall a \in Z_p^*)\ a^{p-1} = 1 \pmod{p}$

For general n, we have Euler's Theorem:

$$(\forall n)(\forall a \in Z_n^*)\ a^{\varphi(n)} = 1 \pmod{n}$$

where $Z_n^* = \{a : \gcd(a,n) = 1\}$

= multiplicative group modulo n

$$\varphi(n) = |Z_n^*|$$

Example: $Z_{10}^* = \{1, 3, 7, 9\}$

$\varphi(10) = 4$

$3^4 = 1 \pmod{10}$

Thus $\varphi(n)$ is well-defined for all n, &

$\text{order}_n(a)$ is also well-defined.

Can we say more?

Example: mod $p = 7$

	1	2	3	4	5	6	7 …	
1	1	1	1	1	1	1	1 …	order(1) = 1
2	2	4	1	2	4	1	2 …	order(2) = 3
3	3	2	6	4	5	1	3 …	order(3) = 6
4	4	2	1	4	2	1	4 …	order(4) = 3
5	5	4	6	2	3	1	5 …	order(5) = 6
6	6	1	6	1	6	1	6 …	order(6) = 2

Fermat

Def: $\langle a \rangle = \{a^i : i \geq 0\}$ = subgroup generated by a

Example: $\langle 2 \rangle = \{2, 4, 1\}$ (in $\mathbb{Z}_7^*$)

Theorem: $\text{order}(a) = |\langle a \rangle|$

Theorem: If p prime: $\text{order}_p(a) \mid (p-1)$.

Theorem: $|\langle a \rangle| \;\Big|\; |\mathbb{Z}_n^*|$

or: $\text{order}_n(a) \mid \varphi(n)$ equivalently.

TOPIC:

FILE UNDER:

DATE

Generators

Def: If $\text{order}_p(g) = p-1$

then g is a generator of Z_p^*.

(i.e. $\langle g \rangle = Z_p^*$)

Theorem: If p is a prime and

g is a generator mod p, then

$$g^x = y \pmod{p}$$

has a unique solution x $(0 \le x < p-1)$

for each $y \in Z_p^*$.

Def: x is the "discrete logarithm"

of y, base g, modulo p.

x	=	1	2	3	4	5	6
g^x	=	3	2	6	4	5	1

for $g=3$, modulo 7.

Theorem: $\mathbb{Z}_n^*$ has a generator (i.e. $\mathbb{Z}_n^*$ is cyclic) iff n is

$$2,\ 4,\ p^m,\ \text{or } 2p^m$$

for some prime p & $m \geq 1$.
odd

Theorem: If p is prime, the number of generators mod p is $\varphi(p-1)$

Example: $p = 11$

$\mathbb{Z}_{11}^*$ has $\varphi(10) = 4$ generators (they are 2, 6, 7, and 8).

How to find a generator mod a prime p?

In general, seems to require knowledge of factorization of $p-1$.

While factoring is hard, we can create primes for which factoring $p-1$ is trivial.

TOPIC:

FILE UNDER:

DATE:

Def: If p & q are both primes &

$$p = 2q + 1$$

then p is a "safe prime" and

q is a "Sophie Germain prime".

Examples: $p = 23$, $q = 11$ $\quad p = 11$, $q = 5$

$p = 59$, $q = 29$ $\quad \ldots$

Theorem: If p is a safe prime

then $p - 1 = 2 \cdot q$

so $(\forall a \in \mathbb{Z}_p^*)$ $\text{order}_p(a) \in \{1, 2, q, 2q\}$.

It is not hard to find safe primes. ("Probability" that a prime p is safe is $\approx 1/\ln(p)$, empirically.)

Can test if g is a generator mod $p = 2q+1$ easily:

check that $g^{p-1} = 1 \pmod p$ $\quad$ ✓ by Fermat

& $g^2 \neq 1 \pmod p$ $\quad [\text{order}_p(g) \neq 2]$

& $g^q \neq 1 \pmod p$ $\quad [\text{order}_p(g) \neq q]$

then $\text{order}_p(g) = p - 1$.

TOPIC:

DATE

FILE UNDER:

We can use "generate & test" again: (for "safe prime" p)

do $g \xleftarrow{R} Z_p^*$

until $\text{order}_p(g) = p-1$

Generators are quite common:

Theorem: If $p = 2q+1$ is a "safe prime"

then # generators mod p

$$= \varphi(p-1)$$

$$= q-1 \quad \text{(almost half of them!)}$$

(In general:

Theorem: If p prime, then

generators mod p

$$= \varphi(p-1)$$

$$\geq \frac{p-1}{6 \ln\ln(p-1)}$$

)

So generate & test works well for finding generators modulo a safe prime p, or modulo any prime p for which you know $\varphi(p-1)$.

TOPIC:

FILE UNDER:

DATE:

- **Common public-key setup:**

  Public system parameters

p	large prime (e.g. 1024 bits)
g	generator mod p

  Alice choose x $0 \leq x < p-1$ as her secret key.

  Alice publishes $y = g^x \pmod{p}$ as her public key.

- Secrecy of x protected by difficulty of computing discrete log

$$\log_{g,p}(y) = x$$

- Commonly assumed that discrete log problem (DLP) is infeasible for p large & random, or p large safe prime.

  (Appears to be roughly as hard as factoring a large integer of the same size as p. This is observation, not a theorem.)