Today:

- Quick review of finite fields
- Shamir Secret Sharing Scheme
- Groups

Recall:

Def: A field is defined by $(S, +, \cdot)$ s.t.

- $S$ is a set containing "0" & "1"
- $(S, +)$ is an abelian (commutative) group w. identity 0:

group laws
- $(a+b)+c = a+(b+c) \quad \forall a,b,c \in S$ (associative)
- $a+0 = 0+a = a \quad \forall a \in S$ (identity 0)
- $\forall a \in S \ \exists b \in S$ s.t. $a+b = 0$ (inverse)
- $a+b = b+a \quad \forall a,b \in S$ (commutative)

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

$S^* = S \setminus \{0\}$

group laws
- $(a \cdot b) \cdot c = a \cdot (b \cdot c) \quad \forall a,b,c \in S^*$ (associative)
- $a \cdot 1 = 1 \cdot a = a \quad \forall a \in S^*$ (identity 1)
- $\forall a \in S^* \ \exists b \in S^*$ s.t. $a \cdot b = 1$ (inverse)
- $a \cdot b = b \cdot a$ (commutative)

Common Examples: $\mathbb{R}$ (reals), $\mathbb{C}$ (complex) } infinite fields

In crypto we often work w. finite fields.

where $|S|$ is finite.

Common Example: $(\mathbb{Z}_p, +, \cdot)$ where $+, \cdot$ are mod $p$.

Denoted by $GF[p]$

Galois Field

(Another common example $GF[p^k]$ ← recitation!

in particular $GF[2^k]$.)

Common Operations:

Exponentiation: Given $a, b \in \mathbb{Z}_p$ compute $a^b$

efficiently in time $O(\log p)$

Idea: Compute $a, a^2, a^4, a^8, \ldots, a^{2^k}$ $\quad k = \log p$.

if $b = (b_{k-1} \ldots b_0) \in \{0,1\}^k$ then

($b = (b_{k-1} \ldots b_0)$: binary representation)

$$a^b = a^{\sum_{i=0}^{k-1} 2^i b_i} = \prod_{i=0}^{k-1} (a^{2^i})^{b_i}$$

## Computing mult. inverse efficiently

**Thm** (Fermat's Little Thm):

$$\forall \text{prime } p \quad \forall a \in \mathbb{Z}_p^* \quad (= \{1, \ldots, p-1\}) \qquad a^{p-1} = 1 \bmod p$$

**Corollary** $a^{-1} = a^{p-2} \bmod p$

↖ can be computed efficiently by repeated squaring

## Generating Large Primes

Choose a random $k$-bit number $n \in \{0,1\}^k$ and check if it is prime.

- Using Fermat's little Thm: Choose random $a \in \{1, \ldots, n-1\}$ and check that $a^{n-1} = 1 \bmod n$.
  This test $a^{n-1} = 1 \bmod n$ works w.h.p. for <u>random</u> $n$.
- Miller Rabin gave primality test that works <u>for every</u> $n$.
- [Agrawal-Kayal-Saxena 2002]: gave a <u>deterministic</u> primality test

This is efficient because primes are dense.

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

$\Rightarrow$ We expect to hit a prime after $\approx 0.69k$ tries.

## Shamir Secret Sharing

A secret sharing scheme allows a user to "share" their secret $\Delta$ among a set of $n$ players s.t.

- Any $t$ or more of the players can reconstruct $\Delta$.
- Any set of $< t$ players learn nothing about $\Delta$.

Formally: A $(n,t)$ secret sharing scheme consists of 2 alg's: (Share, Reconstruct).

- Share takes as input a secret $\Delta$ and outputs $n$ "shares" $(\Delta_1, \ldots, \Delta_n)$.
- Reconstruct takes as input $t$ shares $(\Delta_i)_{i \in I}$ for $I \subseteq [n]$, $|I| = t$ and outputs $\Delta$.

Security: $\forall I \subseteq [n]$, $|I| < t$, the distribution of $\{s_i\}_{i \in I}$ is independent of $s$.

Easy cases:

$t = 1$: $s_i = s$

$t = n$: Suppose (for simplicity) that $s$ is a single bit.

Share($s$): Choose at random $s_1, \ldots, s_n \in \{0,1\}$ s.t. $s_1 \oplus \ldots \oplus s_n = s$.

(This can be done by choosing $s_1, \ldots, s_{n-1} \in \{0,1\}$ at random and setting $s_n = s \oplus (s_1 \oplus \ldots \oplus s_{n-1})$)

What about $1 < t < n$?

Shamir's Scheme

(How to Share a Secret, 1979)

Suppose $s \in GF[p]$ for prime $p$.

Share($\Delta$):

- Let $a_0 = \Delta$
- Choose at random $a_1, \ldots, a_{t-1} \in GF(p)$
- Let $f(x) = \sum_{i=1}^{t-1} a_i x^i$
- Let $\Delta_i = (i, \underbrace{f(i)}_{y_i})$ $\quad \forall i \in [n]$

Reconstruct$\left((x_i, y_i)_{i=1}^{t}\right)$: (w.l.o.g. $I = \{1, \ldots, t\}$).

Note that $f(x) = \sum_{i=1}^{t} f_i(x) \cdot y_i$

where $f_i(x) = \begin{cases} 1 & \text{if } x = x_i \\ 0 & \forall x \in \{x_j\}_{j \in [t] \setminus \{i\}} \end{cases}$

↑ unique deg $\leq t$ poly that satisfies these constraints.

Note that $f_i(x) = \dfrac{\prod_{j \in [t] \setminus \{i\}} (x - x_j)}{\prod_{j \in [t] \setminus \{i\}} (x_i - x_j)}$

To reconstruct the secret $\Omega$, compute

$$\Omega = f(0) = \sum_{i=1}^{t} y_i \frac{\prod_{j \in [t] \setminus \{i\}} (-x_j)}{\prod_{j \in [t] \setminus \{i\}} (x_i - x_j)}$$

Theorem: Shamir's secret sharing scheme is information theoretically secure, i.e., an adv. with $< t$ shares has no info about $\Omega$.

Proof: Fix any $\Omega$.

For a randomly chosen deg $t-1$ curve $f$ s.t. $f(0) = \Omega$, for any non zero (and distinct) $x_1, \ldots, x_t$, it holds that $f(x_1), \ldots, f(x_t)$ are uniformly and independently distributed in $GF[p]$ (independent of $\Omega$).

# Group Theory for Dummies

In cryptography we often use finite groups (i.e., a set with a single operation)

**Common Groups**: $\mathbb{Z}_p^*$, $\mathbb{Z}_n^*$, $Q_p$, $Q_n$, Elliptic curves ← not today

$\mathbb{Z}_p^*$ = mult. group w. elements $\{1, \ldots, p-1\}$ and mult. mod $p$.
($p$ prime)

$$\mathbb{Z}_n^* = \{ a \in \{1, \ldots, n-1\} \text{ s.t. } \gcd(a, n) = 1 \}$$

$n = p \cdot q$ product of two primes (used in RSA)

Note $|\mathbb{Z}_p^*| = p - 1$

$$|\mathbb{Z}_n^*| = n - p - q + 1 = (p-1)(q-1) \triangleq \varphi(n)$$

Euler's function

**Def**: The **order** of a group is the number of elements in the group.

Order of $\mathbb{Z}_n^*$ is hard to compute given only $n$ (this is used in RSA)