Crypto Engineering '20

✦

Finite fields / extension fields

Pierre Karpman

pierre.karpman@univ-grenoble-alpes.fr

https://www-ljk.imag.fr/membres/Pierre.Karpman/tea.html

2020–09–22

Why do we care?

Extension fields (esp. of the form $\mathbb{F}_{2^n}$) are useful to:

- Build polynomial MACs
- Define matrices "over bytes" or *nibbles* (4-bit values)
  - Used e.g. in the AES
- Etc.

Those of the form $\mathbb{F}_{p^2}$, $\mathbb{F}_{p^6}$, ... often underly the arithmetic done in elliptic curve cryptography or when using pairings

Generally useful when working over (binary) discrete data $\rightsquigarrow$ they're the "right" abstraction

Roadmap

Linear-Feedback Shift Registers

Finite fields extensions

Implementation of FF arithmetic

Linear-Feedback Shift Registers

Finite fields extensions

Implementation of FF arithmetic

Linear-Feedback Shift Registers

LFSR (type 1, "Galois")

An LFSR of length n over a field $\mathbb{K}$ is a map
$\mathcal{L} : [s_{n-1}, s_{n-2}, \ldots, s_0] \mapsto$
$[s_{n-2} + s_{n-1}r_{n-1}, s_{n-3} + s_{n-1}r_{n-2}, \ldots, s_0 + s_{n-1}r_1, s_{n-1}r_0]$ where the s_i, $r_i \in \mathbb{K}$

LFSR (type 2, "Fibonacci")

An LFSR of length n over a field $\mathbb{K}$ is a map
$\mathcal{L} : [s_{n-1}, s_{n-2}, \ldots, s_0] \mapsto$
$[s_{n-2}, s_{n-3}, \ldots, s_0, s_{n-1}r_{n-1} + s_{n-2}r_{n-2} + \ldots + s_0 r_0]$ where the s_i, $r_i \in \mathbb{K}$

Theorem: The two above definitions are "equivalent"

Characterization

An LFSR is fully determined by:

- Its base field $\mathbb{K}$
- Its state size n
- Its feedback function $(r_{n-1}, r_{n-2}, \ldots, r_0)$

An LFSR may be used to generate an infinite sequence (U_m) (valued in $\mathbb{K}$):

1. Choose an initial state $S = [s_{n-1}, \ldots, s_0]$
2. $U_0 = S[n-1] = s_{n-1}$
3. $U_1 = \mathcal{L}(S)[n-1]$
4. $U_2 = \mathcal{L}^2(S)[n-1]$, etc.

Some properties

In all of the following we assume that $\mathbb{K}$ has a finite number of elements

- The sequence generated by an LFSR is periodic (Q: Why?)
- Some LFSRs map non-zero initial states to the all-zero one (Q: Give an example?)
- Some LFSRs generate a sequence of maximal period when initialised to any non-zero state (Q: What is it?)
- It is very easy to recover the feedback function of an LFSR from (enough outputs of) its generated sequence (Q: How many? How?)

A simple case: binary LFSRs

Let's focus on:

- LFSRs of type 1
- Over $\mathbb{F}_2$

$\mathcal{L}$ becomes:

1. Shift bits to the left
2. If the (previous) msb was 1
    1. Add (XOR) 1 to some state positions (given by the feedback function)

Some formalism

The feedback function of an LFSR can be written as a polynomial:

- $(r_{n-1}, r_{n-2}, \ldots, r_0) \equiv Q := X^n + r_{n-1}X^{n-1} + \ldots + r_1 X + r_0$

Same for the state:

- $(s_{n-1}, s_{n-2}, \ldots, s_0) \equiv S := s_{n-1}X^{n-1} + \ldots + s_1 X + s_0$

$\mathcal{L}$ corresponds to the map $S \times X \mod Q$

Example:

- Take $\mathcal{L}$ of length 4 over $\mathbb{F}_2$ and feedback polynomial $X^4 + X + 1$
- $\Rightarrow \mathcal{L} : (s_3, s_2, s_1, s_0) \mapsto (s_2, s_1, s_0 + s_3, s_3)$

Why should I care about those?

- Useful as a basis for PRNGs / stream ciphers (in the olden times, mostly)
- One way to define/compute with extension fields
- It's beautiful?

Linear-Feedback Shift Registers

Finite fields extensions

Implementation of FF arithmetic

Finite fields: prime fields recap

- Motivation: a rich field structure over a finite set
- Idea: take the integers and reduce modulo N
  - Operations work "as usual"
  - Over a finite set
- Problem: have to ensure invertibility of all elements
  - Necessary condition N has to be prime
  - (Otherwise, $N = pq \Rightarrow p \times q = 0 \mod N \Rightarrow$ neither is invertible)
  - In fact also sufficient: $\mathbb{Z}/p\mathbb{Z}$ is a field (also noted $\mathbb{F}_p$) iff. p is prime

Fields ⇒ polynomials

- One can define the polynomials $\mathbb{F}_p[X]$ over a finite field
- One can divide polynomials (e.g. $(X^2 + X)/(X + 1) = X$)
- $\Rightarrow$ notion of remainder (e.g. $(X^2 + X + 1)/(X + 1) = (X, 1)$
- $\Rightarrow$ can define multiplication in $\mathbb{F}_p[X]$ modulo a polynomial Q
  - If $\deg(Q) = n$, operands are restricted to a finite set of poly. of deg $< n$

Finite fields with polynomials

- $\mathbb{F}_p[X]/Q$ is a finite set of polynomials
- With addition, multiplication working as usual (again) $\rightsquigarrow$ get a ring
- To make it a field: have to ensure invertibility of all elements
  - Necessary condition: Q is *irreducible*, i.e. has no non-constant factors (Q is "prime")
  - In fact also sufficient: $\mathbb{F}_p[X]/Q$ is a field iff. Q is irreducible *over* $\mathbb{F}_p$ (constructive proof: use the extended Euclid algorithm)
  - Theorem: irreducible polynomials of all degrees exist over any given finite field

Quick questions

- How many elements does a field built as $\mathbb{F}_p[X]/Q$ have, when $\deg(Q) = n$?
- Describe the cardinality of finite fields that you know how to build
- Let $\alpha \in \mathbb{F}_q \equiv \mathbb{F}_p[X]/Q$. what is the result of $\alpha + \alpha + \ldots + \alpha$ (addition of p copies of α)?

Characteristic

Characteristic of a field

The *characteristic* of a field $\mathbb{K}$, noted $\mathrm{char}(\mathbb{K})$, is the min. $n \in \mathbb{N}$ s.t. $\forall x \in \mathbb{K}$, $\sum_{i=1}^{n} x = 0$, or 0 if no such n exists

- Prime fields $\mathbb{F}_p$ have characteristic p
- Extension fields $\mathbb{F}_{p^e}$ have characteristic p
- In characteristic two ("even characteristic"), $+ \equiv -$

We may say that the characteristic of a field $\mathbb{F}_q$ is:

- "small", if e.g. $= 2, 3, \ldots$
- "medium" if e.g. $q = p^6, p^{12}, \ldots$
- "large" if e.g. $q = p, p^2$

Quick remarks

- Two finite fields of equal cardinality are *unique up to isomorphism*
- But different choices for Q may be possible $\Rightarrow$ different *representations* $\rightsquigarrow$ important for (explicit) implementations
- One can build extension towers: extensions over fields that were already extension fields, iterating the same process as for a single extension

Linear-Feedback Shift Registers

Finite fields extensions

Implementation of FF arithmetic

How to implement finite field operations?

Some options (not the only ones):

- $\mathbb{F}_p$:
  - Addition: add modulo
  - Multiplication: multiply modulo
  - Inverse: use the extended Euclid algorithm or the little Fermat Theorem
- $\mathbb{F}_{p^e}$:
  - Represent elements as polynomials, then
  - Addition: add modulo, coefficient-wise
  - Multiplication: multiply polynomials modulo (w.r.t. polynomial division) $\rightsquigarrow$ can use LFSRs
  - Inverse: use the extended Euclid algorithm (for polynomials)

Multiplication in $\mathbb{F}_{2^n}$

We now focus on characteristic two for simplicity

- $\alpha \in \mathbb{F}_{2^n} \equiv \mathbb{F}_2[X]/Q$ is "a polynomial over $\mathbb{F}_2$ of deg $< n$"
- So $\alpha = \alpha_{n-1}X^{n-1} + \ldots + \alpha_1 X + \alpha_0$
- So we can multiply α by $X \Rightarrow \alpha_{n-1}X^n + \ldots + \alpha_1 X^2 + \alpha_0 X$
- But this may be of deg $= n$, so "not in $\mathbb{F}_{2^n}$"
- So we reduce the result modulo

$$Q = X^n + q_{n-1}X^{n-1} + \ldots + q_1 X + q_0,$$

the defining polynomial of $\mathbb{F}_{2^n}$

Reduction: two cases

Case 1: $\deg(\alpha X) < n$

- There's nothing to do

Case 2: $\deg(\alpha X) = n : \alpha X = X^n + \ldots + \alpha_0 X$

- Then $\deg(\alpha X - Q) < n$
- And $\alpha X - Q$ is precisely the remainder of $\alpha X \div Q$
- (Think how if $a \in]\!]N, 2N[\![$, $a \mod N = a - N$)

Multiplication + reduction: alternative view

$(\alpha_{n-1}, \ldots, \alpha_1, \alpha_0) \times X \mod (q_n, q_{n-1}, \ldots, q_1, q_0) =$

- $(\alpha_{n-2}, \ldots, \alpha_1, \alpha_0, 0)$ if $\alpha_{n-1} = 0$
- $(\alpha_{n-2} - q_{n-1}, \ldots, \alpha_1 - q_2, \alpha_0 - q_1, -q_0)$ if $\alpha_{n-1} = 1$
- (or $(\alpha_{n-2} + q_{n-1}, \ldots, \alpha_1 + q_2, \alpha_0 + q_1, q_0)$ as we're in characteristic two)
- or $(\alpha_{n-2} + q_{n-1}\alpha_{n-1}, \ldots, \alpha_1 + q_2\alpha_{n-1}, \alpha_0 + q_1\alpha_{n-1}, q_0\alpha_{n-1})$ $\Rightarrow$ the result of one step of LFSR with feedback polynomial equal to $(-)Q$!

Summary

- An element of $\mathbb{F}_{2^n} \equiv \mathbb{F}_2[X]/Q$ is a polynomial
- ...is the state of an LFSR with feedback polynomial Q
- Multiplication by X is done $\mod Q$
- ...is the result of clocking the LFSR once
- Multiplication by X^2 is done by clocking the LFSR twice, etc.
- Multiplication by $\beta_{n-1}X^{n-1} + \ldots + \beta_1 X + \beta_0$ is done "the obvious way", using distributivity

A note on representation

It is convenient to write $\alpha = \alpha_{n-1}X^{n-1} + \ldots + \alpha_1 X + \alpha_0$ as the integer $a = \alpha_{n-1}2^{n-1} + \ldots + \alpha_1 2 + \alpha_0$

- Example: $X^4 + X^3 + X + 1$ "=" $27 = \texttt{0x1B}$

Examples in $\mathbb{F}_{2^8} \equiv \mathbb{F}_2[X]/X^8 + X^4 + X^3 + X + 1$

Example 1:

- $\alpha = X^5 + X^3 + X$ (`0x2A`), $\beta = X^2 + 1$ (`0x05`)
- $\alpha + \beta = X^5 + X^3 + X^2 + X + 1$ (`0x2F`)
- $\alpha\beta = X^2\alpha + \alpha = X^7 + X^5 + X^3$ (`0xA8`) $+ X^5 + X^3 + X = X^7 + X$ (`0x82`)

Example 2:

- $\alpha = X^5 + X^3 + X$, $\gamma = X^4 + X$ ($\texttt{0x12}$)
- $\alpha\gamma = X^4\alpha + X\alpha$
  - $X^4\alpha = X(X(X^7 + X^5 + X^3))$
  - $X(X^7 + X^5 + X^3) =$ $(X^8 + X^6 + X^4) + (X^8 + X^4 + X^3 + X + 1) = X^6 + X^3 + X + 1$
  - $X(X^6 + X^3 + X + 1) = X^7 + X^4 + X^2 + X$
- $= X^7 + X^4 + X^2 + X$ ($\texttt{0x96}$) $+ X^6 + X^4 + X^2$ ($\texttt{0x54}$) $=$ $X^7 + X^6 + X$ ($\texttt{0xC2}$)

Other implementation possibilities

- Precompute the full multiplication table $\rightsquigarrow O(q^2)$ space (quickly impractical)
- Precompute a log table (e.g. using Zech's representation) $\rightsquigarrow O(q)$ space (reasonable for small q)
- Use efficient polynomial arithmetic + reduction, for instance:
  - `pclmulqdq` for extensions of $\mathbb{F}_2$
  - Kronecker substitution in other small characteristics
- Sometimes, only implementation by a constant matters