Implementation Security In Cryptography

Lecture 08: Field Isomorphism

Recap

- In the last lecture
  - AES Algorithm

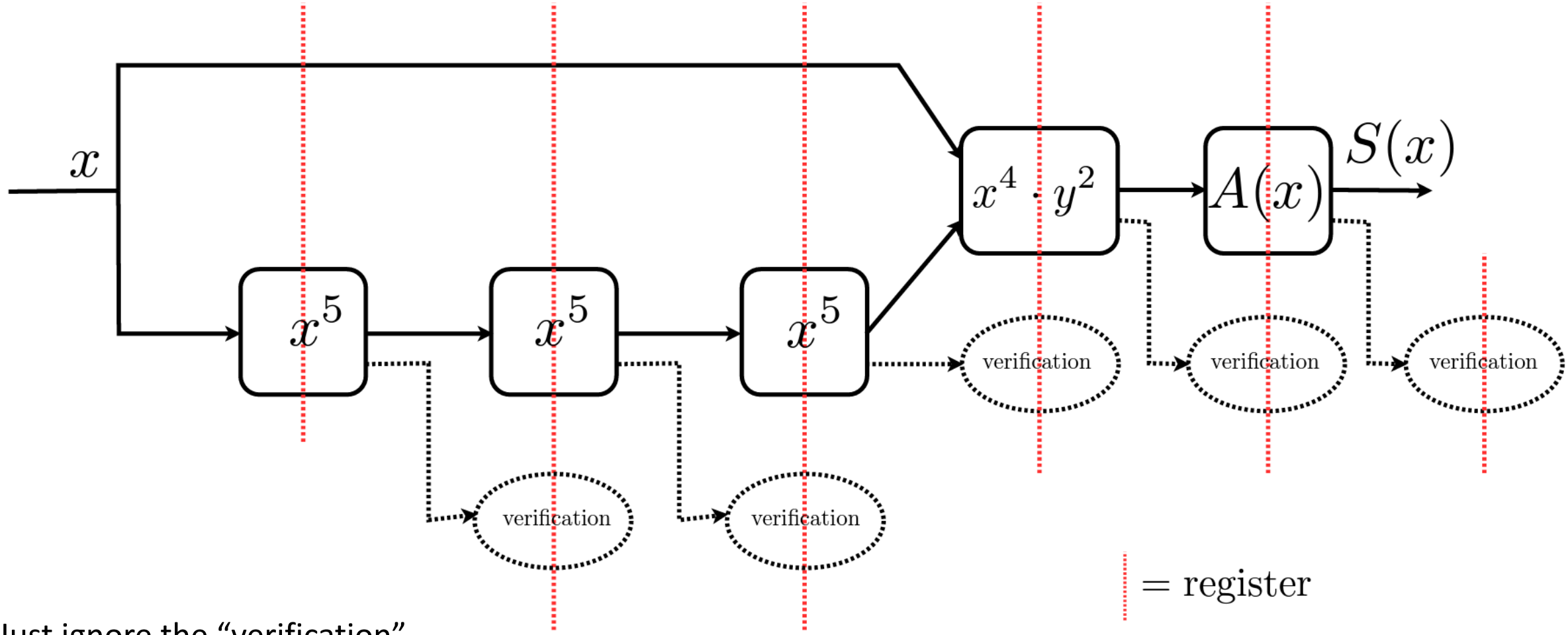
S-box Mystery

- Use a table to do the byte substitution
  - Is this a great approach?
  - Although, used in many software implementations
  - This has quite a lot of security issues.
  - Also, AES S-Box has a nice math
    - **How to implement this in hardware/software??**

How to Compute Inverse in a Finite Field

- Several ways exist
  - Option 1: Extended Euclidean algorithm
    - Your same old GCD finder with a twist — it can find multiplicative inverse
    - Hardly used
  - Option 2: Any idea? — there was a hint in the previous class
  - Option 3: Exploiting field isomorphisms
    - **The most compact way…**

S-Box Hardware — One Popular Way



- Just ignore the “verification”

How to Compute Inverse in a Finite Field

- Several ways exist
  - Option 1: Extended Euclidean algorithm
    - Your same old GCD finder with a twist — it can find multiplicative inverse
    - Hardly used
  - Option 2: Any idea? — there was a hint in the previous class

  - Option 3: Exploiting field isomorphisms
    - **The most compact way…**

Main Idea

- Option 3: Exploiting field isomorphisms
  - A finite field can have different forms — e.g. depending on the polynomial used for reduction.
  - The computation (multiplication, inversion) is less costly in some forms
  - We, therefore, go to a specific form (called composite field) where these operations can be represented in terms of *smaller* operations
    - e.g. inversions in GF(2^8) can be represented as inversions in GF(2^2) — much easy to compute.
  - What we need
    - A systematic way to represent elements in any field.
    - A systematic way to map between two forms.

Example GF(2^4)

- Irreducible Polynomial: x^4+x+1
- Elements:

Example GF(2^4)

- Irreducible Polynomial: x^4+x+1
- Elements:

$1 : x$
$2 : x^2$
$3 : x^3$
$4 : x^4 = x + 1$
$5 : x^2 + x$
$6 : x^3 + x^2$
$7 : x^3 + x + 1$
$8 : 1 + x^2$
$9 : x + x^3$
$10 : x^2 + x + 1$
$11 : x^3 + x^2 + x$
$12 : x^3 + x^2 + x + 1$
$13 : x^3 + x^2 + 1$
$14 : x^3 + 1$
$15 : 1$
$0 : 0$

- Notice the role of FLT!!

Root of a Polynomial in Finite Field

- Some element in the (extension) field that satisfies p(x) = 0
- Example: consider $x^4 + x + 1$; equating this to 0, we get $x^4 = x + 1$.
  So for any element α, for which $\alpha^4 = \alpha + 1$ is satisfied is a root!!
- There are 4 roots in this case: find them!!

$1 : x$
$2 : x^2$
$3 : x^3$
$4 : x^4 = x + 1$
$5 : x^2 + x$
$6 : x^3 + x^2$
$7 : x^3 + x + 1$
$8 : 1 + x^2$
$9 : x + x^3$
$10 : x^2 + x + 1$
$11 : x^3 + x^2 + x$
$12 : x^3 + x^2 + x + 1$
$13 : x^3 + x^2 + 1$
$14 : x^3 + 1$
$15 : 1$
$0 : 0$

Root of a Polynomial in Finite Field

- Some element in the (extension) field that satisfies p(x) = 0
- Example: consider $x^4 + x + 1$; equating this to 0, we get $x^4 = x + 1$.
  So for any element α, for which $\alpha^4 = \alpha + 1$ is satisfied is a root!!
- There are 4 roots in this case: $x, x + 1, x^2, x^2 + 1$

$1 : x$
$2 : x^2$
$3 : x^3$
$4 : x^4 = x + 1$
$5 : x^2 + x$
$6 : x^3 + x^2$
$7 : x^3 + x + 1$
$8 : 1 + x^2$
$9 : x + x^3$
$10 : x^2 + x + 1$
$11 : x^3 + x^2 + x$
$12 : x^3 + x^2 + x + 1$
$13 : x^3 + x^2 + 1$
$14 : x^3 + 1$
$15 : 1$
$0 : 0$

Primitive Element of a Field

- Consider the field GF(2^n).
- There is an element α such that every non-zero element can be written in terms of the form of α^k .
- This element is called the generator or primitive element of the group.
- Not every element in a field is a primitive element.
  - Example: Try with $x^2 + x$ for the finite field of the last slide..

Primitive Polynomial

- A primitive polynomial is the monic polynomial of minimum degree such that the **primitive element is a root.**
- A primitive polynomial is always irreducible but not vice-versa.
- Over GF(2^n), there are $\phi(2^n - 1)/n$ primitive polynomials, where ϕ is the Euler's Totient function.
  - The polynomial we say for $GF(2^4)$ was primitive!!
  - AES irreducible polynomial is not a primitive polynomial of $GF(2^8)$
  - Then, how can I represent all the elements?

Another Example

- Lets change the polynomial for $GF(2^4)$ to $p(x) = x^4 + x^3 + x^2 + x + 1$
  - This one is not primitive but irreducible.
  - You can generate the elements by yourself!!!
  - Let's see the roots $x, x^2, x^3, x^3 + x^2 + x + 1$
    - Let's consider x^2 — how many field elements it can generate?
      - $(1,\ x^2)\ (2,\ x^3 + x^2 + x + 1)\ (3,\ x)\ (4,\ x^3)\ (5,\ 1)$
      - So this is not a primitive element!!!

Bases of the Binary Field

- Such fields are represented using two types of bases:
  - Polynomial base: Let p(x) be an irreducible polynomial over GF(2^n), and let α be the root of p(x). Then the set: $\{1, \alpha, \alpha^2, \cdots \alpha^{n-1}\}$ is called the polynomial base.
  - Normal base: Let p(x) be an irreducible polynomial over GF(2^n), and let α be the root of p(x). Then the set: $\{\alpha, \alpha^2, \alpha^{2^2}, \cdots \alpha^{2^{n-1}}\}$ is called the normal base, if the m elements are linearly independent.

Polynomial Representation

- Any element in the field can be expressed in terms of its bases.
- For example in the field $GF(2^n)$, an element can be expressed wrt. its polynomial bases as:

$$a(\alpha) = a_{n-1}\alpha^{n-1} + a_{n-1}\alpha^{n-2} + a_0$$

- Normal Basis:
  - $a(\alpha) = a_{n-1}\alpha^{2^{n-1}} + a_{n-1}\alpha^{2^{n-2}} + a_0\alpha$

Can you tell me the polynomial basis for $GF(2^4)$ with $p(x) = x^4 + x^3 + x^2 + x + 1$?

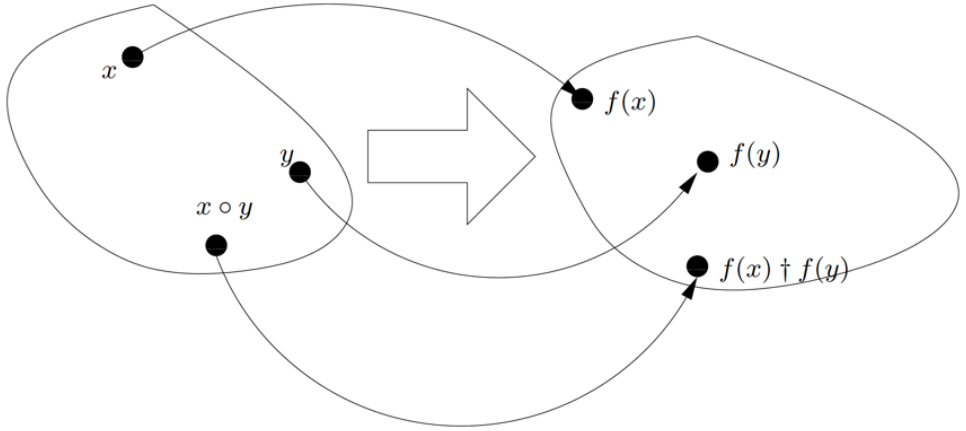
Another Example

- Lets change the polynomial for $GF(2^4)$ to $p(x) = x^4 + x^3 + x^2 + x + 1$
  - This one is not primitive but irreducible.
  - You can generate the elements by yourself!!!
  - Let's see the roots $x, x^2, x^3, x^3 + x^2 + x + 1$
    - Let's consider x^2 — how many field elements it can generate?
      - $(1,\ x^2)\ (2,\ x^3 + x^2 + x + 1)\ (3,\ x)\ (4,\ x^3)\ (5,\ 1)$
      - So this is not a primitive element!!!

Let's Pause

- What's going on?? Why this math??
- Well, as you shall see next $GF(2^4)$remains $GF(2^4)$ irrespective of which irreducible poly you choose…
- Then what is the issue??
  - Turns out that the choice of the field matters while you implement it in hardware…
    - Number of gates vary etc…
- Eventually what we want is an efficient finite field inverse circuit for AES.

Isomorphism



For two groups G1 and G2, a surjective function G1 to G2 is said to be a homomorphism iff

$$f(x \circ y) = f(x) \dagger f(y).$$

Note, the operators on the left and right are not the same.

An injective (one-to-one) homomorphism is called an isomorphism.

The idea of isomorphism can be extended to rings and fields. In these extensions the only difference is that the latter two are defined wrt. two operators, say (+,.). Thus, we say f: R1→R2 is say a field isomorphism iff: f(a+b)=f(a)+f(b), and f(a.b)=f(a).f(b) for every a and b in R1.

Example in GF(2^4)

0	z^2	z^3	z^3+z^2
1	z^2+1	z^3+1	z^3+z^2+1
z	z^2+z	z^3+z	z^3+z^2+z
$z+1$	z^2+z+1	z^3+z+1	z^3+z^2+z+1

There are 3 irreducible polynomials of degree 4, which can be used to construct the above field elements: $f_1(z) = z^4 + z + 1$, $f_2(z) = z^4 + z^3 + 1$, $f_3(z) = z^4 + z^3 + z^2 + z + 1$.

The fields are denoted as F_1, F_2, and F_3 respectively.

The resulting fields all have 16 elements, as shown above.

However, the operations are different.

Like the same operation, $z \,.\, z^3$ would result in $z^4 = z + 1, z^3 + 1, z^3 + z^2 + z + 1$ in the 3 fields.

Defining Isomorphism

- The fields are isomorphic and one can establish a mapping between say F_1 and F_2, by computing $c \in F_2,\ st.\ f_1(c) \equiv 0 \left(mod\ f_2\right)$.
- The mapping $z \rightarrow c$ is thus used to construct the isomorphism, say T: F1→F2
- An example for c could be $c = z^2 + z$. To verify compute:

$$f_1\left(z^2+z\right) = \left(z^2+z\right)^4 + \left(z^2+z\right) + 1 = z^8 + z^4 + z^2 + z + 1\ (mod\ f_2)$$

Now, note that for $mod\ f_2$, we substitute $z^4 = z^3 + 1$.

$$z^4 = z^3 + 1 \Rightarrow z^5 = z^4 + z = z^3 + z + 1 \Rightarrow z^6 = z^4 + z^2 + z = z^3 + z^2 + z + 1$$

$$\Rightarrow z^8 = z^6 + 1 = z^3 + z^2 + z.$$

$$\text{Thus, } f_1(c) = z^8 + z^4 + z^2 + z + 1 \equiv 0 (\text{mod } f_2)$$

Check on Homomorphism

- Consider two elements $e_1 = z^2 + z, e_2 = z^3 + z$.
- Product in field F_1: $(z^2 + z)(z^3 + z) = z^5 + z^4 + z^3 + z^2$
- In field F_1: $z^4 = z + 1 \Rightarrow z^5 = z^2 + z$.
  - Thus, the product is: $z^2 + z + z^3 + z + 1 + z^2 = z^3 + 1.$
- The same operation can also be performed in the field F_2.
- Compute,

$$T(e_1) = (z^2 + z)^2 + (z^2 + z) mod(z^4 + z^3 + 1) = z^4 + z^2 + z^2 + z = z^3 + z + 1.$$

- Likewise, $T(e_2) = (z^2 + z)^3 + (z^2 + z) mod(z^4 + z^3 + 1)$

$= z + 1$

Check on Homomorphism

Multiplying in the field F_2 : $T(e_1).T(e_2) = (z+1)(z^3+z+1) \bmod (z^4+z^3+1) = z^2$.

This can be seen as the mapped result from F_1 :

$$T(z^3+1) = (z^2+z)^3+1 = (z^6+z^5+z^4+C) \bmod (z^4+z^3+1)$$

$$= (z^3+z^2+z+1)+(z^3+z+1)+(z^3+1)+(z^3+1)$$

$$= z^2$$