OOKMARK [2][]Outline0.2Tower Field DecompositionsOOKMARK [2][]Outline0.2Tower Field Decompositions

Thesis Progress Report

Christopher A. Wood

April 13, 2013

Agenda

Isomorphic Function Generation - Primitive Irreducible Polynomials

The isomorphic mappings are constructed as follows (assume we're constructing a function for mapping $GF(2^{nm}) \rightarrow GF((2^n)^m)$):

- Find two generators α and β ($\alpha \in GF(2^{nm})$ and $\beta \in GF((2^n)^m)$), where α and β are roots of the same *primitive* irreducible polynomial.
- Map α^k to β^k for $1 \leq k \leq 2^{nm}$.
- Multiplication and addition homomorphism is guaranteed.
  - $\alpha^i \times \alpha^j = \alpha^{i+j} = \beta^{i+j} = \beta^i \times \beta^j$
  - $\alpha^t = \alpha^i + \alpha^j \rightarrow \beta^t = \beta^j + \beta^j$

Isomorphic Function Generation - Irreducible Polynomials

- Find two generators α and β ($\alpha \in GF(2^{nm})$ and $\beta \in GF((2^n)^m)$).
- Map α^k to β^k for $1 \leq k \leq 2^{nm}$ (multiplication homomorphism holds)
- For all $0 \leq i \leq 2^{nm} - 1$ check to see if $\alpha^i + 1 \rightarrow \beta^i + 1$.
- Multiplication and addition homomorphism is now guaranteed.
  - $\alpha^i \times \alpha^j = \alpha^{i+j} = \beta^{i+j} = \beta^i \times \beta^j$
  - $\alpha^t = \alpha^i + \alpha^j = \alpha^i \times (1 + \alpha^{j-i}) \rightarrow \beta^t = \beta^i \times (1 + \beta^{j-i}) = \beta^i + \beta^j$

Matrix Transformation **T**

The matrix **T** can be generated with the following algorithm.

- Let β be a generator of $GF((2^n)^m)$ such that $\alpha^i \in GF(2^{nm})$ is mapped to β^i for all $0 \leq i \leq 2^{nm} - 1$ (α forms a basis of $GF((2^n)^m)$).
- Compute $\alpha^0, \alpha^1, \alpha^2, \dots, \alpha^{nm-1}$.
- Define the columns of **T** as the transpose of each *nm*-dimensional bit vector of these powers:

$$\mathbf{T} = \left[\begin{array}{ccc} (\alpha^{nm-1})^T & \dots & (\alpha^1)^T (\alpha^0)^T \end{array} \right]$$

An Example

- $\alpha = x$ and $\beta = xy$
- $(x^7 + x^6 + x^5 + x^2 + x + 1) \rightarrow [(x^3 + x^2 + x + 1)y + (x^3 + x^2 + 1)]$
- $\mathbf{T} = \left[\begin{array}{ccc} (xy^{nm-1})^T & \ldots & (xy^1)^T (xy^0)^T \end{array} \right]$

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

Another Example

- $\alpha = x$ and $\beta = xy$ (same homomorphic mapping)
- $(x^6) \rightarrow [(x^2)y + (x^3 + x^2 + 1)]$
- $\mathbf{T} = \begin{bmatrix} (xy^{nm-1})^T & \dots & (xy^1)^T (xy^0)^T \end{bmatrix}$

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

Different Tower Field Decompositions

$$(1) GF(2^{16}) \rightarrow GF((2^8)^2)$$
$$(2) GF(2^{16}) \rightarrow GF((2^4)^4)$$
$$(3) GF(2^{16}) \rightarrow GF((2^2)^8)$$

We need only study these decompositions - optimal tower-field decompositions for smaller fields are in the literature.

Multiplicative Inverse Calculations

The derivation gets messy very quick...

$$(b*x^3+c*x^2+d*x+e)*(f*x^3+g*x^2+h*x+i) = k*(x^4+A*x^3+B*x^2+C*x+D)+1$$

$$f \to \frac{1}{x^3\left(e+dx+cx^2+bx^3\right)}\left(1-ei+Dk-ehx-dix+Ckx\right) \left(-egx^2-dhx^2-cix^2+Bkx^2-dgx^3-chx^3-bix^3+Akx^3-cgx^4\right) \left(-bhx^4+kx^4-bgx^5\right)$$

Are there easier ways to calculate the inverse?

Finding Capable Polynomials

- Primitive polynomials always work for the mapping
  - Let α be a primitive root of the field $F_2[x]/P(x)$ and $P(\alpha) = 0$
  - $P(x)$ is therefore a *primitive polynomial*
  - Powers α (which are linearly independent) can be used to form a standard basis
- This didn't seem to work with non-primitive polynomials (i.e. $(x^8 + x^4 + x^3 + x + 1)$).
  - **Question**: Why not? Group homomorphism between the elements holds...

Choosing the Right Irreducible Polynomials

- Exhausively search for all (primitive) irreducible polynomials
- For each polynomial $P(x)$, generate the transformation matrix and estimate the complexity of the multiplicative inverse calculation
  - The polynomials $P(x)$, $P(y)$ and $Q(y)$ determine the complexity of this mathematication operation.
  - $\alpha^{-1} = (bx + c)^{-1} = b(b^2B + bcA + c^2)^{-1} + (c + bA)(b^2B + bcA + c^2)^{-1}$
- Choose the polynomial $P(x)$ that yields the lowest “cost”

Choosing the Right Transform

- Let $\mathbf{T}^*$ be the optimal transformation matrix in the set of transformations $\mathscr{T}$.
- The "cost" of transforms is the number of 1s in the matrix $\mathbf{T}^*$.
- The "cost" of the inverse is dependent on the polynomial selection.

$$T^* = \min_{T_i \in \mathscr{T}} \{C(transform) + C(inverse) + C(invTransform)\}$$

Exhaustively Searching All S-boxes

- Loop over invertible binary matrices and all constants for affine transformation
- For each valid mapping, measure the cryptographic strength using the Boolean function analysis software
- Pick the one with the best properties

An Interesting Case

- Nyberg's Power Mapping: $F(x) = x^{2^k+1}$
- These functions are 2-differentially unfiform with a $\mathscr{N}_l$ equal to precisely $2^{n-1} - 2^{\frac{n-1}{2}}$.
    - That's better than the inverse mapping $F(x) = x^{-1}$
- In a normal basis, this reduces to squaring (which is free) and multiplications
- For hardware, does this yield a more efficient **and** more secure mapping for 16-bit S-boxes?

Combinational Implementations - XOR

X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	0

```
Y <= (NOT(X1) AND (X2)) OR (X1 AND NOT(X2))
```

- Walk the truth table output Y and insert the appropriate literal assignments into the DNF formula
- I have the code to accomplish this task...

Research Goal

What is the resource consumption and security tradeoff for Boolean function implementations of S-boxes versus those based on mathematical operations?

What's next?

Action Items

- Generate the list of all transformation matrices **T**
- List of all (primitive) irreducible polynomials up to degree 16
- Exhaustively generate all S-boxes based on the inverse mapping $F(x) = x^{-1}$
- Thesis chapter on composite field arithmetic decomposition and inverse derivations using composite fields

Next meeting: **4/15/13**