Thesis Progress Report #7

Christopher A. Wood

May 29, 2013

Agenda

Updates

- Magma?

Counting the number of gates

Main idea:

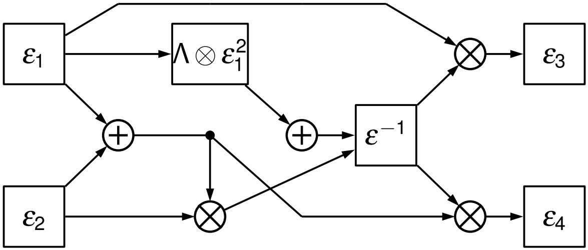
- Decompose operations in $GF(2^{16})$ to $GF(2)$.
- Implement $GF(2)$ operations using simple logic gates.

Key

- $GF(2^{16})/GF(2^8) : s(y) = y^2 + \Psi y + \Lambda$
- $GF(2^8)/GF(2^4) : r(x) = x^2 + \Theta x + \Pi$
- $GF(2^4)/GF(2^2) : q(w) = w^2 + \Omega w + \Sigma$
- $GF(2^2)/GF(2) : p(v) = v^2 + \Gamma v + \Delta$ $(\Gamma = 1, \Delta = 1)$
- $\theta \in GF(2^{16})$, $\zeta \in GF(2^8)$, $\varepsilon \in GF(2^4)$, $\delta \in GF(2^2)$, $\gamma \in GF(2)$

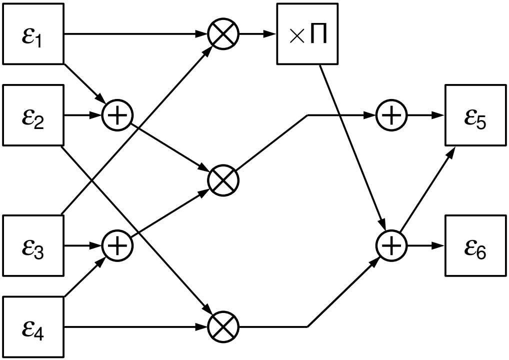
Multiplicative inverse in $GF(2^{16})$

$$\zeta^{-1} = (\varepsilon_1 y + \varepsilon_2)^{-1}$$



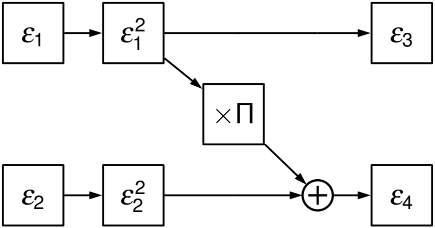
Multiplication in $GF(2^8)$

$$\zeta_1 \times \zeta_2 = (\varepsilon_1 y + \varepsilon_2) \times (\varepsilon_3 y + \varepsilon_4) = (\varepsilon_5 y + \varepsilon_6)$$



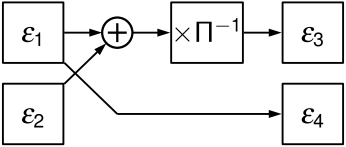
Squaring in $GF(2^8)$

$$\zeta^2 = (\varepsilon_1 y + \varepsilon_2)(\varepsilon_1 y + \varepsilon_2) = (\varepsilon_3 y + \varepsilon_4)$$



Scaling in $GF(2^8)$

$$\zeta \times \Lambda = (\varepsilon_1 y + \varepsilon_2) \times \Lambda = (\varepsilon_3 y + \varepsilon_4)$$



Subfield operations

Perform the algebra, minimize the arithmetic, create the circuit, count the gates

Basis changes to degree 2 extension tower field representations

Magma!

Optimizing linear transformations

$$
\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}
$$

Equivalently...

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} x_0 + x_1 + x_3 + x_4 + x_5 \\ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \\ x_1 + x_3 + x_4 + x_5 \\ x_0 + x_1 + x_3 + x_6 \\ x_0 + x_3 + x_4 + x_6 \\ x_0 + x_1 + x_3 + x_4 + x_6 \\ x_0 + x_1 + x_3 \\ x_0 + x_1 + x_3 + x_4 + x_5 + x_6 + x_7 \end{pmatrix}$$

A better solution

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} x_9 + x_{10} \\ x_6 + x_8 + x_{12} \\ x_8 + x_{10} \\ x_{11} \\ x_0 + x_3 + x_4 + x_6 \\ x_4 + x_{11} \\ x_9 \\ x_7 + x_{11} + x_{12} \end{pmatrix}$$

with $x_8 = x_1 + x_3$, $x_9 = x_0 + x_8$, $x_{10} = x_4 + x_5$, $x_{11} = x_6 + x_9$, and $x_{12} = x_2 + x_{10}$

Optimization

Main idea:

- Factor out bits that can share a gate
- Cast the new shared gate as a new variable in the linear system
- Repeat factorization until no more shares can be found

Fun fact:

- This is formalized as the *Shortest Linear Program* problem - NP-hard
- Complexity proved by considering the decision variant - does there exist a linear program with at most k lines which computes the function? - and reducing from VERTEX-COVER
- Approximation algorithms have ratios of at least $3/2$ (i.e. they cannot yield near-optimal solutions)

Optimization

Morale: heuristics are needed!

- Paar: Greedy factoring
- Peralta: Greedy factoring with (Euclidean) norm-based tie breaker (details discussed in the thesis)

Question: which one works best for 16×16 matrices (linear transformations)?

Exhaustive search for the optimal linear program

... or, perform an exhaustive search similar to Canright

Our contribution: reimplement Canright's exhaustive search and then *parallelize* it

Problem: Exhaustive search is recursive, so how can we partition the work among multiple cores/threads? ☹

Solution: *fork/join recursive actions* ☺

Demonstration

Demo (in Java)

Cryptographically significant power mappings

All S-boxes are based off of some *nonlinear power mapping*: $f(x) = x^d$

Name	**Exponent** (d)
Inverse	$-1 \equiv 2^n - 2$
Gold	$2^k + 1$, $\gcd\{k, n\} = 1$ for some $1 \leq k \leq 2^n - 1$
Kasami	$2^{2k} - 2^k + 1$, $\gcd\{k, n\} = 1$ for some $1 \leq k \leq 2^{n-1} - 1$
Dobertin	$2^{4k+3k+2k+k} - 1$ over $GF(2^s)$ with $s = 5k$
Niho	$2^m + 2^{m/2} - 1$ over $GF(2^s)$ with $s = 2m + 1$ and m even
Welch	$2^m + 3$ over $GF(2^s)$ with $s = 2m + 1$

What are our options for $GF(2^{16})$?

$$d \in \{3, 1023, 63, 255, 15, 16383, \mathbf{65534}, 4095\}$$

- No Kasami, Gold, Welch, and Niho exponents exist for $n = 16$
- We need only study the *inverse* and *Dobbertin* exponents

Action: Complete security analysis by Monday, 6/3

Affine transformation update

- Boolean function-related properties are unaffected by affine transformations
- Need to select one with optimal algebraic complexity
  - $S = A \circ P$ (maximum algebraic complexity is $n+1$ for $GF(2^n)$) - efficient
  - $S = A \circ P \circ A$ (maximum algebraic complexity is $2^n - 1$ for $GF(2^n)$) - inefficient

Action Items

- Quantified security results for all Dobbertin mappings and the inverse mapping
- Magma code to count the number of gates needed for multiplicative inverse calculation
- Continued writing for thesis (draft of all relevant chapters by Monday)

Next meeting: **6/3/13**