Finding differential trails using an evolutionary algorithm

Tim van Dijk
tim.vandijk96@gmail.com

Institute for Computing and Information Sciences – Digital Security
Radboud University Nijmegen

Lunch colloquium
December 7, 2018

Outline

Research internship

- Part of the TRU/e master.
- 2.5 days a week for one semester.
- Supervised by Joan Daemen.
- Overarching theme: cryptography over GF(P)
    - Bitsliced $+, \cdot$ and ^{2} over GF(3), GF(5) and GF(7) for ARM.
    - Multidimensional discrete Fourier transform with full precision.
    - Finding MAC collisions though differential cryptanalysis.
    - Key recovery after finding a collision.

Cryptography in GF(P) - What?

- Normally, in symmetric cryptography we work over GF(2).
- Here, we work over a Galois Field of size P with P a prime.
- This means we work with digits in $\{0, \ldots, P-1\}$ instead of bits that are either 0 or 1.
- In GF(P) things often behave a little differently, for example addition no longer is the same as subtraction.

Cryptography over GF(P) - Why?

- Not only functions over GF(2) and GF(2^n) have been studied extensively.
- Interesting differences that to a certain extent provide richer functionality than binary fields.
  - GF(P): Perfect Nonlinear Functions.
  - GF(2): Almost Perfect Nonlinear Functions.
- Try to capitalize on the large amount of theoretical results over GF(P).
- Generalization might give new insights.
- Applicable to coding theory and signal modulation techniques?

Cryptography over GF(P) - How?

- Specialized hardware - ternary computers are virtually nonexistent, but again, it might be useful in telecommunications.
- In practice, most data will continue to be represented in bit strings.
- Software emulation - $\lceil \log_2 P \rceil$ bits required to store a digit in GF(P).

8-digit value in GF(7)

3	0	5	2	6	1	4	3

Binary array of 8-bit values

1	0	0	0	0	1	0	1	least-significant bits
1	0	1	1	1	0	0	1	middle bits
0	0	1	0	1	0	1	0	most-significant bits

Bitsliced operations for ARM

Bitslicing is a technique where a computation is:

1. reduced to elementary operations (e.g. OR, AND, XOR);
2. executed in parallel, with as many simultaneous instances a there are bits in a register.

To work, data needs to be transposed:

Normal				Bitsliced			
r_0	a_2	a_1	a_0	r_0	c_0	b_0	a_0
r_1	b_2	b_1	b_0	r_1	c_1	b_1	a_1
r_2	c_2	c_1	c_0	r_2	c_2	b_2	a_2

Bitsliced operations for ARM

Normal				Bitsliced			
r_0	a_2	a_1	a_0	r_0	c_0	b_0	a_0
r_1	b_2	b_1	b_0	r_1	c_1	b_1	a_1
r_2	c_2	c_1	c_0	r_2	c_2	b_2	a_2

For values a, b and c, compute in parallel: $X_0 + X_0 \cdot X_1$.

```
AND     R3, R0, R1      ; R3 := X0 * X1
EOR     R3, R0, R4      ; R3 := X0 + (X0 * X1)
```

Can sometimes be used to dramatically increase throughput at the cost of increased latency.

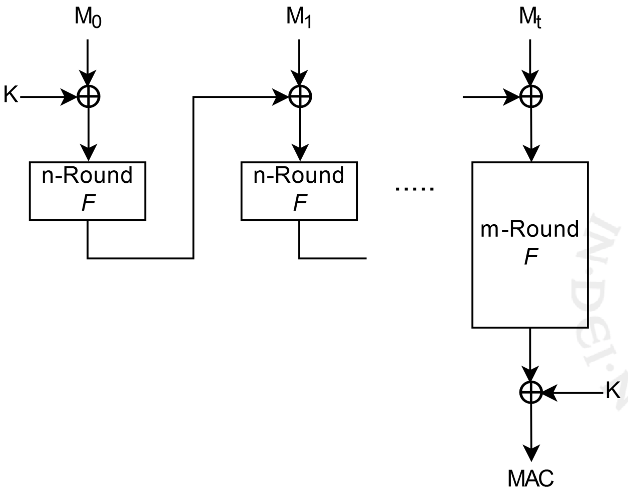
Bitsliced operations for ARM

- Addition, multiplication and squaring in GF(3), GF(5) and GF(7).
- Represent the output in terms of a function of the input bits.
- Difficult to do by hand $\rightarrow$ use logic synthesis tools.
- Implemented in assembly for ARM Cortex M4.

Multidimensional discrete Fourier transform

- GF(P) version of the Walsh-Hadamard transform.
- Important in linear cryptanalysis.
- Used to compute the correlation of a function with all linear functions at once.
- In the binary case, correlation is between -1 and 1.
- In the non-binary case, it deals with complex numbers.
- Two versions:
  - a straightforward one that uses floating points. Numbers are represented as $a + bi$.
  - one that uses polynomials of form $\sum a_j \omega^j$ with $\omega = e^{\frac{2\pi i}{p}}$
- The latter has full precision.

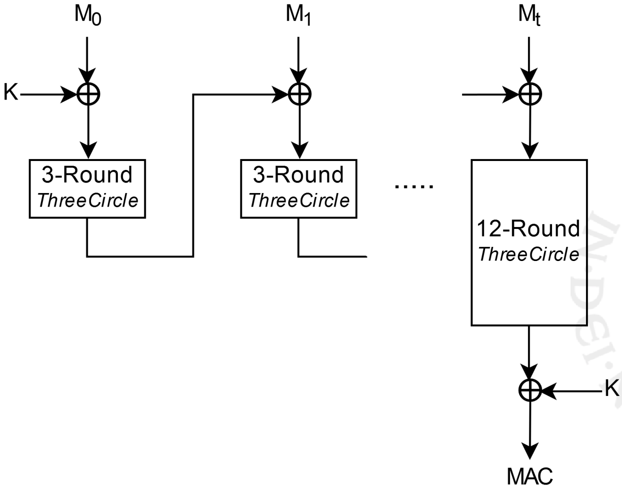
MACs using (round reduced) transformations



The cryptographic transformation ThreeCircle

- A permutation designed by Joan for me to experiment on.
- Operates on a state of 160 digits over GF(3).
- State a arranged in 5 32-digit lanes, we write $a = (a_0, a_1, a_2, a_3, a_4)$.
- Classical iterated structure: it iterates a round function R_i 12 times to the state.

MACs using (round reduced) ThreeCircle



The round function of ThreeCircle

Non-linear step χ

- $a_y \leftarrow a_y + a_{y+1} a_{y+1} \lll 1$ for all y in parallel

Mixing step θ

- $p \leftarrow a_0 + a_1 + a_2 + a_3 + a_4$
- $e \leftarrow p \lll 12 + p \lll 17$
- $a_y \leftarrow a_y + (e \lll y)$ for all y

Transposition step π

- $a_y \leftarrow a_{y+1}$ for all y in parallel

Transposition step ρ

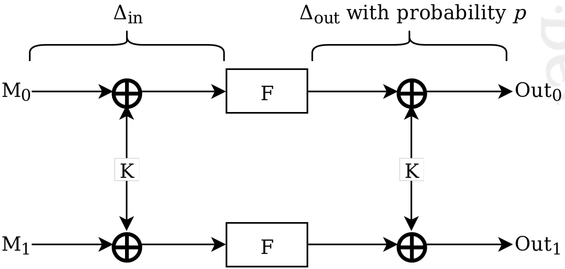
- $a_y \leftarrow a_y \lll r_y$ for all y with $r = (0, 2, 6, 11, 19)$

Differential cryptanalysis

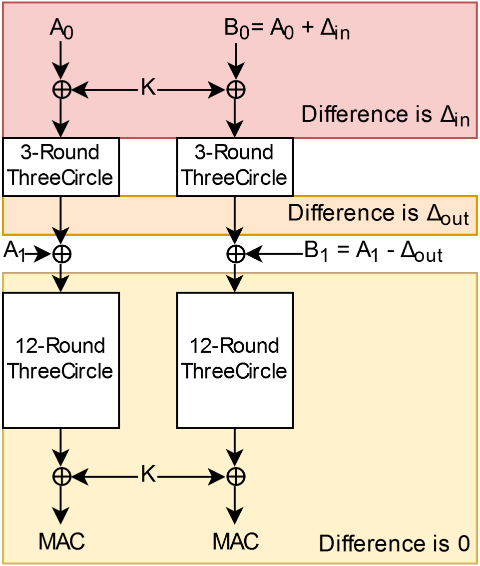
- The study of how differences in information input can affect the resultant difference at the output.
- We want to use it to find colliding MACs.
- Trail: knowledge of how an input difference propagates and the probability of it happening.
- Differential: same as trail, but we do not care about the difference between each round.

Differential cryptanalysis - Differentials

- Assume we have the following differential: $(\Delta_{in}, \Delta_{out})$ with associated differential probability $DP(\Delta_{in}, \Delta_{out}) = p$.
- We start with two messages such that $\Delta_{in} = M_1 - M_0 = (M_1 + K) - (M_0 + K)$
- With probability $p = DP(\Delta_{in}, \Delta_{out})$, we have that: $\Delta_{out} = F(M_1 + K) - F(M_0 + K) = (F(M_1 + K) + K) - (F(M_0 + K) + K)$



Differential cryptanalysis - Finding collisions



How do differences propagate in ThreeCircle?

Non-linear layer

- Each non-zero digit in the differential causes 3 equally likely branches.
- Branches lower the differential probability.

Mixing layer

- Only has effect when there are non-zero digits in differential of the parities.
- Does not cause branching, but causes diffusion (i.e. more non-zero digits in the differential).

Transposition layer

- Moves the digits around.
- Does not cause branching, but spreads the active digits.

Finding trails

- Finding best trails is an open problem.
- Finding a trail that spans 1 round often is doable by hand.
- Finding a trail that spans multiple rounds gets *very* complex *very* fast.
- Essentially an optimization problem: we want to find an input/output difference that has an as high as possible differential probability.
- Problem: search space is *huge*.

Evolutionary algorithms

- An evolutionary algorithm (EA) is a generic population-based optimization algorithm.
- An EA uses mechanisms inspired by biological evolution, such as reproduction, mutation, recombination and selection.
- Candidate solutions play the role of individuals in a population, and the fitness function determines the quality of the solutions.
- It often works well on NP-problems.
- Based on underlying assumptions that are not necessarily true, most notably the building block hypothesis.

Evolutionary algorithms - Implementation

1. Generate the initial population of individuals randomly.
2. Evaluate the fitness of each individual.
3. Repeat until termination:
    - Select the best-fit individuals for reproduction.
    - Breed new individuals through crossover and mutation.
    - Evaluate the individual fitness of new individuals.
    - Replace least–fit population with new individuals.

Evolutionary algorithms - fitness function

- Many design decisions when implementing an EA.
- Perhaps the most important one is the fitness function.
- Single figure of merit that summarizes how close a solution is to achieving a set of aims.
- If chosen poorly, the algorithm will converge on a poor solution or not at all.
- We want to minimize the number of branches $\rightarrow$ minimize the number of active digits going into χ.
- Use sum of weights going to χ. Problem: noisy.

Evolutionary algorithms - Tweaks to make things work

- The fitness function is noisy since many trails are possible for a given Δ_{in}, therefore I run the fitness function many times and return weight of best trail.
- Empirically, I know Δ_{in}'s of good trails tend to have most digits set to 0, so to speed up the process I create the initial population such that most digits are 0.

Results

- I could not find a good trail for 3 round ThreeCircle by hand.
- We anticipated that the best trail would have a DP of $\frac{1}{3^{26}}$.
- I used an EA to find a trail with DP $\frac{1}{3^{16}}$.
- The trail I found should not exist and reveals a weakness in (the rotation constants of) ThreeCircle.

Conclusion

- It might be worth pursuing the use of evolutionary algorithms to find trails for differential cryptanalysis.