A New Design Approach in Symmetric Cryptography

Arnab Roy
University of Innsbruck

ALPSY 2025, Obergurgl

Moving away from functional box

- **Our motivation**: Algebraic (Arithmetization Oriented) design requires polynomial based design approach
- Understand and study the polynomial instantiations in a compact way
- Towards polynomial based construction
  - How to define a (suitable) polynomial system?
  - How to characterise the polynomials defining such a system?
  - How to instantiate?

How do we construct block ciphers?

SPN Network

- Let $f : \mathbb{F}_q \mapsto \mathbb{F}_q$ be *permutation polynomial*

$$\mathcal{S} : \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix} \mapsto \begin{bmatrix} f(x_1) \\ f(x_2) \\ \cdot \\ \cdot \\ f(x_n) \end{bmatrix}$$

- Let $A_{n \times n} \in GL_n(\mathbb{F}_q)$ i.e. an invertible matrix over $\mathbb{F}_q$
- Iterate: $\mathcal{S} \circ A \circ \mathcal{S} \circ \cdots \circ \mathcal{S}$
- HereIgnoring here the key and constant addition (can be included in linear transformation with slight modification)

How do we construct block ciphers?

Fesitel Network

- Let $p : \mathbb{F}_q^n \mapsto \mathbb{F}_q^n$ for $n \geq 1$ be a polynomial (may or may not be permutation)
- Balanced Feistel e.g. $n = 2$
  - Let $F : \mathbb{F}_q^2 \mapsto \mathbb{F}_q^2$ be such that

$$F : \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mapsto \begin{bmatrix} x_1 \\ x_2 + p(x_1) \end{bmatrix}$$

  - Let $A : [x_1 \quad x_2] \mapsto [x_{\sigma(1)} \quad x_{\sigma(2)}]$ where $\sigma \in S_2$ and $\sigma \neq \text{id}$
  - Iterate : $\mathcal{S} \circ A \circ \mathcal{S} \circ \cdots \circ \mathcal{S}$
- Similarly we can define other Feistel Networks (balanced and unbalanced)

Why?

- *Any function over $\mathbb{F}_q$ can be represented with a polynomial*
    - Current approach do not characterise polynomials but study a function w.r.t (known) cryptanalytic properties e.g. *differential and linear* properties
- Start with polynomial-based approach
    - Efficient polynomial evaluation : Low multiplicative complexity (in AO primitives, SCA resilient design)
    - Polynomial with necessary (cryptanalytic) properties
    - Properties of iterating polynomial system
- Aim for an ***algebraically structured*** way

Polynomial based approach

- A topic in Mathematics: polynomial dynamical system (over finite fields)
- Iterative polynomial system (over finite field)
- Example of studied properties
  - Randomness (using **discrepancy** notion)
  - Period (with specific polynomial e.g. $f(x) = x^3 + c$)
  - Degree growth
  - …
- Provides a good starting point

Triangular Dynamical System

- Introduced by Ostafe and Shparlinski (2010)

$$
\begin{aligned}
f_1(x_1, \ldots, x_n) &= x_1 \cdot g_1(x_2, \ldots, x_n) + h_1(x_2, \ldots, x_n) \\
f_2(x_1, \ldots, x_n) &= x_2 \cdot g_1(x_3, \ldots, x_n) + h_1(x_3, \ldots, x_n) \\
&\cdots\cdots \\
&\cdots\cdots \\
f_{n-1}(x_1, \ldots, x_n) &= x_{n-1} \cdot g_{n-1}(x_n) + h_{n-1}(x_n) \\
f_n(x_1, \ldots, x_n) &= x_n
\end{aligned}
$$

- $g_i, f_i \in \mathbb{F}_q[x_1, \ldots, x_n]$ for finite $n \in \mathbb{N}$
- The TDS is defined by $\mathscr{F} = \{f_1, \ldots, f_n\} \subset \mathbb{F}_q[x_1, \ldots, x_n]$

Triangular dynamical system

- Shows polynomial degree growth under iteration
- PRNG with $\mathscr{F}$ was investigated using the discrepancy notion
- Polynomial degree growth $\Longrightarrow$ low discrepancy
- A hash function based on TDS was proposed

Generalised triangular dynamical system

- A generalisation of TDS [Joint work with Matthias Steiner, SAC'24]

$$
\begin{aligned}
f_1(x_1, \ldots, x_n) &= p(x_1) \cdot g_1(x_2, \ldots, x_n) + h_1(x_2, \ldots, x_n) \\
f_2(x_1, \ldots, x_n) &= p(x_2) \cdot g_1(x_3, \ldots, x_n) + h_1(x_3, \ldots, x_n) \\
&\cdots\cdots \\
&\cdots\cdots \\
f_{n-1}(x_1, \ldots, x_n) &= p(x_{n-1}) \cdot g_{n-1}(x_n) + h_{n-1}(x_n) \\
f_n(x_1, \ldots, x_n) &= p(x_n)
\end{aligned}
$$

- Aim: define a permutation with $\mathcal{F}$
- $p_i \in \mathbb{F}_q[x_i]$ are permutations; $g_i, h_i \in \mathbb{F}_q[x_{i+1}, \ldots, x_n]$ are such that g_i have no zeros
- The GTDS is defined by $\mathcal{F} \subset \mathbb{F}_q[x_1, \ldots, x_n]$

Invertibility: polynomial characterisation

- For given $\beta = (\beta_1, \ldots, \beta_n) \in \mathbb{F}_q^n$
- Consider f_i for $i = n, \ldots, 1$
  - $p_n(x_n) = \beta_n \implies x_n = p_n^{-1}(\beta_n)$
  - $p_{n-1}(x_{n-1})g_{n-1}(x_n) + h_{n-1}(x_n) = \beta_{n-1} \implies p_{n-1}(x_{n-1}) = \dfrac{\beta_{n-1} - h_{n-1}(x_n)}{g_{n-1}(x_n)}$
  - And so on
- Finding $g_i \in \mathbb{F}_q(x_{i+1}, \ldots, x_n)$ with no zeros is non-trivial in general
- When q is prime a trivial instantiation is: $g(x) = x^2 + a \cdot x + b$ s.t. $b^2 - 4a$ is non-square modulo q
- More general g_i can be build in from g

GTDS instantiations

- **SPN** and **partial SPN**
  - $g_i = 1, h_i = 0, \forall i$
- **Generalised Feistel**
  - $p_i(x_i) = x_i, g_i = 1$
  - Example
    - Feistel with contracting RF
    - Feistel with expanding RF
    - …
- **Balanced Feistel**
  - Can be composition of more than one $\mathscr{F}$ (with same GTDS structure but different instantiations)

Recall GTDS

$$
\begin{aligned}
f_1(x_1, \ldots, x_n) &= p(x_1) \cdot g_1(x_2, \ldots, x_n) + h_1(x_2, \ldots, x_n) \\
f_2(x_1, \ldots, x_n) &= p(x_2) \cdot g_1(x_3, \ldots, x_n) + h_1(x_3, \ldots, x_n) \\
&\cdots\cdots \\
&\cdots\cdots \\
f_{n-1}(x_1, \ldots, x_n) &= p(x_{n-1}) \cdot g_{n-1}(x_n) + h_{n-1}(x_n) \\
f_n(x_1, \ldots, x_n) &= p(x_n)
\end{aligned}
$$

Other instantiations

- GTDS gives **Horst scheme** [GHRSWW '22, '23]
  - $\begin{bmatrix} x_L \\ x_R \end{bmatrix} \mapsto \begin{bmatrix} x_R \\ x_L \cdot g(x_R) + h(x_R) \end{bmatrix}$ where $g, h \in \mathbb{F}_q[x]$ such that g has no zeros
  - Independent work from us at the same time
- Horst variations: **Griffin** and **Reinforced Concrete**
  - A mapping $\mathbb{F}_p^3 \mapsto \mathbb{F}_p^3$ defined as

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \mapsto \begin{bmatrix} x_1^d \\ x_2 \cdot x_1^2 + a_1 \cdot x_1 + b_1 \\ x_2^2 + a_2 \cdot x_2 + b_2 \end{bmatrix}$$

  - p, a_i, b_i, d are integers such that p is prime, $\gcd(d, p-1) = 1$ and $b_i^2 - 4a_i$ is a non-square modulo p

GTDS: Motivation and consequence

- **Disclaimer** : it was neither the intention nor the motivation to define arbitrary SK primitive with polynomials (and linear transformations)
- **Motivation**
  - Systematically investigate efficient AO primitive constructions
  - Example criteria: Efficient polynomial evaluation (e.g. w.r.t bilinear gates)
  - A polynomial based design approach
- **Consequence**
  - New constructions beyond Feistel, SPN and Lai-Massey, can be derived using GTDS
  - A compact way to study a large set of cryptographic permutations and hash function
  - Cryptanalytic properties in connection with polynomials (more work needed)

New construction from GTDS

Arion (keyed) permutation

- First design utilising GTDS at round level [Joint work with Matthias Steiner and Stefano Trevisani]
- Arion GTDS is defined as

$$f_i(x_1, \ldots, x_n) = x_i^{d_1} \cdot g_i(\sigma_{i+1,n}) + h(\sigma_{i+1,n}) \quad 1 \le i \le n-1$$
$$f_n(x_1, \ldots, x_n) = x^e$$

- Here $\sigma_{i+1,n} = \sum_{j=i+1}^{n} f_j(x_1, \ldots, x_n) + x_j$
- $g_i, h_i \in \mathbb{F}_q[x_{i+1}, \ldots, x_n]$ are degree 2 polynomials such that g_i have no zeros
- q is prime, $1 < d_1, d_2 < q-1$ be integers such that gcd$(d_i, q-1) = 1$ and $e \cdot d_2 = 1 \pmod{q}$

Conclusion

- Open problems
  - Utilising GTDS beyond AO primitives, e.g. over small field
  - More *generic cryptanalysis* of GTDS and tighten cryptanalytic bound
  - Non-trivial degree growth bound
  - Utilising for HW friendly instantiation
  - Extending GTDS for non-invertible systems

THAT YOU!