Algebraic Cryptanalysis of Full Ciminion

Augustin Bariant

Inria, Paris, France
augustin.bariant@inria.fr

Abstract. With the increasing interest for advanced protocols for Multi Party Computation, Fully-Homomorphic Encryption or Zero Knowledge proofs, a need for cryptographic algorithms with new constraints has emerged. These algorithms, called Arithmetization-Oriented ciphers, seek to minimize the number of field multiplications in large finite fields $\mathbb{F}_{2^n}$ or $\mathbb{F}_p$. Among them, Ciminion is an encryption algorithm proposed by Dobraunig *et al.* in Eurocrypt 2021.
In this paper, we show a new univariate modelization on a variant of Ciminion proposed by the designers. This instance restricts the attacker to at most $2^{s/2}$ data, where s is the security level. Because the designers chose to reduce the number of rounds in that specific attacker model, we are able to attack the cipher for large security levels. We also propose some slight modifications of Ciminion that would overcome this vulnerability.

1 Introduction

Some recent advanced protocols, such as Multi-Party Computation (MPC), Fully-Homomorphic Encryption (FHE) or Zero-Knowledge proofs (ZK), have become the object of attention in modern cryptography. Some ZK-proof systems and MPC protocols operate on large finite fields $\mathbb{F}_q$ with q prime or power of 2 [BCCT12,DPSZ12,DZ13,BBHR18]. In these protocols, the communication cost depends roughly on the number of field multiplications required by the proof or the MPC function to evaluate. As such, standard implementations need to be converted into sequences of finite field operations, with a preferably low number of multiplications. Multiple works were conducted to reduce the cost of the AES implementation in such protocols [DLT14,GRR$^+$16].

In 2015, LowMC was the first attempt to propose a design aiming at minimizing the number of boolean multiplications [ARS$^+$15]. In 2016, Albretch *et al.* designed MiMC [AGR$^+$16], a family of cryptographic functions operating directly on the native finite field $\mathbb{F}_q$ of the protocol, which significantly reduced the number of field multiplications. This paved the way for a new class of cryptographic functions: *Arithmetization-Oriented* ciphers, operating on $\mathbb{F}_q$ with large q. Multiple such ciphers were later introduced, such as Jarvis [AD18], Vision, Rescue [AAB$^+$20], Poseidon [GKR$^+$21], Ciminion [DGGK21], Anemoi [BBC$^+$22], Griffin [GHR$^+$22] and Hydra [GØSW23]. All *Arithmetization-Oriented* ciphers use finite field multiplications to provide non-linearity, which have strong differential and linear properties. For instance, the mapping $x \rightarrow x^3$ has a differential uniformity of 2 on the large field. Consequently, statistical attacks often perform poorly against *arithemization-oriented* ciphers and the most threatening attacks become algebraic attacks, as highlighted by different works [ACG$^+$19,BBLP22]. In particular, Ciminion was subject to algebraic cryptanalysis in several works: Bariant *et al.* show an algebraic representation of the cipher breaking security claims for very large security levels [BBLP22]. Later, Zhang *et al.* provide a cryptanalysis of Ciminion against higher order differential and integral attacks [ZLLL23]. In addition, they show that under weak round constants, a subkey recovery attack can be mounted on Aiminion, an aggressive evolution of Ciminion.

Contribution In this paper, we present an algebraic attack against a variant of Ciminion using univariate solving. This attack breaks the security claims of the designers for large security levels. We then suggest a patch to apply to Ciminion in order to avoid the attack.

Outline Section 2 presents the notations and describes the cipher Ciminion. Section 3 explains algebraic attacks. Section 4 presents our univariate attack against Ciminion. We eventually present in Section 5 some Ciminion modifications for better security confidence.

2 Preliminaries

2.1 Notations

$2 \leq \omega \leq 2.372$ is the matrix multiplication exponent. In this paper, q is either a large prime or a power of 2. $\mathbb{F}_q$ denotes the finite field with q elements.

2.2 Ciminion

Ciminion is an Arithmetization-Oriented encryption scheme presented at Eurocrypt 2021 [DGGK21]. Ciminion operates on large fields $\mathbb{F}_q$ with $q \geq 2^{64}$. The state of Ciminon is composed of three elements of $\mathbb{F}_q$. The specifications of Ciminion are presented on figures 1,2 and 3. Two permutations are employed: p_C and p_E of respectively N and R rounds. The round function for round i is denoted f_i. It uses four round constants RC_l, with $l = i$ for p_C , and $l = i + N - R$ for p_E. $RC4$ is assumed to be different from 0 or 1. The round function and the rol function are based on Toffoli gates and are of degree 2. Because both these functions use Toffoli gates rather than low degree Sboxes, the inverse of these functions are also of degree 2.

Ciminion truncates one output element of each permutation p_E, to prevent the recovery of intermediate states. Since the guess of a truncated element allows to recover the intermediate states and thus the round keys K_1 and K_2, the security of Ciminion cannot exceed $\log_2(q)$ by design. The designers claim a

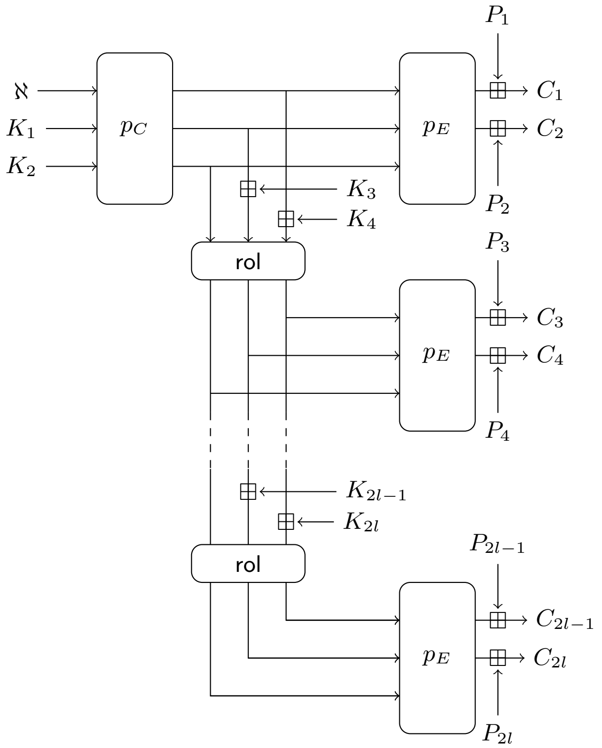

Fig. 1. The Ciminion Encryption over $\mathbb{F}_p$ (replace $+$ by $\oplus$ over $\mathbb{F}_{2^n}$). The figure is taken from [BBLP22].

security level of $64 \leq s \leq \log_2(q)$ for three instances presented in Table 1, where the number of rounds depends on the security level s.

As it is the case in standard Arithmetization-Oriented ciphers, statistical attacks perform poorly because of the strong cryptographic properties (e.g. linear, differential ...) of the multiplication in large fields. The number of rounds of Ciminion was therefore chosen to provide security against algebraic attacks, by insuring that the degree of the involved functions is too high for any algebraic attack.

3 Algebraic Attacks

In this section, we describe two types of algebraic attacks discussed in the paper: polynomial solving and interpolation attacks.

3.1 Attacks using polynomial solving

An algebraic attack using polynomial solving on a cipher can be decomposed into two main steps:

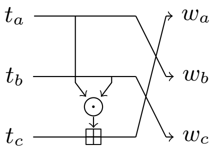

Fig. 2. rolling function rol.

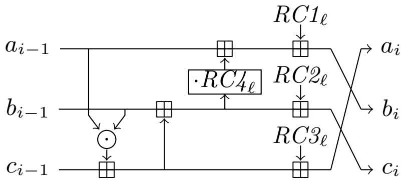

Fig. 3. Ciminion round function f_i.

Instance	p_C rounds	p_E rounds
Standard	$s+6$	$\lceil \frac{s+37}{12} \rceil$
Data limit $2^{s/2}$	$\lceil \frac{2(s+6)}{3} \rceil$	$\lceil \frac{s+37}{12} \rceil$
Conservative	$s+6$	$\lceil \frac{3}{2} \cdot \frac{s+37}{12} \rceil$

Table 1. Number of rounds of p_C and p_E for the instances proposed by the designers of Ciminion, for a security level of $64 \leq s \leq q$.

- **Modelization:** The attacker represents the cipher with a system of polynomial equations, such that a solution of the polynomial system contains secret data, such as the encryption key or an internal state. In the case of hash functions, a solution could for example contain information on a preimage or a solution to the CICO problem.
- **System solving:** The attacker then solves the polynomial system using state-of-the-art techniques, such as polynomial root finding algorithms for univariate polynomials or Groebner basis algorithms for multivariate systems.

Both steps should be analyzed carefully when mounting an algebraic attack. On the one hand, the modelization step is highly cipher dependant, and heavily impacts the complexity of the attack. Some ciphers might possess different modelizations with different solving time complexities, such as Anemoi [BBC+22] or Ciminion. Efficient modelizations are found through cryptanalysis. On the other hand, the system solving step often relies on existing generic algorithms, such as Groebner basis based algorithms.

Let us denote n the number of variables of the polynomial system that we want to solve. For simplicity, and since it corresponds to our attacks, we will only consider *well-defined* systems, that is with as many equations as variables:

$$\begin{cases} P_1(X_1, \dots X_n) = 0 \\ P_2(X_1, \dots X_n) = 0 \\ \qquad\qquad \vdots \\ P_n(X_1, \dots X_n) = 0 \end{cases}$$

The complexity of polynomial solving highly depends on the structure of the equations. For simplicity, we will consider that the system of polynomial equations behaves like a random polynomial system. The Fröberg conjecture states that such a polynomial system is regular [Frö85], ie for all i, $gP_i \in \langle P_1, \ldots, P_{i-1} \rangle \Rightarrow g \in \langle P_1, \ldots, P_{i-1} \rangle$. The regularity of polynomial systems is a property that allows to accurately estimate the complexity of system solving. Polynomial systems representing ciphers are often unregular [DG10,DY13,Sau22], but it is standard to study the complexity of solving regular systems for estimating an algebraic attack complexity. Let us denote d_i the degree of P_i and:

$$D = 1 + \sum_{i=1}^{n} d_i - 1 \qquad\qquad d = \prod_{i=1}^{n} d_i$$

For more detailed insights on these parameters, we refer to previous works, such as [BBLP22].

Depending on the value of n, the solving technique varies:

$\boldsymbol{n = 1}$ In that case, the system consists of a single univariate polynomial P of degree $d = d_0$. Using fast FFT-based polynomial multiplications which cost $O(d \log(d) \log(\log(d)))$ field operations, we can find the roots of the polynomial P in $\mathbb{F}_q$ with $\mathcal{O}(d \log(d)(\log(d) + \log(q)) \log(\log(d)))$ field multiplications, with the following steps:

1. Compute $Q = X^q - X \mod P$. This requires $\log(q)$ multiplications of polynomials of degree d, for a total complexity of $O(d \log(q) \log(d) \log(\log(d)))$ field operations.
2. Compute $R = \gcd(P, Q)$. One can easily note that $R = \gcd(P, X^q - X)$, therefore the roots of R are also roots of P and are all in $\mathbb{F}_q$. This costs $O(d \log^2(d) \log(\log(d)))$ field operations.
3. Factor R. In general, R is of very low degree because P has few roots in the field, so this step has a negligible complexity.

$\boldsymbol{n = 2}$ Compute the resultant of the two bivariate polynomials, a univariate polynomial of degree $d = d_1 d_2$. Computing the resultant costs $d \times (\min d_1, d_2)^{1-1/\omega}$, using fast linear algebra [Vil18]. The roots of the resultant are exactly the values x such that there exists a y such that (x, y) is a solution of the system. Using univariate root finding, computing solutions from the resultant then costs $\mathcal{O}(d \log(d)(\log(d) + \log(q)) \log(\log(d)))$.

$\boldsymbol{n \geq 3}$ The most common and fastest way to solve this type of polynomial system is by computing a Groebner basis of the system. It is composed of three steps:

1. Compute the Groebner basis of the polynomial system in the *graded reverse lexicographic (grevlex)* order. To this day, the fastest algorithm is Faugère F5 algorithm [Fau02]. It runs in $\mathcal{O}(\binom{D_{reg}}{n+D_{reg}}^{\omega})$.

2. Convert the *grevlex* Groebner basis into a *lexicographic (lex)* Groebner basis, using an order change algorithm. The most famous one is FGLM algorithm [FGLM93]. Recent works have however improved the FGLM algorithm using Hermite Normal Forms [BNS22]. Since the sparsity of the the multiplication matrix is $O(d^{1-1/n})$ [FM17], the complexity of this step is $\mathcal{O}(d^{\omega-\omega/n+1/n})$, improving on previous complexity esimations of FGLM variants [FM17,FGHR14].
3. Given the computed *lex* Groebner basis, compute the solutions in the field using successive univariate polynomial solving and variable replacements. This step is of negligible complexity compared to the two other steps in practice.

In the context of encryption schemes, this type of attack usually requires a low number of data, since one plaintext/ciphertext pair gives sufficient information to recover few candidates for the key.

3.2 Interpolation attack

The interpolation attack represents some data accessible to the attacker (e.g. keystream, ciphertext...) as a polynomial of secret data (key, inner state...) and other data known by the attacker (nonce, plaintext...). By generating enough data, the attacker can perform a Lagrange interpolation to retrieve the key-dependant coefficients of the polynomial, and eventually retrieve key information.

Because of this attack, the number of rounds is chosen so that the degree of the involved polynomial exceeds 2^s. Recall that the round function is of degree 2, and r rounds of the function is roughly of degree 2^{r-2} from any input to any output. For extra security, the designers chose the number of rounds of p_C (N) to be more $s+6$ in the standard attacker model. The drawback of this attack is the large number of data needed by the attacker to mount the attack. Because of this very reason, in their analysis, the designers of Ciminion suggested that if the attacker is limited to $2^{s/2}$ cipher calls, they can relax the constraint on the number of rounds in a manner that ensures that the degree of the polynomial still exceeds $2^{s/2}$. They proposed a *limited data* variant of Ciminion for this case. As a security margin, they chose respectively $\lceil \frac{2(s+6)}{3} \rceil$ and $\lceil \frac{s+37}{12} \rceil$ as the number of rounds of p_C and p_E. In this settings, the polynomial $P_K(\aleph)$ is of degree $2^{\lceil \frac{2(s+6)}{3} \rceil + \lceil \frac{s+37}{12} \rceil - 1}$, which is less than 2^s for large security levels s.

4 Our attack using univariate polynomial solving

This attack breaks the security claims of the designers in the *limited data* variant, in which the attacker can not query more than $2^{s/2}$ data. This attack is a known-plaintext attack using one data of two key stream blocks S_1 and S_2. It is an equivalent key-recovery attack in the sense that it allows to recover an arbitrary number of subkeys. Because the subkey generation is performed from a strong sponge function, recovering the master key from the subkeys is difficult and not performed here. The variable X represents the third truncated output of the first final permutation layer p_E.

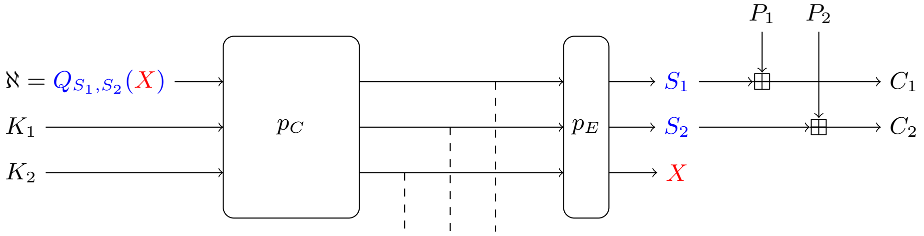

Fig. 4. Our modelization of Ciminion.

Recovery of K_1 and K_2. We represent the nonce $\aleph$ as a polynomial on X. The nonce $\aleph$ and X are separated by $\lceil\frac{2(s+6)}{3}\rceil + \lceil\frac{s+37}{12}\rceil \leq \frac{3s}{4} + 10$ rounds. As shown in previous papers [ZLLL23,BBLP22], the first input element of a r-round Ciminion permutation as a function of the output elements is of degree 2^{r-1}. This implies that Q_{S_1,S_2} is of degree $2^{\lceil\frac{2(s+6)}{3}\rceil+\lceil\frac{s+37}{12}\rceil-1} \approx 2^{0.75s+6.1}$. The truncated element X is a root of the polynomial $Q_{S_1,S_2} - \aleph$. Recovering X allows to recover K_1 and K_2 by inverting p_E and p_C from the output (S_1, S_2, X).

The univariate root finding algorithm being quasi-linear in the degree of the polynomial, it implies that for large security levels s, we expect the root finding algorithm to run faster than 2^s operations. Let us take the case of $s = log_2(q) = 256$. Q_{S_1,S_2} is of degree $2^{174+25-1} = 2^{198}$, and the root finding algorithm is of complexity $\mathcal{O}(d\log(d)(\log(d) + \log(q))\log(\log(d))) \approx c \times 2^{198} \times 198 \times (198 + 256) \times 7.6 \approx c \times 2^{217.4}$ field operations, where c is the constant behind the $\mathcal{O}$.

In the case of $s = log(q) = 128$, Q_{S_1,S_2} is of degree $2^{90+14-1} = 2^{103}$. The root finding algorithm costs approximately $\mathcal{O}(d\log(d)(\log(d) + \log(q))\log(\log(d))) = c \times 2^{103} \times 103 \times (103 + 128) \times 6.7 \approx c \times 2^{120.3}$ field operations.

Supposing that a Ciminion encryption costs approximately c operations, this attack breaks the security claims for $s = \log(q) \geq 93$.

Recovery of K_i for $i \geq 3$ After the recovery of K_1 and K_2, we query further keystream elements S_i for $i \geq 3$, under the same nonce. The inner state before the first p_E is known, as depicted in green in Figure 5. We denote Y the truncated output of the second permutation p_E. We can compute the polynomial $\tilde{Q}_{S_3,S_4}(Y)$ representing the first inner state element. This inner state element is not dependant on K_3 and K_4 and is known from the recovery of K_1 and K_2. We denote it α. The truncated output of the second p_E is a root of $\tilde{Q}_{S_3,S_4}(Y) - \alpha$, which is a polynomial of degree $2^{r_E} + 2^{r_E-1} \approx 2^{r_E+0.6}$ where $r_E = \lceil\frac{s+37}{12}\rceil$ is the number of rounds of p_E. The recovery of Y is of negligible complexity compared to the first step. This allows to recover the inner state before the second p_E permutation and therefore to recover K_3 and K_4. Ultimately, (K_{2i+1}, K_{2i+2}) for $i \geq 2$ can be recovered in a similar manner if the keystream is long enough.

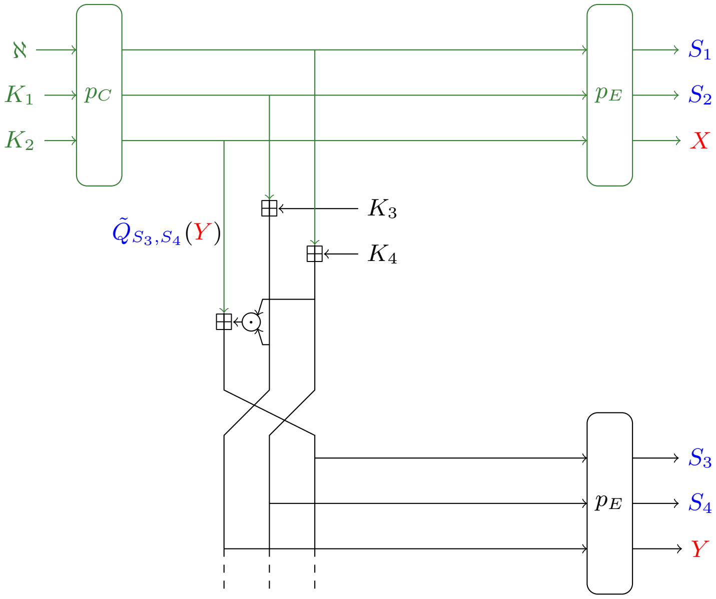

Fig. 5. Recovery of keystream elements K_3 and K_4. The green wires denote the known internal state elements from the recovery of K_1 and K_2.

Aiminion Aiminion is an aggressive evolution of Ciminion presented in appendix of the original Ciminion paper [DGGK21]. Since it uses a key addition before the keystream, it is impossible to express the nonce only with the keystreams S_1, S_2 and the truncated element X. Instead, the unkown subkeys K_3 and K_4 would be involved in the formula. We did not manage to overcome this difficulty to mount an attack.

5 Suggested modifications of Ciminion

Protection against the univariate solving attack The univariate solving attack presented in Section 4 relies on the backward computation of p_C from the single guess of X. To provide protection against this attack, a cheap modification is to perform a feedforward after p_C, by XORing at least one key elements K_0, K_1 or both to one or several outputs of p_C. A variant of this feedforward would be to add new key elements to the output of p_C, although this costs extra key scheduling. This way, it is no longer possible to compute p_C backward with the sole knowledge of X.

Acknowledgement This work was supported by the French DGA.

References

AAB$^+$20. Abdelrahaman Aly, Tomer Ashur, Eli Ben-Sasson, Siemen Dhooghe, and Alan Szepieniec. Design of symmetric-key primitives for advanced cryptographic protocols. *IACR Transactions on Symmetric Cryptology*, 2020(3):1–45, 2020.

ACG$^+$19. Martin R. Albrecht, Carlos Cid, Lorenzo Grassi, Dmitry Khovratovich, Reinhard Lüftenegger, Christian Rechberger, and Markus Schofnegger. Algebraic cryptanalysis of STARK-friendly designs: Application to MARVELlous and MiMC. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology – ASIACRYPT 2019, Part III*, volume 11923 of *Lecture Notes in Computer Science*, pages 371–397, Kobe, Japan, December 8–12, 2019. Springer, Heidelberg, Germany.

AD18. Tomer Ashur and Siemen Dhooghe. MARVELlous: a STARK-friendly family of cryptographic primitives. Cryptology ePrint Archive, Report 2018/1098, 2018. `https://eprint.iacr.org/2018/1098`.

AGR$^+$16. Martin R. Albrecht, Lorenzo Grassi, Christian Rechberger, Arnab Roy, and Tyge Tiessen. MiMC: Efficient encryption and cryptographic hashing with minimal multiplicative complexity. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 191–219, Hanoi, Vietnam, December 4–8, 2016. Springer, Heidelberg, Germany.

ARS$^+$15. Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 430–454, Sofia, Bulgaria, April 26–30, 2015. Springer, Heidelberg, Germany.

BBC$^+$22. Clémence Bouvier, Pierre Briaud, Pyrros Chaidos, Léo Perrin, and Vesselin Velichkov. Anemoi: Exploiting the link between arithmetization-orientation and CCZ-equivalence. Cryptology ePrint Archive, Report 2022/840, 2022. `https://eprint.iacr.org/2022/840`.

BBHR18. Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable, transparent, and post-quantum secure computational integrity. Cryptology ePrint Archive, Report 2018/046, 2018. `https://eprint.iacr.org/2018/046`.

BBLP22. Augustin Bariant, Clémence Bouvier, Gaëtan Leurent, and Léo Perrin. Algebraic attacks against some arithmetization-oriented primitives. *IACR Transactions on Symmetric Cryptology*, 2022(3):73–101, 2022.

BCCT12. Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In Shafi Goldwasser, editor, *ITCS 2012: 3rd Innovations in Theoretical Computer Science*, pages 326–349, Cambridge, MA, USA, January 8–10, 2012. Association for Computing Machinery.

BNS22. Jérémy Berthomieu, Vincent Neiger, and Mohab Safey El Din. Faster change of order algorithm for gröbner bases under shape and stability assumptions. In *Proceedings of the 2022 International Symposium on Symbolic and Algebraic Computation*, pages 409–418, 2022.

DG10. Vivien Dubois and Nicolas Gama. The degree of regularity of HFE systems. In Masayuki Abe, editor, *Advances in Cryptology – ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 557–576, Singapore, December 5–9, 2010. Springer, Heidelberg, Germany.

DGGK21. Christoph Dobraunig, Lorenzo Grassi, Anna Guinet, and Daniël Kuijsters. Ciminion: Symmetric encryption based on Toffoli-gates over large finite fields. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021, Part II*, volume 12697 of *Lecture Notes in Computer Science*, pages 3–34, Zagreb, Croatia, October 17–21, 2021. Springer, Heidelberg, Germany.

DLT14. Ivan Damgård, Rasmus Lauritsen, and Tomas Toft. An empirical study and some improvements of the MiniMac protocol for secure computation. In Michel Abdalla and Roberto De Prisco, editors, *SCN 14: 9th International Conference on Security in Communication Networks*, volume 8642 of *Lecture Notes in Computer Science*, pages 398–415, Amalfi, Italy, September 3–5, 2014. Springer, Heidelberg, Germany.

DPSZ12. Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 643–662, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Heidelberg, Germany.

DY13. Jintai Ding and Bo-Yin Yang. Degree of regularity for HFEv and HFEv-. In Philippe Gaborit, editor, *Post-Quantum Cryptography - 5th International Workshop, PQCrypto 2013*, pages 52–66, Limoges, France, June 4–7, 2013. Springer, Heidelberg, Germany.

DZ13. Ivan Damgård and Sarah Zakarias. Constant-overhead secure computation of Boolean circuits using preprocessing. In Amit Sahai, editor, *TCC 2013: 10th Theory of Cryptography Conference*, volume 7785 of *Lecture Notes in Computer Science*, pages 621–641, Tokyo, Japan, March 3–6, 2013. Springer, Heidelberg, Germany.

Fau02. Jean Charles Faugere. A new efficient algorithm for computing gröbner bases without reduction to zero (f 5). In *Proceedings of the 2002 international symposium on Symbolic and algebraic computation*, pages 75–83, 2002.

FGHR14. Jean-Charles Faugère, Pierrick Gaudry, Louise Huot, and Guénaël Renault. Sub-cubic change of ordering for gröbner basis: a probabilistic approach. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, pages 170–177, 2014.

FGLM93. Jean-Charles Faugere, Patrizia Gianni, Daniel Lazard, and Teo Mora. Efficient computation of zero-dimensional gröbner bases by change of ordering. *Journal of Symbolic Computation*, 16(4):329–344, 1993.

FM17. Jean-Charles Faugère and Chenqi Mou. Sparse fglm algorithms. *Journal of Symbolic Computation*, 80:538–569, 2017.

Frö85. Ralf Fröberg. An inequality for hilbert series of graded algebras. *Mathematica Scandinavica*, 56(2):117–144, 1985.

GHR$^+$22. Lorenzo Grassi, Yonglin Hao, Christian Rechberger, Markus Schofnegger, Roman Walch, and Qingju Wang. A new feistel approach meets fluid-SPN: Griffin for zero-knowledge applications. Cryptology ePrint Archive, Report 2022/403, 2022. `https://eprint.iacr.org/2022/403`.

GKR$^+$21. Lorenzo Grassi, Dmitry Khovratovich, Christian Rechberger, Arnab Roy, and Markus Schofnegger. Poseidon: A new hash function for zero-knowledge proof systems. In Michael Bailey and Rachel Greenstadt, editors, *USENIX Security 2021: 30th USENIX Security Symposium*, pages 519–535. USENIX Association, August 11–13, 2021.

GØSW23. Lorenzo Grassi, Morten Øygarden, Markus Schofnegger, and Roman Walch. From farfalle to megafono via ciminion: The PRF hydra for MPC applications. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part IV*, volume 14007 of *Lecture Notes in Computer Science*, pages 255–286, Lyon, France, April 23–27, 2023. Springer, Heidelberg, Germany.

GRR$^+$16. Lorenzo Grassi, Christian Rechberger, Dragos Rotaru, Peter Scholl, and Nigel P Smart. Mpc-friendly symmetric key primitives. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 430–443, 2016.

Sau22. Jan Ferdinand Sauer. Gröbner basis-attacking a tiny sponge. Available online at `https://asdm.gmbh/2021/06/28/gb_experiment_summary/`; retrieved on August 19, 2022, 2022.

Vil18. Gilles Villard. On computing the resultant of generic bivariate polynomials. In *Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation*, ISSAC '18, page 391–398, New York, NY, USA, 2018. Association for Computing Machinery.

ZLLL23. Lulu Zhang, Meicheng Liu, Shuaishuai Li, and Dongdai Lin. Cryptanalysis of ciminion. In Yi Deng and Moti Yung, editors, *Information Security and Cryptology*, pages 234–251, Cham, 2023. Springer Nature Switzerland.