Improved Resultant Attack against Arithmetization-Oriented Primitives

Augustin Bariant[1], Aurélien Boeuf[2], Pierre Briaud[3],
Maël Hostettler[4], Morten Øygarden[3], Håvard Raddum[3]

[1]ANSSI [2]INRIA,
[3]Simula UiB [4]Télécom SudParis

August 2025
CRYPTO2025, Santa Barbara

Arithmetization-Oriented (AO) primitives

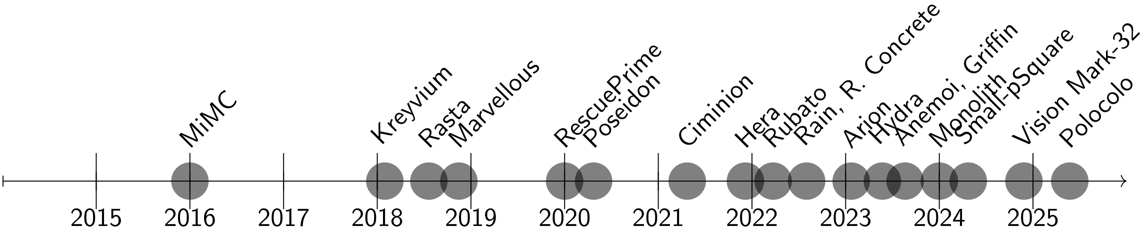
Traditional primitives

- Designed for bit-oriented platforms.
- Operate on bit sequences.
- Low resource consumption (time, etc.).
- Several decades of cryptanalysis.

Arithmetization-Oriented primitives

- Designed for advanced protocols.
- Operate on large finite field elements.
- Low number of field multiplications.
- $\leq$ 10 years of cryptanalysis.

Non-exhaustive timeline based on stap-zoo.com:



Targets of this paper

We focus on these hash functions

Anemoi	Griffin	ArionHash	Rescue-Prime
Crypto2023	Crypto2023	arXiv	ePrint2020

Targets of this paper

We focus on these hash functions

Anemoi	Griffin	ArionHash	Rescue-Prime
Crypto2023	Crypto2023	arXiv	ePrint2020

Anemoi, Griffin, ArionHash: Improved full-round attacks

Rescue-Prime: $\Downarrow$ First full-round break of an instance

Targets of this paper

We focus on these hash functions

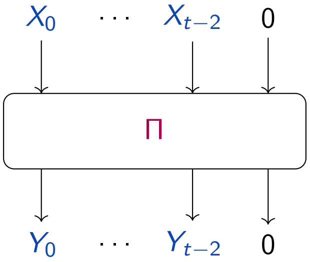
Anemoi	Griffin	ArionHash	Rescue-Prime
Crypto2023	Crypto2023	arXiv	ePrint2020
Improved full-round attacks			⇓ First full-round break of an instance

- Based on the Sponge construction
- We target the underlying permutation of each hash function
- Many different instances for each permutation family
- Based on SBoxes of the form
  - $x \to x^{\alpha}$
  - $x \to x^{1/\alpha}$

Cryptanalysis of AO permutations

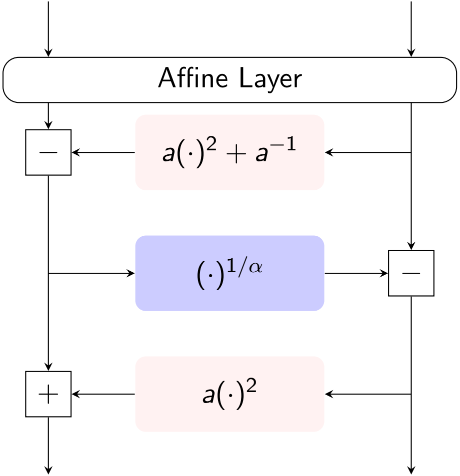
CICO-1 problem against AO permutations on $\mathbb{F}_q^t$

Find $(X_0, \dots X_{t-2}, Y_0, \dots Y_{t-2}) \in \mathbb{F}_q^{2t-2}$ s.t. $\Pi(X_0, \dots X_{t-2}, 0) = (Y_0, \dots Y_{t-2}, 0)$.



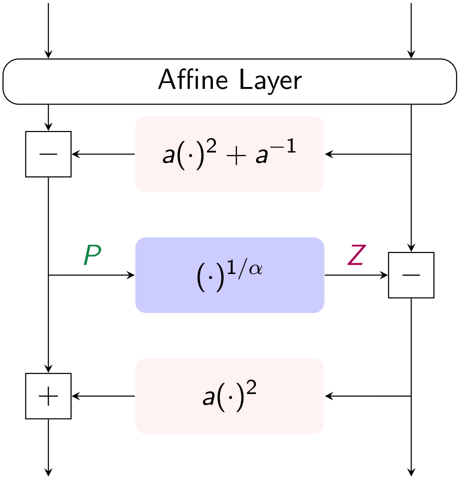
- For a sponge of capacity 1, solving a `CICO-1` gives a collision to the hash function.
- Foundation to further study generic `CICO-`c problem.
- Best attacks against primitves using SBox of the form $x \to x^{1/\alpha}$: algebraic attacks.
  - Freelunch attack [BBL+, CRYPTO'24]
  - Resultant attack [YZY+, AC'24]

Example: Anemoi-π round function



Example: Anemoi-π round function

$P^{1/\alpha} = Z$

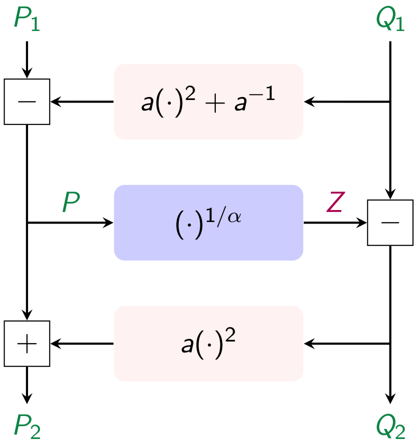


$P = Z^{\alpha}$

$(.)^{1/\alpha}$ is the only high degree operation $\Rightarrow$ One extra variable per round

Example: In detail construction

Focus on the non-linear layer :



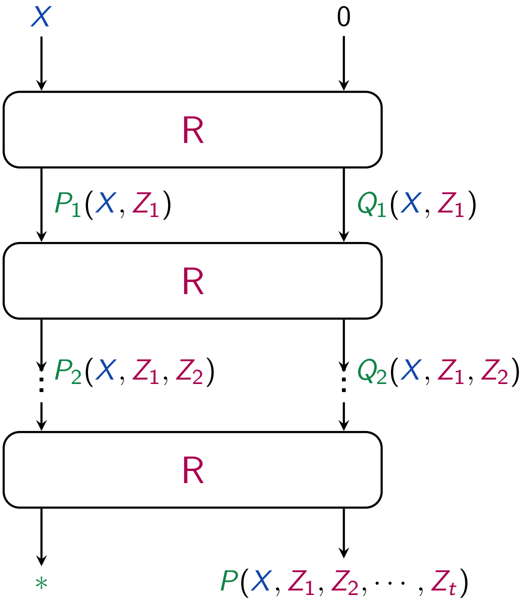
Construction for one round

- $P = P_1 - aQ_1{}^2 - a^{-1} = Z^\alpha$

$$\begin{cases} P_2 = P_1 - aQ_1{}^2 - a^{-1} + a(Q_1 - Z)^2 \\ Q_2 = Q_1 - Z \\ Z^\alpha = P_1 - aQ_1{}^2 - a^{-1} \end{cases}$$

- One extra equation of degree α in Z
- Low algebraic degree of each variable

Iterative construction



$$\begin{cases} {Z_1}^{\alpha} - U_1(X) = 0 \\ {Z_2}^{\alpha} - U_2(X, Z_1) = 0 \\ {Z_3}^{\alpha} - U_3(X, Z_1, Z_2) = 0 \\ \vdots \\ {Z_t}^{\alpha} - U_t(X, Z_1, Z_2, \cdots, Z_{t-1}) = 0 \\ P(X, Z_1, Z_2, \ldots, Z_{t-1}, Z_t) = 0 \end{cases}$$

Properties

- Highly-structured system
- 0-dimensional ideal
- We can construct U_i s.t $\deg_{Z_j}(U_i) < \alpha$

Gradual reduction / Construction cost

Generic complexities

- Univariate polynomial multiplication
  Given $P, Q \in \mathbb{F}_p[X]$ s.t $\deg(P), \deg(Q) \leq d$
  Computing PQ costs $\mathcal{M}(d) = \mathcal{O}(d \log(d) \log(\log d))$ by FFT.
- Multivariate polynomial multiplication (Kronecker trick)
  Given $P, Q \in \mathbb{F}_p[X_1, \cdots, X_n]$ s.t $\deg_{X_i}(P) = \alpha_i$ and $\deg_{X_i}(Q) = \beta_i$
  Computing $P \times Q$ costs $\mathcal{M}\left(\prod_{i=1}^{n}(\alpha_i + \beta_i + 1)\right)$

Gradual reduction / Construction cost

Generic complexities

- Univariate polynomial multiplication
  Given $P, Q \in \mathbb{F}_p[X]$ s.t $\deg(P), \deg(Q) \leq d$
  Computing PQ costs $\mathcal{M}(d) = \mathcal{O}(d \log(d) \log(\log d))$ by FFT.
- Multivariate polynomial multiplication (Kronecker trick)
  Given $P, Q \in \mathbb{F}_p[X_1, \cdots, X_n]$ s.t $\deg_{X_i}(P) = \alpha_i$ and $\deg_{X_i}(Q) = \beta_i$
  Computing $P \times Q$ costs $\mathcal{M}\left(\prod_{i=1}^{n}(\alpha_i + \beta_i + 1)\right)$

Consequences

- Chaining multivariate multiplications is costly: $deg_{X_i}(PQ) = deg_{X_i}(P) + deg_{X_i}(Q)$
- Solution : gradual reduction after each multiplication s.t $\deg_{X_i}(PQ) < \alpha$ for $i > 1$

Gradual reduction / Construction cost

Multiplication/Reduction

Given $P, Q \in \mathbb{F}_p[X, Z_1, \cdots, Z_n]$ s.t $\deg_{Z_i}(P) < \alpha$ and an ideal $\mathcal{P}_n = \{Z_i^\alpha - U_i(X, Z_1, \cdots, Z_i) \mid i \in [\![1, n]\!]\}$

- Compute $PQ \mod (\mathcal{P}_n)$ i.e in $\mathbb{F}_p[X, Z_1, \cdots, Z_n]/(\mathcal{P}_n)$
- We first compute PQ in $\mathbb{F}_p[X, Z_1, \cdots, Z_n]$ so $\deg_{Z_i}(PQ) \leq 2\alpha - 2$
- We then use a specialized recursive reduction algorithm with complexity $\tilde{\mathcal{O}}\left(d_x(2\alpha - 1)^n\right)$ to reduce the n-variate polynomial s.t d_x is the largest X-degree among the polynomials manipulated in the algorithm

Gradual reduction / Construction cost

Multiplication/Reduction

Given $P, Q \in \mathbb{F}_p[X, Z_1, \cdots, Z_n]$ s.t $\deg_{Z_i}(P) < \alpha$ and an ideal $\mathcal{P}_n = \{Z_i^{\alpha} - U_i(X, Z_1, \cdots, Z_i) \mid i \in [\![1, n]\!]\}$

- Compute $PQ \mod (\mathcal{P}_n)$ i.e in $\mathbb{F}_p[X, Z_1, \cdots, Z_n]/(\mathcal{P}_n)$
- We first compute PQ in $\mathbb{F}_p[X, Z_1, \cdots, Z_n]$ so $\deg_{Z_i}(PQ) \leq 2\alpha - 2$
- We then use a specialized recursive reduction algorithm with complexity $\tilde{\mathcal{O}}\left(d_x(2\alpha - 1)^n\right)$ to reduce the n-variate polynomial s.t d_x is the largest X-degree among the polynomials manipulated in the algorithm

$\Rightarrow$ Overall attack in roughly $\tilde{\mathcal{O}}\left(d_I(2\alpha - 1)^n\right)$
n new variables and d_I the degree of the ideal.

What are resultants?

Definition (Resultants)

Let R be a ring and $P(x) = \sum_{i=0}^{d} a_i x^i \in R[x]$ and $Q(x) = \sum_{i=0}^{d'} b_i x^i \in R[x]$

$$res(P, Q) = \begin{vmatrix} a_0 & a_1 & \dots & a_d & 0 & \dots & 0 \\ 0 & a_0 & a_1 & \dots & a_d & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & a_0 & a_1 & \dots & a_d \\ b_0 & b_1 & b_2 & \dots & b_{d'} & 0 & \dots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \ddots & 0 & b_0 & b_1 & b_2 & \dots & b_{d'} \end{vmatrix}$$

Why resultants?

Most important property of resultants

For $P, Q \in \mathbb{K}[x]$ where $\mathbb{K}$ is a field

$$res(P(x), Q(x)) = 0 \iff \gcd(P(x), Q(x)) \neq 1$$

- $P(x)$ and $Q(x)$ might have a common root
- In general, $res(P, Q) = 0 \iff P$ and Q have a non-trivial common factor

Why resultants?

Most important property of resultants

For $P, Q \in \mathbb{K}[x]$ where $\mathbb{K}$ is a field

$$res(P(x), Q(x)) = 0 \iff \gcd(P(x), Q(x)) \neq 1$$

- $P(x)$ and $Q(x)$ might have a common root
- In general, $res(P, Q) = 0 \iff P$ and Q have a non-trivial common factor

Example: solving bivariate system

Let $P, Q \in \mathbb{F}_q[x, y]$

- P and Q : polynomials in y with coefficients in $\mathbb{F}_q[x]$, i.e. $P, Q \in \mathbb{F}_q[x][y]$
- Find a root $\alpha \in \mathbb{F}_q$ of $res(P, Q)$
  - $res(P, Q)(\alpha) = 0$, so $\gcd(P(\alpha, y), Q(\alpha, y)) \neq 1$ (as polynomials in y)
- There probably exists a common root $\beta \in \mathbb{F}_q$ s.t. $P(\alpha, \beta) = Q(\alpha, \beta) = 0$

Solving generic polynomial systems with resultants

$$\mathcal{P} = \begin{cases} P_1(x_1, \dots x_n) = 0 \\ \vdots \\ P_n(x_1, \dots x_n) = 0 \end{cases}$$

Idea: Eliminate the variable x_n and produce $n-1$ polynomials in $x_1, \dots x_{n-1}$

- Interpret $P_1, \dots P_n$ as polynomials in x_n over $\mathbb{F}_q[x_1, \dots x_{n-1}]$
- Compute $Q_i = res(P_i, P_n) \in \mathbb{F}_q[x_1, \dots x_{n-1}]$ for $i = 0, \dots n-1$
- Solve $\mathcal{P}' = \{Q_1(x_1, \dots x_{n-1}) = 0, \dots Q_{n-1}(x_1, \dots x_{n-1}) = 0\}$

Solving generic polynomial systems with resultants

$$\mathcal{P} = \begin{cases} P_1(x_1, \dots x_n) = 0 \\ \vdots \\ P_n(x_1, \dots x_n) = 0 \end{cases}$$

Idea: Eliminate the variable x_n and produce $n-1$ polynomials in $x_1, \dots x_{n-1}$

- Interpret $P_1, \dots P_n$ as polynomials in x_n over $\mathbb{F}_q[x_1, \dots x_{n-1}]$
- Compute $Q_i = res(P_i, P_n) \in \mathbb{F}_q[x_1, \dots x_{n-1}]$ for $i = 0, \dots n-1$
- Solve $\mathcal{P}' = \{Q_1(x_1, \dots x_{n-1}) = 0, \dots Q_{n-1}(x_1, \dots x_{n-1}) = 0\}$

Issue: The degrees of the Q_i increase significantly compared to the P_i

- $\deg(Q_i) = \deg(P_i) \times \deg(P_n)$
- The ideal degree increases by $\deg(P_n)^{n-2}$: many parasite solutions
- Complexity estimation: at least $\deg(P_i)^{n(n-1)/2+1}$ operations
  - Costlier than Groebner bases when $n \geq 3$

Resultants in our context

[YZY+, AC'24]

$$
\begin{cases}
Z_1{}^{\alpha} - U_1(X) = 0 \\
Z_2{}^{\alpha} - U_2(X, Z_1) = 0 \\
Z_3{}^{\alpha} - U_3(X, Z_1, Z_2) = 0 \\
\vdots \\
Z_{t-1}{}^{\alpha} - U_{t-1}(X, Z_1, Z_2, \cdots, Z_{t-2}) = 0 \\
Z_t{}^{\alpha} - U_t(X, Z_1, Z_2, \cdots, Z_{t-2}, Z_{t-1}) = 0 \\
P(X, Z_1, Z_2, \ldots, Z_{t-1}, Z_t) = 0
\end{cases}
$$

Resultants in our context

[YZY+, AC'24]

$$\begin{cases} {Z_1}^{\alpha} - U_1(X) = 0 \\ {Z_2}^{\alpha} - U_2(X, Z_1) = 0 \\ {Z_3}^{\alpha} - U_3(X, Z_1, Z_2) = 0 \\ \vdots \\ {Z_{t-1}}^{\alpha} - U_{t-1}(X, Z_1, Z_2, \cdots, Z_{t-2}) = 0 \\ {Z_t}^{\alpha} - U_t(X, Z_1, Z_2, \cdots, Z_{t-2}, Z_{t-1}) = 0 \\ P(X, Z_1, Z_2, \ldots, Z_{t-1}, Z_t) = 0 \end{cases} \rightsquigarrow \begin{cases} {Z_1}^{\alpha} - U_1(X) = 0 \\ {Z_2}^{\alpha} - U_2(X, Z_1) = 0 \\ {Z_3}^{\alpha} - U_3(X, Z_1, Z_2) = 0 \\ \vdots \\ {Z_{t-1}}^{\alpha} - U_{t-1}(X, Z_1, Z_2, \cdots, Z_{t-2}) = 0 \\ \tilde{P}(X, Z_1, Z_2, \ldots, Z_{t-1}) = 0 \end{cases}$$

$$\tilde{P} = res_{Z_t}(P, {Z_t}^{\alpha} - U_t)$$

Special Resultant

Very structured Sylvester matrix

Using $P = \sum_{i=0}^{d} a_i {Z_t}^i, a_i \in \mathbb{F}_p[X, Z_1, Z_2, \dots, Z_{t-1}]$

- Computing the naive determinant costs $\mathcal{O}\left((d+\alpha)^3\right)$

$$res_{Z_t}(P, {Z_t}^{\alpha} - U_t) = \begin{vmatrix} -U_t & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & -U_t & 0 & \dots & 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 1 & 0 \\ 0 & \dots & 0 & -U_t & 0 & \dots & 0 & 1 \\ a_0 & a_1 & \dots & a_d & 0 & \dots & \dots & 0 \\ 0 & a_0 & a_1 & \dots & a_d & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & a_0 & a_1 & \dots & a_d \end{vmatrix}$$

Special Resultant

Very structured Sylvester matrix

Using $P = \sum_{i=0}^{d} a_i {Z_t}^i,\ a_i \in \mathbb{F}_p[X, Z_1, Z_2, \ldots, Z_{t-1}]$

- Computing the naive determinant costs $\mathcal{O}\left((d+\alpha)^3\right)$

$$
res_{Z_t}(P, {Z_t}^{\alpha} - U_t) = \begin{vmatrix}
-U_t & 0 & \ldots & 0 & 1 & 0 & \ldots & 0 \\
0 & -U_t & 0 & \ldots & 0 & 1 & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 1 & 0 \\
0 & \ldots & 0 & -U_t & 0 & \ldots & 0 & 1 \\
a_0 & a_1 & \ldots & a_d & 0 & \ldots & \ldots & 0 \\
0 & a_0 & a_1 & \ldots & a_d & \ddots & & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\
0 & \ldots & \ldots & 0 & a_0 & a_1 & \ldots & a_d
\end{vmatrix}
$$

$\Rightarrow$ Combinations of rows to reduce the determinant's size

Special resultants

Special Toeplitz matrix

- The presented matrix is also a special Toeplitz matrix
- In practice, we use $\alpha = 3$ which makes this computation cheap
- For larger α the overhead is roughly of α^2

$$res_{Z_t}(P, {Z_t}^{\alpha} - U_t) = \begin{vmatrix} a_0 & U_t a_{\alpha-1} & \dots & U_t a_2 & U_t a_1 \\ a_1 & a_0 & U_t a_{\alpha-1} & \dots & U_t a_2 \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ a_{\alpha-2} & \dots & a_1 & a_0 & U_t a_{\alpha-1} \\ a_{\alpha-1} & a_{\alpha-2} & \dots & a_1 & a_0 \end{vmatrix}.$$

$\Rightarrow$ We compute a $\alpha \times \alpha$ resultant instead

Experimental results ($\alpha = 3$)

Cipher	t	Number of rounds (r)					
		7	8	9	10	11	
Anemoi	2	49m	10h	-	-	-	[YZY+]
		9.5s	**1m25s**	**13m51s**	**2h38m**	**1d22h**	**Ours**
		6	7	8			
Griffin	12	1m	3h32m	-			[BBL+]
		10s	**5m30s**	**4h20m**			**Ours**
		4	5	6			
Rescue	3	15m	1d	-			[YZY+]
		2.4s	**6m6s**	**2d4h**			**Ours**

Theoretical complexities (full-round instances)

Cipher	Security	Parameters				
		$\alpha = 3$	$\alpha = 5$	$\alpha = 7$	$\alpha = 11$	
Anemoi	128	110	133	141	158	[YZY+]
		80	**96**	**103**	**111**	**Ours**
		$t = 3$	$t = 4$	$t = 8$	$t \geq 12$	
Griffin	128	120	112	76	64	[BBL+]
		96	**87**	**63**	**55**	**Ours**
		$\alpha = t = 3$				
Rescue	512	-				-
		475				**Ours**

Conclusion

Insights on AO design criteria

- AO hash functions should not base their security on Gröbner basis methods
- Instead, conservatively consider the ideal degree d_I as a lower bound for the best attack

Future works

- Utilizing better algorithm for generic resultant computations
- Moving from CICO-1 to CICO-2

Thank you for your attention !

The reduction algorithm

Algorithm 1 $\mathrm{Reduce}_k(g(X, Z_1, \ldots, Z_k), \mathcal{P}_k)$

Input: A polynomial $g \in \mathbb{F}_q[X, Z_1, \ldots, Z_k]$, where $\deg_{Z_i}(g) < 2\alpha - 1$ for $1 \leq i \leq k$, and a reduced polynomial system $\mathcal{P}_k$

Output: The normal form of g with respect to $\mathcal{P}_k$

if $k = 0$ **then**
  return g
end if
write g as $g = \sum_{i=0}^{2\alpha-2} g_i(X, Z_1, \ldots, Z_{k-1}){Z_k}^i$
$\rho \leftarrow \mathrm{Reduce}_{k-1}(g_{\alpha-1}, \mathcal{P}_{k-1}) \cdot {Z_k}^{\alpha-1}$
for $i = 0$ to $\alpha - 2$ **do**
  $\rho \leftarrow \rho + \mathrm{Reduce}_{k-1}(g_i + \mathrm{Reduce}_{k-1}(g_{\alpha+i}, \mathcal{P}_{k-1}) \cdot f_k, \mathcal{P}_{k-1}) \cdot {Z_k}^i$
  ▷ $2\alpha - 1$ calls to Reduce_{k-1} in total
end for
return ρ

Example for $\alpha = 3$ and $d = 6$

$$\cdot \times U_t +$$

$$\left|\begin{array}{ccc:ccc:ccc}
1 & 0 & 0 & -U_t & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & -U_t & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & -U_t & 0 & 0 & 0 \\
\hdashline
0 & 0 & 0 & 1 & 0 & 0 & -U_t & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & -U_t & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -U_t \\
\hdashline
a_6 & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 & 0 & 0 \\
0 & a_6 & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 & 0 \\
0 & 0 & a_6 & a_5 & a_4 & a_3 & a_2 & a_1 & a_0
\end{array}\right|$$

Example for $\alpha = 3$ and $d = 6$

$$\cdot \times U_t +$$

$$\left|\begin{array}{ccc:ccc:ccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hdashline
0 & 0 & 0 & 1 & 0 & 0 & -U_t & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & -U_t & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -U_t \\
\hdashline
a_6 & a_5 & a_4 & \bar{a}_3 & \bar{a}_2 & \bar{a}_1 & a_0 & 0 & 0 \\
0 & a_6 & a_5 & a_4 & \bar{a}_3 & \bar{a}_2 & a_1 & a_0 & 0 \\
0 & 0 & a_6 & a_5 & a_4 & \bar{a}_3 & a_2 & a_1 & a_0
\end{array}\right|$$

Example for $\alpha = 3$ and $d = 6$

$$
\left|\begin{array}{ccc:ccc:ccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hdashline
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\hdashline
a_6 & a_5 & a_4 & \bar{a}_3 & \bar{a}_2 & \bar{a}_1 & \tilde{a}_0 & U_t\bar{a}_2 & U_t\bar{a}_1 \\
0 & a_6 & a_5 & a_4 & \bar{a}_3 & \bar{a}_2 & \tilde{a}_1 & \tilde{a}_0 & U_t\bar{a}_2 \\
0 & 0 & a_6 & a_5 & a_4 & \bar{a}_3 & \tilde{a}_2 & \tilde{a}_1 & \tilde{a}_0
\end{array}\right|
$$

We are left with a $\alpha \times \alpha$ determinant of a Toepliz matrix !