CRYPTANALYSIS OF A POLYNOMIAL-BASED REDACTABLE BLOCKCHAIN

ALEXANDER USHAKOV

Abstract. We analyze a recently published construction of redactable blockchains based on multivariate polynomial equations. We show that its public integrity conditions reveal sufficient algebraic structure to recover the relevant private evaluation points efficiently. As a consequence, the blockchain unauthorized redaction problem reduces to univariate root finding over a finite field followed by solving a linear system. We also discuss an alternative attack based on linearization.

Keywords. Redactable blockchains, multivariate polynomial equations, cryptanalysis, algebraic attacks, hidden evaluation points, post-quantum cryptography.
2020 Mathematics Subject Classification. 94A60, 68P25, 13P10, 11T71.

1. Introduction

Redactable blockchains are designed to preserve the integrity of a blockchain while allowing an authorized party to modify previously recorded data. Such constructions are typically based on a trapdoor mechanism: verification is public, whereas modifying a block without changing the subsequent chain requires knowledge of secret information.

An early construction of redactable blockchains was proposed by Ateniese, Magri, Venturi, and Andrade [1]. Their approach uses chameleon hash functions to allow authorized modifications of previously recorded blocks while preserving the integrity of the blockchain. A different construction, based on the RSA problem, was proposed by Grigoriev and Shpilrain [3].

More recently, Demin, Ovchinnikov, and Shpilrain [2] proposed a post-quantum construction based on multivariate polynomial equations over a finite field. The security of the scheme is intended to rely on the difficulty of solving certain systems of polynomial equations. In the advanced version of the construction, the public integrity conditions are obtained by evaluating a private polynomial at a collection of secret points

$$t_0, \dots, t_d$$

and publishing the resulting expanded polynomial equations. The evaluation points themselves, as well as the private polynomials used to construct the integrity conditions, remain hidden.

The authors of [2] consider several natural attacks on the scheme. In particular, their Attack 1 amounts to fixing a desired new content and solving the resulting system of $d+1$ polynomial equations in the $d+1$ unknown coefficients of a new suffix. Experimental evidence based on Gröbner basis computations suggests that solving this system directly may be computationally expensive.

Date: September 20, 2026.

In this paper, we show that the public polynomial system has a special algebraic structure that makes this problem considerably easier. After fixing the prefix, the new content, and the prefix of the subsequent block, the jth public integrity equation has the form

$$Q_j(\bar{c}) = T_j(L_j(\bar{c})),$$

where T_j is a univariate polynomial and

$$L_j(c_0, \ldots, c_d) = c_0 + c_1 t_j + \cdots + c_d t_j^d$$

is a linear form determined by the secret evaluation point t_j.

This representation allows us to recover all relevant hidden evaluation points efficiently from the public integrity equations. Once the points t_j are known, the multivariate system separates into a collection of independent univariate polynomial equations over the finite field. Solving these equations yields values

$$\alpha_j = P_{\bar{c}}(t_j),$$

and the coefficients of a valid suffix $P_{\bar{c}}$ are then obtained by solving a linear system. Thus, the blockchain unauthorized redaction problem reduces to univariate root finding over a finite field followed by solving a linear system, rather than to solving a generic multivariate polynomial system.

We also revisit Attack 4 from [2], which uses information obtained from legitimate block redactions to recover the private polynomial H. In the current version of the construction, the resulting equations are nonlinear in the coefficients of H. We observe, however, that these equations are of bounded degree and can be linearized by introducing variables for the corresponding monomials.

The main contribution of this paper is therefore to show that the algebraic structure introduced by the hidden evaluation points remains visible in the public integrity equations. This structure can be exploited to transform the blockchain unauthorized redaction problem into substantially simpler algebraic problems.

2. Description of the advanced scheme

2.1. **Setup.** Let n be a prime and let

$$\mathbb{F} = \mathbb{Z}_n.$$

The prime n is chosen so that $n \equiv 3 \pmod 4$. Hence -1 is not a quadratic residue in $\mathbb{F}$, and therefore

$$u^2 + 1 \neq 0 \qquad \text{for every } u \in \mathbb{F}.$$

Each *block* B_i is a triple

$$(P_i(t), C_i(t), S_i(t))$$

of polynomials of degree at most d over $\mathbb{F}$, where

- P_i is the *prefix* of the block,
- C_i is the *content* of the block, and
- S_i is an auxiliary *suffix* that allows an authorized party to modify the content without changing the subsequent blocks.

The construction uses private polynomials

$$G(x,y), \qquad H(x,y), \qquad R(u,z),$$

where $R(u,z)$ is nonlinear in u but linear in z. These polynomials define the private polynomial

$$F(x,y,z) = G(x, R(H(x,y),z)).$$

In [2], the authors suggest using

$$R(u,z) = u + u^2 z + z, \tag{1}$$

which gives

$$F(x,y,z) = G\big(x, H(x,y) + H(x,y)^2 z + z\big).$$

In addition, the central authority chooses distinct private elements

$$t_0, \ldots, t_d \in \mathbb{F}.$$

These points determine the private polynomial

$$M(t) = \prod_{j=0}^{d} (t - t_j).$$

2.2. **Public integrity condition.** The *integrity condition* for a block (P_i, C_i, S_i) in the chain $\{(P_i, C_i, S_i)\}_i$ is given by

$$F(P_i, C_i, S_i) \equiv_{M(t)} P_{i+1}. \tag{2}$$

Making this condition public in its original form would reveal the structure of F. Therefore, the authors translate (2) into a system of polynomial congruences in the coefficients of P_i, C_i, S_i, and P_{i+1}.

Since

$$M(t) = \prod_{j=0}^{d} (t - t_j)$$

has distinct linear factors, condition (2) is equivalent to

$$\bigwedge_{j=0}^{d} F(P_i, C_i, S_i) \equiv_{t-t_j} P_{i+1},$$

or, equivalently,

$$\bigwedge_{j=0}^{d} F(P_i(t_j), C_i(t_j), S_i(t_j)) \equiv_n P_{i+1}(t_j). \tag{3}$$

Write

$$\begin{aligned} P_i(t) &= \sum_{k=0}^{d} A_k t^k, \qquad & C_i(t) &= \sum_{k=0}^{d} B_k t^k, \\ S_i(t) &= \sum_{k=0}^{d} C_k t^k, \qquad & P_{i+1}(t) &= \sum_{k=0}^{d} D_k t^k, \end{aligned} \tag{4}$$

where

$$\overline{A} = (A_0, \dots, A_d), \quad \overline{B} = (B_0, \dots, B_d), \quad \overline{C} = (C_0, \dots, C_d), \quad \overline{D} = (D_0, \dots, D_d) \tag{5}$$

stand for the coefficient vectors of P_i, C_i, S_i, and P_{i+1}, respectively.

Substituting (4) into (3) and expanding yields a system of $d+1$ polynomial congruences

$$\bigwedge_{j=0}^{d} E_j(\overline{A}, \overline{B}, \overline{C}, \overline{D}) \equiv_n 0. \tag{6}$$

These congruences are public and are called the *public integrity conditions*. The polynomials F, G, H and the evaluation points $t_0, \dots, t_d$ remain private.

Notice that the congruences (6) are linear in $D_0, \dots, D_d$. Therefore, given P_i, C_i, S_i, the coefficients of P_{i+1} can be computed by solving a system of $d+1$ linear equations over $\mathbb{F}$.

2.3. **Authorized redaction.** Suppose that the central authority wants to replace the content C_i by a new content C_i' while keeping P_i and P_{i+1} unchanged. For the trapdoor

$$F(x, y, z) = G\big(x, H(x, y) + H(x, y)^2 z + z\big),$$

it is sufficient to find a polynomial S_i' of degree at most d satisfying

$$H(P_i, C_i) + H(P_i, C_i)^2 S_i + S_i \equiv_{M(t)} H(P_i, C_i') + H(P_i, C_i')^2 S_i' + S_i'. \tag{7}$$

For the central authority, which knows H and $M(t)$, this is a linear congruence in the coefficients of S_i'. Since $n \equiv 3 \pmod 4$, we have

$$H(P_i(t_j), C_i'(t_j))^2 + 1 \neq 0, \qquad j = 0, \dots, d.$$

Hence $H(P_i, C_i')^2 + 1$ is invertible in $\mathbb{F}[t]/(M(t))$, and (7) has a unique solution modulo $M(t)$. Since $\deg M = d+1$, this solution has a unique representative S_i' of degree at most d.

The resulting modified block

$$(P_i, C_i', S_i')$$

satisfies the same integrity condition with P_{i+1} and therefore does not require modification of the subsequent blocks.

2.4. **Blockchain unauthorized redaction problem.** Suppose that an unauthorized user wants to replace the content C_i of a block (P_i, C_i, S_i) by a prescribed polynomial C_i', while keeping P_i and P_{i+1} unchanged. To accomplish this, the user needs to find a new suffix S_i' whose coefficients satisfy the public integrity conditions (6). This computational task gives rise to the following algorithmic problem.

BLOCKCHAIN UNAUTHORIZED REDACTION PROBLEM (BURP(n, d, P, C', P')).

In: A prime $n \in \mathbb{N}$, an integer $d < n$, the system (6), and polynomials $P(t), C'(t), P'(t) \in \mathbb{Z}_n[t]$ of degree at most d.

Question: Find a polynomial $S'(t) \in \mathbb{Z}_n[t]$ of degree at most d such that (P, C', S', P') satisfies (6).

By the authorized redaction procedure described above, every instance of BURP arising from the proposed construction has at least one solution. Indeed, for every prescribed content C', equation (7) has a solution S' of degree at most d, and this solution satisfies the integrity condition (2), and hence the public integrity conditions (6). The main result of this paper can therefore be formulated as follows.

Theorem 2.1. *Assume that* $\deg_z F < n$. *Every instance of the Blockchain Unauthorized Redaction Problem arising from the proposed construction can be solved in probabilistic polynomial time.*

The next two sections develop the algorithm and prove Theorem 2.1.

2.5. **Notation.** Since we work with polynomials in the variable t of degree at most d, each uniquely determined by its $d+1$ coefficients, it is convenient to introduce the following notation. For $\overline{a} = (a_0, \ldots, a_d) \in \mathbb{F}^{d+1}$, define

$$P_{\overline{a}}(t) = a_d t^d + \cdots + a_1 t + a_0.$$

Let us reformulate the integrity condition (2) using this notation. A general block is represented by the triple

$$B_i = (P_{\overline{a}}(t), P_{\overline{b}}(t), P_{\overline{c}}(t)),$$

and suppose that the next block B_{i+1} has prefix $P_{\overline{d}}(t)$. The integrity condition then takes the form

$$F(P_{\overline{a}}(t), P_{\overline{b}}(t), P_{\overline{c}}(t)) - P_{\overline{d}}(t) \equiv_{M(t)} 0.$$

Regarding t and the coefficients $a_0, \ldots, a_d, b_0, \ldots, b_d, c_0, \ldots, c_d, d_0, \ldots, d_d$ as indeterminates, the left-hand side defines a polynomial in $4d+5$ variables

$$t, a_0, \ldots, a_d, b_0, \ldots, b_d, c_0, \ldots, c_d, d_0, \ldots, d_d.$$

Denote the left-hand side by

$$E(\overline{a}, \overline{b}, \overline{c}, \overline{d}, t) = F(P_{\overline{a}}(t), P_{\overline{b}}(t), P_{\overline{c}}(t)) - P_{\overline{d}}(t).$$

Substituting $t \leftarrow t_j$ yields the polynomial

$$E_j(\overline{a}, \overline{b}, \overline{c}, \overline{d}) = E(\overline{a}, \overline{b}, \overline{c}, \overline{d}, t_j)$$

in the $4d+4$ variables. Hence, in this notation, the blockchain unauthorized redaction problem can be formulated as follows.

Blockchain Unauthorized Redaction Problem (BURP$(n, d, \{E_j\}, \overline{a}, \overline{b}, \overline{d})$).

In: A prime $n \in \mathbb{N}$, an integer $d < n$, polynomials $E_0, \ldots, E_d$, and vectors $\overline{a}, \overline{b}, \overline{d} \in \mathbb{F}^{d+1}$.

Question: Find a vector $\overline{c} \in \mathbb{F}^{d+1}$ such that $E_j(\overline{a}, \overline{b}, \overline{c}, \overline{d}) \equiv_n 0$ for every $j = 0, \ldots, d$.

For each $j = 0, \ldots, d$, the expression $E_j(\overline{a}, \overline{b}, \overline{c}, \overline{d})$ is a polynomial in the variables $\overline{c} = (c_0, \ldots, c_d)$. Denote this polynomial by

$$Q_j(\overline{c}) = E_j(\overline{a}, \overline{b}, \overline{c}, \overline{d}).$$

Thus, the blockchain unauthorized redaction problem amounts to solving the system of polynomial congruences

$$\bigwedge_{j=0}^{d} Q_j(\overline{c}) \equiv_n 0, \tag{8}$$

in the unknown vector $\overline{c} = (c_0, \ldots, c_d) \in \mathbb{F}^{d+1}$.

The attack described in [2, Attack 1] translates directly into the problem of solving the system (8). In the following sections, we exploit the special structure of this system to show that a solution can be found efficiently.

3. Recovering the points $t_0, \dots, t_d$

Consider an instance $(n, d, \{E_j\}_{j=0}^{d}, \overline{a}, \overline{b}, \overline{d})$ of BURP. In this section, we show that every evaluation point t_j corresponding to a nonzero polynomial Q_j can be recovered efficiently. If Q_j is identically zero, the corresponding integrity condition imposes no restriction and can be discarded.

3.1. **The polynomial $Q_j(\overline{c})$ as a composition.** After substituting $\overline{a}$, $\overline{b}$, and $\overline{d}$ into the jth public integrity condition, we obtain

$$Q_j(\overline{c}) = F(P_{\overline{a}}(t_j), P_{\overline{b}}(t_j), P_{\overline{c}}(t_j)) - P_{\overline{d}}(t_j).$$

Hence, $Q_j(\overline{c})$ has the form

$$Q_j(\overline{c}) = T_j\big(L_j(\overline{c})\big), \tag{9}$$

where

$$T_j(z) = F\big(P_{\overline{a}}(t_j), P_{\overline{b}}(t_j), z\big) - P_{\overline{d}}(t_j)$$

is a univariate polynomial and

$$L_j(c_0, \dots, c_d) = c_0 + c_1 t_j + \cdots + c_d t_j^d = P_{\overline{c}}(t_j)$$

is a linear form in $c_0, \dots, c_d$.

Although F and t_j are private, the polynomial T_j can be recovered directly from the public polynomial Q_j. Indeed, setting $c_0 = z$ and $c_1 = \cdots = c_d = 0$ gives

$$L_j(z, 0, \dots, 0) = z,$$

and therefore

$$T_j(z) = Q_j(z, 0, \dots, 0). \tag{10}$$

3.2. **Recovering the evaluation point t_j.** It follows from (9) that t_j satisfies the polynomial identity

$$Q_j(c_0, \dots, c_d) = T_j(c_0 + c_1 t_j + \cdots + c_d t_j^d) \tag{11}$$

over $\mathbb{F}$.

For the parameters suggested in the construction, where $n \approx 2^{20}$, one can recover t_j simply by enumerating all $t \in \mathbb{F}$ and checking the polynomial identity

$$Q_j(c_0, \dots, c_d) = T_j(c_0 + c_1 t + \cdots + c_d t^d).$$

Since the correct value $t = t_j$ satisfies this identity by construction, the enumeration always produces at least one solution.

It is possible that more than one value $t \in \mathbb{F}$ satisfies the above identity. This does not cause a problem for our purposes. Indeed, if t_j' is another such value, then

$$T_j(c_0 + c_1 t_j + \cdots + c_d t_j^d) = T_j(c_0 + c_1 t_j' + \cdots + c_d (t_j')^d)$$

as polynomials. Consequently,

$$Q_j(\overline{c}) \equiv_n 0$$

is equivalent to either representation. Thus, for solving the BURP, we may use any value satisfying (11). Thus, for the suggested parameter $n \approx 2^{20}$, the evaluation point t_j can already be recovered by exhaustive search. In the next subsection, we give a direct polynomial-time recovery method that does not depend on the size of n.

3.3. **Direct recovery of t_j from partial derivatives.** The preceding exhaustive search relies on the relatively small value of n suggested in [2]. We now show that the relevant evaluation points can be recovered in polynomial time without enumerating the elements of $\mathbb{F}$. Differentiating (9) with respect to c_k gives

$$\frac{\partial Q_j}{\partial c_k} = T_j'(L_j(\overline{c}))t_j^k.$$

In particular,

$$\frac{\partial Q_j}{\partial c_1} = t_j \frac{\partial Q_j}{\partial c_0}. \tag{12}$$

Therefore, if $\partial Q_j/\partial c_0$ is not the zero polynomial, then t_j can be recovered directly from the public polynomial Q_j. Indeed, choose any monomial μ whose coefficient in $\partial Q_j/\partial c_0$ is nonzero. Let a and b be the coefficients of μ in $\partial Q_j/\partial c_0$ and $\partial Q_j/\partial c_1$, respectively. By (12), $b = t_j a$, and hence

$$t_j = ba^{-1} \in \mathbb{F}.$$

Suppose now that $\partial Q_j/\partial c_0$ is the zero polynomial. Then $\partial Q_j/\partial c_k$ is the zero polynomial for every $k = 0, \ldots, d$. Since $P_{\overline{c}}(t_j)$ is linear in each c_k, we have

$$\deg_{c_k} Q_j \le \deg_z F < n$$

for every k. It follows that Q_j is constant:

$$Q_j(\overline{c}) = a$$

for some $a \in \mathbb{F}$. Since the considered instance of BURP is assumed to have a solution, the congruence

$$Q_j(\overline{c}) \equiv_n 0$$

must be satisfiable. Hence $a = 0$, and Q_j is identically zero. Consequently, this congruence imposes no restriction on $\overline{c}$ and can be removed from the system (8).

4. Computing $\overline{c}$ from roots of the polynomials T_j

Let $J \subseteq \{0, \ldots, d\}$ be the set of indices for which Q_j is not identically zero. For $j \notin J$, the congruence $Q_j(\overline{c}) \equiv_n 0$ imposes no restriction and can be removed from the system (8). Once the polynomials $T_j(z)$ and the corresponding evaluation points t_j have been recovered for $j \in J$, the system (8) takes the form

$$\bigwedge_{j \in J} T_j(c_0 + c_1 t_j + \cdots + c_d t_j^d) \equiv_n 0. \tag{13}$$

Since the instance is admissible, there exists $\overline{c} \in \mathbb{F}^{d+1}$ satisfying (8). Consequently, for every $j \in J$,

$$T_j(P_{\overline{c}}(t_j)) = Q_j(\overline{c}) \equiv_n 0,$$

and hence T_j has at least one root in $\mathbb{F}$.

The roots of T_j in $\mathbb{F}$ can be found in probabilistic polynomial time using standard algorithms for polynomial factorization over finite fields, such as the Cantor–Zassenhaus algorithm. For the parameters proposed in [2], where $n \approx 2^{20}$, even direct enumeration of all elements of $\mathbb{F}$ is feasible. For each $j \in J$, choose one root

$$\alpha_j \in \mathbb{F}.$$

Now, to solve the system (13), it is sufficient to find any $c_0, \ldots, c_d$ satisfying the linear system

$$\bigwedge_{j \in J} c_0 + c_1 t_j + \cdots + c_d t_j^d \equiv_n \alpha_j.$$

This system has at most $d+1$ equations in $d+1$ unknowns. Since the points t_j, $j \in J$, are pairwise distinct, its coefficient matrix has full row rank. Hence, the system is consistent, and a particular solution $\bar{c} = (c_0, \ldots, c_d)$ can be found efficiently by Gaussian elimination. For every $j \in J$,

$$Q_j(\bar{c}) = T_j(\alpha_j) \equiv_n 0.$$

For $j \notin J$, the polynomial Q_j is identically zero. Therefore, $\bar{c}$ is a solution to the Blockchain Unauthorized Redaction Problem.

Remark 1. If none of the polynomials $Q_0, \ldots, Q_d$ is identically zero, then $J = \{0, \ldots, d\}$ and the above linear system consists of exactly $d+1$ equations in $d+1$ unknowns. In this case, the computation of $\bar{c}$ can equivalently be viewed as polynomial interpolation: we seek the unique polynomial $P_{\bar{c}}(t)$ of degree at most d satisfying

$$P_{\bar{c}}(t_j) = \alpha_j, \qquad j = 0, \ldots, d.$$

Proof of Theorem 2.1. For every nonzero polynomial Q_j, the corresponding evaluation point t_j can be recovered efficiently as described above, and the univariate polynomial T_j is obtained from

$$T_j(z) = Q_j(z, 0, \ldots, 0).$$

Since the instance is admissible, each T_j, $j \in J$, has a root in $\mathbb{F}$. Such a root can be found in probabilistic polynomial time.

The resulting linear system in $c_0, \ldots, c_d$ has full row rank and can be solved efficiently by Gaussian elimination. The resulting vector $\bar{c}$ satisfies every congruence in (8), and hence determines a valid suffix.

All operations involved are deterministic polynomial-time operations, except for finite-field root finding, which can be performed in probabilistic polynomial time. This proves the theorem. □

5. Computing $\bar{c}$ by recovering the function H

In this section, we revisit the line of attack described in [2, Attack 4]. Assuming that the evaluation points $t_0, \ldots, t_d$ have been recovered, we show that the information obtained from legitimate redactions can be linearized into a system of linear congruences involving at most 54 unknowns. In particular, if this system has full rank, the secret polynomial H can be recovered efficiently.

By design, the function H is a linear combination of the monomials $x^i y^j$ with private coefficients $b_{ij} \in \mathbb{F}$:

$$H(x,y) = \sum_{i=0}^{d_3} \sum_{j=0}^{d_4} b_{ij} x^i y^j,$$

where $d_3 = d_4 = 2$. Thus, H is determined by nine unknown coefficients b_{ij}. We show how legitimate block redactions produce linear information about these coefficients and describe conditions under which this information is sufficient to recover H.

For each redaction (P_i, C_i', S_i') of a block (P_i, C_i, S_i), the identity (7) holds. Equivalently, for every $j = 0, \ldots, d$,

$$\begin{aligned} &H(P_i(t_j), C_i(t_j)) + H(P_i(t_j), C_i(t_j))^2 S_i(t_j) + S_i(t_j) \\ &\quad \equiv_n H(P_i(t_j), C_i'(t_j)) + H(P_i(t_j), C_i'(t_j))^2 S_i'(t_j) + S_i'(t_j). \end{aligned} \tag{14}$$

The congruence (14) is quadratic in the unknown coefficients b_{ij} of H, but it can be linearized. For each quadratic monomial $b_{ij}b_{kl}$, introduce a new variable

$$b_{ijkl} = b_{ij} b_{kl}.$$

Since

$$b_{ij} b_{kl} = b_{kl} b_{ij},$$

the variables b_{ijkl} and b_{klij} represent the same quadratic monomial. Since H has nine unknown coefficients, there are

$$\binom{9+1}{2} = 45$$

distinct quadratic monomials in these coefficients.

Replacing each occurrence of $b_{ij}b_{kl}$ in (14) by the corresponding variable b_{ijkl} increases the number of variables from 9 to

$$9 + 45 = 54,$$

and transforms (14) into a linear congruence.

Consequently, observed legitimate redactions yield a system of linear congruences in these 54 variables, which can be analyzed efficiently by linear algebra over $\mathbb{F}$. A linear system of rank 54 certainly determines all the linearized variables uniquely. However, such full rank is not necessary for recovering H, since the auxiliary variables b_{ijkl} are not independent but satisfy the relations

$$b_{ijkl} = b_{ij} b_{kl}.$$

Therefore, the coefficients of H may in principle be recovered from a linear system of smaller rank.

Once the function H is recovered, the required polynomial $S' = P_{\bar{c}}$ can be computed by solving (7) for S'.

One possible way to increase the resistance of the scheme to this attack is to increase the number of distinct monomials that must be represented by independent variables after linearization. This can be achieved, for example, by increasing d_3 and d_4 and/or by increasing $\deg_u(R)$. As a result, the linearized system contains more unknowns, so this particular attack may require a larger number of independent congruences, and hence potentially more observed legitimate redactions, in order to recover H.

References

[1] G. Ateniese, B. Magri, D. Venturi, and E. Andrade, *Redactable blockchain—or—rewriting history in Bitcoin and friends*, Proceedings of the 2017 IEEE European Symposium on Security and Privacy (EuroS&P), IEEE, 2017, pp. 111–126. https://doi.org/10.1109/EuroSP.2017.37

[2] A. Demin, A. Ovchinnikov, and V. Shpilrain, *Redactable blockchains and polynomial equations*, Advances in Applied Mathematics **181** (2026).

[3] D. Grigoriev and V. Shpilrain, *RSA and redactable blockchains*, International Journal of Computer Mathematics: Computer Systems Theory, **6** (2021), no. 1, pp. 1–6. https://doi.org/10.1080/23799927.2020.1842808

Department of Mathematical Sciences, Stevens Institute of Technology, Hoboken NJ 07030

Email address: aushakov@stevens.edu