DEFEATING THE BUELLENS-BLACKBURN ATTACKS ON WALNUTDSA$^{\text{TM}}$

IRIS ANSHEL, DEREK ATKINS, DORIAN GOLDFELD, AND PAUL E. GUNNELLS
SECURERF CORPORATION
100 BEARD SAWMILL RD #350, SHELTON, CT 06484

Abstract. The Walnut Digital Signature Algorithm (WalnutDSA) is a group-theoretic, public-key method that is part of the NIST Post-Quantum Cryptography standardization process. Buellens and Blackburn proposed a series of attacks that demonstrated that the parameters proposed to NIST were too small, resulting in a too-small search space that could enable a practical forgery attack. We show that with a modest parameter-only increase we can block these attacks to the desired security level without a significant impact in the performance while making WalnutDSA completely secure against these attacks.

1. Introduction

The digital signature algorithm known as WalnutDSA$^{\text{TM}}$ was introduced in [1]. It is a group theoretic protocol which uses non linear operations in the Artin braid group B_N [3] together with operations in $GL(N, \mathbb{F}_q)$, the $N \times N$ matrix group over the finite field $\mathbb{F}_q$ with q elements.

Hart et al [7] proposed a practical forgery attack on WalnutDSA$^{\text{TM}}$ which required that the private keys (w, w') be equal (see §5 for how to construct w, w'). The Hart et al attack is blocked by the fact that it produces forged signatures that are very long, but the attack can also be easily defeated by increasing parameter sizes (see [2]).

Blackburn and Buellens [5] modified the Hart et al attack [7] so that the attack works even if $w \neq w'$.
The attack was mounted against the WalnutDSA$^{\text{TM}}$ NIST submission, which uses an older version of WalnutDSA$^{\text{TM}}$ which suggested running the protocol on $B_8, \mathbb{F}_{32}$ for 128-bit security and $B_8, \mathbb{F}_{256}$ for 256-bit security. Specifically, the attack in [5] showed that these parameter choices were too small.

Buellens-Blackburn [5] present attacks that produce forgeries whose lengths are the same or even shorter that legitimate signatures but the attacks are still exponential in running time and can be completely thwarted by running the protocol on $B_{10}, \mathbb{F}_{M_{31}}$, where M_{31} is the Mersenne prime $2^{31}-1$ for 128-bit security and $B_{10}, \mathbb{F}_{M_{61}}$ for 256-bit security. Furthermore, even with these increased parameter sizes, the high efficiency and low power consumption advantages of WalnutDSA$^{\text{TM}}$ for constrained devices are still retained.

Additionally, Blackburn and Buellens [5] present a birthday attack on the "Reversing E-Multiplication" (REM) problem which is a hard problem underlying the WalnutDSA$^{\text{TM}}$ protocol. This attack is again exponential in running time and is completely thwarted by running the WalnutDSA$^{\text{TM}}$ protocol on $B_{10}, \mathbb{F}_{M_{31}}$, for 128-bit security and $B_{10}, \mathbb{F}_{M_{61}}$ for 256-bit security.

2. Brief Introduction to WalnutDSA$^{\text{TM}}$

A core tool in group theoretic cryptography is the fact that an element of a group can be rewritten (using the relations in the group) so that the original expression of the element cannot be recovered. Consider, for example (for $N \geq 2$), the N-strand braid group with Artin generators $\{b_1, b_2, \ldots, b_{N-1}\}$, subject to the following relations:

$$b_i b_{i+1} b_i = b_{i+1} b_i b_{i+1}, \qquad (i = 1, \ldots, N-2), \tag{1}$$

$$b_i b_j = b_j b_i, \qquad (|i-j| \geq 2). \tag{2}$$

Let $\mathcal{R}\colon B_N \to B_N$ denote a rewriting algorithm. Well known examples are the Birman-Ko-Lee canonical form [4] or the Dehornoy handle reduction algorithm [6]. The security of WalnutDSA$^{\text{TM}}$ is based on the hard problems known as Reversing E-multiplication (REM) as well as the cloaked conjugacy search problem. E-multiplication, in its simplest form, is a function which on input of a braid element in B_N outputs a pair consisting of a matrix in $GL(N, \mathbb{F}_q)$ together with a permutation in S_N. E-multiplication is based on the colored Burau representation of the B_N [8]. Cloaking elements of B_N are defined to be braids whose output on E-multiplication is the pair consisting of the identity matrix and the identity permutation.

Fix a hash function H. In brief, the protocol begins with a message m which is first hashed to $H(m)$ and then encoded as an element $E(H(m)) \in B_N$. The signer's private key consists of two nontrivial elements in B_N, denoted w, w' (satisfying certain technical properties), and the signer's public key will be an $N \times N$ matrix over a finite field together with a permutation on N symbols, i.e., an element in the symmetric group S_N. The signed message will be a braid in B_N of the form

$$\mathcal{R}\big(v_1 \cdot w^{-1} \cdot v \cdot E(H(m)) \cdot w' \cdot v_2\big),$$

where $\mathcal{R}$ denotes a rewriting algorithm on B_N and $v, v_1, v_2 \in B_N$ are appropriate cloaking elements. Signature verification can be executed rapidly by performing E-multiplication on the signature.

3. Colored Burau Representation of the Braid Group

Each braid $\beta \in B_N$ determines a permutation in S_N (group of permutations of N letters) as follows: For $1 \leq i \leq N-1$, let $\sigma_i \in S_N$ be the i^{th} simple transposition, which maps $i \to i+1,\ i+1 \to i$, and leaves $\{1, \ldots, i-1, i+2, \ldots, N\}$ fixed. Then σ_i is associated to the Artin generator b_i. Further, if $\beta \in B_N$ is written as in (**??**), we take β to be associated to the permutation $\sigma_\beta = \sigma_{i_1} \cdots \sigma_{i_k}$. A braid is called pure if its underlying permutation is trivial (i.e., the identity permutation).

Let $\mathbb{F}_q$ denote the finite field of q elements, and for variables $t_1, t_2, \ldots, t_N$, let

$$\mathbb{F}_q[t_1, t_1^{-1}, \ldots, t_N, t_N^{-1}]$$

denote the ring of Laurent polynomials in $t_1, t_2, \ldots, t_N$ with coefficients in $\mathbb{F}_q$. Next, we introduce the colored Burau representation

$$\Pi_{CB}\colon B_N \to GL\Big(N, \mathbb{F}_q[t_1, t_1^{-1}, \ldots, t_N, t_N^{-1}]\Big) \times S_N.$$

First, we define the $N \times N$ colored Burau matrix (denoted CB) of each Artin generator as follows[**?**].

$$CB(b_1) = \begin{pmatrix} -t_1 & 1 & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix}, \tag{3}$$

For $2 \le i \le N-1$, the matrix $CB(b_i)$ is defined by

$$CB(b_i) = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & t_i & -t_i & 1 & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix}, \tag{4}$$

where the indicated variables appear in row i, and if $i = 1$ the leftmost t_1 is omitted.

We similarly define $CB(b_i^{-1})$ by modifying (4) slightly:

$$CB(b_i^{-1}) = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & 1 & -\frac{1}{t_{i+1}} & \frac{1}{t_{i+1}} & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix},$$

where again the indicated variables appear in row i, and if $i = 1$ the leftmost 1 is omitted.

Recall that each b_i has an associated permutation σ_i. We may then associate to each braid generator b_i (respectively, inverse generator b_i^{-1}) a colored Burau/permutation pair $(CB(b_i), \sigma_i)$ (resp., $(CB(b_i^{-1}), \sigma_i)$). We now wish to define a multiplication of such colored Burau pairs. To accomplish this, we require the following observation. Given a Laurent polynomial $f(t_1, \ldots, t_N)$ in N variables, a permutation in $\sigma \in S_N$ can act (on the left) by permuting the indices of the variables. We denote this action by $f \mapsto {}^{\sigma}f$:

$${}^{\sigma}f(t_1, t_2, \ldots, t_N) = f(t_{\sigma(1)}, t_{\sigma(2)}, \ldots, t_{\sigma(N)}).$$

We extend this action to matrices over the ring of Laurent polynomials in the t_i by acting on each entry in the matrix, and denote the action by $M \mapsto {}^{\sigma}M$. The general definition for multiplying two colored Burau pairs is now defined as follows: given $b_i^{\pm}, b_j^{\pm}$, the colored Burau/permutation pair associated with the product $b_i^{\pm} \cdot b_j^{\pm}$ is

$$(CB(b_i^{\pm}), \sigma_i) \cdot (CB(b_j^{\pm}),\ \sigma_j) = \Big(CB(b_i^{\pm}) \cdot ({}^{\sigma_i}CB(b_j^{\pm})),\ \sigma_i \cdot \sigma_j\Big).$$

We extend this definition to the braid group inductively: given any braid

$$\beta = b_{i_1}^{\epsilon_1} b_{i_2}^{\epsilon_2} \cdots b_{i_k}^{\epsilon_k},$$

we can define a colored Burau pair $(CB(\beta), \sigma_\beta)$ by

$$(CB(\beta), \sigma_\beta) = (CB(b_{i_1}^{\epsilon_1}) \cdot {}^{\sigma_{i_1}}CB(b_{i_2}^{\epsilon_2}) \cdot {}^{\sigma_{i_1}\sigma_{i_2}}CB(b_{i_3}^{\epsilon_3}))\ \cdots\ {}^{\sigma_{i_1}\sigma_{i_2}\cdots\sigma_{i_{k-1}}}CB(b_{i_k}^{\epsilon_k}),\ \ \sigma_{i_1}\sigma_{i_2}\cdots\sigma_{i_k}).$$

The colored Burau representation is then defined by

$$\Pi_{CB}(\beta) := (CB(\beta), \sigma_\beta).$$

One checks that Π_{CB} satisfies the braid relations and hence defines a representation of B_N.

4. E-Multiplication and Cloaking Elements

In brief, E-Multiplication is an action of a group of ordered pairs associated with B_N on a direct product of two groups. Given an element $\beta \in B_N$, we can associate with β both the colored Burau matrix $CB(\beta)$ (whose entries are Laurent polynomials in N variables) and the natural permutation σ_β of the braid which is an element in S_N. Since permutations themselves act on the colored Burau matrices, the ordered pairs $(CB(\beta), \sigma_\beta)$ form a group under the semi-direct product operation. By fixing a field $\mathbb{F}_q$, and a collection of N invertible elements in $\mathbb{F}_q$, $\{\tau_1, \ldots, \tau_N\}$, termed t-values, we can define the right action of $(CB(\beta), \sigma_\beta)$ on the ordered pair $(M, \sigma) \in GL_N(\mathbb{F}_q) \times S_N$:

$$(M, \sigma) \star (CB(\beta), \sigma_\beta) \; = \; \left(M \cdot {}^{\sigma}\big(CB(\beta)\big) \downarrow_{t\text{-values}}, \sigma \circ \sigma_\beta\right),$$

where the $\downarrow_{t\text{-values}}$ indicates the polynomials are evaluated at the t-values. While the Laurent polynomials which would naturally occur as entries of the colored Burau matrices would become computationally unmanageable, the generators b_i of B_N have sparse colored Burau matrices, and, hence, E-Multiplication can be evaluated very efficiently and rapidly.

The above discussion of an infinite group acting on a finite group necessitates the existence of stabilizing elements in the group B_N. With this in mind, we have the following:

Definition (Cloaking element) *Let $m \in GL(N, \mathbb{F}_q)$ and $\sigma \in S_N$. An element v in the pure braid subgroup of B_N (i.e., the permutation associated to v is the identity) is termed a cloaking element of (m, σ) if it satisfies $(m, \sigma) \star v = (m, \sigma)$.*

Thus a cloaking element will essentially disappear when E-Multiplication is evaluated. Since stabilizing elements of a group action form a subgroup, the following proposition is immediate:

Proposition 4.1. *The set of braids that cloak a specific ordered pair (m, σ) forms a subgroup of B_N.*

It should be remarked that when cloaking elements are constructed in the manner above, such elements only depend on the permutation σ. Thus, with a small abuse of language, we can say the element v cloaks for the permutation σ without any ambiguity.

Definition (κ cloaking) *Given an element $\beta \in B_N$, the output of κ iterations of randomly inserting cloaking elements into the braid β is defined to be a κ–cloaking of β and is denoted by $\kappa(\beta)$.*

5. WalnutDSA$^{\text{TM}}$ Signature Generation and Verification

For $\beta \in B_N$ let $\mathcal{P}(\beta)$ denote the E-multiplication of β against the identity element, i.e.,

$$\mathcal{P}(\beta) = (\mathrm{Id}_N, \mathrm{Id}_{S_N}) \star \beta$$

where Id_N is the $N \times N$ identity matrix and Id_{S_N} is the identity element in the symmtric group S_N. The Signer's private key consists of two random freely reduced braids $w, w' \in B_N$. The Signer's public key is $\big(\mathcal{P}(w),\, \mathcal{P}(w')\big)$.

Fix a hash function H. To sign a message $m \in \{0,1\}^*$ the Signer performs the following steps:

Digital Signature Generation:

1. Compute $H(m)$.

2. Generate cloaking elements v, v_1, and v_2 such that

– v cloaks $(\mathrm{Id}_N, \mathrm{Id}_{S_N})$,

– v_1 cloaks $\mathcal{P}(w)$.

– v_2 cloaks $\mathcal{P}(w')$.

3. Generate the encoded message $E(H(m))$.

4. Compute $\mathrm{Sig} = \mathcal{R}\big(\kappa\left(v_1 \cdot w^{-1} \cdot v \cdot E(H(m)) \cdot w' \cdot v_2\right)\big)$, which is a rewritten braid.

5. The final signature for the message m is the ordered pair $(H(m), \mathrm{Sig})$.

Signature Verification: The signature (m, Sig) is verified as follows:

1. Generate the encoded message $E(H(m))$.

2. Evaluate $\mathcal{P}(E(H(m)))$.

3. Evaluate the E-Multiplication $\mathcal{P}(w) \star \mathrm{Sig}$.

4. Test the equality

$$\mathrm{Matrix}\Big(\mathcal{P}(w) \star \mathrm{Sig}\Big) \overset{?}{=} \mathrm{Matrix}\Big(\mathcal{P}\big(E(H(m))\big)\Big) \cdot \mathrm{Matrix}\Big(\mathcal{P}\big(w'\big)\Big), \tag{5}$$

where Matrix denotes the matrix part of the ordered pair in question, and the multiplication on the right is the usual matrix multiplication. The signature is valid if and only if (5) holds and the signature has length $\leq 2L$ where L is a certain positive integer such that all valid WalnutDSA$^{\mathrm{TM}}$ signatures have length in the range $[L, 2L]$.

6. The Buellens-Blackburn Forgery Attacks

Blackburn and Buellens [5] present two forgery attacks on WalnutDSA$^{\mathrm{TM}}$. The first is modeled on the factorization attack of Hart et al attack [7] while the second is a collision search attack.

Factorization Attack: In order to remove the assumption that the private keys (w, w') may not be equal which is required for the Hart et al attack [7] Blackburn and Buellens point out that if one has 3 messages m, m_1, m_2 with $h = \text{Matrix}\big(\mathcal{P}\left(E(H(m)\right)\big),\ h_1 = \text{Matrix}\big(\mathcal{P}\left(E(H(m_1)\right)\big),\ h_2 = \text{Matrix}\big(\mathcal{P}\left(E(H(m_2)\right)\big)$, and three private keys w_1, w_2, w_3 then the following holds:

- *If $h = h_1^{-1}$ and s_1 is a valid signature for m_1 under the public key $(\mathcal{P}(v_1), \mathcal{P}(v_2))$ then s_1^{-1} is a valid signature for h under the public key $(\mathcal{P}(v_2), \mathcal{P}(v_1))$;*

- If $h = h_1 h_2$ and s_1, s_2 are valid signatures for m_1, m_2 under the public keys $(\mathcal{P}(v_1), \mathcal{P}(v_2))$, $(\mathcal{P}(v_2), \mathcal{P}(v_1))$, respectively, then $s_1 {\cdot} s_2$ is a valid signature for m under the public key $(\mathcal{P}(v_1), \mathcal{P}(v_3))$.

Assume the attacker knows message signature pairs (m_i, s_i) and associated matrices $h_i = \text{Matrix}\big(\mathcal{P}\left(E(H(m_i)\right)\big)$, $(i = 1, 2, \ldots, r)$ that are valid under the same public key $(\mathcal{P}(v_1), \mathcal{P}(v_2))$. Then it easily follows that for an odd number of factors

$$s_{i_1} \cdot s_{i_2}^{-1} \ \cdots \ s_{i_{L-1}}^{-1} \cdot s_{i_L}$$

is a valid signature under the public key $(\mathcal{P}(v_1), \mathcal{P}(v_2))$ for any message m satisfying

$$\text{Matrix}\big(\mathcal{P}\left(E(H(m)\right)\big) = h_{i_1} \cdot h_{i_2}^{-1} \ \cdots \ h_{i_{L-1}}^{-1} \cdot h_{i_L}.$$

At this point the attacker may implement the Hart et al factorization attack [7], but the forged signatures will be way too long and the running time will still be exponential so the attack will be completely thwarted on $B_{10}, \mathbb{F}_{M_{31}}$ and $B_{10}, \mathbb{F}_{M_{61}}$ with security levels at least $2^{128}, 2^{256}$, respectively.

Collision Search Attack:

The second forgery attack introduced in [5] seeks to find two messages m_1, m_2 such that

$$\mathcal{P}\left(E(H(m_1)\right) = \mathcal{P}\left(E(H(m_2)\right).$$

Finding such a collision breaks EUF-CMA. The method used in [5] to find a collision is based on the Oorschot and Wiener algorithm [9] which is a parallelizing collision search algorithm built upon the Pollard rho-method. Clearly, to determine the probability of a collision it is enough to find the size of $\mathcal{P}\Big(E\left(\{0,1\}^*\right)\Big)$. Let P_N denote the pure braid subgroup of B_N whose index in B_N is $N!$. Since the encoding function E takes values in P_N it is enough to obtain the size of $\mathcal{P}(P_N)$ which they estimate as $\approx q^{(N-2)^2+1}$ in [5].

As also pointed out in [5] any braid output by the encoding mechanism E is a product of the image (under $\mathcal{P}$) of the encoding braids used and, thus, it is essential that the subspace spanned by said images is sufficiently large. An example of an encoding that yields sufficient security, and hence, defeats this avenue of attack is given as follows. Let $N = 12$ and let S be the periodic sequence of tuples

$$\{(5,7,9,11), (4,6,8,10), (3,5,7,9), (2,4,6,8), (1,3,5,7), (2,4,6,8), (3,5,7,9), (4,6,8,10), ...\}.$$

One can check that this dimension is 122, so using $q = 32$ or 256 results in sufficiently large spaces. For the case of $N = 10$, S can be the sequence $\{(3,5,7,9), (2,4,6,8), (1,3,5,7), (2,4,6,8), ...\}$ which results in a dimension of 82.

7. Defeating the Buellens-Blackburn Forgery Attacks

The forgery attacks presented by Blackburn and Buellens in [5] are all exponential in running time as explained above. They can be completely thwarted by running the protocol on $B_{10}, \mathbb{F}_{M_{31}}$ for 128-bit security and $B_{10}, \mathbb{F}_{M_{61}}$ for 256-bit security.

8. Defeating the Buellens-Blackburn REM Attack

The third attack discussed in [5] is an exponential attack to reverse E-Multiplication (REM) which is a hard problem underlying the security of WalnutDSA^{TM}. The running time of this attack is estimated as $q^{N/2-1}$ in [5]. Here is a direct quote from the Blackburn-Beullens paper [5].

"There does not seem to be a better way to block the attack other than just increasing the parameters to ensure that $q^{N/2-1}$ is higher than the desired security level. One way to do this is to take $N = 10, q = 2^{32}$ to achieve 128 bits of security, and $N = 10, q = 2^{64}$ for 256 bits of security."

Recall that the T-values for E-Multiplication are just a subset of N invertible elements in $\mathbb{F}_q$ denoted $\{\tau_1, \tau_2, \ldots, \tau_N\}$. The Buellens-Blackburn REM attack assumed that for two integers a, b with $1 \le a < b \le N$ we specify that $\tau_a = \tau_b = 1$. If we instead we choose τ_a, τ_b so that $\tau_a \cdot \tau_b = -1$ then the running time of the REM attack is much higher. In fact, an additional factor of $\sqrt{q} \cdot \sqrt{x}$ is added to the runtime, where x is a parameter in their attack (they set $x = 60$ for $N = 8$ and we expect $x = 96$ for $N = 10$). This results in an (unverified) search time of at least $\sqrt{x} \cdot q^{(N-1)/2}$.

Here again, the parameters $B_{10}, \mathbb{F}_{M_{31}}$ for 128-bit security and $B_{10}, \mathbb{F}_{M_{61}}$ for 256-bit security effectively defeat the REM attack.

Increasing N and q does affect the performance of WalnutDSA. In a software implementation, each E-Multiplication step requires N multiplications and $2N$ additions within $\mathbb{F}_q$. This means that increasing N from 8 to 10 changes the number of basic operations from 8 to 10 multiplications and 16 to 20 additions, a 25% increase in the number of operations per E-Multiplication.

Increasing N also affects the length of the signature. The length increase can be obtained heuristically through testing. Using $N = 8$ the average length of a signature was 1399 Artin generators whereas increasing to $N = 10$ increased the length to 1909, a 36% increase in signature length (and an equivalent increase in signature verification time due to the 36% increase in the number of E-Multiplications required).

It should be noted that the increase of N also affects the signature storage size, because with $N = 8$ each generator only needs 4 bits, whereas 5 bits are required for $N = 10$. This increases the storage requirements by an additional 25%, for a total storage increase of 70%.

Increasing N and q affect the public key size, because the matrix is an $N \times N$ matrix over $\mathbb{F}_q$, which requires $N^2 \log_2(\mathrm{q})$ bits for each matrix. Increasing from $N = 8,\ q = 32$ to $N = 10,\ q = M_{31}$ results in an increase in public key matrices from 320 to 3100 bits each (a 10x increase). However, this 10x increase still results in public keys significantly shorter than the majority of NIST signature candidates.

Finally, increasing q from 32 to M_{31} does change the implementation of operations in $\mathbb{F}_q$. Whereas on F_{32} the operations could be implemented as a table lookup, using M_{31} no longer provides for that option. The primary consideration for performance of $\mathbb{F}_q$ is the state of the multiplier. Specifically, if the platform has a $32 \times 32 \to 64$ bit multiplier then the operation can be performed in only two instructions (multiplication and reduction). Some platforms don't provide this, but do provide a $32 \times 32 \to 32$(high) and $32 \times 32 \to 32$(low) operation. Other platforms truncate the result. And finally, some very small platforms don't provide for a 32-bit multiplier at all. The resulting performance degredation is determined by the available multiplier. We note that even on small platforms like an ARM Cortex M4, the multiplier is sufficient to compute the result in the single multiply instruction. The use of Mersenne primes like M_{31} affords a simple reduction methodology, which is simply a shift, addition, and possibly overflow subtraction.

All in, the signature verification times of WalnutDSA on the NIST test platform increased from 160,000 to 230,000 cycles due to these changes, a performance degredation of only 43%.

9. Conclusion

WalnutDSA is a group-theoretic, public-key method that is part of the NIST Post-Quantum Cryptography standardization process. Blackburn and Buellens showed a series of attacks against the parameters proposed for the NIST process, showing weaknesses in the search space and a practical forgery attack that produces signatures that are short enough to be considered valid. We have shown that with a modest parameter increase from $N = 8$ to $N = 10$ and from $q = 32$ to $q = M_{31} = 2^{31} - 1$ we can block these attacks to the desired security level without a significant decrease in performance of WalnutDSA, rendering WalnutDSA completely secure against this attack.

Specifically we find that with these changes the signature storage size increased by 70%, the public key storage increased by 10x, and on the NIST test platform signature verification time only increased by 43%. Moreover, WalnutDSA still runs efficiently on all embedded platforms tested.

References

[1] I. Anshel, D. Atkins, D. Goldfeld, P.E. Gunnells, WalnutDSA$^{\mathrm{TM}}$: a quantum-resistant digital signature algorithm. Cryptology ePrint Archive, Report 2017/058 (2017).

[2] I. Anshel, D. Atkins, D. Goldfeld, P.E. Gunnells, Defeating the Hart, Kim, Micheli, Pascuel-Perez, Petit, Quek Attack on WalnutDSA$^{\mathrm{TM}}$, To appear.

[3] E. Artin, Theory of braids, *Ann. of Math.* (2) 48 (1947), 101–126.

[4] J. Birman; K. H. Ko; S. J. Lee, A new approach to the word and conjugacy problems in the braid groups, Adv. Math. 139 (1998), no. 2, 322–353.

[5] S. Blackburn, W. Buellens, Practical attacks against the Walnut digital signature scheme, Cryptology ePrint Archive, Report 2018/318 (2018).

[6] P. Dehornoy, A fast method for comparing braids, Adv. Math. 125 (1997), no. 2, 200–235.

[7] D. Hart, D. Kim, G. Micheli, G. Pascual-Perez, C. Petit, Y. Quek. A Practical Cryptanalysis of WalnutDSA TM. In: Abdalla M., Dahab R. (eds) *Public-Key Cryptography – PKC 2018. PKC 2018.* Lecture Notes in Computer Science, vol 10769. Springer, Cham. (2018)

[8] H.R. Morton, The multivariable Alexander polynomial for a closed braid, Low-dimensional topology, (Funchal, 1998), 167–172, Contemp. Math., 233, Amer. Math. Soc., Providence, RI, 1999.

[9] P.C. Van Oorschot, M.J. Wiener, Parallel collision search with cryptanalytic applications. Journal of cryptology 12(1), 1–28 (1999).

E-mail address: IANSHEL@SECURERF.COM, DATKINS@SECURERF.COM, DGOLDFELD@SECURERF.COM, PGUNNELLS@SECURERF.COM