DEFEATING THE HART, KIM, MICHELI, PASCUEL-PEREZ, PETIT, QUEK ATTACK ON WALNUTDSA^{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. Prior to its submission to NIST, Hart *et al* published an attack that, when it produces a signature forgery, it is found to be orders of magnitude longer than a valid signature making it invalid due to its length. In addition to being identified as a forgery by our current method, we show that with a modest parameter-only increase we can block this attack to the desired security level without a significant impact on the performance while making WalnutDSA completely secure against this attack.

1. Introduction

The digital signature algorithm known as WalnutDSA^{TM} was introduced in [1]. It is a group theoretic protocol which uses non linear operations in the Artin braid group B_N [2] 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.

Recently, Hart et al [5] proposed a practical forgery attack on WalnutDSA^{TM}. As pointed out by the authors, the attack can be defeated by increasing the parameter sizes, and that even in the range where the attack is successful, it produces forgeries that are many orders of magnitude larger than the signatures allowed in the protocol, i.e., the attack is blocked because the WalnutDSA^{TM} protocol specifies a length limit on the signatures.

We show in this paper that the run time of the attack is exponential and can be easily defeated while still retaining the high efficiency and low power consumption advantages of WalnutDSA^{TM} for constrained devices. For example, the attack can be completely thwarted and a 2^{128} (respectively 2^{256}) security level can be maintained by running WalnutDSA^{TM} on the braid group B_{10} and the finite field $\mathbb{F}_{M_{31}}$, where M_{31} is the Mersenne prime $2^{31} - 1$ (respectively B_{10}, M_{61}).

2. Brief Introduction to WalnutDSA^{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 [3] or the Dehornoy handle reduction algorithm [4]. 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 [6]. 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 $2 \leq i \leq N-1$, 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

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

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 Hart, Kim, Micheli, Pascuel-Perez, Petit, Quek Attack

The Hart et al attack [5] is a universal forgery attack that works in the special case when the two private keys w, w' are equal. The attack is based on a solution of the group factorization problem in $GL(N, \mathbb{F}_q)$.

Definition (Group Factorization Problem) *Let G be a finitely generated group with generators $\{g_1, \ldots, g_r\}$. Given $h \in G$ find a small integer L and sequences $(m_1, \ldots, m_L) \in \{1, 2, \ldots, r\}^L$ and*

$(\epsilon_1, \ldots \epsilon_L) \in \{\pm 1\}^L$ *such that*

$$h = \prod_{i=1}^{L} g_{m_i}^{\epsilon_i}.$$

We now explain how a solution to the group factorization problem can be used to forge signatures. Assume an attacker is in possession of many messages m_i and WalnutDSA$^{\text{TM}}$ signatures s_i with $i \in I$ in a finite indexing set. Let $E(H(m_i))$ denote the encoding of the hash of the message m_i into the braid group B_N and define $g_i := \text{Matrix}\big(\mathcal{P}(E(H(m_i))\big) \in GL(N, \mathbb{F}_q)$.

Assume that the attacker wants to forge a signature for a message m. Let $h = \text{Matrix}\big(\mathcal{P}(E(H(m))\big)$. Suppose the attacker can find $\epsilon_{i_j} \in \{\pm 1\}$ and a small positive integer L such that

$$h = \prod_{j=1}^{L} g_{i_j}^{\epsilon_{i_j}}$$

where $i_j \in I$ for $j = 1, 2, \ldots, L$. Then as shown in [5] a valid signature for m is given by $s = \prod_{j=1}^{L} s_{i_j}^{\epsilon_{i_j}}$.

The basic strategy for the attack is to build forgeries iteratively using a nested sequence of subgroups. In particular, there is a chain of subgroups $A_1 \subset A_2 \subset \cdots \subset A_{N-1}$ in $GL(N, \mathbb{F}_q)$, and a corresponding sequence of subgroups $P_1 \subset P_2 \subset \cdots \subset P_{N-1}$ of the braid group B_N. The two are related in that the matrix part under E-multiplication of any braid in P_i lands in A_i. The main step of the attack attempts to improve a partial solution of the problem in A_i, P_i to a more complete one in the smaller subgroups A_{i-1}, P_{i-1}. An essential role in building and improving solutions is played by the *distinguished point method*, which is a general collision attack on all one-way functions that has nothing to do with E-multiplication in particular.

7. Defeating the attack

In the Hart et al paper [5], the time complexity, memory complexity, and signature length are carefully estimated. Assume we are running WalnutDSA$^{\text{TM}}$ on the braid group B_n and finite field $\mathbb{F}_q$. They show that the running time complexity of the algorithm is

$$\approx 2 \cdot \gamma \cdot q^{\frac{N-1}{2}},$$

the memory complexity is

$$\log_2(q) \cdot N^2 q^{\frac{N-1}{2}},$$

while the forged signature length is

$$\ell \cdot q\big(\log_\gamma(q)\big)^{2N-3} N!\,(N-2)!,$$

where ℓ is the length of the original signature. Here the constant $\gamma \geq 1$ can be chosen by the attacker. They point out that if the parameters N, q are chosen as $q = 2^{16}$ and $N = 14$ then their attack is defeated with time complexity 2^{100}. It is clear that if we choose $q = M_{31} = 2^{31} - 1$, $N = 10$ then the attack is completely defeated with security level $> 2^{128}$ while if we choose $q = M_{61} = 2^{61} - 1$, $N = 10$, then we achieve security level at least 2^{256}. Even with much smaller choices of q, N the attack is still defeated because the forged signatures produced are significantly longer than the actual signatures.

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 $32 \times 32 \rightarrow 64$ 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 $32 \times 32 \rightarrow 32$(high) and $32 \times 32 \rightarrow 32$(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%.

8. Conclusion

WalnutDSA is a group-theoretic, public-key method that is part of the NIST Post-Quantum Cryptography standardization process. Hart *et al* published a practical signature forgery attack that could produce a signature forgery that is orders of magnitude longer than a valid signature (which is considered invalid due to its length). 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 this attack 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] Anshel, I., Atkins, D., Goldfeld, D., Gunnells, P.E.:WalnutDSA^{TM} : a quantum-resistant digital signature algorithm. Cryptology ePrint Archive, Report 2017/058 (2017).

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

[3] 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.

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

[5] Hart D., Kim D., Micheli G., Pascual-Perez G., Petit C., Quek Y. (2018) 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.

[6] 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.

Email address: IANSHEL@SECURERF.COM, DATKINS@SECURERF.COM, DGOLDFELD@SECURERF.COM, PGUNNELLS@SECURERF.COM