symmetry

MDPI

Article

The Cracking of *WalnutDSA*: A Survey

José Ignacio Escribano Pablos [1,2], María Isabel González Vasco [1], Misael Enrique Marriaga [1] and Ángel Luis Pérez del Pozo [1,*]

1 MACIMTE, U. Rey Juan Carlos, 28933 Móstoles, Spain
2 BBVA Next Technologies, 28050 Madrid, Spain
* Correspondence: angel.perez@urjc.es

Received: 8 July 2019; Accepted: 20 August 2019; Published: 23 August 2019

Abstract: This paper reports on the Walnut Digital Signature Algorithm (*WalnutDSA*), which is an asymmetric signature scheme recently presented for standardization at the NIST call for post-quantum cryptographic constructions. *WalnutDSA* is a group theoretical construction, the security of which relies on the hardness of certain problems related to an action of a braid group on a finite set. In spite of originally resisting the typical attacks succeeding against this kind of construction, soon different loopholes were identified rendering the proposal insecure (and finally, resulting in it being excluded from Round 2 of the NIST competition). Some of these attacks are related to the well-structured and symmetric masking of certain secret elements during the signing process. We explain the design principles behind this proposal and survey the main attack strategies that have succeeded, contradicting its claimed security properties, as well as the recently-proposed ideas aimed at overcoming these issues.

Keywords: *WalnutDSA*; digital signatures; post-quantum cryptography; cryptanalysis

1. Introduction

The (seemingly close) advent of quantum computing is urging the cryptographic community to search for new constructions that may withstand attacks arising from this new computing paradigm. Post-quantum cryptography is a bursting research area in which tools are designed for a scenario where honest users are restricted to classical computation, while the adversary may eventually have access to quantum computing resources. The American National Institute of Standards and Technology (NIST) initiated in December 2016 *"a process to develop and standardize one or more additional public-key cryptographic algorithms [...] that are capable of protecting sensitive government information well into the foreseeable future, including after the advent of quantum computers"* (see [1]).

Walnut Digital Signature Algorithm (*WalnutDSA*) was one of the 20 public key signature schemes presented for standardization at the recent NIST call for post-quantum cryptographic constructions. Different mathematical objects were used in these proposals such as lattice theory, coding theory, algebraic geometry (see for instance [2–4]), and, in the case of *WalnutDSA*, braid groups. After a first round of evaluations, only nine of these proposals remained under consideration. *WalnutDSA* failed to enter the second round, mostly due to a number of attacks that were reported during the one-year evaluation phase.

While it is not unusual that post-quantum cryptographic proposals lack a formal security evaluation within the theoretical framework known as *provable security*, the lack of a rigorous security analysis of *WalnutDSA* has been particularly damaging for the scheme's credit. In particular, it makes it difficult to identify the critical points to fend off in an implementation. As a result, ad-hoc fixes have been proposed by the scheme designers after each published attack. Nevertheless, the effectiveness of these fixes is somewhat hard to judge. Moreover, the actual hardness of the underlying mathematical problems is not well understood. The signature process is simple and symmetric, having two secret group elements acting on the encoded message to be signed. However, this simplicity has been exploited in many of the attacks against the scheme. Unfortunately, several computational problems defined over the main algebraic environment of *WalnutDSA* (i.e., braid groups) have turned out to be hard to exploit cryptographically, mainly because the computational complexity of such problems may be high in a worst-case definition, while it is unclear how to produce hard instances effectively.

In this document, we give a self-contained review of *WalnutDSA*, detailing the proposal and describing the main attacks that have been presented against this construction, as well as the possible fixes, currently under discussion, towards a secure implementation of this signature scheme.

Paper roadmap: We start with a short section reviewing the history of braid group cryptography, followed by a section explaining the basics on signature schemes. Then, we give a comprehensive description of *WalnutDSA* in Section 4. Section 5 is devoted to the various attack strategies deployed against *WalnutDSA*; from the early factoring attacks (see Section 5.1), to collision attacks (Section 5.2), attempts to undermine the (claimed) one-wayness of the underlying *E-multiplication* function (Section 5.3), and finally, the recent (and probably the most devastating) attack aiming at the recovery of an alternative secret key by solving a certain rewriting problem (see Section 5.4). The survey wraps up with a short conclusion section.

2. Braid Group Cryptography

Cryptography based on braid groups was born almost 20 years ago and attracted plenty of attention from group theorists, as well as the cryptographic community. The reasons for this are diverse: the schemes were mathematically appealing and the constructions likely to be efficient enough to be practical. Unfortunately, many problems were brought to light after a thorough scrutiny carried out by pure mathematicians and cryptographers. In this section, we briefly review two of the most prominent proposals within this area and refer the interested reader to the survey on the topic by David Garber [5].

2.1. Cryptographic Constructions Using Braid Groups

The two flagship proposals made for deriving cryptographic constructions using braid groups are a key exchange protocol and a public key encryption scheme.

In 1999, Anshel, Anshel, and Goldfeld [6] introduced a generic two-party key establishment protocol. Their presentation could be translated into various implementations with different algebraic structures as a base (and, of course, security levels). The one using braid groups attracted the most attention. The security of this construction relied on the hardness of the so-called *multiple simultaneous conjugacy search problem* (see below) in the braid group.

Later, at CRYPTO 2000, Ko et al. [7] put forward a braid-based version of the Diffie–Hellman two-party key exchange protocol, as well as an encryption scheme á la ElGamal derived from such a protocol. The main idea behind this construction is as follows: Fix a public braid g. Using this public information and exchanging messages through a public channel, two users may establish a shared high entropy secret. This secret is derived from a braid of the form $(ab)g(ab)^{-1}$, which is constructed by letting each user choose a secret conjugating element (a and b respectively) and publicly interchanging the elements aga^{-1} and bgb^{-1}. Indeed, for this idea to work, the conjugating braids a and b should commute. Furthermore, the hardness of the underlying *conjugacy search problem* (see below) in the braid group is crucial for the security of the scheme, since extracting a or b from the public messages aga^{-1} and bgb^{-1} is enough to deduce the exchanged key.

2.2. Computational Problems in Braid Groups

Many cryptographic proposals (like the ones mentioned above) based their security in computational problems related to the so-called *conjugacy problem* in B_n, the braid group on

$n > 0$ strands. However, assuming that these problems are hard is not always reasonable. Indeed, efficient algorithms for special cases of these problems have been behind the cryptanalysis of most of the cryptographic proposals designed using braid groups. Some examples of such problems are:

- *Conjugacy Decision Problem* (CDP). Given $A, B \in B_n$, determine whether they are conjugate, i.e., whether there exists $X \in B_n$ such that $A = X^{-1}BX$.
- *Conjugacy Search Problem* (CSP). Given $A, B \in B_n$, known to be conjugate, compute $X \in B_n$ such that $A = X^{-1}BX$.
- *Braid Diffie–Hellman Decision Problem* (BDHDP). Given $A, B, C, D \in B_n$, such that there exist $X, Y \in B_n$ satisfying $B = X^{-1}AX$ and $C = Y^{-1}AY$, with $XY = YX$, determine whether $D = Y^{-1}X^{-1}AXY$.
- *Braid Diffie–Hellman Search Problem* (BDHSP). Given $A, B, C \in B_n$, such that there exist $X, Y \in B_n$ satisfying $B = X^{-1}AX$ and $C = Y^{-1}AY$, with $XY = YX$, compute $D = Y^{-1}X^{-1}AXY$.
- *Multiple Simultaneous Conjugacy Search Problem* (MSCSP). Given k pairs of elements $(U_i, W_i) \in B_n^2$, such that they are all conjugates with respect to the same braid, find such a conjugating braid, i.e., compute $X \in B_n$ such that $W_i = X^{-1}U_iX$, for all $i = 1, \ldots, k$.
- *Decomposition Problem* (DP). Let G be a fixed subgroup of B_n. Given $A, B \in B_n$, find $X, Y \in G$ such that $B = XAY$.
- *Root Extraction Problem* (REP). For $A \in B_n$ and $r \in \mathbb{N}$ such that there exists $B \in B_n$ with $A = B^r$, compute such a braid B.

It is easy to see that there are close relations among the above problems. Let us focus on how to solve CSP and CDP. As explained in detail in [5], the basic idea that has proven more fruitful towards a solution for the CSP and CDP problems involves a set I_x for each braid x (typically a subset of the conjugacy class of A), which characterizes the conjugacy class (i.e., A and B are conjugates if and only if $I_A = I_B$). Furthermore, there should be an efficient algorithm to compute a representative $\hat{A} \in I_A$ and a *witness* $X \in B_n$, such that $X^{-1}AX = \hat{A}$. Last, it should be possible to construct the full set I_A in a finite number of steps, starting from any representative $\hat{A}$. Now, given two braids $A, B \in B_n$, specifying an instance of CSP or CDP, one should:

(i) find representatives $\hat{A} \in I_A$ and $\hat{B} \in I_B$;
(ii) compute elements of I_A (storing the corresponding witnesses) until either:

   (a) $\hat{B}$ is found as an element of I_A, proving A and B to be conjugate and providing a conjugating element or
   (b) the entire set I_A is constructed without finding $\hat{B}$, proving that A and B are not conjugate.

Several choices of the special sets I_A can be found in the literature: summit sets, super summit sets, ultra summit sets, reduced supper summit sets, etc. All of them are subsets of the conjugacy class of the corresponding braid A. Of course, choosing a simpler and smaller set results in a more efficient algorithm derived from the above strategy. Using the above technique and other sophisticated geometric techniques, Birman, Gebhardt, and González Meneses [8] provided a polynomial-time algorithm to solve the CSP involving the so-called *periodic braids*. Furthermore, the same authors proved that the problem would be solved for all instances if a polynomial-time algorithm for a special type of braid (*rigid braids*) was found.

However, not only full theoretical solutions for the conjugacy problems have been of interest in the cryptographic context; indeed, heuristic algorithms with a significant success rate suffice to thwart the security of a scheme that is based on one of the above problems (we refer again to [5] for details). As a consequence, all cryptographic proposals built around the above problems are currently considered problematic.

3. Basics on Signature Schemes

In this section, we recall some basic concepts related to public key digital signature schemes and the assessment of provable security for these cryptographic tools. Many of the definitions below are taken from [9,10].

Definition 1. *A digital signature scheme is a triplet of algorithms* $(\mathcal{G}, \boldsymbol{\Sigma}, \mathcal{V})$ *where:*

- $\mathcal{G}$*, the key generation algorithm, is a probabilistic algorithm that takes as input* $\mathbf{1}^{\lambda}$ *(for a security parameter* $\lambda \in \mathbb{N}$*) and returns a pair* $(\mathbf{pk}, \mathbf{sk})$ *of public and secret keys, from a designated key space of polynomial size in* λ*.*
- $\boldsymbol{\Sigma}$*, the signing algorithm, is a probabilistic algorithm that takes as input a given message* $m \in \mathcal{M}_{\lambda}$ *(for a fixed message space) and a secret key* $\mathbf{sk}$ *and returns a signature* `sig` *(also assumed to belong to a prescribed set of polynomial size in* λ*). ss of generality, we can assume that each* $\mathcal{M}_{\lambda}$ *consists of bitstrings of polynomial size in* λ*. In the sequel, we often drop the subscript* λ *for the sake of readability*
- $\mathcal{V}$*, the verification algorithm, is a deterministic algorithm that takes as input a given signature* `sig`*, a message* $m \in \mathcal{M}_{\lambda}$*, and a public key* $\mathbf{pk}$ *and outputs a bit in* $\{0, 1\}$*, checking if* `sig` *is a valid signature of m with respect to* $\mathbf{pk}$*.*

Typically, a correctness requirement is imposed, establishing that $\mathcal{V}$ outputs one if it gets a valid signature as the input. The fact that it should output zero for an invalid signature is typically captured by the different definitions of security.

3.1. Security Notions for Signature Schemes

Prior to giving formal definitions of security notions, we informally list the different adversarial goals and attack models, which attempt to capture the main attack strategies that should be prevented for each specific adversary. Let $\mathcal{A}$ denote a (probabilistic polynomial-time) adversary. We assume that $\mathcal{A}$ pursues one of the following *adversarial goals*:

- Existential Forgery (EF): $\mathcal{A}$ tries to produce a valid signature for a message m, not necessarily adversarial chosen.
- Selective Forgery (SF): $\mathcal{A}$ tries to produce a valid signature for some adversarial chosen fixed message m.
- Universal Forgery (UF): $\mathcal{A}$ aims at producing a valid signature for *any* given message.
- Total Break (TB): $\mathcal{A}$ tries to retrieve, from the public information, a legitimate signer's secret key.

Similarly, in order to capture adversarial capabilities, we distinguish among the following *attack models*:

- No Message Attack (NMA): $\mathcal{A}$ only knows the public parameters (in particular, the public signing key).
- Random Message Attack (RMA): $\mathcal{A}$ is given signatures on a sequence of messages selected uniformly at random.
- Chosen Message Attack (CMA): $\mathcal{A}$ is given access to a signing oracle, which signs any message chosen by $\mathcal{A}$. Queries to this oracle can be adaptive, i.e., $\mathcal{A}$ may adapt the input messages based on previous output signatures.

Formal security notions are introduced by combining adversarial goals and capabilities. For instance, a signature scheme is secure in the sense of UF-NMA if given any probabilistic polynomial-time adversary $\mathcal{A}$, there exists a negligible function of the security parameter bounding the probability of success of a UF attack, provided that $\mathcal{A}$ has access only to public information (NMA). Other security notions are defined analogously; for instance, EUF-CMA captures the fact that a CMA adversary will not be able to produce an existential forgery.

Now, we give precise definitions for the three security notions, which are relevant throughout this work.

Definition 2. *A signature scheme $(\mathcal{G}, \mathbf{\Sigma}, \mathcal{V})$ with message space $\mathcal{M}$ and security parameter λ is said to be universally unforgeable under no-message attacks (UF-NMA) if for any probabilistic polynomial-time adversary $\mathcal{A}$ and $\forall m \in \mathcal{M}$, then:*

$$\Pr\begin{bmatrix}(\mathbf{pk}, \mathbf{sk}) \leftarrow \mathcal{G}(\mathbf{1}^\lambda);\\ \mathtt{sig} \leftarrow \mathcal{A}(\mathbf{pk}, m);\\ \mathcal{V}(m, \mathtt{sig}, \mathbf{pk}) = 1\end{bmatrix} \le \operatorname{negl}(\lambda).$$

Definition 3. *A signature scheme $(\mathcal{G}, \mathbf{\Sigma}, \mathcal{V})$ with message space $\mathcal{M}$ and security parameter λ is said to be universally unforgeable under random-message attacks (UF-RMA) if for any probabilistic polynomial-time adversary $\mathcal{A}$ and $\forall m \in \mathcal{M}$, then:*

$$\Pr\begin{bmatrix}(\mathbf{pk}, \mathbf{sk}) \leftarrow \mathcal{G}(\mathbf{1}^\lambda);\\ \{m_i\}_{i=1}^k \overset{\$}{\leftarrow} \mathcal{M} \setminus \{m\};\\ \{\mathtt{sig}_i\}_{i=1}^k \leftarrow \mathbf{\Sigma}(\mathbf{sk}, \{m_i\}_{i=1}^k);\\ \mathtt{sig} \leftarrow \mathcal{A}(\mathbf{pk}, \{(m_i, \mathtt{sig}_i)\}_{i=1}^k, m);\\ \mathcal{V}(m, \mathtt{sig}, \mathbf{pk}) = 1\end{bmatrix} \le \operatorname{negl}(\lambda).$$

The above definition states that when given a list of message-signature pairs, where the messages are selected uniformly at random, the adversary should still have only a negligible probability of constructing a new valid signature pair.

Definition 4. *A signature scheme $(\mathcal{G}, \mathbf{\Sigma}, \mathcal{V})$ with message space $\mathcal{M}$ and security parameter λ is said to be existentially unforgeable under adaptive chosen-message attacks (EUF-CMA) if for any probabilistic polynomial-time adversary $\mathcal{A}$ with polynomial access to a signing oracle $\mathcal{O}_{\mathbf{sk}}$ that produces valid signatures with respect to a certain secret key $\mathbf{sk}$, then:*

$$\Pr\begin{bmatrix}(\mathbf{pk}, \mathbf{sk}) \leftarrow \mathcal{G}(\mathbf{1}^\lambda);\\ \{\mathtt{sig}_i\}_{i=1}^k \leftarrow \mathcal{O}_{\mathbf{sk}}(\{m_i\}_{i=1}^k);\\ (m, \mathtt{sig}) \leftarrow \mathcal{A}(\mathbf{pk}, \{(m_i, \mathtt{sig}_i)\}_{i=1}^k);\\ \mathcal{V}(m, \mathtt{sig}, \mathbf{pk}) = 1 \ \textit{and} \ m \notin \{m_1, \dots, m_k\}\end{bmatrix} \le \operatorname{negl}(\lambda).$$

In the above definition, the adversary is given access to a signing oracle that produces valid signatures with respect to the key pair under attack and faces the challenge of producing a valid signature for a message. This model is particularly relevant for capturing malleability attacks, which exploit the possibility of deriving new valid signatures from legitimate ones.

The standard security definition for signature schemes is EUF-CMA, which is the strongest among the three notions we have introduced. More precisely, every EUF-CMA scheme is UF-RMA, and in turn, every UF-RMA scheme is UF-NMA.

4. Scheme Description

In this section, we describe the *Walnut Digital Signature Algorithm (WalnutDSA)* introduced in [11]. This construction relies on certain computational properties of nonlinear operations in the *Artin braid group B_N* [12] combined with operations in $\mathrm{GL}_N(\mathbb{F}_q)$, the group of non-singular $N \times N$ matrices with entries in the finite field $\mathbb{F}_q$ with q elements.

Informally, in *WalnutDSA*, the message to be signed is hashed and encoded as a braid in B_N, (see Section 4.1). The private key consists of a pair of braids, while an ordered set of N elements in $\mathbb{F}_q$ and a pair of elements of the set $\mathrm{GL}_N(\mathbb{F}_q) \times S_N$ form the public key (As usual, S_N is the group of permutations of $\{1, 2, \dots, N\}$). Key generation is described in detail in Section 4.2. In order to render brute force attacks ineffective, the key space is made sufficiently large by choosing $N \geq 8$ and $q \geq 32$. A signature is built from the encoded message, the private keys, and two additional braids used to obscure the private key. Valid signatures must verify a certain equation involving the public key, the encoded message, and *E-multiplication*, a group-theoretic one-way function introduced in [13]. All these algorithms are precisely described in Section 4.3. Let us start here by describing the mathematical ingredients needed to understand them.

4.1. Message Encoding

WalnutDSA encodes messages as elements in the Artin braid group, which is a nice algebraic and computational habitat.

4.1.1. Braids

Informally, the *braid group with N strands* B_N is a non-Abelian group whose elements can be described as a configuration of N non-intersecting vertical or horizontal strands in three-dimensional space, with ends fixed on two parallel disks. Moreover, the strands flow in one direction without turning back, so that any plane parallel to the disks will intersect each strand exactly once. Multiplication of two braids is defined as concatenation of strands, and two braids are considered equal if one can be continuously transformed into the other, keeping the ends fixed and without intersecting the strands.

More precisely, the braid group with N strands is defined as follows [12]. For $N \geq 2$, B_N is a group generated by the *Artin generators* $\{b_1, b_2, \ldots, b_{N-1}\}$, subject to the following relations:

$$\begin{array}{rcll} b_i\, b_{i+1}\, b_i &=& b_{i+1}\, b_i\, b_{i+1}, & 1 \leq i \leq N-2, \\ b_i\, b_j &=& b_j\, b_i, & |i-j| \geq 2. \end{array} \tag{1}$$

The Artin generator b_i represents the braid where the ith strand crosses over the $(i+1)$th strand. The relation $b_i\, b_{i+1}\, b_i = b_{i+1}\, b_i\, b_{i+1}$ for $1 \leq i \leq N-2$, corresponds to moving the ith strand over the crossing of the $(i+1)$th and the $(i+2)$th strand, and the relations $b_i\, b_j = b_j\, b_i$ for $|i-j| \geq 2$ correspond to the fact that crossings that do not share strands commute.

Any braid $b \in B_N$ can be expressed as a product of the Artin generators and their inverses, that is,

$$b \;=\; b_{i_1}^{e_1}\, b_{i_2}^{e_2} \cdots b_{i_k}^{e_k}, \tag{2}$$

where $1 \leq i_n \leq N-1$ and $e_n \in \{-1, 1\}$. Clearly, the expression for b is not unique since applying (1) yields infinite equivalent expressions.

Let S_N be the symmetric group of order N. There exists a group homomorphism $\sigma : B_N \to S_N$ defined as follows. For each Artin generator b_i, $\sigma_{b_i}(i) = \sigma_{b_i^{-1}}(i) = i+1$ and $\sigma_{b_i}(j) = \sigma_{b_i^{-1}}(j) = j$ for $1 \leq j \leq N-1$, such that $i \neq j$. That is, b_i and b_i^{-1} are mapped into the element in S_N, which interchanges the i^{th} and the $(i+1)^{\text{th}}$ elements of $\{1, 2, \ldots, N\}$ and leaves the rest fixed. Notice that $\sigma_{b_i}^{-1} = \sigma_{b_i}$ for $1 \leq i \leq N-1$. Moreover, for $1 \leq i, j \leq N-1$, $\sigma_{b_i^{\pm 1} b_j^{\pm 1}} = \sigma_{b_i}\, \sigma_{b_j}$. Hence, for any braid $b \in B_N$ as in (2), we have:

$$\sigma_b \;=\; \sigma_{b_{i_1}}\, \sigma_{b_{i_2}} \cdots \sigma_{b_{i_k}}.$$

If σ_b is the identity element of S_N, then b is called a *pure braid*. In other words, a braid is a pure braid if and only if it is in the kernel of σ.

4.1.2. Encoding

WalnutDSA requires the permutation linked to each encoded message to be the identity. Thus, the encoded message must be a pure braid.

The encoding algorithm utilizes the following collection of pure braids:

$$g_{N,i} \;=\; b_{N-1}\,b_{N-2}\cdots b_{i+1}\,b_i^2\,b_{i+1}^{-1}\cdots b_{N-2}^{-1}\,b_{N-1}^{-1}, \qquad 1 \le i \le N-1.$$

This collection of pure braids generates a free subgroup of B_N [14],that is the set of products of $g_{N,i}$, $1 \le i \le N-1$, that satisfy no relations except those implied by the group axioms (e.g., $a\,b = a\,c\,c^{-1}\,b$, but $a \neq b^{-1}$ for $a, b, c \in B_N$) ([15], Chapter 7). Any subset of the above collection of pure braids will generate a free subgroup.

Let $m \in \{0,1\}^*$ be a message, and let $H : \{0,1\}^* \to \{0,1\}^{4\ell}$, $\ell \ge 1$, denote a cryptographically-secure hash function. Fix any four generators $g_{N,j_1}, g_{N,j_2}, g_{N,j_3}, g_{N,j_4}$, and denote by $C_{N,4}$ the free subgroup generated by these four generators. Define the encoding function $E : \{0,1\}^{4\ell} \to C_{N,4}$ as follows. The hashed message $H(m)$ is broken into ℓ 4-bit blocks. For the k^{th} block, the first two bits determine a generator $g_{N,j_{n_k}}$, $1 \le n_k \le 4$, and the next two bits determine an integer $1 \le p_k \le 4$. Then,

$$E(H(m)) = g_{N,j_{n_1}}^{p_1}\, g_{N,j_{n_2}}^{p_2} \cdots g_{N,j_{n_\ell}}^{p_\ell},$$

written in its reduced form, that is products of the form $b_i\,b_i^{-1}$ and $b_i^{-1}\,b_i$, $1 \le i \le N-1$, are erased from the braid (see [16,17] for examples of reduction algorithms). This encoding algorithm ensures that each message is mapped to a unique reduced element of the free subgroup generated by $g_{N,j_1}, g_{N,j_2}, g_{N,j_3}, g_{N,j_4}$.

4.2. Key Generation

The security of *WalnutDSA* relies on E-multiplication, a function that maps braids in B_N to elements in the set $\mathrm{GL}_N(\mathbb{F}_q) \times S_N$. This mapping is based on the colored Burau representation of B_N. We provide some preliminaries before describing the public and private keys in *WalnutDSA*.

4.2.1. Colored Burau Representation of the Braid Groups

Let $\mathcal{L}_{\mathbb{F}_q} \equiv \mathcal{L}_{\mathbb{F}_q}[t_1, t_2, \ldots, t_N]$ denote the ring of Laurent polynomials in the variables $t_1, t_2, \ldots, t_N$ with coefficients in $\mathbb{F}_q$, that is,

$$\mathcal{L}_{\mathbb{F}_q} = \left\{ \sum_{j=0}^{k} a_j\, t_1^{n_{1,j}}\, t_2^{n_{2,j}} \cdots t_N^{n_{N,j}} \;:\; n_{i,j} \in \mathbb{Z},\; a_j \in \mathbb{F}_q,\; k \ge 0 \right\}.$$

For each Artin generator, we define the following $N \times N$ matrices [18]:

$$\mathtt{CB}_{b_1}(t_1) \;=\; \left(\begin{array}{cc|c} -t_1 & 1 & \mathtt{0} \\ 0 & 1 & \mathtt{0} \\ \mathtt{0} & \mathtt{0} & I_{N-2} \end{array}\right),$$

$$\mathtt{CB}_{b_1^{-1}}(t_2) \;=\; \left(\begin{array}{cc|c} -t_2^{-1} & t_2^{-1} & \mathtt{0} \\ 0 & 1 & \mathtt{0} \\ \mathtt{0} & \mathtt{0} & I_{N-2} \end{array}\right),$$

$$\mathtt{CB}_{b_i}(t_i) \;=\; \left(\begin{array}{c|ccc|c} I_{i-2} & \mathtt{0} & \mathtt{0} & \mathtt{0} & \mathtt{0} \\ \hline \mathtt{0} & 1 & 0 & 0 & \mathtt{0} \\ \mathtt{0} & t_i & -t_i & 1 & \mathtt{0} \\ \mathtt{0} & 0 & 0 & 1 & \mathtt{0} \\ \mathtt{0} & \mathtt{0} & \mathtt{0} & \mathtt{0} & I_{N-i-1} \end{array}\right), \qquad 2 \le i \le N-1,$$

$$\mathtt{CB}_{b_i^{-1}}(t_{i+1}) \;=\; \left(\begin{array}{c|ccc|c} I_{i-2} & \mathtt{0} & \mathtt{0} & \mathtt{0} & \mathtt{0} \\ \hline \mathtt{0} & 1 & 0 & 0 & \mathtt{0} \\ \mathtt{0} & 1 & -t_{i+1}^{-1} & t_{i+1}^{-1} & \mathtt{0} \\ \mathtt{0} & 0 & 0 & 1 & \mathtt{0} \\ \mathtt{0} & \mathtt{0} & \mathtt{0} & \mathtt{0} & I_{N-i-1} \end{array}\right), \quad 2 \le i \le N-1,$$

where I_n is the identity matrix of size $n \times n$ and $\mathtt{0}$ is the zero matrix of adequate size.

Let $M \in \mathrm{GL}_N(\mathcal{L}_{\mathbb{F}_q})$ and $\pi \in S_N$. We define:

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

The product of $(A,\ \pi)$ and $(B,\ \tau)$ in $\mathrm{GL}_N(\mathcal{L}_{\mathbb{F}_q}) \times S_N$ is defined as:

$$(A,\ \pi) \cdot (B,\ \tau) = (A \cdot {}^{\pi}B,\ \pi\tau).$$

We have that the elements of $\mathrm{GL}_N(\mathcal{L}_{\mathbb{F}_q}) \times S_N$ form a group under this product operation.

Now, we define the colored Burau representation:

$$\Pi_{\mathrm{CB}} : B_N \to \mathrm{GL}_N(\mathcal{L}_{\mathbb{F}_q}) \times S_N.$$

For any Artin generator b_i, $1 \le i \le N-1$,

$$\Pi_{\mathrm{CB}}(b_i^{\pm 1}) = (\mathtt{CB}_{b_i^{\pm 1}},\ \sigma_{b_i}),$$

and for all b_i and b_j, $1 \le i, j \le N-1$,

$$\Pi_{\mathrm{CB}}(b_i^{\pm 1}\, b_j^{\pm 1}) \ = \ \Pi_{\mathrm{CB}}(b_i^{\pm 1}) \cdot \Pi_{\mathrm{CB}}(b_j^{\pm 1}).$$

More generally, for any braid $b \in B_N$ as in (2),

$$\Pi_{\mathrm{CB}}(b) \ = \ \Pi_{\mathrm{CB}}(b_{i_1}^{e_1}) \cdot \Pi_{\mathrm{CB}}(b_{i_2}^{e_2}) \cdots \Pi_{\mathrm{CB}}(b_{i_k}^{e_k}).$$

It can be verified that Π_{CB} is a homomorphism that preserves the braid relations (1) and, hence, defines a representation of B_N.

4.2.2. E-Multiplication

The key generation in *WalnutDSA* is based on E-multiplication, a group-theoretic one-way function introduced in [13]. Here, we recall its definition.

Fix a finite field $\mathbb{F}_q$ and a set of N non-zero elements in $\mathbb{F}_q$,

$$\mathcal{T} = \{y_1, y_2, \ldots, y_N\} \subset \mathbb{F}_q.$$

For every $M \in \mathrm{GL}_N(\mathcal{L}_{\mathbb{F}_q})$, we define:

$$M \downarrow_{\mathcal{T}} = \ M(y_1, y_2, \ldots, y_N) \ \in \ \mathrm{GL}_N(\mathbb{F}_q).$$

Now, we define E-multiplication.

Definition 5. *E-multiplication, denoted by $\star$, is a right action of the group $\Pi_{\mathrm{CB}}(B_N)$ on the set $\mathrm{GL}_N(\mathbb{F}_q) \times S_N$, defined inductively as follows. Given $(M, \pi) \in \mathrm{GL}_N(\mathbb{F}_q) \times S_N$,*

$$(M, \pi) \star \Pi_{\mathrm{CB}}(b_i^{\pm 1}) \ = \ (M \cdot {}^{\pi}\mathtt{CB}_{b_i^{\pm 1}} \downarrow_{\mathcal{T}},\ \pi\sigma_{b_i}) \ \in \ \mathrm{GL}_N(\mathbb{F}_q) \times S_N, \qquad 1 \le i \le N-1.$$

More generally, for any braid $b \in B_N$ as in (2),

$$(M, \pi) \star \Pi_{\mathrm{CB}}(b) \ = \ (M, \pi) \star \Pi_{\mathrm{CB}}(b_{i_1}^{e_1}) \star \Pi_{\mathrm{CB}}(b_{i_2}^{e_2}) \star \cdots \star \Pi_{\mathrm{CB}}(b_{i_k}^{e_k}),$$

where the operations are done from left to right. Furthermore, for convenience, we will write $(M, \pi) \star b$ instead of $(M, \pi) \star \Pi_{\mathrm{CB}}(b)$.

4.2.3. Key Generation Mechanism

The signer's private key consists of two random braids s_1 and s_2, written in reduced form, such that s_1, s_2, and $s_1 s_2$ are not pure braids. No further prerequisites were made explicit in the original proposal of *WalnutDSA*.

Let $b \in B_N$ be a braid and $\mathcal{T} \subset \mathbb{F}_q$ a fixed set of N non-zero elements. Define:

$$\mathcal{P}(b) \equiv \mathcal{P}_{\mathcal{T}}(b) \;=\; (I_N,\, \iota_N) \star b \;\in\; \mathrm{GL}_N(\mathbb{F}_q) \times S_N,$$

where I_N is the identity $N \times N$ matrix and $\iota_N \in S_N$ is the identity permutation. The signer's public key consists of:

- $\mathcal{T} = \{y_1, y_2, \ldots, y_N\} \subset \mathbb{F}_q$ such that $y_i \neq 0, 1 \leq i \leq N$, and $y_a = y_b = 1$ for some $1 \leq a, b \leq N$,
- $\mathcal{P}(s_1)$, and
- the matrix component of $\mathcal{P}(s_2)$, denoted by $\mathtt{mat}(\mathcal{P}(s_2))$, that is, $\mathtt{mat}(\mathcal{P}(s_2))$.

4.3. Signature Generation and Verification

We now describe *WalnutDSA* in detail.

4.3.1. Cloaking Elements

First, we discuss elements in the subgroup of pure braids that essentially disappear when performing E-multiplication. The purpose of these elements is to *cloak*, or hide, the private key used to construct the signature.

Definition 6. *Let $(M, \pi) \in \mathrm{GL}_N(\mathbb{F}_q) \times S_N$, and let $\mathcal{T}$ be a fixed set of N non-zero elements of $\mathbb{F}_q$. A pure braid $v \in B_N$ is called a cloaking element of (M, π) if:*

$$(M, \pi) \star v = (M, \pi).$$

It is clear from this definition that the set of cloaking elements of (M, π) depends on the set $\mathcal{T}$. The existence of cloaking elements is discussed in the following proposition.

Proposition 1. *Fix integers $N \geq 2$, $1 \leq a, b \leq N$, and fix a set of N non-zero elements $\mathcal{T} = \{y_1, y_2, \ldots, y_N\} \subset \mathbb{F}_q$ such that $y_a = y_b = 1$. Let $(M, \pi) \in \mathrm{GL}_N(\mathbb{F}_q) \times S_N$, b_i, $1 \leq i \leq N-1$, an Artin generator of B_N, and $w \in B_N$ such that:*

$$\pi\sigma_w(i) \;=\; a, \quad \text{and} \quad \pi\sigma_w(i+1) \;=\; b.$$

Then,

$$v \;=\; w\, b_i^2\, w^{-1},$$

is a cloaking element of (M, π).

4.3.2. Signature Generation

Fix a hash function $H : \{0,1\}^* \to \{0,1\}^{4\ell}$, $\ell \geq 1$, and let $s_1, s_2 \in B_N$ be the braids in the private key. In *WalnutDSA*, a signature for the message $m \in \{0,1\}^*$ is the braid:

$$\mathtt{sig} \;=\; v_1 \cdot s_1^{-1} \cdot v \cdot E(H(m)) \cdot s_2 \cdot v_2 \;\in\; B_N,$$

written in reduced form, where $v, v_1, v_2 \in B_N$ are cloaking elements of (I_N, ι_N), $\mathcal{P}(s_1)$, and $\mathcal{P}(s_2)$, respectively.

4.3.3. Signature Verification

The verification algorithm calculates the matrix component of $\mathcal{P}(s_1) \star \texttt{sig}$ and $\mathcal{P}(E(H(m)))$, denoted by $\texttt{mat}(\mathcal{P}(s_1) \star \texttt{sig})$ and $\texttt{mat}(\mathcal{P}(E(H(m))))$, respectively, and accepts the signature if the following equation holds:

$$\texttt{mat}(\mathcal{P}(s_1) \star \texttt{sig}) \;=\; \texttt{mat}(\mathcal{P}(E(H(m)))) \cdot \texttt{mat}(\mathcal{P}(s_2)).$$

5. Cryptanalysis of *WalnutDSA*

5.1. Factoring Attacks

The essential idea behind these attacks is to forge a signature for any given message m solving a factorization problem in groups, defined as follows:

Definition 7 (Factorization problem in groups)**.** *Let G be a group; let $\Gamma = \{g_1, \dots, g_\gamma\}$ be a generating set for G; and let $h \in G$. Find an integer L and sequences $(k_1, \dots, k_L) \in \{1, \dots, \gamma\}^L$ and $(\varepsilon_1, \dots, \varepsilon_L) \in \{\pm 1\}^L$ such that:*

$$h = \prod_{i=1}^{L} g_{k_i}^{\varepsilon_i}.$$

A solution to a specific instance of this problem has been exploited by several authors [19,20] to construct a new valid signature from several valid signatures, in order to violate UF-CMA. More precisely, Hart et al. presented in [19] an efficient method to compute, given a couple of signatures on random messages, a new signature on an arbitrary message. However, these forged signatures were significantly longer than those constructed by the honest signer. The design of *WalnutDSA* was modified by the authors in order to defeat this attack, yet a refinement of this method, presented in Section 3 of [20], rendered this modification insufficient.

5.1.1. Factoring For Universal Forgeries: The Attacks by Hart Et Al., and Beullens and Blackburn

The strategy behind [19] allows for constructing a valid signature for any arbitrary message m (and is thus a universal forgery). More precisely, Proposition 4 in [19] states that, given a finite set of signatures:

$$\mathcal{S} = \{(m_i, \texttt{sig}_i) : \; i \in I\}$$

and taking g_i as the matrix part of $\mathcal{P}(E(H(m_i)))$ for all $i \in I$, it holds that, if the matrix part h of $\mathcal{P}(E(H(m)))$ can be factored with respect to the generating set $\{g_i \mid i \in I\}$, then constructing the very same *word* replacing each g_i with the corresponding braids s_i from $\mathcal{S}$ yields a valid signature for m.

Beullens and Blackburn explained how to exploit this malleability property through the following simple theorem:

Theorem 1 (Theorem 1 from [20])**.** *Consider the version of WalnutDSA, where it holds that $s_1 = s_2$. Suppose m, m_1, m_2 are three messages. Let h, h_1, h_2 be the matrix part of $\mathcal{P}(E(H(m)))$, $\mathcal{P}(E(H(m_1)))$, $\mathcal{P}(E(m_2))$, respectively. Then,*

1. *If $h = h_1^{-1}$ and $\texttt{sig}_1$ is a valid signature for m_1, then $\texttt{sig}_1^{-1}$ is a valid signature for m.*
2. *If $h = h_1 \cdot h_2$ and $\texttt{sig}_1$, $\texttt{sig}_2$ are valid signatures for m_1 and m_2, respectively, then $\texttt{sig}_1 \cdot \texttt{sig}_2$ is a valid signature for m.*

However, the above result is only valid if the public braids s_1 and s_2 coincide, which was only the case in the first versions of the proposal [11]. All in all, a simple variant of the above theorem, presented in [20], evidences that choosing $s_1 \neq s_2$ does not amend the strong malleability inherent to *WalnutDSA*:

Theorem 2 ([20]). *Suppose m, m_1, m_2 are three messages. Let h, h_1, h_2 be the matrix part of $\mathcal{P}(E(H(m)))$, $\mathcal{P}(E(H(m_1)))$, $\mathcal{P}(E(H(m_2)))$, respectively. Let s_1, s_2, $s_3 \in B_N$ be three braids. Then,*

1. *If $h = h_1^{-1}$ and $\mathtt{sig}_1$ is a valid signature for m_1 under the public key $(\mathcal{P}(s_1), \mathcal{P}(s_2))$, then $\mathtt{sig}_1^{-1}$ is a valid signature for m under the public key $(\mathcal{P}(s_2), \mathcal{P}(s_1))$.*
2. *If $h = h_1 \cdot h_2$ and $\mathtt{sig}_1 \cdot \mathtt{sig}_2$ are valid signatures for m_1 and m_2 under the public keys $(\mathcal{P}(s_1), \mathcal{P}(s_2))$ and $(\mathcal{P}(s_2), \mathcal{P}(s_3))$, respectively, then $\mathtt{sig}_1 \cdot \mathtt{sig}_2$ is a valid signature for m under the public key $(\mathcal{P}(s_1), \mathcal{P}(s_3))$.*

Note that the above theorems do not impose a practical restriction on the forged message m, for suitable m_1, m_2 can be constructed for any m in order to mount the UF attack. Still, the forged signatures obtained through these factoring strategies are many orders of magnitude longer than legitimate signatures; thus, imposing length limits on the output signatures (as the authors did in the implementation submitted to the NIST PQCstandardization call) is enough to dodge these attacks.

5.1.2. Factoring Using the Garside Normal Form

Recently, in [21], it was noticed that whenever a product of braids $ABC \in B_N$ is represented in the Garside normal form, parts of the corresponding form of the individual factors A, B, and C are somewhat easy to extract. In particular, the authors of this paper presented an algorithm for recovering, given B, elements A' and C' such that:

- $A = A'$, $C = C'$ up to multiplications with elements in the center of B_N
- $AC = A'C'$.

Note that the center of the group B_N is a cyclic group generated by the square of the so-called *Garside's fundamental braid*, Δ, which is the only positive braid for which any two strands cross exactly once (see [14,22] for a classical introduction and a comprehensive survey on braid groups). This decomposition strategy allows for constructing a universal forgery, as stated in the following result:

Theorem 3 ([21]). *Let $W_1 \cdot E(H(m)) \cdot W_2 \in B_N$ be a valid signature for some message m, and let W_1', $W_2' \in B_N$ such that $W_1' \equiv W_1 \mod \Delta^2$, $W_2' \equiv W_2 \mod \Delta^2$, and $W_1 \cdot W_2 = W_1' \cdot W_2'$. Then,*

$$W_1' \cdot E(H(m')) \cdot W_2'$$

is a valid signature for any message m'.

Note that since the *replaced* braids W_1 and W_2 are in principle independent of the message m, the forged signature need not be longer than a legitimate signature. Furthermore, the complexity of this procedure is essentially that of computing Garside normal forms, which can be done in time $\mathcal{O}(k^2N)$, where k is the number of Artin generators encoding the input braid.

Furthermore, this method fends off the colored Burau representation used in the implementation of *WalnutDSA*; thus, it cannot be prevented by modifying the size of the underlying finite field. The authors of this cryptanalysis suggest that the only way to dodge this attack is to add many concealed cloaking elements to the encoding, which has a significant cost both in signature length and computing time for the generation of signatures. Furthermore, in [23], the authors of the scheme claimed to have experimentally demonstrated that inserting cloaking elements every 7–12 generators into the braid $E(H(m'))$ blocked this attack. However, no details were given on how this strategy was theoretically or empirically assessed.

5.2. Collision Attacks

Imposing implicit limits on the output signature sizes is indeed a valid strategy for preventing factoring attacks, and so, it was promptly noticed by the authors of *WalnutDSA*. However, in Section 4 of [20], it was demonstrated that, through a simple collision method, it was possible to compute *short* forged signatures, yet not on arbitrary messages.

In Section 4 of [20], it was observed that if there exist two messages m_1, m_2 such that $\mathcal{P}(E(H(m_1))) = \mathcal{P}(E(H(m_2)))$, then a valid signature for m_1 is valid for m_2 and vice versa. Breaking the EUF-CMA security notion (see Definition 4) is as simple as finding such two messages m_1 and m_2, since an adversary could query a signature for m_1 and then obtain a signature for m_2.

A generic collision attack is expected to require $|\mathcal{P}(E(\{0,1\}^*))|^{1/2}$ evaluations of function $\mathcal{P} \circ E$. In order to evaluate the feasibility of this attack, it is necessary to estimate the size of $|\mathcal{P}(E(\{0,1\}^*))|$. The authors of *WalnutDSA* considered $q^{N(N-3)} \cdot N!$ a conservative lower bound for values of $\mathcal{P}$. For 128-bit and 256-bit security levels, these values were 2^{216} and 2^{336}, respectively, so it is expected to find a collision after 2^{108} and 2^{168} evaluations of $\mathcal{P} \circ E$. Hence, a generic collision attack is not practical.

In [20], it was shown (by means of computer experiments) that $|\mathcal{P}(E(\{0,1\}^*))|$ is at most q^{13} (lying in an affine subspace over $\mathbb{F}_q$), so a collision is expected to be found after $q^{13/2}$ evaluations of $\mathcal{P} \circ E$. With this new estimate, $2^{32.5}$ and 2^{52} evaluations of $\mathcal{P} \circ E$ are necessary for 128-bit and 256-bit security levels, respectively. Therefore, collision attack is practical in this case.

In order to implement this attack, the authors used a generic collision finding algorithm: the distinguished point algorithm of Van Oorschot and Wiener [24].

This algorithm finds collisions in any function $f : S \to S$ that behaves like a random function [24]. The time complexity for finding a single collision is $\mathcal{O}(\sqrt{|S|})$. A distinguished point is an element of S satisfying some easily testable property (e.g., a fixed number of leading zero bits). The distinguished point algorithm selects a starting point $x_0 \in S$ at random and produces a chain of points $x_i = f(x_{i-1})$ for $i = 1, 2, \ldots,$ until a distinguished point is reached. Then, the starting point x_0, the distinguished point x_k, and the length of the chain are stored. It is expected that after $\mathcal{O}(\sqrt{|S|})$, the current chain will collide with one of the stored chains. Following the chain from that point, the same distinguished point will be reached.

In [20], this algorithm was applied to the function $f = g \circ \mathcal{P} \circ E$ instead of to $f = \mathcal{P} \circ E$, where g is a function that crafts plausible messages, given an output of $\mathcal{P}$. However, no implementation or description of how to build the function g was provided.

Using a standard PC, the algorithm found a collision after $2^{32.2}$ evaluations of f ($2^{32.5}$ evaluations were expected). This took approximately one hour. The two messages found by the algorithm were

$$\begin{cases} m_1 = & \text{“I would like to receive 7181666883746416503free samples of delicious cookies”.} \\ m_2 = & \text{“I pledge to donate 3519533052089988469 USD to Ward Beullens”.} \end{cases}$$

In order to mitigate this practical attack, Beullens and Blackburn [20] recommended to increase the value of q up to $q = 2^{20}$ and $q = 2^{40}$ to accomplish 128-bit and 256-bit security levels, respectively. With these new parameters, the size of the public key is five-times larger and the verification algorithm is 25-times slower for 256 bit.

A better mitigation of this attack is to change the encoding algorithm to output pure braids not restricted to the subgroup generated by $g_{N,1}$, $g_{N,2}$, $g_{N,3}$, and $g_{N,4}$. This change would require $q^{((N-2)^2+1)/2}$ evaluations of $\mathcal{P} \circ E$, and only a minor increase of parameters is needed. It was pointed out in [20] that a 256-bit security level could be accomplished by setting $q = 2^8$ and $N = 8$, making the key size 50%, the signature size 25% larger, and the verification algorithm two-times slower.

The authors of *WalnutDSA* pointed out that any braid output by the encoding mechanism E (see Section 4.1.2) 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 [23]. They further depicted two design strategies towards defeating this attack (see Table 1).

Table 1. Examples of sequences to defeat collision attack (see [23]).

N	Periodic Sequence S	dim S	Recommended q
10	$\{(3,5,7,9),(2,4,6,8),(1,3,5,7),(2,4,6,8),\ldots\}$	82	—
12	$\{(5,7,9,11),(4,6,8,10),(3,5,7,9),(2,4,6,8),(1,3,5,7),$ $(2,4,6,8),(3,5,7,9),(4,6,8,10),\ldots\}$	122	32, 256

5.3. Reversing E-Multiplication

A fundamental hard problem underlying the security of the *Walnut* signature scheme is to break the one-wayness of the function:

$$\begin{aligned} \mathcal{P} : B_N &\to \mathrm{GL}_N(\mathbb{F}_q) \times S_N \\ s &\mapsto (I_N, \iota_N) \star s. \end{aligned}$$

Here, we write $\mathcal{P}$ instead of $\mathcal{P}_{\mathcal{T}}$ with the understanding that the set $\mathcal{T} \subset \mathbb{F}_q$ of non-zero elements is arbitrary, but fixed.

More precisely, the underlying problem is defined as follows.

Definition 8 (Reversing E-Multiplication (REM) problem [20])**.** *Given a pair $(M,\sigma) \in \mathrm{GL}_N(\mathbb{F}_q) \times S_N$, such that $(M,\sigma) = \mathcal{P}(s)$ for some braid $s \in B_N$, find a braid $s' \in B_N$ such that $\mathcal{P}(s') = (M,\sigma)$.*

Observe that if brute force is used to solve the REM problem, then it would take $\mathcal{O}(|\mathcal{P}(B_N)|)$ E-multiplications to find a solution, where $|\mathcal{P}(B_N)|$ is the size of the orbit of (I_N, ι_N).

Recall that the private key consists of two braids $s_1, s_2 \in B_N$, and the corresponding public key consists of $\mathcal{P}(s_1)$ and $\mathtt{mat}(\mathcal{P}(s_2))$, the matrix component of $\mathcal{P}(s_2)$. In [20], it was observed that a valid signature $\mathtt{sig}$ for a message m also satisfies:

$$\mathcal{P}(s_1) \star \mathtt{sig} = \mathcal{P}(E(H(m))) \star s_2. \tag{3}$$

Therefore, not knowing the permutation component of $\mathcal{P}(s_2)$ poses no problem to the attacker since it can be recovered from the permutation component of (3) without necessarily knowing the encrypted message (no message attack). Indeed, since cloaking elements and $E(H(m))$ are required to be pure braids, we have:

$$\sigma_{s_1} \sigma_{\mathtt{sig}} = \sigma_{s_2}.$$

Once σ_{s_2} has been computed, an attacker can solve two instances of the REM problem by finding two braids $s_1', s_2' \in B_N$ such that $\mathcal{P}(s_1) = \mathcal{P}(s_1')$ and $\mathcal{P}(s_2) = \mathcal{P}(s_2')$, which can be used to sign any message (universal forgery). Hence, solving the REM problem means that UF-NMA security (Definition 2) can be violated.

In this section, we describe two algorithms proposed in [20] that solve the REM problem. The first algorithm is a generic birthday attack, while the second exploits the structure of the braid group B_N and is more efficient than the first one.

5.3.1. Generic Birthday Attack

Given a pair $(M,\sigma) \in \mathrm{GL}_N(\mathbb{F}_q) \times S_N$, if we can find two braids $s_1, s_2 \in B_N$ such that:

$$(M,\sigma) \star s_1 = (I_N, \iota_N) \star s_2,$$

then the solution of the REM problem is $s' = s_2 s_1^{-1}$. In [20], it was argued that a naive way of finding s_1 and s_2 by constructing tables with values $(M,\sigma) \star s_1$ and checking if $(I_N, \iota_N) \star s_2$ for random s_2 lying in the table would take $\mathcal{O}(\sqrt{|\mathcal{P}(B_N)|})$ E-multiplications, making this method more efficient than a brute force approach. Nevertheless, a naive approach may require too much storage memory.

This inconvenience can be circumvented by using a distinguished point algorithm (see Section 5.2). In this case, the algorithm is applied to the function:

$$f(x) = \begin{cases} (I_N, \iota_N) \star \mathtt{s}(x) & \text{if} \quad \mathtt{b}(x) = 0, \\ (M, \sigma) \star \mathtt{s}(x) & \text{if} \quad \mathtt{b}(x) = 1, \end{cases}$$

where $\mathtt{b}$ and $\mathtt{s}$ are hash functions that take elements in the orbit of (I_N, ι_N) as input and output a bit or a braid, respectively.

The idea is to find collisions:

$$f(x_1) = f(x_2) \quad \text{such that} \quad \mathtt{b}(x_1) \neq \mathtt{b}(x_2).$$

Hence, if a collision is found such that $b(x_1) = 1$, then $b(x_2) = 0$ and $(M, \sigma) \star \mathtt{s}(x_1) = f(x_1) = f(x_2) = (I_N, \iota_N) \star \mathtt{s}(x_2)$. In this case, a solution of the REM problem is $\mathtt{s}(x_2)\, s(x_1)^{-1}$. On the other hand, if $b(x_1) = 0$, then a solution of the REM problem is $\mathtt{s}(x_1)\, s(x_2)^{-1}$.

As noted in [23], this attack is exponential in running time and can be thwarted by choosing the correct parameters for *WalnutDSA*, in this case $N = 10$, $q = 2^{31} - 1$ for 128-bit security, and $N = 10$, $q = 2^{61} - 1$ for 256-bit security.

5.3.2. Subgroup Chain Attack

This attack exploits the fact that the restriction of $\mathcal{P}$ to pure braids is a group homomorphism, which maps the chain of subgroups:

$$\{\iota_N\} = P_1 \subset P_2 \subset \cdots \subset P_N \subset B_N,$$

to a nice chain of subgroups of $\mathrm{GL}_N(\mathbb{F}_q)$. Here, P_k denotes the intersection of the subgroup of pure braids in B_N and the subgroup generated by $b_1, b_2, \ldots, b_{k-1}$, that is the subgroup of pure braids such that only the first k strands cross over each other. More precisely, for each $1 \leq k \leq N$, $\mathcal{P}$ is a homomorphism from P_k into the subgroup:

$$A_k = \left\{ \begin{pmatrix} X & Y & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I_{N-k} \end{pmatrix} : \; X \in \mathrm{GL}_{k-1}(\mathbb{F}_q), \; Y \in \mathbb{F}_q^{k-1} \right\}.$$

In contrast to the birthday attack, this method solves the REM problem for a pair $(M, \sigma) \in \mathrm{GL}_N(\mathbb{F}_q) \times S_N$, by finding in iterative steps a braid $s \in B_N$ such that $(M, \sigma) \star s = (I_N, \iota_N)$, as follows. First, choose any braid $s' \in B_N$ such that $\sigma_{s'} = \sigma^{-1}$. Therefore, $(M, \sigma) \star s' = (M', \iota_N) \in A_N \times S_N$. Next, find a pure braid $s_N \in P_N$ such that $(M_N, \iota_N) = (M', \iota_N) \star s_N = (M, \sigma) \star s' s_N \in A_{N-1} \times S_N$. The iterative step consists of randomly choosing a target matrix $M_i \in \mathcal{P}(P_i) \cap A_{i-1}$ and then finding a pure braid $s_i \in P_i$ such that:

$$(M, \sigma) \star s'\, s_N\, s_{N-1} \cdots s_i \in A_{i-1} \times S_N.$$

Notice that in each iterative step, the permutation component is ι_N since s_i is a pure braid, and thus, $\sigma_{s_i} = \iota_N$. This process yields a braid $s = s'\, s_N\, s_{N-1} \cdots s_2$ such that $(M, \sigma) \star s = (I_N, \iota_N)$. Then, the solution to the REM problem is s^{-1}.

In [20], it was pointed out that if $M_i \notin \mathcal{P}(P_{i-1})$ for some $2 \leq i \leq N$, then it is not possible to complete the attack, and thus, assuming:

$$\mathcal{P}(P_i) \cap A_{i-1} = \mathcal{P}(P_{i-1}) \tag{4}$$

for each $2 \leq i \leq N$, guarantees that the attack will work. This assumption is not too restrictive since it seems to hold for the proposed parameters for *WalnutDSA*. With (4) in mind, the i^{th} iterative step of this attack can be solved by performing a collision search in the space cosets of A_{i-1} in $A_{i-1}\,\mathcal{P}(P_i)$ with a cost of $\sqrt{|\mathcal{P}(P_i)|/|\mathcal{P}(P_{i-1})|}$ E-multiplications (see Sections 5.2 and 5.3 of [20] for details).

In [20], the running time of this attack was estimated to be $q^{N/2-1}$ whenever E-multiplication uses the set of invertible elements $\mathcal{T} = \{y_1, y_2, \ldots, y_N\} \subset \mathbb{F}_q$ with $y_a = y_b = 1$ for some $1 \leq a, b \leq N$ (see Section 4.2.2). It was noted in [23] that if y_a and y_b are chosen such that $y_a \cdot y_b = -1$, then the running time of the attack is increased to at least $\sqrt{x}\,q^{(N-1)/2}$, where $x = 60$ for $N = 8$ and $x = 96$ for $N = 10$. Moreover, this attack is defeated by taking $N = 10$, $q = 2^{31} - 1$ for 128-bit security,and $N = 10$, $q = 2^{61} - 1$ for 256-bit security.

5.4. Uncloaking Signatures

Kotov, Menshov, and Ushakov presented in [25] a powerful attack against *WalnutDSA*. It is a heuristic attack that works exclusively with braids and does not need to take into account E-multiplication. The authors reported experiments with one hundred random protocol instances with a 100% success rate. It is worth pointing out that the experiments were carried out for three different settings: the 128 and 256-bit security levels from the official specification [26] (where $N = 8$) and the 256-bit security version with $N = 11$, proposed in [27].

In a nutshell, the attack works as follows: An adversary, which collects several arbitrary pairs of messages and valid signatures, is able to compute an alternative secret key such that, when used to sign any message, it produces the same signature as the real secret key. Therefore, this is a very strong attack as it violates a rather weak security notion for signatures (UF-RMA; see Definition 3), that is an adversary with access to signatures for random messages (not adversarially chosen) can produce a valid signature for any message of its choice; that is, it achieves a universal forgery.

Next, we provide a high-level description of the attack:

- **Step 1.** The attacker collects k pairs $\{(m_i, \mathtt{sig}_i)\}_{i=1}^{k}$ where each $\mathtt{sig}_i$ is a valid signature for m_i computed with the same secret key (s_1, s_2). Each signature is a braid with the form:

$$\mathtt{sig}_i = v_1^{(i)} \cdot s_1^{-1} \cdot v^{(i)} \cdot E(H(m_i)) \cdot s_2 \cdot v_2^{(i)}$$

  where $v^{(i)}, v_1^{(i)}, v_2^{(i)}$ are cloaking elements.
- **Step 2.** The attacker, using a heuristic procedure described in [25], is able to remove the cloaking elements from the signatures, that is compute braids $P_i = s_1^{-1} \cdot E(H(m_i)) \cdot s_2$. It is worth pointing out that Kotov, Menshov, and Ushakov reported a high success rate for their uncloaking algorithm, close to 80% or 100%, depending on the type of cloaking elements used (see Table 2).
- **Step 3.** The attacker computes the $k-1$ products $P_i P_{i+1}^{-1}$. Note that these are:

$$\begin{array}{rcl} P_1 P_2^{-1} & = & s_1^{-1} E(H(m_1)) E(H(m_2))^{-1} s_1 \\ & \vdots & \\ P_{k-1} P_k^{-1} & = & s_1^{-1} E(H(m_{k-1})) E(H(m_k))^{-1} s_1 \end{array}$$

  obtaining a system of conjugacy equations in B_N where only s_1 is unknown. In [25], another heuristic algorithm to obtain a solution s_1' of the system (not necessarily equal to s_1) was developed.
- **Step 4.** The attacker sets $s_2' = E(H(m_i))^{-1}\, s_1'\, P_i$ for i of its choice. Under certain conditions, (s_1', s_2') works as an alternative secret key to (s_1, s_2), in the sense that it produces a valid signature for any message. Moreover, *as a braid word*, this signature equals the one produced with the original key. This implies that the attack cannot be avoided by limiting the size of accepted signatures. In order to decide if the alternative key (s_1', s_2') works as intended, Kotov, Menshov, and Ushakov generated signatures for 10 random messages and checked their validity.

Table 2. Percentage of properly-identified cloaking elements v_1, v, v_2 according to [25].

Encoding	Cloaking Elements	128-Bit	256-Bit	256-Bit with $N = 11$
Original	$wb_i^{\pm 2}w^{-1}$	80%	77%	76%
Original	$wb_i^{\pm 4}w^{-1}$	100%	100%	100%
Alternative proposed in [27]	$wb_i^{\pm 2}w^{-1}$	77%	81%	81%
Alternative proposed in [27]	$wb_i^{\pm 4}w^{-1}$	97%	98%	100%

In [25], a 100% success rate of the full attack was reported. One interesting fact is that the attack did not need many message/signature pairs in order to succeed: Kotov, Menshov, and Ushakov affirmed that, in all their experiments, six successfully uncloaked signatures were enough to get five conjugacy equations and a valid alternative secret key. Average running times for the full attack are shown in Table 3.

Table 3. Average running time (in seconds) for the full attack according to [25].

Encoding	Cloaking Elements	128-Bit	256-Bit	256-Bit with $N = 11$
Original	$wb_i^{\pm 2}w^{-1}$	18.8	120.8	213.0
Original	$wb_i^{\pm 4}w^{-1}$	17.4	112.4	185.6
Alternative proposed in [27]	$wb_i^{\pm 2}w^{-1}$	78.7	264.9	1674.9
Alternative proposed in [27]	$wb_i^{\pm 4}w^{-1}$	66.2	224.6	1323.3

With respect to possible countermeasures against their attack, Kotov, Menshov, and Ushakov themselves made several proposals. The first one is to artificially introduce many so-called *critical letters* in the secret braids (locating critical letters is one of the main ingredients in the uncloaking algorithm). In addition, they proposed using many more cloaking elements (around 30) on each side of the signature. Nevertheless, they pointed out that it is not even clear if this measure would be useful as it does not neutralize their attack [28] against Kayawood [29], another braid-based protocol. Finally, Kotov et al. recommended short conjugators for constructing cloaking elements, making them less visible.

The proponents of *WalnutDSA* recognize the weakness of their original implementation against the uncloaking attack and put forward in [23] a countermeasure against it. Namely, they introduced the concept of concealed cloaking elements and proposed to add six of them to the computation of each signature, which translated into a 6.7% increase of the signature size. Kotov, Menshov, and Ushakov questioned the effectiveness of the approach in the NIST PQC project discussion forum [27], pointing out that their algorithms were designed taking into account the existence of precisely three cloaking elements, but could be modified to deal with more of them.

6. Final Remarks

WalnutDSA is a beautifully-designed signature scheme, conceived in the remarkable mathematical scenario of braid groups. Despite the inspiring ideas involved in the construction of this scheme, the many attacks explained in this survey demonstrate that there is still a long way to go before a suitable key generation/parameter selection process is identified. We believe that it will be rather difficult to fix the security problems described, which may be an unavoidable consequence of the adept and symmetric signature procedure. A formal security analysis, as well as a deeper understanding of the actual relation between the cryptanalytic goals and the affiliated mathematical problems are essential ingredients for a secure implementation of *WalnutDSA*. Maybe a promising idea is to start by identifying the concrete cost of a forgery. For instance, a first step would be to assess whether a forger can be used in a black-box manner to reverse the related E-multiplication procedure (i.e., to solve the REM problem). Once such a result is at hand, the next step would be to look for solid instances of REM that could be used for secure key generation.

Author Contributions: All authors contributed equally to this survey, searching for related results, selecting relevant information and writing and reviewing the draft.

Funding: This research was funded by NATO Science for Peace and Security Programme, grant number G5448 and by MINECO under Grant MTM2016-77213-R.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Announcing Request for Nominations for Public-Key Post-Quantum Cryptographic Algorithms. Available online: https://csrc.nist.gov/News/2016/Public-Key-Post-Quantum-Cryptographic-Algorithms (accessed on 19 December 2016).
2. Persichetti, E. Efficient One-Time Signatures from Quasi-Cyclic Codes: A Full Treatment. *Cryptography* **2018**, *2*, 30. [CrossRef]
3. Hoffstein, J.; Howgrave-Graham, N.; Pipher, J.; Whyte, W. Practical Lattice-Based Cryptography: NTRUEncrypt and NTRUSign. In *The LLL Algorithm—Survey and Applications*; Nguyen, P.Q., Vallée, B., Eds.; Information Security and Cryptography; Springer: Berlin, Germany, 2010; pp. 349–390. [CrossRef]
4. Jalali, A.; Azarderakhsh, R.; Kermani, M.M.; Campagna, M.; Jao, D. Optimized Supersingular Isogeny Key Encapsulation on ARMv8 Processors. *IACR Cryptol. ePrint Arch.* **2019**, *2019*, 331.
5. Garber, D. *Braid Group Cryptography*; World Scientific: Singapore, 2007
6. Anshel, I.; Anshel, M.; Goldfeld, D. An algebraic method for public-key cryptography. *Math. Res. Lett.* **1999**, *6*, 287–292. [CrossRef]
7. Ko, K.; Lee, S.; Cheon, J.; Han, J.; Kang, J.; Park, C. New Public-Key Cryptosystem using Braid Groups. In *Advances in Cryptology, Proceedings of CRYPTO 2000*; Lecture Notes in Computer Science; Springer: Santa Barbara, CA, USA, 2000; Volume 1880, pp. 166–183.
8. Birman, J.; Gebhardt, V.; González-Meneses, J. Conjugacy in Garside groups I: Periodic braids. *J. Algebra* **2007**, *2*, 746–776. [CrossRef]
9. Katz, J. *Digital Signatures*; Springer: Berlin, Germany, 2010.
10. Goldwasser, S.; Bellare, M. *Lecture Notes on Cryptography*; MIT: Hong Kong, China, 2001.
11. Anshel, I.; Atkins, D.; Goldfeld, D.; Gunnells, P.E. WalnutDSA^{TM}: A Quantum Resistant Digital Signature Algorithm. *IACR Cryptol. ePrint Arch.* **2017**, *2017*, 58.
12. Artin, E. Theory of braids. *Ann. Math.* **1947**, *48*, 101–126. [CrossRef]
13. Anshel, I.; Anshel, M.; Goldfeld, D.; Lemieux, S. Key agreement, the Algebraic Eraser^{TM}, and Lightweight Cryptography. In *Algebraic Methods in Cryptography, Contemp. Math.*; American Mathematical Society: Providence, RI, USA, 2006; Volume 418, pp. 1–34.
14. Birman, J.S.; Cannon, J. *Braids, Links, and Mapping Class Groups, Annals of Mathematics Studies*; Princeton University Press: Princeton, NJ, USA, 1974.
15. Artin, M. *Algebra*; Prentice Hall: Upper Saddle River, NJ, USA, 1991.
16. Birman, J.S.; Ko, K.H.; Lee, S.J. A new approach to the word and conjugacy problems in the braid groups. *Adv. Math.* **1998**, *139*, 322–353. [CrossRef]
17. Dehornoy, P. A fast method for comparing braids. *Adv. Math.* **1997**, *125*, 200–235. [CrossRef]
18. Morton, H.R. The multivariable Alexander polynomial for a closed braid. In *Lower Dimensional Topology, (Funchal, 1998)*; American Mathematical Society: Providence, RI, USA, 2006; Volume 233, pp. 167–172.
19. Hart, D.; Kim, D.; Micheli, G.; Pascual-Perez, G.; Petit, C.; Quek, Y. A Practical Cryptanalysis of WalnutDSA TM. In *Proceedings of the Public-Key Cryptography—PKC 2018—21st IACR International Conference on Practice and Theory of Public-Key Cryptography, Rio de Janeiro, Brazil, 25–29 March 2018*; Part I—Lecture Notes in Computer Science; Abdalla, M., Dahab, R., Eds.; Springer: Berlin, Germany, 2018; Volume 10769; pp. 381–406. [CrossRef]
20. Beullens, W.; Blackburn, S.R. Practical Attacks Against the Walnut Digital Signature Scheme. In *Proceedings of the Advances in Cryptology—ASIACRYPT 2018—24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, 2–6 December 2018*; Part I—Lecture Notes in Computer Science; Peyrin, T., Galbraith, S.D., Eds.; Springer: Berlin, Germany, 2018; Volume 11272, pp. 35–61. [CrossRef]

21. Merz, S.; Petit, C. Factoring Products of Braids via Garside Normal Form. In *Public Key Cryptography (2)*; Lecture Notes in Computer Science; Springer: Berlin, Germany, 2019; Volume 11443, pp. 646–678.
22. Paris, L. Braid groups and Artin groups. *arXiv* **2007**, arXiv:math.GR/0711.2372
23. Anshel, I.; Atkins, D.; Goldfeld, D.; Gunnells, P.E. Defeating the Hart et al, Beullens-Blackburn, Kotov-Menshov-Ushakov, and Merz-Petit Attacks on WalnutDSA (TM). *IACR Cryptol. ePrint Arch.* **2019**, *2019*, 472.
24. van Oorschot, P.C.; Wiener, M.J. Parallel Collision Search with Cryptanalytic Applications. *J. Cryptol.* **1999**, *12*, 1–28. [CrossRef]
25. Kotov, M.; Menshov, A.; Ushakov, A. An attack on the Walnut digital signature algorithm. *Des. Codes Cryptogr.* **2019**, 1–20. [CrossRef]
26. Anshel, I.; Atkins, D.; Goldfeld, D.; Gunnells, P.E. The Walnut Digital Signature Algorithm^{TM} Specifcation. Submitted to NIST PQC Project. 2017. Available online: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions (accessed on 7 July 2019).
27. Comments to WalnutDSA^{TM} Proposal to NIST PQCProject. Available online: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/official-comments/WalnutDSA-official-comment.pdf (accessed on 7 July 2019).
28. Kotov, M.; Menshov, A.; Ushakov, A. Attack on Kayawood Protocol: Uncloaking Private Keys. *IACR Cryptol. ePrint Arch.* **2018**, *2018*, 604.
29. Anshel, I.; Atkins, D.; Goldfeld, D.; Gunnells, P.E. Kayawood, a Key Agreement Protocol. *IACR Cryptol. ePrint Arch.* **2017**, *2017*, 1162.