WALNUT DIGITAL SIGNATURE ALGORITHM

Dorian Goldfeld

SecureRF Corporation

NATO Post Quantum Cryptography Workshop, September 27, 2016

INTRODUCING WALNUTDSA

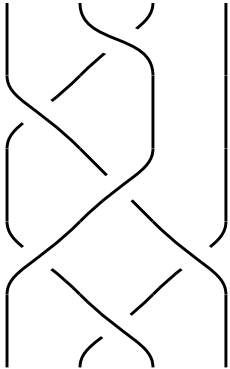
(joint work with Iris Anshel, Derek Atkins, Paul Gunnells)

We introduce WalnutDSA a group theoretic digital signature algorithm with the following features:

- *Very fast signature verification. Running time is linear in key/signature size.*
- *Security is based on the hard problems of solving a novel equation over the braid group and reversing a certain representation of the braid group.*
- *WalnutDSA appears resistant to known attacks in group theoretic cryptography.*
- *Quantum Resistant*

BRAIDS

Introduced by Emil Artin in 1921, a braid is a configuration of strands of the form:



THE ARTIN BRAID GROUP

For, $N \geq 2$, let B_N denote the N-strand braid group with Artin generators $\{b_1, b_2, \ldots, b_{N-1}\}$, subject to the following relations:

$$b_i b_{i+1} b_i = b_{i+1} b_i b_{i+1}, \qquad (i = 1, \ldots, N-2),$$
$$b_i b_j = b_j b_i, \qquad (|i-j| \geq 2).$$

Every $\beta \in B_N$ is of the form

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

where $i_j \in \{1, \ldots, N-1\}$, and $\epsilon_j \in \{\pm 1\}$.

PERMUTATION ASSOCIATED TO A BRAID

Each braid $\beta \in B_N$ determines a permutation in S_N.

Let $\sigma_i \in S_N$ be the i^{th} simple transposition, which maps

$$i \to i+1, \qquad i+1 \to i, \qquad (1 \le i \le N-1).$$

Then σ_i is associated to the Artin generator b_i.

The permutation associated to $b_{i_1}^{\epsilon_1}\ b_{i_2}^{\epsilon_2}\ \cdots\ b_{i_k}^{\epsilon_k}$ is just $\sigma_{i_1}^{\epsilon_1}\ \sigma_{i_2}^{\epsilon_2}\ \cdots\ \sigma_{i_k}^{\epsilon_k}$

COLORED BURAU REPRESENTATION OF B_N

Each $b_i^{\pm 1}$ is associated with the ordered pair $\left(CB(b_i)^{\pm 1}, \sigma_i\right)$ where σ_i is the transposition $(i, i+1)$,

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

By letting permutations act on the left on the matrices $CB(b_i)$ (by permuting the variable entries), the ordered pairs $\left(CB(b_i)^{\pm 1}, \sigma_i\right)$ form a semi-direct product which satisfy the braid relations. This gives a representation of B_N.

Laurent Polynomial entry: Short random word of length 10 in $B_4 \rtimes S_4$. What E-multiplication erases!

$$
-\frac{1}{t[3]}\left(1-t[1]+t[1]\,t[2]-\frac{1-t[1]}{t[3]}+\frac{(1-t[1])\,t[2]}{t[3]}+\right.
$$

$$
\frac{1}{t[1]}
$$

$$
\left(\frac{1}{t[2]}\right.
$$

$$
\left(-t[1]\left(1-t[1]+t[1]\,t[2]-\frac{1-t[1]}{t[3]}+\frac{(1-t[1])\,t[2]}{t[3]}\right)+\right.
$$

$$
t[1]\left(1-t[1]+t[1]\,t[2]-\frac{1-t[1]}{t[3]}+\frac{(1-t[1])\,t[2]}{t[3]}+\right.
$$

$$
t[3]\left(-\frac{(1-t[1])\,t[2]}{t[3]}+\frac{-\frac{1-t[1]}{t[3]}+\frac{(1-t[1])\,t[2]}{t[3]}}{t[4]}\right)-
$$

$$
\left.\left.\left.\frac{-\frac{1-t[1]}{t[3]}+\frac{(1-t[1])\,t[2]}{t[3]}}{t[4]}\right)\right)-\right.
$$

$$
t[1]\left(1-t[1]+t[1]\,t[2]-\frac{1-t[1]}{t[3]}+\frac{(1-t[1])\,t[2]}{t[3]}+\right.
$$

$$
t[3]\left(-\frac{(1-t[1])\,t[2]}{t[3]}+\frac{-\frac{1-t[1]}{t[3]}+\frac{(1-t[1])\,t[2]}{t[3]}}{t[4]}\right)-
$$

$$
\left.\left.\left.\left.\frac{-\frac{1-t[1]}{t[3]}+\frac{(1-t[1])\,t[2]}{t[3]}}{t[4]}\right)\right)-\frac{-\frac{1-t[1]}{t[3]}+\frac{(1-t[1])\,t[2]}{t[3]}}{t[4]}\right),
$$

SECURE RF
Securing the Internet of Things®

VERIDIFY

E-MULTIPLICATION (denoted $\star$)

- $\mathbb{F}_q$ = finite field of q elements.
- $T = \{\tau_1, \ldots, \tau_N\} \subset (\mathbb{F}_q^\times)^N$ = set of T-values.
- $m \in GL(N, \mathbb{F}_q)$, $\sigma \in S_N$.

E-multiplication by one Artin generator

$$(m, \sigma) \star b_i \;=\; \left(m \cdot \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & \tau_{\sigma(i)} & -\tau_{\sigma(i)} & 1 & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix}, \;\; \sigma \cdot (i, i+1) \right).$$

By iterating this computation we can compute the E-Multiplication of (m, σ) with an arbitrary braid element (finite product of Artin generators and their inverses).

REVERSING E-MULTIPLICATION IS HARD

- Braid group B_N and symmetric group S_N with $N \geq 8$.
- Finite field $\mathbb{F}_q$ with $q \geq 32$.
- $\beta \in B_N$.
- $(m, \sigma) = (Id, Id) \star \beta \in GL(N, \mathbb{F}_q) \times S_N$.

Conjecture: *It is infeasible to determine β from (m, σ) if the normal form of β is sufficiently long.*

Quantum Resistance: *As the length of the word β increases, the complexity of the Laurent polynomials occurring in the E-multiplication increases exponentially. It does not seem to be possible that E-Multiplication exhibits any type of simple periodicity, so it is very unlikely that inverting E-Multiplication can be achieved with a polynomial quantum algorithm.*

CLOAKING ELEMENTS

Let $m \in GL(N, \mathbb{F}_q)$ and $\sigma \in S_N$. An element v in the pure braid subgroup of B_N is termed a cloaking element of (m, σ) if

$$\boxed{(m, \sigma) \star v = (m, \sigma).}$$

The cloaking element is defined by the property that it essentially disappears when performing E-Multiplication.

Remark: *When E-Multiplication is viewed as a right action of B_N on $GL(N, \mathbb{F}_q) \times S_N$ then cloaking elements are stabilizers which form a subgroup of B_N.*

CLOAKED CONJUGACY SEARCH PROBLEM (CCSP))

- The braid group B_N and symmetric group S_N with $N \geq 8$.
- $Y, v_1, v_2 \in B_N$.
- Assume v_1 cloaks $(\mathrm{Id}, \mathrm{Id})$ and v_2 cloaks $(\mathrm{Id}, \mathrm{Id}) \star Y$.

Conjecture (CCSP): *Assume $A \in B_N$ and $Y^{-1}\, v_1\, A\, Y\, v_2$ are known. Then it is infeasible to determine Y if the normal forms of Y, v_1, v_2 are sufficiently long.*

QUANTUM RESISTANCE OF CCSP

Hidden Subgroup Problem *The Hidden Subgroup Problem asks to find an unknown subgroup $H \leq G$ using calls to a known function on G which takes distinct constant values on distinct cosets of G/H.*

• Shor's quantum attack breaking RSA and other public key protocols such as ECC are essentially equivalent to the fact that there is a successful quantum attack on the Hidden Subgroup Problem for finite cyclic groups.

• Since the braid group does not contain any non-trivial finite subgroups at all, there does not seem to be any viable way to connect to connect CCSP with HSP.

WALNUTDSA KEY GENERATION

WalnutDSA allows a signer with a fixed private/public key pair to create a digital signature associated to a given message which can be validated by anyone who knows the public key of the signer and the verification protocol.

Public Information:

- An integer $N \geq 8$ and associated braid group B_N.
- A rewriting algorithm $\mathcal{R}\colon B_N \to B_N$.
- A finite field $\mathbb{F}_q$ of $q \geq 32$ elements.
- T-values $= \{\tau_1, \tau_2, \ldots, \tau_N\}$ a set of invertible elements in $\mathbb{F}_q$.

Signer's Private Key: $\mathrm{Priv(S)} \in B_N$.

Signer's Public key: $\mathrm{Pub(S)} = \big(\mathrm{Id}, \mathrm{Id}\big) \star \mathrm{Priv(S)},$

SECURE RF
Securing the Internet of Things®

VERIDIFY

WALNUTDSA SIGNATURE GENERATION

- $\mathcal{M}$ is a hash of a message.
- $E(\mathcal{M})$ is an encoding of $\mathcal{M}$ into the pure braid subgroup of B_N.

Step 1: Generate the cloaking elements v_1 and v_2 which cloak $\big(\mathrm{Id}, \mathrm{Id}\big)$ and $\mathrm{Pub}(S)$, respectively.

Step 2: Compute $\mathrm{Sig} = \mathcal{R}\big(\mathrm{Priv}(\mathrm{S})^{-1} \cdot v_1 \cdot E(\mathcal{M}) \cdot \mathrm{Priv}(\mathrm{S}) \cdot v_2\big)$, a rewritten braid.

Step 3: The final signature for the message $\mathcal{M}$ is the ordered pair $(\mathrm{Sig},\ \mathcal{M})$.

WALNUTDSA SIGNATURE VERIFICATION

The signature $(\mathrm{Sig}, \mathcal{M})$ *is verified as follows:*

Step 1: Generate the encoded message $E(\mathcal{M})$.

Step 2: Evaluate $\mathrm{Pub}(\mathrm{E}(\mathcal{M})) := \big(\mathrm{Id}, \mathrm{Id}\big) \star E(\mathcal{M})$.

Step 3: Evaluate $\mathrm{Pub}(\mathrm{S}) \star \mathrm{Sig}$.

Step 4: *Verify the equality*

$$\mathrm{MatrixPart}\big(\mathrm{Pub}(\mathrm{S}) \star \mathrm{Sig}\big) = \mathrm{MatrixPart}\big(\mathrm{Pub}(\mathrm{E}(\mathcal{M}))\big) \cdot \mathrm{MatrixPart}\big(\mathrm{Pub}(\mathrm{S})\big),$$

where the matrix multiplication on the right is performed over the finite field. The signature is valid if this equality holds. If the results are not equal then the signature validation has failed.

LINEAR RUNNING TIME OF SIGNATURE VERIFICATION

- The bulk of Signature Verification only requires E-Multiplication computations.

- E-multiplication by one Artin generator can be performed in one clock cycle.

- Most of the running time of Signature Verification is taken up by computing L successive E-Multiplications, each by one Artin generator, where L is the number of Artin generators of the Signature (Sig).

- WalnutDSA Signature Verification is extremely fast and the running time is linear in the Signature length.

GROVER'S QUANTUM SEARCH ALGORITHM (GQSA)

- GQSA finds an element in an unordered N element set in time $\mathcal{O}\left(N^{\frac{1}{2}}\right)$.

- GQSA can find the private key in a cryptosystem with a square root speed-up in running time and cuts the security in half.

- GQSA can be defeated by increasing the key size.

- WalnutDSA Signature Verification runs in linear time in the signature length. GQSA can be defeated by doubling the signature length, which results in double the computation time.

- By comparison ECDSA runs in quadratic time. Defeating GQSA requires a four fold increase in computation time.

SECURITY DISCUSSION

• The recent attack of Ben-Zvi–Blackburn–Tsaban (BBT) (*"A practical cryptanalysis of the Algebraic Eraser,"* CRYPTO 2016, see also *"Defeating the Ben-Zvi, Blackburn, and Tsaban Attack on the Algebraic Eraser"*) does not seem to apply to WalnutDSA because the signature is a braid and there are no commuting subgroups involved, Hence, the linear algebraic attacks as in BBT to solve CCSP or to forge a signature are not applicable.

• Length attacks of the type proposed by Myasnikov–Ushakov (2009) or Garber-Kaplan-Teicher-Tsaban-Vishnu (2005) do not appear to facilitate solving CCSP if the keys are sufficiently large. First of all, as pointed out by Gunnells (2011), the length attack only works effectively for short conjugates. Secondly, the placement of the unknown cloaking elements v_1, v_2 in the braid word $Y^{-1}\, v_1\, A\, Y\, v_2$ seems to completely thwart any type of length attack for the conjugacy problem.

SECURITY DISCUSSION

• Note that if the cloaking elements v_1, v_2 are trivial then CCSP reduces to the ordinary conjugacy search problem (CSP). If an attacker can determine the cloaking elements v_1, v_2 then it is easy to see that CCSP again reduces to CSP and fast methods for solving for Y were obtained in Gebhardt *"A new approach to the conjugacy problem in Garside groups," (2005)* provided the super summit set of the conjugate Y was not too large.

Performance of WalnutDSA Verification

$B_8F_{32}, 2^{128}$ Security level (equivalent to ECC P256)

Platform	Clock	WalnutDSA				ECDSA			Gain
	MHz	ROM[1]	RAM[1]	Cycles	Time[2]	ROM[1]	RAM[1]	Time[2]	(Time)
MSP430	8	3244	236	370944	46	[3] 20-30K	2-5K	1000-3000	21-63x
8051	24.5	3370	312	864101	35.3				
ARM M3	48	2952	272	275563	5.7	[4] 7168	540	233	40.8x
FPGA	50			2500	0.05			[5] 2.08	41.6x

[1] ROM/RAM in Bytes

[2] Time is in milliseconds.

[3] C.P.L. Gouvêa and J. López, *Software implementation of Pairing-Based Cryptography on sensor networks using the MSP430 micro controller*, Progress in Cryptology, Indocrypt 2009

[4] Wenger, Unterluggauer, and Werner in 8/16/32 *Shades of Elliptic Curve Cryptography on Embedded Processors in Progress in Cryptology*, Indocrypt 2013

[5] Jian Huang, Hao Li, and Phil Sweany, *An FPGA Implementation of Elliptic Curve Cryptography for Future Secure Web Transaction*, 2007.

SECURE RF
Securing the Internet of Things®

VERIDIFY