Entropoid-based cryptography is group exponentiation in disguise

Lorenz Panny

Institute of Information Science, Academia Sinica, Taipei, Taiwan
lorenz@yx7.cc

Abstract. A recent preprint [3] suggests the use of exponentiation in a non-associative algebraic structure called *entropoid* to construct post-quantum analogues of DLP-based cryptosystems. In this note, we show a polynomial-time reduction from the entropoid version of DLP to the conventional DLP in the underlying finite field. The resulting attack takes less than 10 minutes on a laptop against parameters suggested in [3] for 128-bit post-quantum secure key exchange and runs in polynomial time on a quantum computer. We briefly discuss how to generalize the attack to the generic setting.

Keywords: Cryptanalysis, post-quantum cryptography, entropic quasigroup, non-associative exponentiation, finite field, discrete-logarithm problem.

1 Introduction

The quest for drawback-free post-quantum substitutes of vital cryptographic building blocks continues. One approach to replace DLP-based schemes (such as Diffie–Hellman) is to search for algebraic structures supporting a generalized exponentiation operation that commutes — so Alice and Bob can obtain a shared secret — while not being vulnerable to Shor's quantum algorithm.

What [3] proposes is such an algebraic structure: It defines a *non-associative* binary operation, however with a sufficiently strong alternative associativity law to permit defining an exponentiation map that makes exponents commute. More concretely, [3] defines an *entropoid* to be a quasigroup[1] $(G, *)$ where $*$ is *entropic*, i.e., satisfies the pseudo-associativity law

$$(x * y) * (z * w) = (x * z) * (y * w)\,. \qquad (\dagger)$$

(To be precise, [3] requires an additional addition operation on G and explicitly excludes associative or commutative multiplication. We ignore these details as they are not needed in the sequel.)

[*] Date of this document: 2021-05-04.

[1] A *quasigroup* is a set G together with a binary operation $*\colon G \times G \to G$ such that for all $a \in G$, the maps $a * -\colon G \to G$ and $- * a\colon G \to G$ are bijections. (In other words, all left- and right-divisions are possible and uniquely defined.)

The cryptosystem is then based on non-associative exponentiation in $(G, *)$: Besides the number of times an element is multiplied by itself, an exponent must thus also encode how these multiplications are parenthesized. For example, the list of all such generalized exponents up to size 4 may be represented as follows:

$$-,\ \square,\ \square\square,\ (\square\square)\square,\ \square(\square\square),\ \square(\square(\square\square)),\ \square((\square\square)\square),\ (\square\square)(\square\square),\ ((\square\square)\square)\square,\ (\square(\square\square))\square\,.$$

Now, the remarkable (and, from a cryptographer's perspective, intriguing) thing about groupoids satsifying (†) is that the non-associative exponentiation map behaves "as it should"; i.e., for generalized exponents $\mathbf{A}, \mathbf{B}$ as above we have

$$(x^{\mathbf{A}})^{\mathbf{B}} = (x^{\mathbf{B}})^{\mathbf{A}}\,. \qquad (\star)$$

This equation virtually *screams* Diffie–Hellman, and indeed, building analogues of DLP-based systems on top of the commutativity property ($\star$) for entropoids is precisely what [3] proposes.

After laying out the general framework, [3] proceeds to construct a concrete instantiation $\mathbb{E}^*_{(p-1)^2}$ of this idea using an algebraic multiplication law on a subset of $\mathbb{F}_p \times \mathbb{F}_p$. The parameters of the entropoid $\mathbb{E}^*_{(p-1)^2}$ defined in [3] are a (large) prime p together with constants $a_3, a_8, b_2, b_7 \in \mathbb{F}_p$ subject to some mild algebraic constraints. The definition of $\mathbb{E}^*_{(p-1)^2}$ is as follows:

$$\begin{aligned}
\mathbb{E}^*_{(p-1)^2} \;&=\; \Big(\mathbb{F}_p \setminus \{-a_3/a_8\}\Big) \times \Big(\mathbb{F}_p \setminus \{-b_2/b_7\}\Big) \\
(x_1, x_2) * (y_1, y_2) \;&=\; \Big(\frac{a_3(a_8 b_2 - b_7)}{a_8 b_7} + a_3 x_2 + \frac{a_8 b_2}{b_7} y_1 + a_8 x_2 y_1, \\
&\qquad -\frac{b_2(a_8 - a_3 b_7)}{a_8 b_7} + \frac{a_3 b_7}{a_8} y_2 + b_2 x_1 + b_7 x_1 y_2\Big)\,.
\end{aligned}$$

Notice that

$$\mathbf{1} \;=\; (1/b_7 - a_3/a_8,\ 1/a_8 - b_2/b_7)$$

is a left-neutral element of $(\mathbb{E}^*_{(p-1)^2}, *)$.

2 Reduction to finite-field DLP

In this section, we demonstrate an attack against the concrete instantiation $\mathbb{E} := \mathbb{E}^*_{(p-1)^2}$ proposed by [3]. Section 2.1 will discuss how that attack should generalize to the entropoid cryptography concept in a generic setting.

The hidden group. First, it follows from [6, Theorem 1] that we can recover an abelian group structure $(\mathbb{E}, \cdot)$ on the set $\mathbb{E}$ characterized the property

$$(x * \mathbf{1}) \cdot y = x * y\,.$$

It is not hard to check using (†) that $(\mathbb{E},\cdot)$ is in fact an abelian group with identity element $\mathbf{1}$, and that $\sigma\colon \mathbb{E}\to\mathbb{E},\ x\to x*\mathbf{1}$ is an automorphism of order 2 of both $(\mathbb{E},*)$ and $(\mathbb{E},\cdot)$. Thus, we have established that

$$x*y = x^\sigma\cdot y\,.$$

Notably, the non-associative non-commutative structure of $(\mathbb{E},*)$ is really just the abelian group structure of $(\mathbb{E},\cdot)$ with one input twisted by an automorphism.

Maps to finite fields. Concretely, the automorphism σ and the newly recovered abelian group structure on $\mathbb{E}$ are

$$\begin{aligned}
\sigma((x_1,x_2)) &= \Big(\frac{a_8}{b_7}x_2+\frac{a_8^2b_2-a_3b_7^2}{a_8b_7^2},\ \frac{b_7}{a_8}x_1+\frac{a_3b_7^2-a_8^2b_2}{a_8^2b_7}\Big)\,;\\
(x_1,x_2)\cdot(y_1,y_2) &= \Big(b_7x_1y_1+\frac{a_3b_7}{a_8}x_1+\frac{a_3b_7}{a_8}y_1+\frac{a_3^2b_7-a_3a_8}{a_8^2},\\
&\qquad a_8x_2y_2+\frac{a_8b_2}{b_7}x_2+\frac{a_8b_2}{b_7}y_2+\frac{a_8b_2^2-b_2b_7}{b_7^2}\Big)\,.
\end{aligned}$$

The group $(\mathbb{E},\cdot)$ is easily seen to decompose as a direct product as there are no interactions at all between the first and second component.

Furthermore, as suggested by the classification of affine algebraic groups of dimension one, each component of $(\mathbb{E},\cdot)$ ought to be isomorphic to $(\mathbb{F}_p^\times,\cdot)$, and indeed, a possible isomorphism is given by

$$\iota\colon\quad \mathbb{E}\to\big(\mathbb{F}_p^\times\big)^2,\ (x_1,x_2)\mapsto\big(b_7x_1+a_3b_7/a_8,\ a_8x_2+a_8b_2/b_7\big)\,.$$

Newfound associativity. Rewriting $x*y$ as $x^\sigma\cdot y$ reveals that the choice of parenthesization of a non-associative exponentiation in $(\mathbb{E},*)$ matters much less than it seems at first: Computing a few examples (or, more formally, induction) using the property $\sigma^2=id$ quickly reveals that *any* non-associative power of an element $x\in\mathbb{E}$ can simply be written in the form

$$(x^\sigma)^i\cdot x^j$$

with $i,j\in\mathbb{Z}_{\geq 0}$; exponentiations now taking place in $(\mathbb{E},\cdot)$. We may thus recover the constants i,j corresponding to Alice's private-key operation $x\mapsto x^{\mathbf{A}}$ in order to evaluate that map on arbitrary elements of $\mathbb{E}$ other than the generator $g\in\mathbb{E}$ chosen in the cryptosystem. This involves a multidimensional discrete-logarithm computation in $(\mathbb{E},\cdot)$, which is polynomial-time on a quantum computer and can be reduced to DLPs in the finite field $\mathbb{F}_p$ and some linear algebra classically:

- Map $g, g^\sigma, g^{\mathbf{A}}$ to $\mathbb{F}_p$ via ι: $(\alpha_1,\alpha_2)=\iota(g)$, $(\beta_1,\beta_2)=\iota(g^\sigma)$, $(\gamma_1,\gamma_2)=\iota(g^{\mathbf{A}})$.
- Pick a generator κ of the group $\mathbb{F}_p^\times$ and compute the discrete logarithms $r_i=\log_\kappa(\alpha_i)$, $s_i=\log_\kappa(\beta_i)$, $t_i=\log_\kappa(\gamma_i)$ in $\mathbb{F}_p$.
- Solve the linear system $\big(i\ j\big)\begin{pmatrix} r_1 & r_2\\ s_1 & s_2\end{pmatrix}=\big(t_1\ t_2\big)$ modulo $p-1$ for $(i\ j)\in\mathbb{Z}^2$.
- Evaluate Alice's private-key map $x\mapsto x^{\mathbf{A}}$ by computing $x\mapsto\iota^{-1}\big((x^\sigma)^i\cdot x^j\big)$.

Representation of private keys. Our reduction as described above does not strictly solve the DELP problem exactly as given in [3, Definition 23], since that formulation assumes a specific way of writing down generalized exponents. However, we argue that this detail is a distraction: The DELP from [3] is in its current version already satisfied with equivalent keys — as it should, since the mapping from private to public keys is non-injective, so recovering the exact private key is information-theoretically impossible anyway — and thus there is no reason an attacker wouldn't be happy with any representation of the private key that allows them to *compute the private-key operation* in polynomial time.

In any case, it appears feasible (albeit perhaps somewhat tedious) to devise an algorithm for recovering a private key in the style of [3] from the representation of the private key obtained in the attack above.

2.1 The general case

The main structure result for entropoids is the following theorem, which was independently (and with slightly different conditions) proved by Murdoch [4], Toyoda [6], and Bruck [1]:

Theorem 1. *For every entropic quasigroup $(G, *)$, there exists an abelian group $(G, \cdot)$, commuting automorphisms σ, τ of $(G, \cdot)$, and an element $c \in G$, such that*

$$x * y = x^\sigma \cdot y^\tau \cdot c\,.$$

Thus, like we have observed in the example above (with $\tau = id$ and $c = 1$), the composition law in *any* entropic quasigroup comes from a multiplication in an abelian group that is twisted by automorphisms and translated by a constant.

As before, this implies that any non-associative power of an element $x \in G$ can in fact be written as a product combination in $(G, \cdot)$ of elements of the form x^ξ and c^γ where $\xi, \gamma \in \langle \sigma, \tau \rangle$. The classification of finite abelian groups implies that there exists a small[2] subset of such elements that suffices to span the entire subquasigroup $\langle g \rangle_*$ generated by $g \in G$, and again, recovery of the exponents corresponding to Alice's private-key operation consists of a multidimensional discrete-logarithm computation (which is polynomial-time quantumly).

It therefore appears that, barring unforeseen complications in the details of the argument sketched in this section, all instantiations of the entropoid framework should be breakable in polynomial time on a quantum computer.

3 Attack implementation

We have fully implemented the reduction described in the preceding in `sage` [2] and verified that it succeeds against the proof-of-concept `sage` implementation of entropoid Diffie–Hellman that was (commendably!) provided in [3].

[2] Polynomially-sized in $\log |G|$.

The reduction itself consists of polynomially many algebraic operations in $\mathbb{F}_p$ and requires negligible time in practice. Since the sizes of p suggested by [3] are relatively small (between 128 and 512 bits), the CADO-NFS software [5] can solve the resulting DLP instances within at most a couple of days on a high-end desktop computer. For the largest proposed key-exchange instantiation with claimed 256-bit classical and 128-bit post-quantum security, CADO-NFS computes the DLPs arising from the reduction in less than 10 minutes on a laptop with a 4-core i5-6440HQ processor and 16 gigabytes of memory.

Attack code is available at `https://yx7.cc/files/entropoid-attack.tar.gz`. (Note that the prime p is chosen smaller in this example so that `sage`'s default method resolves the DLPs resulting from the reduction quickly. The same code can handle large sizes if the DLP computations are outsourced to CADO-NFS.)

References

[1] Richard H. Bruck. "Some Results in the Theory of Quasigroups". In: *Transactions of the American Mathematical Society* 55.1 (1944), pp. 19–52.

[2] The Sage Developers. *SageMath, the Sage Mathematics Software System*. Version 9.2. 2020. URL: https://sagemath.org.

[3] Danilo Gligoroski. *Entropoid Based Cryptography*. IACR Cryptology ePrint Archive 2021/469. 2021. URL: https://ia.cr/2021/469.

[4] David C. Murdoch. "Quasi-Groups Which Satisfy Certain Generalized Associative Laws". In: *American Journal of Mathematics* 61.2 (1939), pp. 509–522.

[5] The CADO-NFS Development Team. *CADO-NFS, An Implementation of the Number Field Sieve Algorithm*. Version 2.3.0. 2017. URL: http://cado-nfs.gforge.inria.fr.

[6] Kôshichi Toyoda. "On axioms of linear functions". In: *Proceedings of the Imperial Academy* 17.7 (1941), pp. 221–227.