On lineariazation attack of entropic quasigroups cryptography

Daniel Nager
daniel.nager@gmail.com

November 14, 2022

Abstract

In [Pan21] a linearization attack is proposed in order to break the cryptosystem proposed in [Gli21]. We want to propose here an analysis of linearization using the algorithm and cryptosystem proposed in [NJ21], and set as a general solution not to reuse keys at the protocol level, both of digital signature and secret agreement to avoid the linearization strategy. Furthermore we will show that an entropic quasigroup's linearized form is indeed a valid start to set up a valid cryptosystem.

1 Algorithm description

Let's define in simple terms the algorithm described in [NJ21].

We're working with a set G, with a finite number of elements and an entropic operation $*$, defining entropic as satisfying $(a * b) * (c * d) = (a * c) * (b * d)$, $a, b, c, d \in G$.

Let's explain a mixing function based on $(G, *)$, $R = m(T, K)$. We define $T = [t_i]$, $K = [k_i]$ tables of elements in G, so $t_i \in G$, $k_i \in G$, $i \in 1..n$. This function is the same as the one proposed in [NJ21].

The hard problem we're working with is in $R = m(T, K)$, finding K knowing T and R.

To make it simpler to read let's set $n = 4$, the size of the tables, so we have two initial tables:

$T = [t_1, t_2, t_3, t_4]$
$K = [k_1, k_2, k_3, k_4]$

And from these two tables we create an initial mixing state, applying $*$ position-wise:

$V = [t_1 * k_1, t_2 * k_2, t_3 * k_3, t_4 * k_4]$

And we start mixing, the procedure is to select two positions in the range 1..4 of the table in a pseudorandom deterministic way and operate both with $*$ and placing the result in one of both selected positions. For example, if we have:

$V = [v_1, v_2, v_3, v_4]$

we can iterate (the operation is arbitrary but fixed and known at each step):

$V \leftarrow [v_1, v_2 * v_4, v_3, v_4]$

were we've selected in this case $v_2 \leftarrow v_2 * v_4$.

This can be done as many times as needed with different positions of the state table, so we repeat $v_i \leftarrow v_i * v_j$ with different i and j, chosen randomly in a deterministic way at each step.

Finally, after the pseudorandom mixing we operate K again to the state to get the result:

$R = [v_1 * k_1, v_2 * k_2, v_3 * k_3, v_4 * k_4]$

Now, it's proven in [NJ21] that the operation $R = m(T, K)$ is as well entropic if $*$ is.

2 Linearization analysis

At this point let's see what happens if we have found a linearization of $*$. This is the attack proposed in [Pan21] to the algorithm proposed in [Gli21].

So we have $a * b = \sigma(f) \cdot g$, $f \in G$, $g \in G$, $\sigma(\sigma(x)) = x$, and $f = \iota(a)$, $g = \iota(b)$. $(G, \cdot)$ is an abelian group, σ an automorphism of order 2 of this group, and ι is an isomorphism $\iota : (G, *) \to (G, \cdot)$.

Now, with this linearization we can redo and trace the mixing procedure $R = m(T, K)$ but in $(G, \cdot)$ instead of in $(G, *)$, applying $v_i = \sigma(v_i) \cdot v_j$. In the example above it will be $v_2 \leftarrow \sigma(v_2) \cdot v_4$. We're ignoring the isomorphism as is not relevant, so $\iota(v) = v$.

Finally, we get with $n = 4$, 4 equations each one corresponding to a position in the result table R, that, moving out known values to the left side of each equation, in the right side we have products of elements of the form:

$\sigma(k_i)^{e_i^p} \cdot k_i^{f_i^p}$

So for each result position we can get an equation of the following form:

$r_p = \prod_{i=1..4} (\sigma(k_i)^{e_i^p} \cdot k_i^{f_i^p})$

and an equation system considering $p = 1 \dots 4$, of 4 equations:

$$r_1 = \sigma(k_1)^{e_1^1} \cdot (k_1)^{f_1^1} \cdot \sigma(k_2)^{e_2^1} \cdot (k_2)^{f_2^1} \cdot \sigma(k_3)^{e_3^1} \cdot (k_3)^{f_3^1} \cdot \sigma(k_4)^{e_4^1} \cdot (k_4)^{f_4^1}$$
$$r_2 = \sigma(k_1)^{e_1^2} \cdot (k_1)^{f_1^2} \cdot \sigma(k_2)^{e_2^2} \cdot (k_2)^{f_2^2} \cdot \sigma(k_3)^{e_3^2} \cdot (k_3)^{f_3^2} \cdot \sigma(k_4)^{e_4^2} \cdot (k_4)^{f_4^2}$$
$$r_3 = \sigma(k_1)^{e_1^3} \cdot (k_1)^{f_1^3} \cdot \sigma(k_2)^{e_2^3} \cdot (k_2)^{f_2^3} \cdot \sigma(k_3)^{e_3^3} \cdot (k_3)^{f_3^3} \cdot \sigma(k_4)^{e_4^3} \cdot (k_4)^{f_4^3}$$
$$r_4 = \sigma(k_1)^{e_1^4} \cdot (k_1)^{f_1^4} \cdot \sigma(k_2)^{e_2^4} \cdot (k_2)^{f_2^4} \cdot \sigma(k_3)^{e_3^4} \cdot (k_3)^{f_3^4} \cdot \sigma(k_4)^{e_4^4} \cdot (k_4)^{f_4^4}$$

Now, if we consider $\sigma(k_i)$ and k_i as lineary unrelated, they must be treated as diferent unknowns, so in the example we have 4 equations and 8 unknowns.

Also all the e_i^p and f_i^p are known but in general the linear relation of $e_i^{p_1}$ and $f_i^{p_1}$, and $e_i^{p_2}$ and $f_i^{p_2}$, for two positions p_1 and p_2 is not the same due to the pseudorandom mixing, so gaussian-like elimination cannot eliminate $\sigma(k_i)$ and k_i at once.

The best method at first glance to solve the problem is bruteforce half of k_i's and then do a gaussian-like elimination on the rest, considering exponentials.

3 Conclusion

The previous analysis is correct if we only hold a single application of $R = m(T, K)$, so it's a requirement not to reuse keys, in particular not to encrypt two different T with the same K, an this can be done at the protocol level, where we use m to build a key agreement and a signature protocol. Let's add that not reusing keys also addresses the break in [Jia21].

As a proposal of a protocol to do signatures, we can profit from the following formula:

$$m(m(C, H), m(K, Q)) = m(m(C, K), m(H, Q))$$

Then $\langle C, m(C, K)\rangle$ are the signer credentials, and $\langle m(H, Q), m(K, Q)\rangle$ the signature. Q must be different for each signature, while K is always the same. H is the hash to sign and C a constant value.

The secret agreement protocol is the same as in [NJ21].

Now, we assert that with any quasigroup expressed as $a * b = \sigma(f) \cdot \tau(g) \cdot h$, or one reduced variant with $\sigma = \text{id}$, $\tau = \text{id}$ or $h = 1$, we can derive a signature and key agreement scheme, as we can do it even with this linearized form.

This is a method to create cryptosystems once an automorphism on some abelian group is found. This construction of course doesn't exclude the fact that the cryptosystem can be broken in other ways.

4 Cryptosystem building example

Let's add an example of linearization that leads to a possible cryptosystem.

Let's work in $\mathbb{F}_{p^n}$, and define an abelian group $(G, \cdot)$, where $a \cdot b = ab + a + b$, $a, b \in \mathbb{F}_{p^n}$.

Without the need of an isomorphism let's define an automorphism $\sigma(a) = a^p$, $a \in \mathbb{F}_{p^n}$. σ is an automorphism of $(G, \cdot)$.

The resulting linearization and entropic operation is:

$a * b = \sigma(a) \cdot b$

So we can conclude that $\sigma(k)$ and k, for a given value k, are not lineary related in $(G, \cdot)$. In particular $\sigma(k) \cdot k = k^p \cdot k \neq k^{p+1}$, since the power in $\sigma(k)$ is in $\mathbb{F}_{p^n}$ and not in $(G, \cdot)$.

This is a simple illustrative example that cannot be broken by linearization, although is possible it can be broken in another ways.

References

[Gli21] Danilo Gligoroski. *Entropoid Based Cryptography.* Cryptology ePrint Archive, Paper 2021/469. 2021. URL: `https://eprint.iacr.org/2021/469`.

[Jia21] "Danny" Niu Jianfang. *Xifrat Cryptanalysis - Compute the Mixing Function Without the Key.* Cryptology ePrint Archive, Paper 2021/487. 2021. URL: `https://eprint.iacr.org/2021/487`.

[NJ21] Daniel Nager and "Danny" Niu Jianfang. *Xifrat - Compact Public-Key Cryptosystems based on Quasigroups.* Cryptology ePrint Archive, Paper 2021/444. 2021. URL: `https://eprint.iacr.org/2021/444`.

[Pan21] Lorenz Panny. *Entropoids: Groups in Disguise.* Cryptology ePrint Archive, Paper 2021/583. 2021. URL: `https://eprint.iacr.org/2021/583`.