Wave cryptanalysis and implementation tools

Quentin Assémat

June 9, 2022

Abstract

Wave's signature scheme presents many implementation difficulties. These potential errors can compromise the correctness of the distributions and therefore the security of the scheme. The aim of this work is to build a tool to detect these errors made by a well-intentioned developer. As this is an analysis tool and not a cryptanalysis, we have access to the private key and notably to the secret permutation used. We will see in the document that we focus on the so-called matched pairs since it is the sensitive content to be hidden in a signature file and that an error in the management of the distributions will be mostly found in the distribution of these pairs.

1 Wave in a nutshell

If we step back from Wave's signature algorithm, we can describe it in a simplified way as follows:

Algorithm 1 Wave signature algorithm

Require: the secret key sk, the number of signature to be produced N, and Sign() a signing procedure.
Ensure: Generate a signature file with N signatures.

$\mathcal{E} = \emptyset$
for $0i < n$ **do**
  $s \xleftarrow{\$} \mathbb{F}_3^{n-k}$
  $e \leftarrow \text{Sign}(sk, s)$
  $\mathcal{E} = \mathcal{E} \cup \{e\}$
END FOR
RETURN $\mathcal{E}$

In this algorithm, given a ternary $[n, k]$-code of (public) parity check matrix H and a syndrome $s \in \mathbb{F}_3^{n-k}$, the signature algorithm returns an error $e \in \mathbb{F}_3^n$ of Hamming weight w such that $eH^T = s$. The call Sign() makes use of a secret key sk known only by the legitimate user. The reader may refer to [DAST18] for more details on the signature algorithm.

The following algorithm simply generates words of weight w uniformly.

Algorithm 2 Word generation

Require: the number of word to be produced n, the target weight w.
Ensure: Generate a word file with n signature.
$\mathcal{W} = \emptyset$
for $0i < n$ **do**
$w \leftarrow \textsc{WordGen}(n, w)$
$\mathcal{W} = \mathcal{W} \cup \{w\}$
END FOR
RETURN $\mathcal{W}$

With an ideal implementation of WAVE, the output of algorithm 1 and 2 are unconditionally indistinguishable. In practice, because implementations use finite precision arithmetic, one only gets a computational indistinguishability.

2 Purpose of the tool

Purpose. The implementatio of WAVE invovles various random distributions. If the internal randomness is not correctly drawn, information on the secret key may leak, even though the produced signatures are perfectly valid [1]. The purpose of the tool provided here is to check whether a given implementation is compliant to the specification and produces correctly distributed signatures.

Limitations. However, it should be noted that this tool does not offer any guarantee that an implementation is correct. It is surely possible to make an implementation whose purpose is to validate the various statistical tests present while presenting a weakness. However, we believe that this tool will allow a developer seeking to follow the specifications of WAVE to detect a potential error in the management of the internal distributions

Computational advantage. Since the security proofs relies on the indistinguishability, we give ourselves the knowledge of the private key, which gives us a considerable (computational) advantage over an attacker in a "real" attack scenario. Even with this knowledge, a correctly generated signature file is not distinguishable from a w word file.

Challenge. The challenge our tool is trying to solve is to distinguish the random word generation algorithm (2) with the Wave signature algorithm (1), degraded (with a weak signing procedure $\textsc{Sign}_{\textsc{Degraded}}$) or not. We will illustrate with different potential errors that are all detected by this tool. This analysis also confirms the validity of the reference Wave implementation.

[1] One car notice that all the exemple algorithm (except the random word generation) in the github repository (`https://github.com/quentinassemat/wave-analysis`) produce valid signatures even though some of them are far from being secure.

3 Using "Wave tools"

3.1 Organisation of the code

The starting point of the construction of this tools was the actual reference implementation which is designed to generate a lot of signatures (`https://wave.inria.fr/en/implementation/`). We used this implementation to generate large files signatures (correctly distributed), but also signatures files with a biased distribution in order to test our tool. We will analyse these in the following. To analyse these signatures files, We used the bitsliced ternary arithmetic, the manipulation of secret key from the Wave reference implementation. In the files `analysis.c` lies all the performed tests which theoretical foundations are detailed in appendix.

3.2 How to use it ?

Normally a simple "*make all*" instruction will compile our source code. We provide some examples files of signatures and of uniformly distributed word of weight w to explain the structure of our data storage. The typical use will be the following : `./analysis-i <parameters id (i.e. usually 128g)> -f <path to signature file (i.e. usually ./Data/128g/sign_128g_0_0-499.dat)> -k <(optional) seed of the secret key>`

3.3 Example of bad implementation detection

Let's first use our tools on a files of uniformly distributed word of weight w. Thus we run the following : `./analysis -i 128g -f ../Data/128g/w > ../Wave_tools_output/word.out`. Our files of 10000 word will be analysed in a few second (depending on the number of pairs choosen for the order 2 analysis). The output will normaly look like something close to that :

```
Order 1 analysis
Order 2 analysis
Bernoulli/Chernoff test:
alpha : 0.050000, eps : 0.013581
Rejection among matched: 0 0 0 0 4 3 0 6 3
Rejection among random/non matched: 0 0 0 0 5 2 0 7 4
test 1 ... OK
Rejection among matched: 0 0 0
Rejection among random/non matched: 0 0 0
test 2 ... OK
.
.
.
Independance test :
Rejection : 0.000000
Rejection rejet_matched : 0.000000
Rejection rejet_rand : 0.000000
```

```
Rejection rejet_matched_rand : 0.000000
Tested pairs:
matched: (7134, 2623) (3037, 8099) (6899, 1625)
random/not matched: (8194, 8348) (741, 2238) (801, 3682)
```

All of our output (on the various signature/word tested files are available on the public github repositery `https://github.com/quentinassemat/wave-analysis` in the ouptut folder.

We see that all the tests are passed and that the statistical distance are very low and similar among the matched pairs and among non-matched pairs. Let's now test a real Wave signature file. We also see that all the tests are passed and that the statistical distance are very low and similar among the matched pairs and among non-matched pairs (*cf.* the output files on the repositery).

Now let's see some implementations errors that can be detected thanks to this tool. One of the simplification of the Wave signature algorithm would be to remove rejection sampling. Let's analyse a signature file generated as such. We then see that lot's of the test used detect that the signature file is not correctly distributed (*cf.* the output files on the repositery). Another simplification could be to take a dirac distribution centered to the typical weight (to prevent from order 1 attack only since order 2 attack implies large computation). This simplification is also detected (*cf.* the output files on the repositery). Another issue could be an index error in the developpment leading to a shifted distribution. This error is also detected (*cf.* the output files on the repositery).

4 Acknowledgment

In the context of my research internship at INRIA, under the supervision of Nicolas Sendrier and Thomas Debris-Alazard, I was brought to work on the realization of this tool of analysis of the implementations of the signature scheme of Wave. I would like to thank them for their help and support throughout this project.

List of Algorithms

References

[DAST18] Thomas Debris-Alazard, Nicolas Sendrier, and Jean-Pierre Tillich. Wave: A new family of trapdoor one-way preimage sampleable functions based on codes. Cryptology ePrint Archive, Paper 2018/996, 2018. `https://eprint.iacr.org/2018/996`.

A Appendix : theoretical construction of the statistical tests

A.1 Notation and definitions

Permutations and matched pairs. We previously seen that our tool have access to the secret key sk. As a consequence, withtout loosing any generality, we can see suppose in the following analysis that the matched pairs are the $(i, i + \frac{n}{2})$ for $0 \leqslant i \leqslant \frac{n}{2} - 1$. The reader may refer to [DAST18] more extended details on the secret permutation masking the matched pairs.

In the following analysis, the code length will be n, k the rank. We modelise the signature scheme as a probabilistic oracle. Indeed we note $(X_\ell)_{1\leqslant\ell\leqslant N}$ i.i.d random variable (representing the set of signature) identicaly distributed with the distribution $\mathcal{D}$ in $\mathbb{W}(n, w)$ the set of word of lenght n and of Hamming weight w. Let us recall that the Waved signature Scheme use the algebraic field $\mathbb{F}_3$ and then if $x \in \mathbb{W}(n, w)$, we can write $x = (x_0, .., x_{n-1})$ with $x_i \in \mathbb{F}_3$. We might sometimes write $x^{(\ell)} = (x_0^{(\ell)}, .., x_{n-1}^{(\ell)})$ to highlight that we deal with the ℓ-th signature.

Definition 1. Let's first start with some notations. Let $i \neq j \in \{0, ..., n-1\}$.

- $\mathcal{D}_{\text{target}}$ is the distribution of $(x_i, x_j) \in \mathbb{F}_3^2$ when $x = (x_0, ..., x_{n-1}) \sim \mathcal{U}$ the uniform distribution over $\mathbb{W}(n, w)$. This distribution is indenpendent of (i, j).
- $p_A \triangleq \Pr((x_i, x_j) \in A \mid x \sim \mathcal{U})$
- $p_{a,b} \triangleq \Pr((x_i, x_j) = (a, b) \mid x \sim \mathcal{U})$

The ideal case. Let's first analyse the distribution of a pair of index when the signature are distributed with the uniform distribution.

Lemma 1. *Let $i \neq j \in \{0, ..., n-1\}$ and $x \sim \mathcal{U}$. We then have :*

1. $p_{0,0} = p_0$
2. $p_{0,1} = p_{0,2} = p_{1,0} = p_{2,0} = \frac{1}{4} * p_1$
3. $p_{1,1} = p_{1,2} = p_{1,2} = p_{2,2} = \frac{1}{4} * p_2$

Where $p_0 \triangleq \frac{\binom{n-2}{w}}{\binom{n}{w}} = \frac{(n-w)(n-w-1)}{n(n-1)}$, $p_1 \triangleq \frac{2*\binom{n-1}{w-1}}{\binom{n}{w}} = \frac{2w(n-w)}{n(n-1)}$ *and* $p_2 \triangleq \frac{\binom{n-2}{w-2}}{\binom{n}{w}} = \frac{w(w-1)}{n(n-1)}$.

The pratical case. Our goal is to ensure with statistical test that the distribution $\mathcal{D}$ to be tested over $\mathbb{W}(n, w)$ is the uniform distribution $\mathcal{U}$. Moreover our prime goal is to protect the secret of the mathed index/pairs. For that purpose, we will work within the framework of statistical tests, and place ourselves in the adapted formalism that we recall here.

Statistical hypothesis. We define some null hypothesis wether focused on one pair, or on a set of pairs (possibly the matched ones). For our tests we note in the sequel α the level i.e. the probality of false rejection of the null hypothesis. In that purpose, let $x \sim \mathcal{D}$, where $\mathcal{D}$ the distribution to be tested and $\mathcal{K} \subset \{0, ..., n-1\}^2$.

$$\mathrm{H}_{0,A}(i,j): \quad \Pr((x_i, x_j) \in A \mid x \sim \mathcal{D}) = p_A.$$

$$\mathrm{H}_{0,\mathcal{K}}: \quad \forall (i,j) \in \mathcal{K}, (x_i,\ x_j) \sim \mathcal{D}_{\text{target}}.$$

$$\mathrm{H}_0: \quad \forall i \in \{0, ..., \tfrac{n}{2}\}, (x_i,\ x_{i+fracn2}) \sim \mathcal{D}_{\text{target}}.$$

A.2 Individual test

In this section we first focus on the statistical tests leading to reject or not null hypothesis of the form $\mathrm{H}_{0,A}(i,j)$. We will build more global test using these individual tests.

A.2.1 Bernoulli/Chernoff test

In order to check this distribution in the ouput of the implementation of the wave signature files, especially in the matched pairs, we do the following tests :

Test 1. *By browsing the N signatures of the files, we can estimate the distribution of the pair (x_i, x_j). Let be $A \subset \mathbb{F}_3^2$ a subset. The probability $\Pr((x_i, x_j) \in A \mid x \sim \mathcal{D})$ is the parameter of the Bernoulli variable $\mathbb{1}_{(x_i^{(\ell)}, x_j^{(\ell)}) \in A}$. Then we estimate with :*

$$\overline{p}_A = N^{-1} * \sum_{\ell=1}^{N} \mathbb{1}_{(x_i^{(\ell)}, x_j^{(\ell)}) \in A}$$

We can wrote thanks to the Hoeffding inequality the following :

$$\sup_{p \in [0,1]} \mathbb{P}(|\overline{p}_A - p_A| > \varepsilon) \leqslant 2\exp(-2N\varepsilon^2)$$

And then with $\varepsilon = \sqrt{\frac{1}{2N}\log(\frac{2}{\alpha})}$,
$\mathcal{I}_{\alpha,A} = [\overline{p}_A \pm \varepsilon]$ is a confidence interval for p_A of level α. We then reject the null hypothesis $H_{0,A}(i,j)$ if $p_A \notin \mathcal{I}_{\alpha,A}$ i.e. our test of level α is :

$$\phi((X_\ell)_{1 \leqslant \ell \leqslant N})_{i,j} = \mathbb{1}_{p_{a,b} \notin \mathcal{I}_{\alpha,A}} \quad \textit{(1 is rejection of the null hypothsesis)}$$

We will use this test with various idea for the subset A. For example we will test each probability $p_{a,b}$ with $A = \{a, b\}$ for all $a, b \in \mathbb{F}_3$, but we will also study the distribution with $A = \{(0,1),(1,0),(0,2),(2,0)\}$, $A = \{(1,1),(2,2)\}$, $A = \{(1,2),(2,1)\}$.

A.2.2 Khi-Square test

The Khi-square test allows us to have a more global study on the distribution. It seems justified to use it in spite of its asymptotic character because we often have a large number of signatures for our analyses.

Test 2. *By browsing the N signatures of the files, we can estimate the distribution of the pair (x_i, x_j). More precisely we compute the following for all $(a, b) \in \mathbb{F}_2^3$:*

$$\overline{p}_{a,b} = N^{-1} * \sum_{\ell=1}^{N} \mathbb{1}_{(x_i^{(\ell)}, x_j^{(\ell)})=(a,b)}$$

Then we can define the χ^2 statistic :

$$T = \sum_{a,b \in \mathbb{F}^3} N * \frac{(\overline{p}_{a,b} - p_{a,b})^2}{p_{a,b}}$$

Indeed, under the null-hyppothesis, this statistic follow a khi-square distribution with 8 degrees of freedom. We can then build an α level test using the quantile of ordre $1 - \alpha$ i.e.

$$\phi((X_\ell)_{1 \leqslant \ell \leqslant N})_{i,j} = \mathbb{1}_{T > F^{-1}{}_{X^2(8)}(1-\alpha)}$$

A.2.3 Metrics

Another idea to get a finer insight into the law of a pair is the use of various metrics between laws of probability such as statistical distance. We propose here the use of three metrics. Let us remind that $\mathcal{D}_{\text{target}} = (p_{a,b})_{a,b \in \mathbb{F}_3}$ the target distribution corresponding to uniformly distributed random words of weight w and $\mathcal{D} = (q_{a,b})_{a,b \in \mathbb{F}_3}$ the distribution for the studied pair.

$$\text{Statistical distance:} \qquad d_{\text{stat}}(\mathcal{D}, \mathcal{D}_{\text{target}}) = \sum_{a,b \in \mathbb{F}_3} |q_{a,b} - p_{a,b}|$$

$$\text{Kullback–Leibler divergence:} \qquad d_{\text{kl}}(\mathcal{D}, \mathcal{D}_{\text{target}}) = \sum_{a,b \in \mathbb{F}_3} q_{a,b} \log(\frac{q_{a,b}}{p_{a,b}})$$

$$\text{Hellinger distance:} \qquad d_{\text{hell}}(\mathcal{D}, \mathcal{D}_{\text{target}}) = \frac{1}{\sqrt{2}} \sqrt{\sum_{a,b \in \mathbb{F}_3} (\sqrt{q_{a,b}} - \sqrt{p_{a,b}})^2}$$

Motivations. We understand intuitively by noticing that for each of these metrics $d(\mathcal{D}, \mathcal{D}_{\text{target}}) = 0 \implies \mathcal{D} = \mathcal{D}_{\text{target}}$, that the closer the distribution, the lower the d metric. Experimentally, we will compute these metrics with the observed distribution (i.e. a statistical estimation of the distribution), but this fact is verified and these metrics are good distinguishers between distributions.

Limitations. However, it is complicated to establish a simple decision rule and then a proper statistical test since it is difficult to estimate a priori the value that these distances will take even under the null hypothesis.

A.3 Global test

In this subsection we build test to reject H_0 or $H_{0,K}$ from the previous tests. Indeed, from any individual test $\phi_{i,j}$ we can derive several tests concerning all the pairs in our sample $\mathcal{K} \subset \{(i,j),\ 0 \leqslant i,j \leqslant n-1\}$.

Consider the following two situations. In the first case a small number of matched pairs are treated with a great error in the distribution and the others follow the correct distribution. It is therefore appropriate to reject the null hypothesis as soon as a small number of unit tests with a low alpha level reject the pair in question. However, if we only do this gglobal test another disturbing situation goes unnoticed.

We can indeed imagine that all the pairs almost follow the correct distribution but that a very small bias sets in. In order to reject this situation we must also reject the overall test when a large number of pairs are rejected with a fairly large risk. There is therefore a balance to be found between the threshold of the number of unit tests that reject at which we reject at the global level, and the level of these unit tests.

This leads to the following study to construct a class of statistical tests of the same level, from alpha level unit tests. This test class will allow us to detect the two situations described above by varying the alpha level of the unit tests and the corresponding threshold.

Test 3. *Let $\phi_{i,j}$ for $(i,j) \in \mathcal{K}$ a series of test where $\mathcal{K}$ is the set of tested pairs . Let $t \leqslant |\mathcal{K}|$ be the threshold for the number of individual rejection among the tested pairs. Let α be the level of the individual test. We reject the null hypothesis $H_{0,\mathcal{K}}$ since at least t number of tested pairs are rejected i.e. :*

$$\phi((X_j)_{1\leqslant j\leqslant i})_{\mathcal{K},t} = \max_{\mathcal{J}\subset\mathcal{K},|\mathcal{J}|=t} \prod_{(i,j)\in\mathcal{J}} \phi((X_j)_{1\leqslant j\leqslant i})_{i,j}$$

Let us try to analyse the level of such a test. We place ourselves under the null hypothesis. We also assume that under the null hypothesis the rejection probability is the same for all test and correpond to the level of the test.

$$P_{H_0}(\phi = 1) = \sum_{l\geq t} \binom{\frac{n}{2}}{l} \alpha^l (1-\alpha)^{|\mathcal{K}-l|}$$

In this way we can empirically determine the α test level and the t threshold in order to achieve the type of overall statistical test we want while still reaching the desired level for the overall test. To do this we note the following limit which gives us a form of balancing of these parameters and which will allow us to easily find the parameters to have the desired overall test level.

Lemma 2. *(Experimental and unproven result, but useful to find the different parameters.)*

$$\lim_{n\to+\infty} \sum_{k=\alpha n}^{n} \binom{n}{k} \alpha^k (1-\alpha)^{(n-k)} = 0.5$$

Thus for a unit test level α it is sufficient to take t slightly away from $\alpha\frac{n}{2}$ in order to obtain an interesting global test level (typically 0.01).Thus experimentally with Wave's parameters with 128 bits of security $(n, w, k_U, k_V, d = 8492, 7980, 3558, 2047, 81)$ we obtain (for $\mathcal{K}$ corresponding to the set of matched pairs which implies $|\mathcal{K}| = \frac{n}{2}$) :

Unit test level α	Threshold t	Overall test level $P_{H_0}(\phi = 1)$
0.05	$246 = (\frac{n}{2}) * \alpha + 34$	0.01
0.5	$2199 = (\frac{n}{2}) * \alpha + 76$	0.01
0.95	$4066 = (\frac{n}{2}) * \alpha + 33$	0.01

A.4 Aditionnal empirical test with statistical distance

Although the statistical distances seem to allow a fine analysis of the distributions, it seems difficult to construct a statistical test with these metrics. Therefore, we simply calculate the empirical statistical distances to the target distribution and encourage all implementers of the Wave signature scheme to do the statistical distance calculations for their signature file but also for a word file of uniformly distributed weight w and compare the resulting values, both of which should be very small. This approach is not intended to replace the theoretical values obtained, but rather to provide a second security.

A.5 Higher order test

All the tests we have carried out are limited to an analysis that can be described as first-order since it does not observe correlations between the different pairs at any point. It is obviously possible to analyse and mount an attack of higher orders (in fact any implementation whose distributions are well realised up to order n only is theoretically fragile against an attack at order $n+1$). However, most of the errors of a well-intentioned developer will be found in these first tests and it is very expensive computationally speaking to set up tests on higher orders (we content ourselves with analysing about ten pairs in our tests with a calculation of the covariance matrix and a chi-square adequacy test).

A.6 Implementation choice

As this analysis is intended for well-intentioned developers and we assume that potential wave implementation errors are most pronounced in the distribution of matched pairs, we focus our testing on these pairs. In order to analyse whether a global bias is still present we also analyse $\frac{n}{2}$ random (a priori unmatched) pairs and apply to this set of pairs the same tests to which the matched pairs are subjected. Indeed all the performed test use

$$\mathcal{K}_{random} = \{(i, j),\ i \xleftarrow{\$} \{1, ..., n\}, j \xleftarrow{\$} \{1, ..., n\}\}$$

or

$$\mathcal{K}_{matched} = \{(i, i + \tfrac{n}{2}), 0 \leqslant i \leqslant \tfrac{n}{2} - 1\}$$