Rainbow

Eduardo Skapinakis: 98811
Luis Dias: 98819

May 2021

But mark! what arch of varied hue
From heaven to earth is bowed?
Haste, ere it vanish, haste to view
The Rainbow in the cloud

Hemans, The Rainbow

Abstract

Multivariate public key cryptography is one of the main approaches to guarantee the security of communication in a post-quantum world (see [PBB10]).

In this report we will study one of the most promising candidates in the area, the Rainbow signature scheme, first introduced by J. Ding and D. Schmidt in 2005 (see [DS05]).

The implementation of this signature scheme can be found in [MS21]

Contents

1 Introduction

We will start by introducing the key concepts regarding a signature scheme. The interested reader may refer to [Sti95] for a further reading on the topic.

Definition 1.1 (Signature Scheme)**.** A signature scheme is a method of signing a message stored in electronic form. To mimic the process of physically signing a document signature scheme consists of two components: a signing algorithm, where a user can sign a document, and a verification algorithm, which gives the option to verify a signature.
More formally, A signature scheme is a five-tuple, (P, A, K, S, V), where the following four conditions are satisfied:

- P, A and K are finite sets of possible messages, signatures and keys, respectively
- For each $k \in K$, there is a signing algorithm $sig_k : P \to A \in S$ and a corresponding verification algorithm $ver_k : P \times A \to \{0, 1\} \in V$
- For each pair $sign_k, v_k$, for every message $x \in P$ and every signature $y \in A$, $ver_K(x, y) = 1$ if and only if $y = sig_K(x)$

We are, of course, interested in systems where one can easily sign and verify a signature, so sig_k and ver_k should be computable in polynomial time, but it is impossible, or not feasible, to forge the signature of someone else.

Remark 1.2. If we have a public key cryptosystem, we can convert it to a signature scheme by letting the signature be the decryption of the document sent, and the verification be the assertion of whether or not the the signature, when encrypted again, matches the document sent.
Since only the holder of the secret key can apply the operation of decryption to the message, but anyone with the public key can encrypt it, we see that the condition are verified.
There is, however, a *problem* with these systems, which is that anyone can simulate a message, by computing $x = e_k(y)$ for some y. In this case, y is the signature of x, so the message (x, y) is a valid (document,signature) pair.
To eliminate this method of forging, we can use hash functions in conjunction with signature schemes.

2 Oil and Vinegar signature scheme

In order to explain Rainbow, we first need to mention Oil and Vinegar signature schemes. These are characterized by three things:

- k a finite field
- F a non-linear based on multivariable polynomials, with easy to compute F^{-1}
- L_1, L_2 linear invertible maps to mask the structure of F

This way we have a public key

$$\widetilde{F} = L_1 \circ F \circ L_2$$

and a private key

$$(L_1,\ F,\ L_2)$$

In summary, given a document y, it's signature is x a solution of $\tilde{F}(x) = y$ that is hard to compute knowing only $\widetilde{F}$.
Let k be a finite field and $o, v \in \mathbb{N}$. Consider polynomials $F_1, ..., F_o \in k[X_1, ..., X_o, X'_1, ...X'_v]$ of degree two of the form

$$F_l = \sum_{\substack{1\leq i\leq o\\ 1\leq j\leq v}} a_{l,i,j} X_i X'_j + \sum_{1\leq i,j\leq v} b_{l,i,j} X'_i X'_j + \sum_{1\leq i\leq o} c_{l,i} X_i + \sum_{1\leq j\leq v} d_{l,j} X'_i + e_l$$

with $l \in \{1, ..., o\}$. The variables X_i and X'_j are called the oil variables and the vinegar variables, respectively. We now define $F : k^{o+v} \to k^o$ such that for $x = (x_1, ..., x_o, x'_1, ..., x'_v) \in k^{o+v}$

$$F(x) = (F_1(x), ..., F_o(x))$$

In general, F need not be injective so it may not have an inverse. Hence we consider an inverse $F^{-1} : Im(F) \to k^{o+v}$ of F a function such that for every $y \in Im(F)$ we have $F(F^{-1}(y)) = y$. We have an easy way of computing F^{-1}:

1. We randomly guess the values for the $\{x'_j\}_{j=1,...,v}$ vinegar variables
2. We now have have a system of o linear equations over k so we solve them to find $\{x_i\}_{i=1,...,o}$
3. If the system does not have a solution, go back to 1

With probability close to 1, the system in step 2 is solvable so we can easily find a solution.
To mask the structure of F, we use $L_1 : k^o \to k^o, L_2 : k^{o+v} \to k^{o+v}$ which are invertible linear maps. They essentially represent a change of basis so it is difficult to tell the oil and vinegar variables apart. We then combine these maps to create

$$\widetilde{F} = L_1 \circ F \circ L_2$$

which is map that appears to be a random multivariable polynomial.

2.1 Balanced Oil and Vinegar

The above case with $o = v$, was cracked by Kipnis and Shamir [KS99] using matrices related to the bilinear forms defined by quadratic polynomials.

2.2 Unbalanced Oil and Vinegar

We always consider $o < v$. It was shown in [KPG99] that a specific attack has a complexity of roughly $o^4|k|^{v-o-1}$ when $v \approx o$. So by varying v and o we can make the scheme secure.
However, the document that we want to sign has size o but the signature is of size $o + v$ and since $v > o$ the signature is at least twice the size of the document, which is unpractical.

3 Rainbow

3.1 General Construction of Rainbow

Let K be a finite field and $S = \{1, \ldots, n\}$. We set $u \geq 1$, which will represent the number of "layers" our Rainbow will have, and set $v_1, \ldots, v_u$ to be such that $0 < v_1 < v_2 < \cdots < v_u = n$.
We then define the sets

$$S_l = \{1, 2, \ldots, v_l\} \quad O_l = S_{l+1} \setminus S_l$$

and set $o_l = v_{l+1} - v_l$. Note that v_l and o_l are the cardinality of S_l and O_l, respectively. Moreover, the sets S_l and O_l represent the Vinegar and Oil variables of each layer l.
Then, for the "layer" $l \in \{1, ..., u\}$ and "sub-layer" $k \in O_l$ we define, with $X = (X_1, ..., X_n)$,

$$F_{l,k}(X) = \sum_{\substack{1 \leq i \leq o \\ 1 \leq j \leq v}} \alpha_{i,j}^{l,k} X_i X_j + \sum_{i,j \in S_l} \beta_{i,j}^{l,k} X_i X_j + \sum_{i \in S_{l+1}} \gamma_i^{l,k} X_i + \eta^{l,k}$$

with coefficients in K. Note that $S_{l+1} = S_l \cup O_l$ so $F_{l,k}$ has the same form of a Vinegar-Oil polynomial.
This way, we define the map $F : K^n \to K^{n-v_1}$,

$$F(x) = (F_1(x), \ldots, F_u(x))$$

where $F_l = (F_{l,1},\ ...,\ F_{l,o_l})$.
Luckily, this map can easily be "inverted".
Let y be such that we want to solve $F(x) = y$:

1. First, choose $x_1, \ldots, x_{v_1}$, the vinegar variables of the first layer, at random.
2. We now have a system of o_1 linear equations, given by the polynomials in F_1, which we solve.
3. With the obtained values for $x_{v_1+1}, \ldots, x_{v_2}$ we get o_2 linear equations in the variables O_2 from F_2.
4. By repeating this process, we can find all the values of x

The only problem would be if the first values of x, that we guessed, were not a part of a solution for the system. Therefore, we we add the instruction: *if, at any time, the system is not solvable, we repeat the random guessing of the first the vinegar variables.*
From [Pat96], we know that we can expect to succeed with a very high probability, if the number of layers isn't too large. Hence, the algorithms is very likely to not repeat itself to often, looking for the right vinegar variables.

Remark 3.1. When $u = 1$, the Rainbow signature scheme is reduced to an Oil and Vinegar one.

3.2 Rainbow signature scheme

Employing the construction above, we define the Rainbow signature scheme as follows:

Key Generation:
The *private key* consists of two invertible linear maps, $L_1 : K^{n-v_1} \to K^{n-v_1}$ and $L_2 : K^n \to K^n$ and the map $F = (F_1(x), \ldots, F_u(x)))$. The *public key* consists of the field K and the composed map $\widetilde{F}(x) = L_1 \circ F \circ L_2$.

Signature Generation:
To sign a document, y, of size m, we start by padding it with a control character, until its size is a multiple of $n - v_1$. This way, we create $k = (n - v_1)/m$ sections of y.
Our signature will then be a tuple $x = (x_1, \ldots, x_k)$, where each x_i solves the system $\widetilde{F}(x_i) = y_i$.
Note that, since $\tilde{F}$ is not injective, x is not unique.

Verification:
To verify the authenticity of a signature, we just have to check if $\widetilde{F}(x_i) = y_i$, for every section y_i.

References

[DS05] Jintai Ding and Dieter Schmidt. Rainbow, a new multivariable polynomial signature scheme. In John Ioannidis, Angelos D. Keromytis, and Moti Yung, editors, Applied Cryptography and Network Security, Third International Conference, ACNS 2005, New York, NY, USA, June 7-10, 2005, Proceedings, volume 3531 of Lecture Notes in Computer Science, pages 164–175, 2005.

[MS21] Luis Miguelote and Eduardo Skapinakis. Rainbow. `https://github.com/eskapinakis/Rainbow`, 2021.

[Pat96] Jacques Patarin. Hidden fields equations (hfe) and isomorphisms of polynomials (ip): Two new families of asymmetric algorithms. In International Conference on the Theory and Applications of Cryptographic Techniques, pages 33–48. Springer, 1996.

[PBB10] Albrecht Petzoldt, Stanislav Bulygin, and Johannes Buchmann. Selecting parameters for the rainbow signature scheme. In Nicolas Sendrier, editor, Post-Quantum Cryptography, Third International Workshop, PQCrypto 2010, Darmstadt, Germany, May 25-28, 2010. Proceedings, volume 6061 of Lecture Notes in Computer Science, pages 218–240. Springer, 2010.

[Sti95] Douglas Stinson. Cryptography Theory and Practice. CRC Press, 1995.