An algebraic attack for forging signatures of MPPK/DS

Hao Guo

March 29, 2023

Abstract

We give an algebraic attack to forge the signature of a scheme called MPPK/DS, which can be achieved by solving a linear system in 5 variables with coefficients on $\mathbb{Z}/2^x q\mathbb{Z}$ for some odd prime q and $x \geq 1$.

For the signature scheme MPPK/DS[1], the public key is given by four polynomials in $(\mathbb{Z}/\phi(p)\mathbb{Z})[x_0, x_1, \ldots, x_m]$[1], namely $P(x_0, x_1, \ldots, x_m), Q(x_0, x_1, \ldots, x_m)$ and $N_0(x_1, \ldots, x_m), N_n(x_0, x_1, \ldots, x_m)$.

The message/signature pair is (μ, A, B, C, D, E), and it is valid if and only if for a random chosen $r_1, \ldots, r_n \in \mathbb{Z}/\phi(n)\mathbb{Z}$,

$$A^{Q(\mu,r_1,\ldots,r_m)} = B^{P(\mu,r_1,\ldots,r_m)} C^{N_0(r_1,\ldots,r_m)} D^{N_n(\mu,r_1,\ldots,r_m)} E \pmod{p}$$

Here we give a way to forge the signature of the message $\mu = \mu_0$, given the public key polynomials P, Q, N_0, N_n. Firstly, choose a multiplicative generator g of $GF(p)^*$. We can assume $A = g^a$, $B = g^b$, $C = g^c$, $D = g^d$, $E = g^e$. Then we only need to solve

$$aQ(\mu_0, x_1, \ldots, x_m) = bP(\mu_0, x_1, \ldots, x_m) + cN_0(x_1, \ldots, x_m) + dN_n(\mu_0, x_1, \ldots, x_m) + e \pmod{\phi(p)}$$

For any $x_1, \ldots, x_n$. The construction of public key P, Q, N_0, N_n ensures that a nontrivial solution exists where a, b, c, d, e are not all zero. If we organize both sides by monomials in $x_1, \ldots, x_m$, we get $(l_1+1)(l_2+1)\ldots(l_m+1)$ linear equations on a, b, c, d, e over the ring $\mathbb{Z}/\phi(p)\mathbb{Z}$.

Since p has the form of $p = 2^x q + 1$ where q is a large prime (hence an odd number), we can easily get x and q from p. In fact, divide $p - 1$ by 2 until the result is an odd number, and the number of times of division is just x, the remaining part being q. Using Chinese Remainder Theorem, we can reduce the problem into solving on $\mathbb{Z}/q\mathbb{Z}$ and $\mathbb{Z}/2^x\mathbb{Z}$, and combining them later.

Notice that e only appears in the equation corresponding to the constant term, and the other equations are linear in a, b, c, d. Therefore we deal with

[1] The authors originally wrote $GF(\phi(p))$ which does not hold since the order of a finite field can only be a prime power, and not $\phi(p) = 2^x q$.

e first. We first assume $q|e$ or $2^x|e$ doesn't hold. This holds with probability about $1 - 2^{-31}$ for level 5. On $\mathbb{Z}/2^x\mathbb{Z}$, the unit elements are odd numbers, and the equivalence class under multiplying by a unit is just $0, 1, 2, 4, \ldots, 2^{x-1}$, of which 0 is excluded. Therefore we plug in $e_2 = 1, 2, 4, \ldots, 2^{x-1}$ one by one, until a solution of a_2, b_2, c_2, d_2 exists. On $\mathbb{Z}/q\mathbb{Z}$, the nonzero elements are all equivalent to 1, so we can assume $e_q = 1$ and solve for a_q, b_q, c_q, d_q.

Now we get $a \equiv a_2 \pmod{2^x}$ and $a \equiv a_q \pmod{q}$ for some $a_2, a_q \in \mathbb{Z}$, we can use bezout's lemma to find $a \pmod{2^x q}$. If $m_2 2^x + m_q q = 1$ for some $m_2, m_q \in \mathbb{Z}$, we can get $a = a_2 m_q q + a_q m_2 2^x \pmod{\phi(x)}$. We do this similarly for b, c, d. Finally we calculate $e = e_2 m_q q + e_q m_2 2^x \pmod{\phi(x)}$. Then $(\mu, A, B, C, D, E) = (\mu, g^a, g^b, g^c, g^d, g^e)$ is a signature that can pass the check.

We show the case of Level 5 parameters using Sagemath[2], the code can be viewed on Github.[2]. We choose $x = 32$, $q = 6781572043$ and $p = 29126630140152905729$. To sign the message $\mu = 25519$, the signer's result and our forged result are

$$\begin{cases} a = 23907647448598142180, \\ b = 217585470632989176, \\ c = 21573626300939042408, \\ d = 15485360123797689700, \\ e = 28500975508241867136, \end{cases} \qquad \begin{cases} a' = 23621274555729833555, \\ b' = 15998325096079169 70, \\ c' = 28668653663269643358, \\ d' = 24604005124511413683, \\ e' = 10020988832846971 20. \end{cases}$$

References

[1] Randy Kuang, Maria Perepechaenko, and Michel Barbeau. A new quantum-safe multivariate polynomial public key digital signature algorithm. *Scientific Reports*, 12(1):13168, 2022.

[2] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version x.y.z)*, YYYY. `https://www.sagemath.org`.

[2] `https://github.com/guoh064/Attack_on_MPPKDS/`