
University of Oxford
Mathematical Institute
2nd Oxford
Post-Quantum Cryptography
Summit 2023
PQ SHIELD
National Cyber Security Centre

2nd Oxford Post-Quantum Cryptography Summit 2023

KAZ – SIGN v1.3

MUHAMMAD REZAL KAMEL ARIFFIN[1&2] (“REZAL”)

[1]DEPARTMENT OF MATHEMATICS & STATISTICS, FACULTY OF SCIENCE,

[2]INSTITUTE FOR MATHEMATICAL RESEARCH

UNIVERSITI PUTRA MALAYSIA

UPM UNIVERSITI PUTRA MALAYSIA BERILMU BERBAKTI

DESIGN IDEALISME

2nd Oxford
Post-Quantum Cryptography
Summit 2023

(i) To be based upon a problem that could be proven analytically to require exponential time to be solved;

(ii) To be able to prove analytically that the cryptosystem is indeed resistant towards quantum computers;

(iii) To utilize problems mentioned in point (i) above in its full spectrum without having to induce "weaknesses" in order for a trapdoor to be constructed;

(iv) To use "simple" mathematics in order to achieve maximum simplicity in design, such that even practitioners with limited mathematical background will be able to understand the arithmetic;

DESIGN IDEALISME

2nd Oxford Post-Quantum Cryptography Summit 2023

(v) Achieve 128 and 256-bit security with key length roughly equivalent to the non-quantum secure Elliptic Curve Cryptosystem (ECC);

(vi) To achieve maximum speed upon having simplicity in design and short key length;

(vii) To have a sufficiently large signature space;

(viii) The computation overhead for both signing and verification increases slightly even if the key size increases in the future;

(ix) To be able to be mounted on hardware with ease;

(x) The plaintext to signature expansion ratio is kept to a minimum.

DESIGN IDEALISME

2nd Oxford Post-Quantum Cryptography Summit 2023

(v) Achieve 128 and 256-bit security with key length roughly equivalent to the non-quantum secure Elliptic Curve Cryptosystem (ECC);

(vi) To achieve maximum speed upon having simplicity in design and short key length;

(vii) To have a sufficiently large signature space;

(viii) The computation overhead for both signing and verification increases slightly even if the key size increases in the future;

(ix) To be able to be mounted on hardware with ease;

(x) The plaintext to signature expansion ratio is kept to a minimum.

DESIGN IDEALISME

2nd Oxford
Post-Quantum
Cryptography
Summit 2023

MODULAR REDUCTION PROBLEM (MRP)

Let $N = \prod_{i=1}^{j} p_i$ be a composite number and $n = \ell(N)$. Let p_k be a factor of N. Choose $\alpha \in (2^{n-1}, N)$. Compute $A \equiv \alpha \pmod{p_k}$.

The MRP is, upon given the values (A, N, p_k), one is tasked to determine $\alpha \in (2^{n-1}, N)$.

DESIGN IDEALISME

2nd Oxford
Post-Quantum
Cryptography
Summit 2023

COMPLEXITY OF SOLVING THE MRP

Let $n_{p_k} = \ell(p_k)$ be the bit length of p_k. The complexity to obtain α is $O(2^{n-n_{p_k}})$. When deploying Grover's algorithm on a quantum computer, the complexity to obtain α is $O(2^{\frac{n-n_{p_k}}{2}})$. In other words, if $p_k \approx N^{\delta}$, for some $\delta \in (0,1)$, the complexity to obtain α is $O(N^{1-\delta})$. When deploying Grover's algorithm on a quantum computer, the complexity to obtain α is $O(N^{\frac{1-\delta}{2}})$.

DESIGN IDEALISME

2nd Oxford
Post-Quantum
Cryptography
Summit 2023

THE HERMANN MAY REMARKS (Herrmann and May, 2008)

We will now observe two remarks by Herrmann and May. It discusses the ability and inability to retrieve variables from a given modular multivariate linear equation. But before that we will put forward a famous theorem of Minkowski that relates the length of the shortest vector in a lattice to the determinant (see Hoffstein et al. (2008)).

Theorem 1. *In an ω-dimensional lattice, there exists a non-zero vector v with*

$$\|v\| \leq \sqrt{\omega}\, det(L)^{\frac{1}{\omega}}$$

In lattices with fixed dimension we can efficiently find a shortest vector, but for arbitrary dimensions, the problem of computing a shortest vector is known to be NP-hard under ran-

DESIGN IDEALISME

domized reductions (see Ajtai (1998)). The LLL algorithm, however, computes in polynomial time an approximation of the shortest vector, which is sufficient for many applications.

Remark 1. *Let* $f(x_1, x_2, \ldots, x_k) = a_1x_1 + a_2x_2 + \ldots + a_kx_k$ *be a linear polynomial. One can hope to solve the modular linear equation* $f(x_1, x_2, \ldots, x_k) \equiv 0 \pmod{N}$*, that is to be able to find the set of solutions* $(y_1, y_2, \ldots, y_k) \in \mathbb{Z}_N^k$*, when the product of the unknowns are smaller than the modulus. More precisely, let* X_i *be upper bounds such that* $|y_i| \leq X_i$ *for* $1, \ldots, k$*. Then one can roughly expect a unique solution whenever the condition* $\prod_i X_i \leq N$ *holds (see Herrmann and May (2008)). It is common knowledge that under the same condition* $\prod_i X_i \leq N$ *the unique solution* $(y_1, y_2, \ldots, y_k)$ *can heuristically be recovered by computing the shortest vector in an k-dimensional lattice by the LLL algorithm. In fact, this approach lies at the heart of many cryptographic results (see Bleichenbacher and May (2006); Girault et al. (1990) and Nguyen (2004)).*

DESIGN IDEALISME

We would like to provide the reader with the conjecture and remark given in Herrmann and May (2008).

Conjecture 1. *If in turn we have* $\prod_i X_i \geq N^{1+\varepsilon}$ *then the linear equation* $f(x_1, x_2, \ldots, x_k) = \sum_{i=1}^{k} a_i x_i \equiv 0 \pmod{N}$ *usually has* N^{ε} *many solutions, which is exponential in the bit-size of* N.

Remark 2. *From Conjecture 1, there is hardly a chance to find efficient algorithms that in general improve on this bound, since one cannot even output all roots in polynomial time.*

KAZ-SIGN SYSTEM PARAMETERS

2nd Oxford Post-Quantum Cryptography Summit 2023

System Parameters

From the given security parameter k, determine parameter j. Next generate a list of the first j-primes larger than 2, $P = \{p_i\}_{i=1}^{j}$. Let $N = \prod_{i=1}^{j} p_i$. As an example, if $j = 43$, N is 256-bits. Let $n = \ell(N)$ be the bit length of N. Choose a random prime in $g \in \mathbb{Z}_N$ of order G_g where at most $G_g \approx N^{\delta}$ for a chosen value of $\delta \in (0, 1)$ and $\delta \to 0$. That is, $g^{G_g} \equiv 1 \pmod{N}$. Choose a random prime $R \in \mathbb{Z}_{\phi(N)}$ of order G_R, where $G_R \approx \phi(N)^{\varepsilon}$ for $\varepsilon \to 1$.

CONTINUE…

KAZ-SIGN SYSTEM PARAMETERS

That is, choose R with a large order in $\mathbb{Z}_{\phi(N)}$. Let $n_{G_R} = \ell(G_R)$ be the bit length of G_R. Such R, has its own natural order in $Z_{\phi(G_g)}$. Let that order be denoted as G_{Rg}. We can observe the natural relation given by $R^{G_{Rg}} \equiv 1 \pmod{G_g}$ where $\phi(N) \equiv 0 \pmod{G_g}$ and $\phi(G_g) \equiv 0 \pmod{G_{Rg}}$. Let $n_{\phi(G_g)} = \ell(\phi(G_g))$ be the bit length of $\phi(G_g)$ and $n_{\phi(G_{Rg})} = \ell(\phi(G_{Rg}))$ be the bit length of $\phi(G_{Rg})$. Let q be a random k-bit prime where $(q-1)2^{-1}$ is a prime. The system parameters are $(g, k, q, N, R, G_g, G_{Rg}, n, n_{\phi(G_g)})$.

KAZ-SIGN KEY GENERATION

2nd Oxford Post-Quantum Cryptography Summit 2023

Algorithm 1 KAZ-SIGN Key Generation Algorithm

Input: System parameters $(g, k, q, N, R, G_g, G_{Rg}, n, n_{\phi(G_g)})$.

Output: Public verification key pair, $V = (V_1, V_2)$, and private signing key, α

1: Choose random $\alpha \in (2^{n_{\phi(G_g)}-2}, 2^{n_{\phi(G_g)}-1})$.

2: Compute public verification key, $V_1 \equiv \alpha \pmod{G_{Rg}q}$.

3: Compute public verification key, $V_2 = H(\alpha^4 \pmod{q^2})$.

4: Output public verification key pair, $V = (V_1, V_2)$ and private signing key α.

KAZ-SIGN SIGNING

Algorithm 2 KAZ-SIGN Signing Algorithm

Input: System parameters $(g, k, q, N, R, G_g, G_{Rg}, n, n_{\phi(G_g)})$, private signing key, α, and message to be signed, $m \in \mathbb{Z}_N$

Output: Signature pair, $S = (S_1, S_2)$.

1: Compute the hash value of the message, $h = H(m)$.
2: Choose random $r \in (2^{n_{\phi(G_g)}-2}, 2^{n_{\phi(G_g)}-1})$.
3: Compute $S_1 \equiv r \pmod{G_{Rg}q^2}$.
4: Compute $GS_1 = \gcd(S_1, G_{Rg})$.
5: Compute $GS_{12} = \gcd(r, \phi(G_{Rg}q^2(G_{S1}^{-1})))$.

CONTINUE…

KAZ-SIGN SIGNING

6: **if** $GS_{12} < 100$ **then**
7: Repeat from Step 2.
8: **end if**
9: Compute $S_2 \equiv GS_1(\alpha^{S_1} + h)r^{-1} \pmod{G_{Rg}q^2}$.
10: **if** S_2 does not exist **then**
11: Repeat from Step 2.
12: **end if**
13: Output signature pair, $S = (S_1, S_2)$, and destroy r.

KAZ-SIGN VERIFICATION

2nd Oxford Post-Quantum Cryptography Summit 2023

Algorithm 3 KAZ-SIGN Verification Algorithm

Input: System parameters $(g, k, q, N, R, G_g, G_{Rg}, n, n_{\phi(G_g)})$, public verification key pair, $V = (V_1, V_2)$, message, m, and, signature pair, $S = (S_1, S_2)$.

Output: Accept or reject

1: Compute the hash value of the message to be verified, $h = H(m)$.
2: Compute $GS_{1r} = \gcd(S_1, G_{Rg})$.
3: Compute $\alpha_F \equiv V_1 \pmod{G_{Rg}}$.
4: Compute $w_0 \equiv GS_{1r}(V_1^{S_1} + h)S_1^{-1} \pmod{G_{Rg}q^2}$.
5: Compute $w_1 = w_0 - S_2$.
6: **if** $w_1 = 0$ **then**
7: Reject signature $\perp$
8: **else** Continue step 10
9: **end if**

CONTINUE…

KAZ-SIGN VERIFICATION

2nd Oxford Post-Quantum Cryptography Summit 2023

10: Compute $w_2 \equiv GS_{1r}(\alpha_F^{S_1} + h)S_1^{-1} \pmod{G_{Rg}q^2}$.
11: Compute $w_3 = w_2 - S_2$.
12: **if** $w_3 = 0$ **then**
13: Reject signature $\perp$
14: **else** Continue step 16
15: **end if**
16: Compute $w_4 \equiv S_1S_2 - GS_{1r}h \pmod{q}$
17: Compute $w_5 \equiv GS_{1r}V_1^{S_1} \pmod{q}$
18: Compute $w_6 = w_4 - w_5$
19: **if** $w_6 \neq 0$ **then**
20: Reject signature $\perp$
21: **else** Continue step 23
22: **end if**

CONTINUE…

KAZ-SIGN VERIFICATION

2nd Oxford Post-Quantum Cryptography Summit 2023

23: Compute $w_7 = 2S_1^{-1} \pmod{\left(\frac{\phi(q^2)}{2}\right)}$.

24: Compute $w_{80} \equiv \left((S_1S_2 - GS_{1r}h)(GS_{1r})^{-1}\right)^{2w_7} \pmod{q^2}$ and $w_8 = H(w_{80})$.

25: Compute $w_9 = w_8 - V_2$.

26: **if** $w_9 \neq 0$ **then**

27: Reject signature $\perp$

28: **else** Continue step 30

29: **end if**

CONTINUE…

KAZ-SIGN VERIFICATION

2nd Oxford Post-Quantum Cryptography Summit 2023

30: Compute $z_0 \equiv R^{S_1 S_2} \pmod{Gg}$.
31: Compute $y_1 \equiv g^{z_0} \pmod{N}$.
32: Compute $z_1 \equiv R^{GS_{1r}(V_1^{S_1}+h) \pmod{G_{Rg}}} \pmod{G_g}$.
33: Compute $y_2 \equiv g^{z_1} \pmod{N}$.
34: **if** $y_1 = y_2$ **then**
35: accept signature
36: **else** reject signature $\perp$
37: **end if**

KAZ-SIGN VERIFICATION

2nd Oxford
Post-Quantum
Cryptography
Summit 2023

Steps 4, 5, 6, 7, 8, and 9 during verification are known as the **KAZ-SIGN digital signature forgery detection procedure type – 1**, steps 10, 11, 12, 13, 14 and 15 during verification are known as the **KAZ-SIGN digital signature forgery detection procedure type – 2**, steps 16, 17, 18, 19, 20, 21, and 22 during verification are known as the **KAZ-SIGN digital signature forgery detection procedure type – 3**, and steps 23, 24, 25, 26, 27, 28, and 29 are known as the **KAZ-SIGN digital signature forgery detection procedure type – 4**.

THE DESIGN RATIONALE

2nd Oxford Post-Quantum Cryptography Summit 2023

Proof of correctness (Verification steps 30, 31, 32, 33, 34, 35, 36 and 37)

We begin by discussing the rationale behind steps 30, 31, 32, 33, 34, 35, 36 and 37 with relation to the verification process. Observe the following,

$$g^{z_0} \equiv g^{R^{S_1 S_2}} \equiv g^{R^{rGS_1 r(\alpha^{S_1}+h)(r)^{-1}}} \equiv g^{R^{GS_1 r(V_1^{S_1}+h)}} \equiv g^{z_1} \pmod{N}.$$

because $\alpha \equiv V_1 \pmod{G_{Rg}}$. As such the verification process does indeed provide an indication that the signature is indeed from an authorized sender with the private signing key α.

CONTINUE…

THE DESIGN RATIONALE

2nd Oxford Post-Quantum Cryptography Summit 2023

Proof of correctness (Verification steps 4, 5, 6, 7, 8, and 9: KAZ-SIGN digital signature forgery detection procedure type – 1)

In order to comprehend the rationale behind steps 4, 5, 6, 7, 8, and 9, one has to observe the following,

$$w_0 \equiv GS_{1r}(V_1^{S_1} + h)S_1^{-1} \not\equiv GS_{1r}(\alpha^{S_1} + h)S_1^{-1} \pmod{G_{Rg}q^2}$$

because $\alpha \not\equiv V_1 \pmod{G_{Rg}q^2}$. Hence, $w_1 \neq 0$.

CONTINUE…

THE DESIGN RATIONALE

2nd Oxford Post-Quantum Cryptography Summit 2023

Proof of correctness (Verification steps 10, 11, 12, 13, 14 and 15: KAZ-SIGN digital signature forgery detection procedure type – 2)

In order to comprehend the rationale behind steps 10, 11, 12, 13, 14 and 15, one has to observe the following;

$$w_2 \equiv GS_{1r}(\alpha_F^{S_1} + h)S_1^{-1} \not\equiv GS_{1r}(\alpha^{S_1} + h)S_1^{-1} \pmod{G_{Rg}q^2}.$$

because $\alpha \not\equiv \alpha_F \pmod{G_{Rg}q^2}$ where $\alpha_F \equiv V_1 \pmod{G_{Rg}}$. Hence, $w_3 \neq 0$.

CONTINUE…

THE DESIGN RATIONALE

Proof of correctness (Verification steps 16, 17, 18, 19, 20, 21, and 22: KAZ-SIGN digital signature forgery detection procedure type – 3)

In order to comprehend the rationale behind steps 16, 17, 18, 19, 20, 21, and 22, one has to observe

$$S_1S_2 - GS_{1r}h \equiv GS_{1r}V_1^{S_1} \pmod{q}$$

because $\alpha \equiv V_1 \pmod{q}$. Hence, $w_6 = 0$.

CONTINUE…

THE DESIGN RATIONALE

2nd Oxford Post-Quantum Cryptography Summit 2023

Proof of correctness (Verification steps 23, 24, 25, 26, 27, 28, and 29 : KAZ-SIGN digital signature forgery detection procedure type – 4)

In order to comprehend the rationale behind steps 23, 24, 25, 26, 27, 28, and 29, one has to observe

$$w_{80} \equiv \left((S_1S_2 - GS_{1r}h)(GS_{1r})^{-1}\right)^{2w_7} \equiv (\alpha^{S_1})^{2w_7} \equiv \alpha^4 \pmod{q^2}.$$

By computing $w_8 = H(w_{80})$, we finally have $w_9 = 0$.

CONTINUE…

THE DESIGN RATIONALE

2nd Oxford Post-Quantum Cryptography Summit 2023

Deriving forged signature identifiable by KAZ-SIGN digital signature forgery detection procedure type – 1 and KAZ-SIGN digital signature forgery detection procedure type – 2.

An adversary utilizing a random r_0 computes the corresponding parameter pair given by $(S_1 \pmod{G_{Rg}q^2}, GS_{1r})$. Next, the adversary could compute either one of the following:

1. $S_2 \equiv GS_{1r}(V_1^{S_1} + h)S_1^{-1} \pmod{G_{Rg}q^2}$; or
2. $S_2 \equiv GS_{1r}(\alpha_F^{S_1} + h)S_1^{-1} \pmod{G_{Rg}q^2}$

Since $\alpha \equiv V_1 \equiv \alpha_F \pmod{G_{Rg}}$, the forged signature pair will pass steps 30, 31, 32, 33, 34, 35, 36 and 37. However, the signature pair will fail KAZ-SIGN digital signature forgery detection procedure type – 1 or KAZ-SIGN digital signature forgery detection procedure type – 2.

CONTINUE…

THE DESIGN RATIONALE

2nd Oxford Post-Quantum Cryptography Summit 2023

Deriving forged signature identifiable by KAZ-SIGN digital signature forgery detection procedure type – 3

An adversary utilizing a random r_0 computes the corresponding parameter pair given by $(S_1 \pmod{G_{Rg}q^2}, GS_{1r})$. Next, with a random $x \in \mathbb{Z}_{G_{Rg}q^2}$ and random unknown prime $\rho \approx q$, the adversary could compute either one of the following:

i) $S_2 \equiv GS_{1r}(V_1^{S_1} + h + G_{Rg}x)S_1^{-1} \pmod{G_{Rg}q^2}$; or

ii) $S_2 \equiv GS_{1r}(V_1^{S_1} + h + G_{Rg}x)S_1^{-1} \pmod{G_{Rg}\rho q}$; or

iii) $S_2 \equiv GS_{1r}(\alpha_F^{S_1} + h + G_{Rg}x)S_1^{-1} \pmod{G_{Rg}q^2}$; or

iv) $S_2 \equiv GS_{1r}(\alpha_F^{S_1} + h + G_{Rg}x)S_1^{-1} \pmod{G_{Rg}\rho q}$.

CONTINUE…

THE DESIGN RATIONALE

The forged signature pair will not be able to be detected by either the KAZ-SIGN digital signature forgery detection procedure type – 1 or KAZ-SIGN digital signature forgery detection procedure type – 2. It will also pass steps 30, 31, 32, 33, 34, 35, 36 and 37. However, the signature pair will fail KAZ-SIGN digital signature forgery detection procedure type – 3. This is because, one would obtain either:

i) $S_1 S_2 - G S_{1r} h \equiv G S_{1r} (V_1^{S_1} + G_{Rg} x) \not\equiv G S_{1r} V_1^{S_1} \pmod{q}$; or

ii) $S_1 S_2 - G S_{1r} h \equiv G S_{1r} (\alpha_F^{S_1} + G_{Rg} x) \not\equiv G S_{1r} V_1^{S_1} \pmod{q}$.

As a note, the corresponding parameter S_1 could also be modulo $G_{Rg} \rho q$. Nevertheless, the above output will remain.

CONTINUE…

THE DESIGN RATIONALE

An alternative for the adversary would be to derive the corresponding S_1 modulo $G_{Rg}q^2$ by solving the following relation:

$$S_1 S_2 - G S_{1r} h \equiv G S_{1r} V_1^{S_1} \pmod{G_{Rg} q^2} \quad (1)$$

However, to solve equation (1), the complexity is is $O(q)$. When deploying Grover's algorithm on a quantum computer, the complexity will be $O(q^{0.5})$. Furthermore q is a k-bit prime number (where k is either 128 or 192 or 256 bits). The adversary will not be able to execute the Chinese Remainder Theorem to reduce this complexity.

THE DESIGN RATIONALE

2nd Oxford Post-Quantum Cryptography Summit 2023

Deriving forged signature identifiable by KAZ-SIGN digital signature forgery detection procedure type – 4

An adversary utilizing a random r_0 and random unknown prime $\rho \approx q$ computes the corresponding parameter pair $(S_1 \pmod{G_{Rg}\rho q}, GS_{1r})$. Next, the adversary could compute the following:

$$S_2 \equiv GS_{1r}(V_1^{S_1} + h)S_1^{-1} \pmod{G_{Rg}\rho q}$$

The forged signature pair will not be able to be detected by either the KAZ-SIGN digital signature forgery detection procedure type – 1 or KAZ-SIGN digital signature forgery detection procedure type – 2 or KAZ-SIGN digital signature forgery detection procedure type – 3. It will also pass steps 30, 31, 32, 33, 34, 35, 36 and 37. However, the signature pair will fail KAZ-SIGN digital signature forgery detection procedure type – 4. This is because of the different groups $\mathbb{Z}_{G_{Rg}\rho q}$ and $\mathbb{Z}_{G_{Rg}q^2}$.

CONTINUE…

THE DESIGN RATIONALE

2nd Oxford
Post-Quantum
Cryptography
Summit 2023

Note that, by replacing V_1 with α_F for the above forgery strategy in this section, the forged signature will not pass KAZ-SIGN digital signature forgery detection procedure type – 3. This is because $\alpha_F \not\equiv V_1 \pmod{q}$.

THE DESIGN RATIONALE

Extracting $\alpha \pmod{G_{Rg}q^2}$ from S_2.

Observe that,

$$z_1 \equiv S_1 S_2 - G S_{1r} h \equiv G S_{1r} \alpha^{S_1} \pmod{G_{Rg}q^2}.$$

Since $G_{Rg} \equiv 0 \pmod{G S_{1r}}$, we can have

$$z_2 \equiv z_1 G S_{1r}^{-1} \equiv \alpha^{S_1} \pmod{G_{Rg2}q^2} \tag{2}$$

where $G_{Rg2} = G_{Rg} G S_{1r}^{-1}$. However, $\gcd(S_1, \phi(G_{Rg2}q^2)) \neq 1$. Suppose $z_3 = \gcd(S_1, \phi(G_{Rg2}q^2))$. One can then proceed to compute $z_4 \equiv z_3 S_1^{-1} \pmod{\phi(G_{Rg2}q^2)}$. As a result, one can obtain:

$$z_2^{z_4} \equiv \alpha^{z_3} \pmod{G_{Rg2}q^2} \tag{3}$$

Thus, for both cases (2) and (3), the complexity to obtain α modulo $G_{Rg2}q^2$ is $O(G_{Rg2}q^2)$.

THE DESIGN RATIONALE

Extracting α via KAZ-SIGN digital signature forgery detection procedure type – 4

Through the KAZ-SIGN digital signature forgery detection procedure type – 4, the adversary can proceed to obtain the value $w_{81} \equiv \alpha \pmod{q^2}$ from the extracted value $w_{80} \equiv \alpha^4 \pmod{q^2}$ from a valid signature.

Then, the challenge faced the adversary is to retrieve α from $w_{81} \equiv \alpha \pmod{q^2}$. This is the MRP. Since $G_{Rg}q < q^2$, the complexity of solving the MRP via V_1 is much higher than the complexity of solving the MRP via $\alpha \pmod{q^2}$.

As such, the complexity of solving the MRP via $\alpha \pmod{q^2}$ will be the determining factor in identifying the suitable key length for each security level.

THE DESIGN RATIONALE

2nd Oxford Post-Quantum Cryptography Summit 2023

Modular linear equation of S_2.

In this direction we obtain $r_F \equiv S_1 \pmod{G_{Rg}}$.

From the above, observe that one can analyze S_2 as follows,

$$S_2 \equiv GS_{1r}(\alpha^{S_1} + h)r^{-1} \equiv GS_{1r}(V_1^{S_1} + h)r_F^{-1} \pmod{G_{Rg}}$$

Since $G_{Rg} \equiv 0 \pmod{GS_{1r}}$, it implies

$$(\alpha^{S_1} + h)r^{-1} \equiv (V_1^{S_1} + h)r_F^{-1} \pmod{G_{Rg2}}$$

where $G_{Rg2} = G_{Rg}GS_{1r}^{-1}$. Moving forward we have,

$$r_F\alpha^{S_1} - (V_1^{S_1} + h)r + hr_F \equiv 0 \pmod{G_{Rg2}} \quad (4)$$

CONTINUE…

THE DESIGN RATIONALE

Let $\hat{\alpha}$ be the upper bound for α^{S_1} and $\hat{r}$ be the upper bound for r. From Conjecture 1, if one has the situation where $\hat{\alpha}\hat{r} \gg G_{Rg2}$, then there is no efficient algorithm to output all the roots of (4). That is, (4) usually has G_{Rg2} many solutions, which is exponential in the bit-size of G_{Rg2}.

To this end, since α^{S_1} is exponentially large, it is clear to conclude that $\hat{\alpha}\hat{r} \gg G_{Rg2}$. This implies, there is no efficient algorithm to output all the roots of (4).

THE DESIGN RATIONALE

Another "Expensive" Problem Related To KAZ-SIGN: The Second Order Discrete Logarithm Problem (2-DLP)

Let N be a composite number, g a random prime in $\mathbb{Z}_N$ of order G_g where at most $G_g \approx N^{\delta}$ for $\delta \in (0,1)$ and $\delta \to 0$. That is, $g^{Gg} \equiv 1 \pmod{N}$. Choose a random prime $Q \in \mathbb{Z}_{\phi(N)}$ of order G_Q, where $G_Q \approx \phi(N)^{\varepsilon}$ for $\varepsilon \to 1$. That is, choose Q with a large order in $Z_{\phi(N)}$. Such Q, has it own natural order in $Z_{\phi(G_g)}$. Let that order be denoted as G_{Qg}. We can observe the natural relation given by $Q^{G_{Qg}} \equiv 1 \pmod{G_g}$ and $\phi(N) \equiv 0 \pmod{G_g}$.

Then choose a random integer $x \in \mathbb{Z}_{\phi(G_g)}$ where $x \approx \phi(G_g)$. Suppose from the relation given by

$$g^{Q^x \pmod{\phi(N)}} \equiv A \pmod{N} \tag{5}$$

one has solved the Discrete Logarithm Problem (DLP) upon equation (5) in polynomial time on a classical computer and obtained the value X where $Q^x \not\equiv X \pmod{\phi(N)}$ and $g^X \equiv A \pmod{N}$, The relation $Q^x \not\equiv X \pmod{\phi(N)}$ would result in the non-existence of the discrete logarithm solution for $Q^x \equiv X \pmod{\phi(N)}$.

CONTINUE...

THE DESIGN RATIONALE

The 2-DLP is, upon given the values (A, g, N, Q), one is tasked to determine $x \in \mathbb{Z}_{\phi(G_g)}$ where $x \approx \phi(G_g)$ such that equation (5) holds.

Let $Q^x \equiv T_1 \pmod{\phi(N)}$. From the predetermined order of $g \in \mathbb{Z}_N$, during the process of solving the DLP upon equation (5), a collision would occur prior to the full cycle of g. As such, the process of solving the DLP upon equation (5) to obtain $X \approx N^{\delta}$ would occur in polynomial time on a classical computer. And since $T_1 < \phi(N)$ and $T_1 \approx N_1$, the relation $Q^x \not\equiv X \pmod{\phi(N)}$ will hold.

CONTINUE…

THE DESIGN RATIONALE

Furthering on the discussion, one has the relation $g^{G_g} \equiv 1 \pmod{N}$. As such, from the value $X < G_g$ obtained from equation (5), one can construct the set of solutions given by $T_0 = X + G_g t$ for $t = 0, 1, 2, 3, \ldots$. Now let $Q^x \equiv T_1 \pmod{\phi(N)}$. Following through,since T_1 is an element from the set of solutions, one can have the relation

$$t_{T_1} = \frac{T_1 - X}{G_g}$$

Since $G_g, X \approx N^{\delta}$, and $\phi(N) \approx N$, the complexity to obtain t_{T_1} is $O(N^{1-\delta})$. When deploying Grover's algorithm on a quantum computer, the complexity to obtain t_{T_1} is $O(N^{\frac{1-\delta}{2}})$.

CONTINUE…

THE DESIGN RATIONALE

To this end, note that if one proceeds to solve the DLP upon $Q^x \equiv X \pmod{G_g}$, one can obtain the value $x_0 \equiv x \pmod{G_{Qg}}$. From the preceding sections, this is in fact the MRP. It is easy to see that with correct choice of parameters (x, G_{Qg}), the complexity of 2-DLP and MRP can be made the same. Hence, a more "non-expensive" method in discussing the needs of the KAZ-SIGN is directly via the MRP.

PARAMETER SIZES

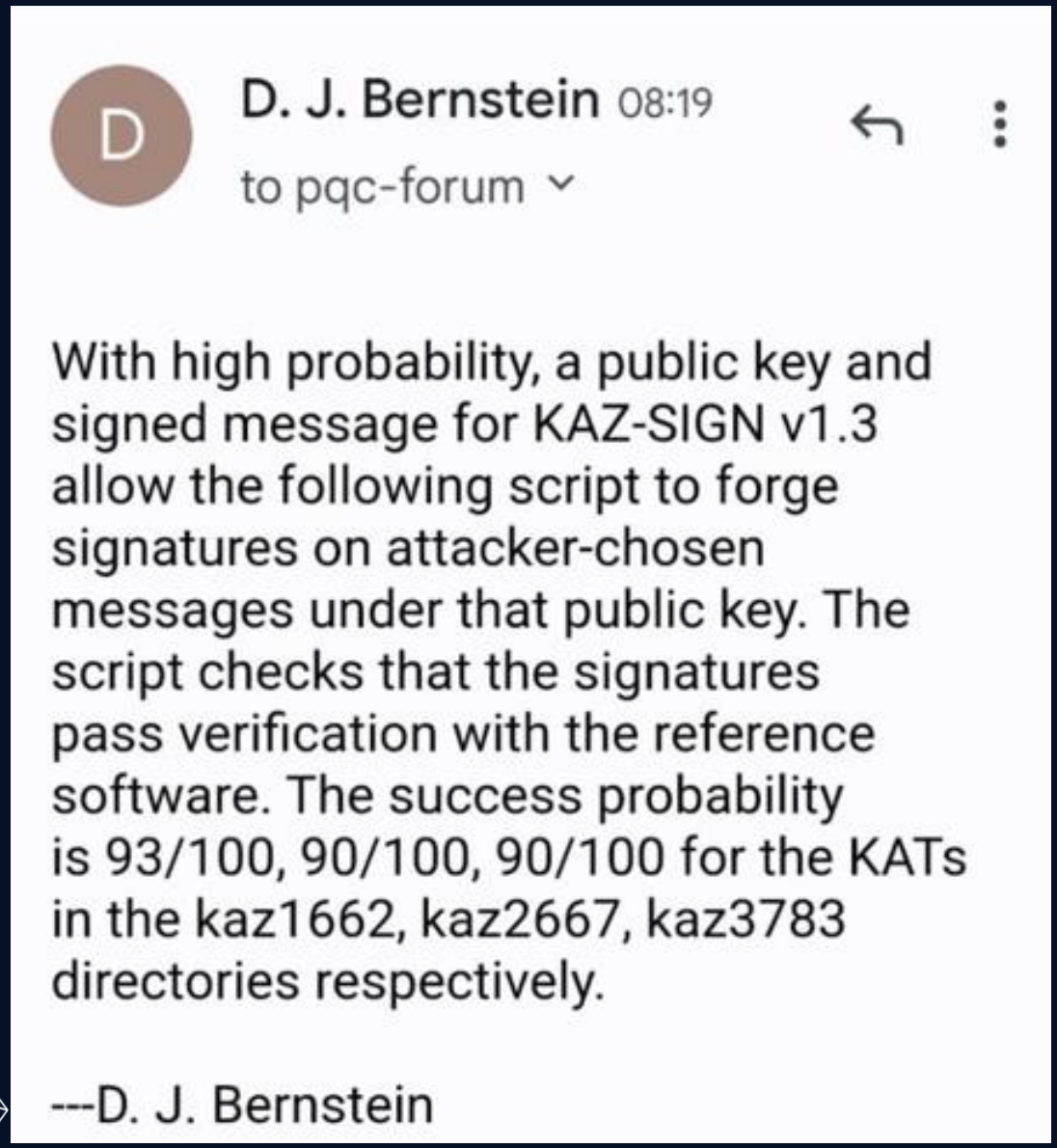
2nd Oxford Post-Quantum Cryptography Summit 2023

We provide here information on size of the key and signature based on security level information from Table 1 (for $\delta = 0.23$) where $\ell(V_2)$ is the length of an output generated by a 256-bit hash function.

NIST Security Level	Number of primes in P	Security level, k	Length of parameter N (bits)	Public key size, (V_1, V_2) (bits)	Private key size, α (bits)	Signature Size (S_1, S_2) (bits)	ECC key size (bits)
1	195	128	1662	$\approx 218 + 256 = 474$	≈ 384	≈ 700	256
3	290	192	2667	$\approx 332 + 256 = 588$	≈ 576	≈ 1046	384
5	390	256	3783	$\approx 450 + 256 = 706$	≈ 768	≈ 1409	521

BERNSTEIN'S COMMENTS ON KAZ-SIGN v1.3

2nd Oxford
Post-Quantum Cryptography
Summit 2023



D. J. Bernstein 08:19

to pqc-forum

With high probability, a public key and signed message for KAZ-SIGN v1.3 allow the following script to forge signatures on attacker-chosen messages under that public key. The script checks that the signatures pass verification with the reference software. The success probability is 93/100, 90/100, 90/100 for the KATs in the kaz1662, kaz2667, kaz3783 directories respectively.

---D. J. Bernstein

2nd Oxford
Post-Quantum
Cryptography
Summit 2023

WAY FORWARD

1) To identify the algebraic structures of the10% KAZ-SIGN valid signature parameters (S_1 , S_2) that could not be utilized to forge signatures under attacker chosen-message for the corresponding public verification key set.

2) To utilize these algebraic structures upon all KAZ-SIGN valid signature parameters (S_1 , S_2) to be generated – with minimum amendment on KAZ-SIGN v1.3. That is, to aspire to only include additional “filtering” procedures during signing and verification and no changes upon keygen and no additional parameters to be introduced (no new keys and signature parameters) and not to increase the key and signature sizes.

3) To make public the findings via publication of KAZ-SIGN v1.4.

ACKNOWLEDGEMENTS

2nd Oxford Post-Quantum Cryptography Summit 2023

1) KAZ Team would like to thank Prof. Abderrahmane Nitaj of Laboratoire de Mathematiques Nicolas Oresme, Universite de Caen Basse Normandie,France for discussions and inputs.

2) KAZ Team would like to thank Prof. D.J. Bernstein of University of Illinois at Chicago, US for inputs that lead to the development of KAZ SIGN till today.

3) KAZ Team would like to thank NIST Teamfor facilitating KAZ SIGN initial submission.

4) KAZ Team would like to thank MI, Oxford (UK) and PQ Shield for organizing this event.

2nd Oxford
Post-Quantum Cryptography
Summit 2023

THANK YOU

TERIMA KASIH