AMSI SUMMER RESEARCH SCHOLARSHIPS 2023–24

SET YOUR SIGHTS ON RESEARCH THIS SUMMER

Post-Quantum Cryptography

Jason Liu

Supervised by Professor Nalini Joshi

University of Sydney

AMSI AUSTRALIAN MATHEMATICAL SCIENCES INSTITUTE

Abstract

Classical cryptographic protocols such as Diffie-Hellman key exchange are expected to become unsafe with the advent of quantum computers, which provide efficient polynomial time attacks using Shor's algorithm. We examine a variant of Diffie-Hellman key exchange based on translation operations in the symmetry group of the fourth Painlevé equation, which may offer exponential time defence against Shor's algorithm.

Contents

1 Introduction

Cryptography is the study of methods of communicating secrets in the presence of a malicious third party whose aim is to learn the secret. Typically, we call the two parties who wish to exchange a secret Alice and Bob, who are communicating over an insecure channel that the attacker has access to. Cryptographic protocols are methods that Alice and Bob can use to communicate their secret without the attacker knowing.

A simple example of a cryptographic protocol is the Cæsar cipher, named after its supposed creator Julius Cæsar, which involves shifting the alphabet and replacing each letter by the letter it is shifted. For instance, a shift of 1 means we encrypt a piece of text, called the plaintext, by writing B in place of each A, C in place of each B, etc. The shifted text, called the ciphertext, can be sent to the other party, who decrypts it by writing A in place of each B, B in place of each C, etc. Since the same key is used to encrypt the plaintext and decrypt the ciphertext, we call such a protocol a symmetric key protocol.

For this type of protocols to work, both Alice and Bob need to know the shift beforehand, which is called the private key, as it is unknown to the attacker. This poses a problem for efficient communication, which required prior communication or the use of codebooks historically. That was until the 1970s, with the advent of computers, when Diffie and Hellman discovered a key exchange protocol that allows users to exchange a secret key over an insecure channel [1]. It relies on performing an operation that can be quickly performed on a computer, but also difficult to reverse for the attacker on a computer. The Diffie-Hellman key exchange remains one of the most commonly used cryptographic protocols in our computers today still.

While computers brought about the first revolution in cryptography, we expect to see a second revolution with the development of quantum computing. By taking advantage of special properties of quantum mechanics, quantum computers are capable of performing certain computations and algorithms at a speed magnitudes faster than classical computers. Of particular importance is Shor's algorithm, discovered in the 1990s, which provides an efficient attack against Diffie-Hellman key exchange, among other classical protocols [6]. While practical quantum computers are still far away, there are stolen data out there that are only classically-protected and are in danger when quantum computers do arrive, so it is of particular significance to investigate and design cryptographic protocols that are quantum-safe as soon as possible.

In this paper, we investigate a mathematical structure that may enable the design of a quantum-safe protocol — the fourth Painlevé equation under translation in its symmetry group. Painlevé equations are a class of differential equations with symmetry groups. In particular, the transformations of the fourth Painlevé equation forms a group and have an associated geometric interpretation. We examine whether adapting Diffie-Hellman key exchange to the context of the fourth Painlevé equation produces an efficient and secure cryptographic protocol.

1.1 Statement of Authorship

The description of Shor's algorithm in Section 3.2 is adapted from Shor [6]. The expository information on Painlevé equations in Section 4 is taken from Joshi [3] and Kajiwara, Noumi, and Yamada [4]. Discussions on the implementation of transformations on Painlevé equations as a cryptographic protocol in Section 5 is our own work.

2 Diffie-Hellman Key Exchange

We give an implementation of Diffie-Hellman key exchange based on the multiplicative group of integers modulo a prime, but in general any group would work (to varying levels of security and efficiency).

Diffie-Hellman key exchange

Shared: prime p and $g \in \mathbb{F}_p$

Alice		**Bob**
$n \in \mathbb{F}_p$		$m \in \mathbb{F}_p$
$N \leftarrow g^n \pmod{p}$		$M \leftarrow g^m \pmod{p}$
	$\xrightarrow{N}$	
	$\xleftarrow{M}$	
$k_a \leftarrow M^n \pmod{p}$		$k_b \leftarrow N^m \pmod{p}$

In this case, we call the shared p, g the public key and the private n, m the private keys. It is straightforward to see that $k_a \equiv g^{mn} \pmod{p} \equiv k_b$, so Alice and Bob have a shared secret. The key exchange is efficient for Alice and Bob since there are efficient algorithms for modular exponentiation, such as repeated squaring. The only information known to the attacker are p, g, N, M, and in order for the attacker to computer the shared secret g^{mn}, they must know n and m. The difficulty of solving for n, m given p, g, N, M is known as the discrete log problem. The best known classical algorithms give only exponential time attack. However, quantum algorithms offer polynomial time attack, which we discuss below.

3 Quantum Computation and Algorithms

3.1 Fundamentals of Quantum Computation

Definition 3.1 (Qubit)**.** A qubit is a basic unit of quantum information whose state is a superposition of the basis states. We write a qubit state as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $|0\rangle, |1\rangle$ are the basis states and $\alpha, \beta \in \mathbb{C}$ are the probability amplitudes, with $|\alpha|^2 + |\beta|^2 = 1$.

The qubit is the quantum analogue of the classical bit with two states 0 and 1, with corresponds to the basis states $|0\rangle$ and $|1\rangle$. We can specify each qubit with a complex number β/α. Single qubits can combine to make multiple-qubit states.

Definition 3.2 (Tensor product of qubits)**.** A tensor product is a binary operator $\otimes$ that satisfies:

1. $(|v_1\rangle + |v_2\rangle) \otimes |w\rangle = |v_1\rangle \otimes |w\rangle + |v_2\rangle \otimes |w\rangle$
2. $|v\rangle \otimes (|w_1\rangle + |w_2\rangle) = |v\rangle \otimes |w_1\rangle + |v\rangle \otimes |w_2\rangle$
3. $(\alpha|v\rangle) \otimes |w\rangle = |v\rangle \otimes (\alpha|w\rangle) = a(|v\rangle \otimes |w\rangle)$

We write the tensor product of two qubits, for example $|1\rangle \otimes |1\rangle$, as $|11\rangle$, or in its base-10 notation $|3\rangle$.

Typically, we work with multiple-qubit systems, so $|0\rangle$ and $|1\rangle$ always refer to $|0\cdots 00\rangle$ and $|0\cdots 01\rangle$ in these contexts.

Definition 3.3 (Quantum register)**.** A system of n qubits is called a quantum register. We write $\{|0\rangle, |1\rangle, \ldots, |2^n - 1\rangle\}$ for the 2^n basis states, and a n-qubit state is denoted $|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |k_i\rangle$, where $0 \leq k_i < 2^n$ and $\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$. We write a system of multiple registers in the form $|\psi_1, \psi_2, \ldots\rangle$.

Notice that we can specify a tensor product of n qubits using n complex numbers, however a n-qubit state can only be specified using $2^n - 1$ complex number.

Definition 3.4 (Quantum entanglement)**.** A multiple-qubit state is entangled if it cannot be written as a tensor product of single qubits.

This is one of the most important properties of qubits that distinguishes it from classical bits. It has the important consequence that measuring one qubit in an entangled quantum state has the effect of collapsing the superposition of other qubits in the state.

Example 3.5. Let $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Whenever the first qubit measures $|0\rangle$, the second qubit must also measure $|0\rangle$, since the only state in the superposition with first qubit $|0\rangle$ also has second qubit $|0\rangle$. Similarly, whenever the first qubit measures $|1\rangle$, the second qubit must also measure $|1\rangle$. Therefore $|\psi\rangle$ is in an entangled state.

We now look at some operations on qubits.

Lemma 3.6 (Quantum measurement)**.** Let $|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |k_i\rangle$ be a n-qubit state. When $|\psi\rangle$ is measured, it takes each outcome $|k_i\rangle$ with probability $|\alpha_i|^2$.

This defines one of the key operations on quantum registers. We cannot know about the state of a quantum register without measuring it, which also collapses its superposition, so this is an irreversible operation.

Definition 3.7 (Quantum Fourier transform)**.** Let $|a\rangle$ be a quantum register and $q \in \mathbb{N}$. The quantum Fourier transform (QFT) on $|a\rangle$ is the following transformation on the register:

$$|a\rangle \mapsto \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} \exp(2\pi i ac/q) |c\rangle$$

This transformation is key to many quantum algorithms. We can verify that it preserves the square of the modulus of the probability magnitudes summing to 1. This is an analogue of the discrete Fourier transform, which helps describe the frequencies encoded in a function. The significance of the QFT will become apparent with its role in Shor's algorithm, and it is also commonly used in other quantum algorithms.

3.2 Shor's Algorithm

We outline a simplified implementation of Shor's algorithm for the discrete log problem [6].

Shor's algorithm

Let $G = \langle g \rangle$, and $g^r = x$, where r is unknown.

1. Start the quantum computer with the state:

$$|0, 0, 0\rangle$$

2. Put the first two registers in the uniform superposition of $|a\rangle, |b\rangle$ with $0 \leq a, b \leq |G|$:

$$\mapsto \frac{1}{|G|} \sum_{a,b=0}^{|G|-1} |a, b, 0\rangle$$

3. Compute $f(a, b) = g^a x^{-b}$ and put it in the third register:

$$\mapsto \frac{1}{|G|} \sum_{a,b=0}^{|G|-1} |a, b, g^a x^{-b}\rangle$$

   This operation entangles the three registers. Note that if $f(a, b)$ is periodic in b, that is $f(a, b) = f(a, b + c)$, then $f(a, b)$ is also periodic in a with a period of cr.

4. Measure the third register. Suppose we get $g^a x^{-b} = g^k$, since the registers are entangled, the state collapses to:

$$\mapsto \frac{1}{\sqrt{|G|}} \sum_{b=0}^{|G|-1} |br + k, b\rangle$$

   where we can discard the third register.

5. Apply the quantum Fourier transform to the two registers:

$$\begin{aligned}&\mapsto \frac{1}{|G|^{\frac{3}{2}}} \sum_{b,c,d=0}^{|G|-1} \exp\left(\frac{2\pi i}{|G|}\left((br+k)c+bd\right)\right)|c,d\rangle\\ &=\frac{1}{|G|^{\frac{3}{2}}} \sum_{b,c,d=0}^{|G|-1} \exp\left(\frac{2\pi i}{|G|}ck\right)\exp\left(\frac{2\pi i}{|G|}b(cr+d)\right)|c,d\rangle\end{aligned}$$

   Note that whenever $cr+d\neq 0$, we have $\sum_{b=0}^{|G|-1}\exp\left(\frac{2\pi i}{|G|}b(cr+d)\right)=0$, as this sum is over the roots of unity. Therefore:

$$=\frac{1}{\sqrt{|G|}}\sum_{d=0}^{|G|-1}\exp\left(\frac{2\pi i}{|G|}ck\right)|c,-cr\rangle$$

6. Measure the state, with equal probability we get $(c,-cr)$ for some c. If c has a multiplicative inverse, divide the second register by $-c$ to obtain r, else repeat from step 1.

In essence, Shor's algorithm finds the discrete log by computing a function using the two given group elements which is doubly periodic with the periods differing by a factor of the discrete log. Using frequency-finding properties of the QFT, the algorithm deduces these two periods and hence the discrete log. The quantum aspect of this algorithm accelerates computation of the function and the QFT through a property called *quantum parallelism*, which essentially is the ability to apply a transformation simultaneously to all basis states in the superposition. In a classical setting, these computations have be done separately for each possible combination.

When it comes to implementing Shor's algorithm on an actual quantumc omputer, efficient QFTs are only known for groups of *smooth* order (i.e. order N where all prime factors of N are less than $\log N$), therefore we need to take the QFT from 0 to some $|G|\leq 2^q\leq 2|G|$. This reduces the probability of obtaining the desired state in step 6 since $\sum_{b=0}^{2^q-1}\exp\left(\frac{2\pi i}{|G|}b(cr+d)\right)$ no longer vanishes, but nevertheless requires only polynomial time to guarantee arbitrarily high certainty of obtaining r.

3.3 Limitations of Shor's Algorithm

The discrete log problem is an instance of the hidden subgroup problem, which involves finding a generating set for a subgroup given a function whose values are constant on cosets of that subgroup. In fact, Shor's algorithm provides a general solution for in polynomial time when the group in question is abelian. The precise reasoning why Shor's algorithm fails with non-abelian groups requires representation theory, so we provide a simplified explanation below.

When we measure the final state, we obtain the kernel of a *character*, a function that takes elements of a group to the traces of its representations. With abelian groups, there is a one-to-one correspondence between its elements and its characters, so by repeatedly finding these kernels we obtain a description of the subgroup. However, with non-abelian groups, there is no longer a one-to-one correspondence between its elements and its characters, so on repetition we might keep finding kernels of characters that correspond to the same element, thereby reducing the probability of obtaining the desired state and pushing the algorithm into exponential time.

An example of a protocol that was designed to be quantum-safe is the supersingular isogeny key exchange (SIKE) protocol [2]. It is similar to the Diffie-Hellman key exchange, but involves the exchange of *isogenous* elliptic curves between Alice and Bob, which are elliptic curves related by group homomorphisms. The protocol is Shor-proof as isogenies of an elliptic curve form a non-abelian group. However, a polynomial time classical attack was discovered against SIKE using auxiliary points on the elliptic curves that must be exchanged during the protocol. The example of the SIKE protocol illustrates the difficulty of balancing efficiency and security in the design of a viable cryptographic protocol.

4 Painlevé Equations and Bäcklund Transformations

4.1 Painlevé Equations

We investigate the potential of *Painlevé equations* and their transformations as structures for cryptographic protocols. Painlevé equations are six families of differential equations whose only movable singularities are poles whose solutions are special transcendental functions, just like elliptic function. In particular, we consider the fourth Painlevé equation.

Definition 4.1 (Fourth Painlevé equation)**.** The fourth Painlevé equation P_{IV} is given by:

$$w'' = \frac{w'^2}{2w} + \frac{3w^3}{2} + 4tw^2 + 2(t^2 - \alpha)w + \frac{\beta}{w}$$

where t is the dependent variable, α, β are parameters, and $'$ denotes differentiation with respect to t,

P_{IV} also has a symmetric form.

Theorem 4.2 (Symmetric form of P_{IV})**.** P_{IV} can be written in the following form:

$$\begin{cases} f_0' = f_0(f_1 - f_2) + \alpha_0 \\ f_1' = f_1(f_2 - f_0) + \alpha_1 \\ f_2' = f_2(f_0 - f_1) + \alpha_2 \end{cases} \qquad f_0 + f_1 + f_2 = t$$

where $\alpha_0, \alpha_1, \alpha_2$ are parameters that sum to some constant c (see [4] for proof).

4.2 Bäcklund Transformations

P_{IV} has *Bäcklund transformations* that commute with differentiation.

Definition 4.3 (Bäcklund Transformation)**.** Let u, t be functions in t and let u_t, v_t, u_{tt}, v_{tt} denote their derivatives with respect to an independent variable t. Suppose they satisfy the following system of equations:

$$\begin{cases} F(u, u_t, \ldots, v, v_t, \ldots) = 0 \\ G(u, u_t, \ldots, v, v_t, \ldots) = 0 \end{cases} \tag{4.1}$$

If by eliminating v we obtain $R(u, u_t, \ldots) = 0$ and by eliminating u we obtain $S(v, v_t, \ldots) = 0$, such that R and S belong to the same family of differential equations, then (4.1) is called a Bäcklund transformation between $R = 0$ and $S = 0$.

Essentially, a Bäcklund transformation takes a differential equation to another differential equation of the same family, and it also relates solutions of these equations. Bäcklund transformations can be interpreted geometrically if we look at their action on $\alpha_0, \alpha_1, \alpha_2$ in the symmetric form. We can scale $\alpha_0, \alpha_1, \alpha_2$ such that $\alpha_0 + \alpha_1 + \alpha_2 = 1$ and consider the triangular lattice with axes $\alpha_0, \alpha_1, \alpha_2$. A point on this lattice has coordinates $(\alpha_0, \alpha_1, \alpha_2)$.

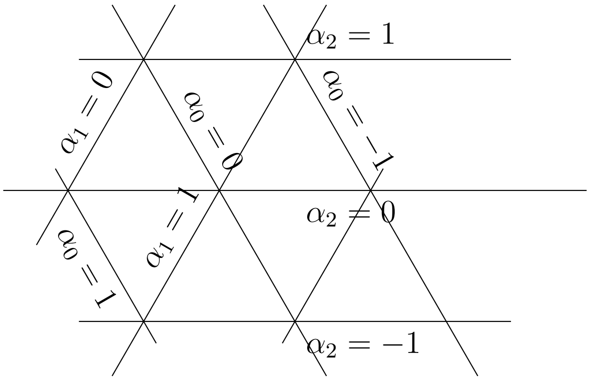

Figure 1: Triangular lattice with axes $\alpha_0, \alpha_1, \alpha_2$

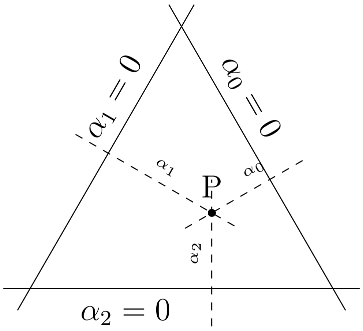

Figure 2: The orthogonal projection from the point to the sides of the triangle with $\alpha_0, \alpha_1, \alpha_2 = 0$ defines its coordinates.

We can define certain operations on this lattice, which, in fact, correspond to Bäcklund transformations on the symmetric form of P_{IV}.

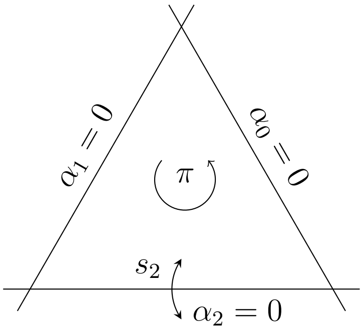

Figure 3: Reflection s_2 across $\alpha_2 = 0$ and rotation π on the lattice

	α_0	α_1	α_2
s_0	$-\alpha_0$	$\alpha_0 + \alpha_1$	$\alpha_0 + \alpha_2$
s_1	$\alpha_0 + \alpha_1$	$-\alpha_1$	$\alpha_1 + \alpha_2$
s_2	$\alpha_0 + \alpha_2$	$\alpha_1 + \alpha_2$	$-\alpha_2$
π	α_1	α_2	α_0

Table 1: Action of each operation on the coordinates

Together, they generate the extended affine Weyl group of type $A_2^{(1)}$, which describes Bäcklund transformations on P_{IV}. Simple computation verifies that this group is indeed non-abelian, which is the desired property against Shor's algorithm.

We can define a translation operation $T := \pi s_2 s_1$, which has the action of translating a point one unit to the right along the α_2 axis, and denote $w_n := T^n(w)$ where w is a solution of some equation in the P_{IV} family. Then we can derive a difference equation on w_n (see [4] for detailed derivation):

$$w_{n+1} + w_n + w_{n-1} = -2t + \frac{n + c_0 + c_1(-1)^n}{w_n}$$

where c_0, c_1 are arbitrary constants. This is known as the first discrete Painlevé equation, which provides a blueprint for a cryptographic protocol.

5 Cryptographic Implementation

5.1 Sketch of Protocol

There are some adjustments that must be made to the difference equation for it to be computationally viable. We can set $t = 0$ for simplicity (other constants would also work) so that the difference equation becomes an equation involving scalars, which allows us to work over a finite field. Also, the presence of the n term is undesirable, since we'd like Alice and Bob to be able to compute iterations on w_n without needing to know the precise "step" in the key exchange. In fact, by scaling $\alpha_0, \alpha_1, \alpha_2 = c$, we can introduce an arbitrary constant in front n, which we can set to 0. These give the difference equation:

$$w_{n+1} = -w_n - w_{n-1} + \frac{c_0 + c_1(-1)^n}{w_n}$$

where w_n are now scalars. We can now outline a model key exchange protocol using the difference equation, where T now represents this iteration.

Translation-based key exchange

Shared: prime p and $w_0, w_1, c_0, c_1 \in \mathbb{F}_p$

Alice		**Bob**
$n \in \mathbb{F}_p$		$m \in \mathbb{F}_p$
$N_1, N_2 \leftarrow w_{n-1}, w_n \pmod p$		$M_1, M_2 \leftarrow w_{m-1}, w_m \pmod p$
	$\xrightarrow{N_1, N_2, n \pmod 2}$	
	$\xleftarrow{M_1, M_2, m \pmod 2}$	
$k_a \leftarrow T^n(M_1, M_2) \pmod p$		$k_b \leftarrow T^m(N_1, N_2) \pmod p$

Indeed, $k_a = w_{n+m} = k_b$, so Alice and Bob have a shared secret. Note the presence of a modular inverse in the difference equation means that we cannot iterate further if we obtain $w_n = 0$ at any step. In practice, there is usually a significant difference in the key size required for the public key p and private keys n, m. For example, in the classical Diffie-Hellman setting, the recommended secure key size is 2048-bits for the public key and 256-bits for the private keys. This means that the probability of obtaining 0 at any step is negligible, and if it does happen, Alice and Bob can always repeat the protocol.

5.2 Practical Considerations

As it stands, there are computational difficulties when it comes to actual implementation of this protocol. In theory, since each iteration involves one multiplication which is polynomial time, and there are $n+m$ iterations required, the overall time complexity is exponential in the size of the private keys. We simulated the key exchange protocol in Magma (see appendix A):

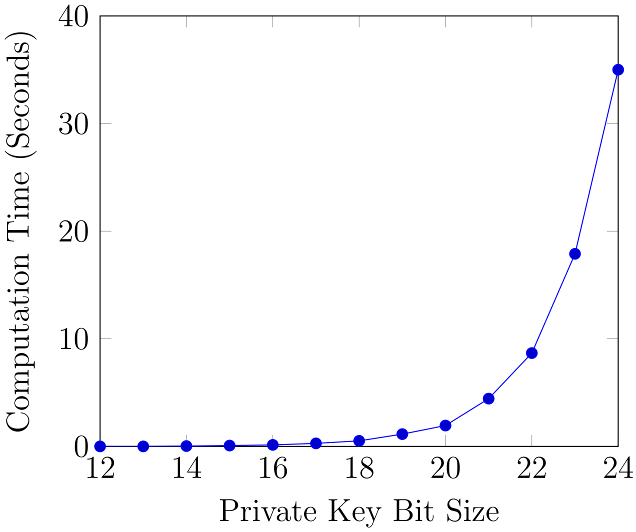

Figure 4: Time taken to compute $k_a = k_b$ for increasing private key bit sizes, given a public key size of 128 bits and random $w_0, w_1, c_0, c_1 \in \mathbb{F}_{2^{128}}$

The chart confirms the exponential time complexity, as the computation time doubles with every additional bit. This is to be expected with the present equation we have derived that can only iterate one step at a time, and an analogue of repeated squaring in the modular exponentiation-based key exchange is required. An operation of this kind on the extended affine Weyl group is not known currently.

6 Conclusion

In this report, we have discussed a new candidate group structure for constructing cryptographic protocols: the extended affine Weyl group that represents Bäcklund transformations on the fourth Painlevé equation. In particular, translation operations in the group give rise to a difference equation on solutions to the fourth Painlevé equation that can be used as a formula for key exchange. However, some kind of doubling formula on translations is required for efficient computation on the users' side, which presents a major barrier for the implementation of the protocol at present.

Future research should prioritise discovering this doubling formula, but other aspects of the protocol's safety should also be investigated. For one, there exist linear attacks on protocols based on non-abelian groups with small nontrivial representations [5], so future work can involve bringing in representation theory and investigating the feasibility of such an attack on extended affine Weyl groups.

7 Acknowledgements

I would like to thank my supervisor Professor Nalini Joshi for her guidance and support throughout the project. I am lucky to have her guide my first steps into mathematical research.

A Magma Code for Estimating Time Complexity

```
FindSecret:=function(w0,w1,c0,c1,p,n)
        F:=FiniteField(p);
        w0:=F ! w0;
        w1:=F ! w1;
        c0:=F ! c0;
        c1:=F ! c1;
        i:=2;
        while i ne n do
                if IsDivisibleBy(i,2) then
                        w2:=-w0-w1+(c0+c1)/w0;
                else
                        w2:=-w0-w1+(c0-c1)/w0;
                end if;
                w0:=w1;
                w1:=w2;
                i:=i+1;
        end while;
        return w1;
end function;

KeySize:=128;
PubKey:=PreviousPrime(2^KeySize);
w0:=RandomBits(KeySize);
w1:=RandomBits(KeySize);
c0:=RandomBits(KeySize);
c1:=RandomBits(KeySize);

for i in [8..24] do
        PriKey:=2^i;
        t:=Cputime();
        FindSecret(w0,w1,c0,c1,PubKey,PriKey);
        PrintFile(”time.txt”,Cputime(t));
end for;
```

References

[1] W. Diffie and M. Hellman. "New directions in cryptography". In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654. DOI: 10.1109/TIT.1976.1055638.

[2] David Jao and Luca De Feo. "Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies". In: *Post-Quantum Cryptography*. Ed. by Bo-Yin Yang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 19–34. ISBN: 978-3-642-25405-5.

[3] Nalini Joshi. "Transformations and Reflection Symmetries". In: *Lecture Notes for Special Topics in Applied Mathematics*. 2023.

[4] Kenji Kajiwara, Masatoshi Noumi, and Yasuhiko Yamada. "Geometric aspects of Painlevé equations". In: *Journal of Physics A: Mathematical and Theoretical* 50.7 (Jan. 2017), p. 073001. ISSN: 1751-8121. DOI: 10.1088/1751-8121/50/7/073001. URL: http://dx.doi.org/10.1088/1751-8121/50/7/073001.

[5] Vitaliĭ Roman'kov and Alexei Myasnikov. *A linear decomposition attack*. 2014. arXiv: 1412.6401 [math.GR].

[6] P.W. Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.