Design and implementation of ECC combined with OPT encryption algorithm

Xiaotian Yang[a], Ran Ma[b], Fei Gao*

Tibet University School, Tibet Lhasa
[a]778920383@qq.com; [b] 824041393@qq.com;

*E-mail:zdkx@utibet.edu.cn

Abstract. 【Objective】 Combining ECC and OPT, a novel elliptic curve algorithm was proposed, which can effectively resist the reverse calculation of quantum computer and ensure the security of private key. 【Method】 On the basis of elliptic curve encryption algorithm, OPT algorithm was introduced to protect the private key. 【Result】 The signature and verification of messages can be successfully completed by the repeated algorithm on Python platform. Finally, the algorithm is proved to be safe under Shor algorithm and quantum computer attack through algorithm theory. 【Conclusion】 ECC combined with OPT encryption algorithm can effectively resist reversible computation of functional quantum devices, prevent eavesdroppers from forging signatures, and ensure the security of modern public key cryptosystems.

1. Introduction

The development of communication and encryption technology has always been the focus of the development of human society. Nowadays, encryption technology is everywhere, such as: daily life, the network world, encryption plays an indispensable role in our world.But there is encryption and decryption, in the protection of privacy at the same time, there are also people covet the privacy of others.The competition between encryption technology and eavesdropping technology can be best described as "The priest climbs a foot, the devil climbs ten.".RSA, a traditional encryption algorithm, is widely used in online payment, communication, email and other fields due to its simplicity and security.But what makes RSA encryption "ever" powerful and secure is that RSA is based on large number factorization or the difficulty of solving discrete logarithms[1].For example, it is easy to multiply two prime numbers, but difficult to find the factors of a large number.Therefore, RSA and other encryption algorithms based on public keys are generally considered to be secure.However, with quantum computing, these complex mathematical problems proved to be easily solved in polynomial time.Functional quantum devices can perform reversible calculations and forge signatures very quickly, and when practical quantum computers become available to eavesdroppers, it will break the security of almost all modern public-key cryptosystems[2][3].

Quantum computers were proposed in 1980 by Feynman[4]In 2016, IBM launched the 5 qubit superconducting quantum platform[5],The international research and development of quantum computer has entered a breakthrough stage.At present, quantum computers are developing rapidly and attracting worldwide attention. Many countries take the research and development of quantum computers as their national development strategy.Although the development of quantum computer will bring new breakthroughs in science, medicine, finance and other aspects in the future, the arrival of

quantum computer also brings unprecedented challenges to traditional cryptography,In October 2019, Google officially published in Nature: Quantum computers in scalable ways[7],The Shor's algorithm is implemented.Shor's algorithm[6]It was proposed in 1994 by mathematician Peter Shor,It's a quantum algorithm for the decomposition of large numbers,The widely used public key encryption algorithm can be broken with quantum computer.And a study by Craig Gidney of Google and Martin Ekera of KTH Royal Institute of Technology in Stockholm, Sweden, showed that it was possible to calculate 2048-bit RSA integers in 8 hours using 20 million noisy qubits[8],This study shows that Shor algorithm has become a reality, indicating that the development of quantum computers will have a huge impact on the whole Internet privacy and data security.

OPT[9][10] encryption algorithm has the characteristics of random key generation, one-time encryption and high efficiency of encryption and decryption operation, which can effectively resist the attack of quantum computer. However, it also has a serious defect, that is, the key is required to be as long as the plaintext, and it is difficult to share the key.In combination with ECC algorithm and OPT algorithm both reference random number to ensure algorithm security, we use OPT algorithm and 256-bit random number encryption to protect private key.The main contributions of this paper are as follows:

- The ECC algorithm is improved by introducing OPT algorithm and combining the advantages of the two algorithms to achieve the coexistence of security and efficiency.It can effectively resist the new encryption algorithm of quantum computer and ensure that the private key is not threatened by quantum computer.
- We use the existing scheme to implement the algorithm in Python platform, build a simple key signature scheme, and verify the feasibility of the scheme through experiments.
- The security of the algorithm is verified by theoretical analysis.

2. ECC combined with OPT elliptic curve encryption algorithm

2.1. The relevant knowledge

Finite field: finite field is the basis of elliptic encryption curve, finite field is the set of finite number of elements, the set is defined as Fp.Meet the following properties;Contains two operations (point addition operation and dot product operation) four propertiesClosure, that is, if a and b are both members of set Fp, then a+ b is also members of set Fp;The identity of addition, that is, the existence of 0 is a+0=a;Multiplicative identity, that is, if there is 1, you get a dot 1 is equal to a;The additive inverse, a is a member of the set Fp, minus a is a member of the set Fp, a plus minus a is equal to 0;So if a is a member of set Fp then a minus 1 is also a member of set Fp.Fp means that there are only P elements and p is prime.The element set is {0,1,2..., p-1}, element addition such that a+b= c (modp).

Elliptic curve:likey**2=x**3+a*x+b .Figure 1 shows two common elliptic curves.

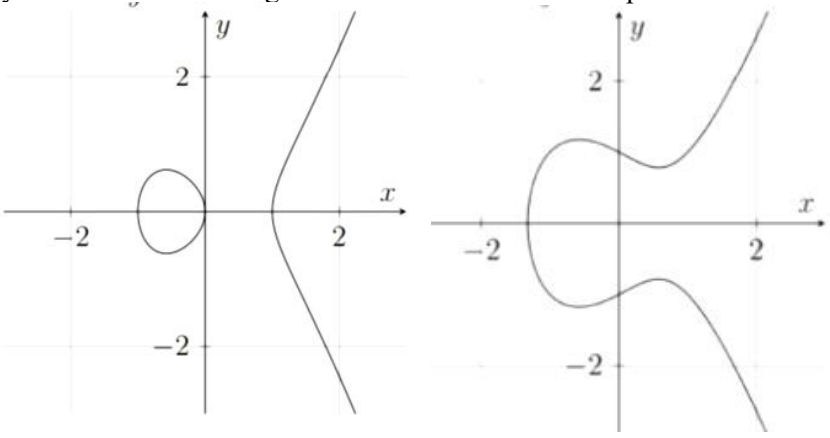

Figure 1. Elliptic curve

2.2. The relevant knowledge

Elliptic curve is widely used because of its point addition operation, point addition: elliptic curve takes any two points P, Q.P=（x1，x2）Q=(x2，y2) So P plus Q is the line that goes through P and Q,Find its third intersection with the elliptic curve, R, and then make that point symmetric with respect to the x axis, R,See Figure 2 below.And the addition of points satisfies the addition closure, addition invertibility, addition associative law and addition distributive law.

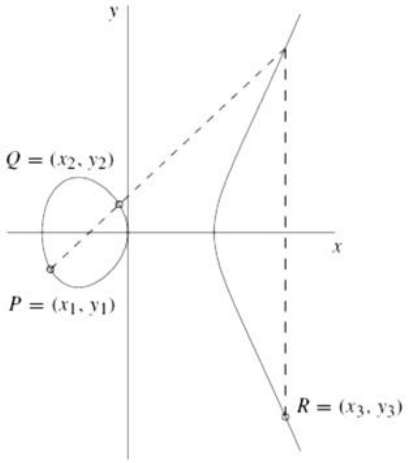

Figure 2. Addition of points on an elliptic curve

Discrete logarithm: Discrete logarithm is the core of elliptic encryption algorithm.The inverse process of scalar multiplication in elliptic curves can be regarded as a discrete logarithm problem.The scalar multiplication of elliptic curves is similar to the associative law of addition of elliptic curves.The popular interpretation of elliptic encryption algorithm is P=eG, the asymmetric algorithm, e*G→P step is easy to get, on the contrary, given P and G it is difficult to deduce e.

Digital signature: Similar to the signature on a document in real life to prove the identity of the signer, the sender achieves password conversion of the digital unit through a specific signature algorithm. This conversion allows the receiver of the data unit to know the source and integrity of the data, and prevents others from forging it.Digital signature verification: the decryption process corresponding to the encryption transformation during signature verification.Digital signatures can provide message authentication, message integrity, and non-repudiation.

2.3. Solution Architecture

ECC algorithm combined with OPT algorithm flow chart is shown in Figure 4： First, a new private key is generated through OPT algorithm (we call it "pseudo private key"), then the message and pseudo private key are signed, and the public key holder decrypts the obtained public key and received signature.Verify the decrypted hash value and the hash value formed by the message to determine whether the authentication is successful.

The "pseudo-key" elliptic algorithm can be thought of as a triplet of (G, E, D).As shown in Figure 3, G is the key generation algorithm, G(pk, sk),F(sk)→sk’,G consists of pk and sk, where pk is the public key, sk is the private key, and sk'is the pseudo private key,First, the public key pk and private key sk are generated through the key generation algorithm G, and then the pseudo private key sk 'corresponding to the private key is generated through algorithm F.User A broadcasts the public key, and user B receives the public key pk from user A.User B uses E(E stands for encryption algorithm, for example, E(pk, m) consists of pk and m, and m refers to the plaintext input) to generate A ciphertext c and send it to user A.User A uses algorithm D, which is the decryption algorithm D(sk ', c) to obtain the corresponding private key and ciphertext as input, and output plaintext D(sk ',c)→

M.Obtain the message m that user B wants to send.The private key is solved by algorithm F to obtain F-1(sk ')→sk(F-1 is the inverse of function F).

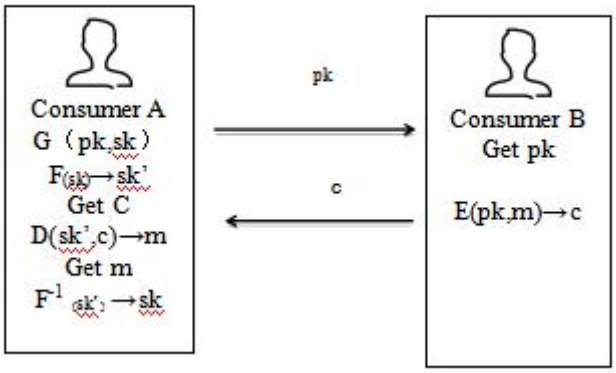

Figure 3. Algorithm schematic diagram

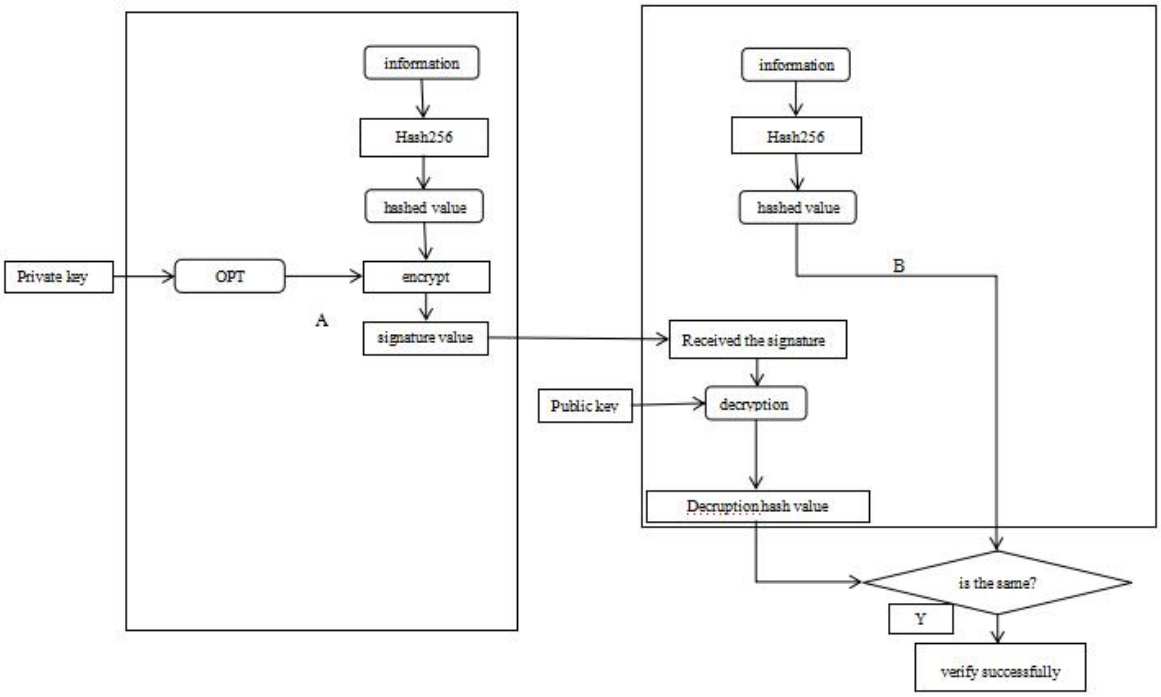

Figure 4. Algorithm flow chart

2.4. Algorithm design principle

The design principle of the algorithm is divided into three parts: pseudo key generation and verification, message signature and signature validity verification.

Pseudo-key generation and verification:

- Pseudo-key generation: e'= e⊕k, pseudo-key e' is generated by xor with a 256-bit random number k. e is the private key provided by the signer, k is the generated random number, e ' is the pseudo-private key, and ⊕is defined as xor operation.
- Generate the corresponding public key: e'G=P, where P is the public key (P is a coordinate point) and G(x,y) is the starting point. Generate the corresponding public key through the pseudo key e' and the starting point.

- (Randomly select a new coordinate point: kG=R, the coordinate of R is (x2,y2), and R is a random number k corresponding to the generated coordinate point. Let r =x2 for convenient observation, then the seat of R is marked as (r,y2).
- Discrete logarithm generation: let uG+vP=kG and uG+vP=kG satisfy the discrete logarithm problem, u and v are two prime numbers of large number decomposition.
- uG+vP=kG →vP=(k-u)G → P=[(k-u)/v]G → e'G=[(k-u)/v]G → e'=(k-u)/v , e=e' ⊕k ⊕k.
- Any point (u, v)(u, v) given by the signer) satisfying the above equation proves to be the holder of the private key e.

Message signatures:

- u=z/s v=k/s ,Let z represent the data processed by the hash function.
- e'=e ⊕k.
- The uG+vP=R=kG→uG+ve'G=kG→u+ve'=k→z/s+re'/s=k→(z+re')/s=u→s=(z+re')/k.r and s are used as message signatures.

Signature validation:

- To calculate u=z/s,v=r/s.
- uG +vP=R is calculated by determining whether the x coordinate of R is equal to r, which means the signature is valid.

2.5. Algorithm implementation

The general outline of the algorithm is shown in Figure 5:

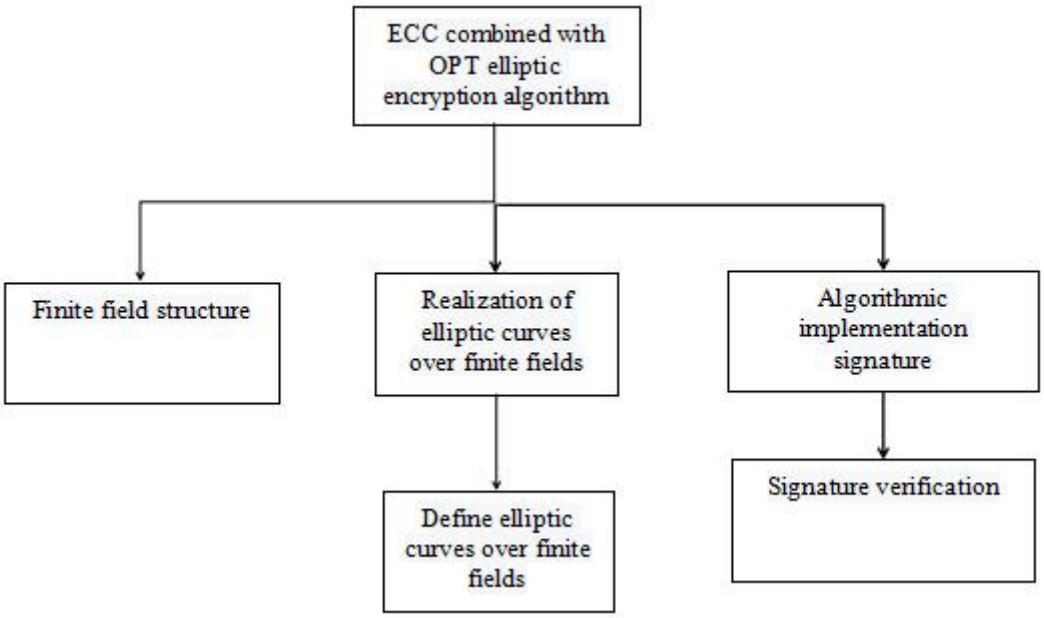

Figure 5. Overall outline of algorithm design

Part of the pseudo-code based on Python code is as follows:

- Point addition on an elliptic curve pseudo-code: four cases of point addition are discussed in the implementation process： 1) The two points are perpendicular to the X-axis.2) The lines of the two points are not perpendicular to the X-axis, and they are two different points.3) The two points are the same.4) Tangent line perpendicular to X-axis:

```
Function< __add__>[self, other]:
Begin
  if<self.a != other.a or self.b != other.b>Then{
     raise ValueError/}
```

```
if< self.x is None>Then{
        output other/}
   if <other.x is None>Then{
        output self/}
       if <self.x == other.x and self.y != other.y>Then{
       output Function<__class__>[None, None, self.a, self.b]/}
   if <self.x != other.x>Then{
        s←(other.y-self.y)/(other.x-self.x)
        ←s**2 -self.x-other.x
        y←s*(self.x-x)-self.y
      output Function<__class__>[x, y, self.a, self.b]/}
   if <self == other>Then{
       s←(3 * self.x ** 2 + self.a) / (2 * self.y)
       x←s ** 2 - 2 * self.x
       y←s * (self.x - x) - self.y
       output Function<__class__>[x, y, self.a, self.b]
   if <self == other and self.y == 0*self.x>Then{
       output Function<__class__>[x, y, self.a, self.b]/}
End
```

- Elliptic curve dot product algorithm pseudo code：

```
Function<__rmul__>[self, coefficient]:
Begin
   coef← coefficient
   current ←self
   output Function<__class__>[x, y, self.a, self.b])
   while< coef> do{
      if <coef & 1>Then{
         result += current
         current += current
          coef >>= 1/}
   output result/}
End
```

- The elliptic curve used to define an elliptic curve over a finite field is y**2=x**3+7,The prime numbers required by a P finite field, where N is the order of the group:

```
      A←0
      B←7
      P←2**256 - 2**32 - 977
      N←0xfffffffffffffffffffffffffffffffebaaedce6af48a03bbfd25e8cd0364141
Function<S256Field>[FieldElement]:
Begin
Function<__init__>[self, num, prime=None]:
Begin
Function<FieldElement>.Function<__init__>[num<-num, prim←P]
End
End
Function<S256Point>[Point]:
Begin
Function<__init__>[self, x, y, a←None, b←None]:
      a, b←S256Field(A), S256Field(B)
          if <type(x) == int>Then{
         Function<S256Field>.Function<__init__>[x←S256Field(x), y←S256Field(y), a←a, b←b) ]/}
          else:
         {Function<S256Field>.Function<__init__>[x←x, y←y, ←a, b←b]/}
End
```

- Verify the signature pseudocode:

```
Function <verify>[ z, sig]:
Begin
    1/s←sig.s**(N - 2)%N
        u←z/s
        v←r/s
        total ← u*G+v*P
        output total.x==sig.r
End
```

- Implement message signing pseudocode:

```
Function <sign>[self, z]:
Begin
    k←Function<deterministic_k>[z]
        r←(k * G).x
        1/k←(k**,N - 2)% N
    S←(z+re)/k
        if <s > N / 2>Then{
        S←N - s
        output Signature(r, s)}
End
```

- Signature validity verification pseudocode:

```
Begin
    P←S256Point(Px,Py )
    Z←z
    R←r
    S←s
    u←( Z *S**N-2%N )% N
    v← (R* S**N%N) % N
    output((u*G + v*R).x.num == R)
End
```

3. Algorithm analysis

3.1. Feasibility verification

The validity of the pseudo-key algorithm is verified by the code.The process is as follows:

1. Experimental message signing: "my secret" as the key input.The decimal value of key E generated by the hash algorithm is:62971298242950415662486979275162298594154135681004836692467839909933090737920.
2. Use my Message as the signature content.The signature hash z generates a hexadecimal fo: 0x231c6f3d980a6b0fb7152f85cee7eb52bf92433d9919b9c5218cb08e79cce78.
3. The generated random number k corresponds to the decimal number: 53141233081433290542791408994650223676848194257993344528424053017128121420544.
4. The hexadecimal value of the pseudo-key e ' generated by the xor key and the random number k is: 0xfe44a2dd99975ae11dc4ff8edc3115c822be60969d6f4c8625330ef8e54e4400.
5. Hexadecimal representation of r (the x coordinate of R) : 0x6238767416e7318ce2fd44b42008bbb9c166e17596fe1a00d9caf4fc41bc202e.
6. The hexadecimal representation of s is,s=(z+re)/k(r and s are signed messages): 0xb59bac6df30fed1aa0a3487b1ec3d19e7e9a3c870cc3375e724b7251a264a9ec.
7. The hexadecimal coordinate of the corresponding public key P is **:** (0caaecdde9c869a116766f6d2a9a63d2ca5c34cefcf5f31f85fa3083b58e192e, 9f7b5994fb8f877178974c7dd78ff27baf733fae08082e844540bacf2b8ed23e).

By substituting the variable public key P into the signature verification program, uG+vP=R was calculated to verify the feasibility of the experiment. The experimental running results are shown in Figure 6:
In order to avoid randomness, English case including digits, underscores, Spaces were selected as keys and information to verify the feasibility of the scheme. The data is shown in Table 1:

Table 1. Corresponding table of different signature verification

e content	generated e	Key e 'generated by OPT algorithm	Signature z content	z	The random number k	Verify validity
123456	1155649646363205395315214381160641773401175759601929706972358053595967610358897	0x89bf38413ed1742501442b9c5deb819a81d389668bcd787fccb0c1fbc836ab18	54321	0x4556c961133d7a593d089834b9099654673d8ab3b8325123bb4f8a240af86d24	5371267512446375345770424556439957451352265294879993394970935519123171692528281	True
xiaotian	1553257431731176132151798793386244334736461221308044453138568646232380270216	0x6ad056cbf5bb2ce93e1bd9e1ff44f7dc7644194bb2db3d4b12a0fac50f8b26a1	give me money	0x8a72b7ca097bf6051af084aec507ecb30209951d6e21efa48329bc9f18d14e31	4783082836841848577224918083236274945659912210547002385330722368203576369156161	True
yy@123_	5483931355940176263880023499333830982969574584552962016147280035758555055491185	0x47132b12a22965e480818c29479645bb40fbae203da7cef88c0540c679e3bcf	I love you!	0xfbb344d409200cf36cc78e91fe34cbdd24dd23ed30bbffe53cfc5b4e4a11f5db	5667471952824463764537959923340350359140561952707320267261753225906467750216160	True

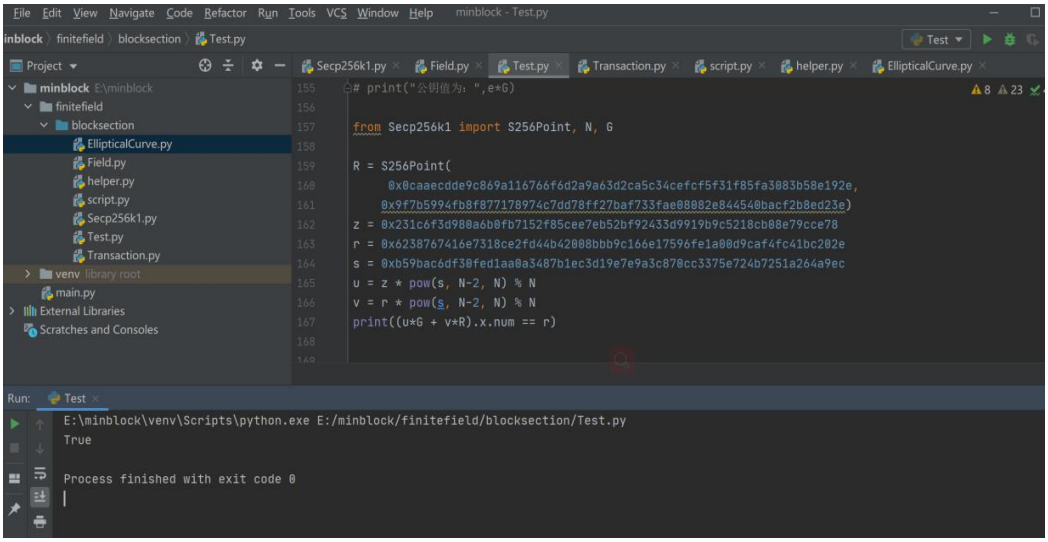

Figure 6. Signature validity verification screenshot

3.2. Security verification

1) Anonymity

Without the signer exposing the key, a third party cannot obtain the real key.Even if an eavesdropper can reverse solve the "private key" we use for information exchange with a mature quantum computer, it turns out to be a fake private key.Eavesdroppers also cannot obtain the user's real key, because the generation of fake private keys is random and one-time.In the algorithm, the pseudo-private key is generated by xor between the private key and the random number k, and k is a random number of 256 bits. The randomness of k ensures the variability of the user private key and the anonymity of the user in the communication process.The eavesdropper can't really get the identity of the correct signer to determine the message signature, so the scheme conforms to anonymity.

2) Security

Absolute security: given a ciphertext c, the attacker cannot tell whether the plaintext is m1 or m2.The encryption algorithm E(pk， m)=c,There is m1, m2 belongs to M(M is a set of plaintext) there is c belongs to C(c is a set of ciphertext).P (E(pk, m1)=c)= P (E(pk, m2)=c) that is, the algorithm that encrypts plaintext into ciphertext with equal probability (an attacker cannot distinguish m1 from m2) is absolutely secure.

Quantum attack resistant: traditional elliptic cryptography,According to the eG=[(k-u)/v]G → e=(k-u)/v,Any combination of (u,v) can be proved to be the holder of e only if it satisfies the preceding equation,I don't know what e is by exhausting (u,v) until e is equal to (k-u)/v.In the condition of quantum computer is not mature, it is not practical to exhaust.But it is not safe in mature functional quantum computing devices.Some experts pointed out that Shor's algorithm takes advantage of the parallelism of quantum computing to quickly decompose common difactors and reverse calculate e in only 8 minutes, thus breaking the basis of RSA algorithm.But for the improved elliptic encryption algorithm,e'G=[(k-u)/v]G → e'=(k-u)/v ,e=e' ⊕k⊕k.It is assumed that the inverse calculation of E 'through quantum equipment is 8 minutes, but it is impossible to calculate the random number K of 256 bits through enumerating, and the pseudo-key generation is random and one-time, which is constantly changing with the use of users.Assume that the user does not use the pseudo key after using it once but stays in the last use, and the eavesdropper steals the real private key after cracking the pseudo private key,assume that it takes 8 minutes to reverse the private key2**256*≈8*10**77 分 ≈1.33*10**76hours.It is not feasible to crack the private key in such a long time. It is theoretically verified that the pseudo-key algorithm is anti-quantum computer. By protecting the real key E, it can

ensure that the user's private key is not obtained by the eavessteal, and it can be concluded that the pseudo-key algorithm is safe under the quantum computer.

4. Conclusion

The development of quantum computing has brought convenience as well as great hidden dangers,Anti-quantum encryption algorithms are imminent,The encryption algorithm proposed in this paper can effectively resist the attack of quantum computing, and the scheme protects the user private key from attack in terms of security.In addition to the high security of the above analysis algorithm, ECC characteristics are retained in terms of performance:less storage space, lower CPU overhead and less bandwidth,And the scheme is based on elliptic curve, sharing the same random number, can achieve password security with minimal changes, high practicability,Higher security at the cost of less computing power.Improved the disadvantages of OPT: convenient key sharing.It also provides a new idea for future key security.In the following study, we will further optimize the algorithm and verify the improved scheme from more aspects. There are still many scientific research work to be improved in the future.

References

[1] Neha, T and Ganesan, R. (2014) "Data Security in Cloud Architecture Based on Diffie Hellman and Elliptical Curve Cryptography"[J], IACR Cryptology ePrint Archive, pp. 49.
[2] Shor, P. W. (1997) Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer[J]. SIAM Journal on Computing, 26(5):1484-1509.
[3] Grover, L. K. (1998) A framework for fast quantum mechanical algorithms [C]//Proceedings of the thirtieth annual ACM symposium on Theory of computing.53-62.
[4] Benioff, P. (1980) "The Computer as a Physical System: A Microscopic Quantum Mechanical Hamiltonian Model of Computers as Represented by Turing Machines"[J], Journal of Statistical Physics, 22(5), pp. 563-591.
[5] "IBM Makes Quantum Computing Available on IBM Cloud to Accelerate Innovation"[C], https://www-03.ibm.com/press/us/en/pressrelease/49661.wss, 2016-5-4.
[6] Shor, P W. (1994) "Algorithms for Quantum Computation: Discrete Logarithms and Factoring", Proceedings 35th Annual Symposium on Foundations of Computer Science[J].IEEE, pp. 124-134.
[7] Arute, F., Arya, K., Babbush, R., et al. (2019) Quantum Supremacy Using a Programmable Superconducting Processor[J]. Nature, 574(7779): 505-510.
[8] How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits[C]https://arxiv.org/abs/1905.09749
[9] Bellovin, S. M. (2011) Frank Miller: Inventor of the one-time pad[J]. Cryptologia, 35(3): 203-222.
[10] Xiao, Y., Wang, B., Wang, Z., et al. (2022) One-time pad scheme based on polar code and OFDM for MMW-RoF system at W-band[J]. Optics Express, 30(3): 4412-4423.