Information-Theoretically Secure Trusted Verification Algorithm

Authors: Ning Zhenhu, Ning Zhenhu

Date: 2025-11-25T00:00:00+00:00

Abstract

Cryptography is the cornerstone of trusted computing. Currently, classical cryptographic systems are facing severe challenges from quantum computing. While post-quantum cryptography (PQC) can resist known quantum attacks, as quantum computing continues to develop and improve, the emergence of new quantum attack methods will be difficult to avoid, and whether the security of PQC algorithms can remain effective in the long term is uncertain. Based on this, this paper proposes a trusted verification algorithm with information-theoretic security. The algorithm is designed based on modular arithmetic; its security follows directly from mathematical principles, does not rely on any hard problem assumptions, and provides complete resistance against quantum computing attacks, with the capability to resist both known and unknown quantum attacks.

Full Text

Preamble

Trusted Verification Algorithm with Information-Theoretic Security

Zhen-Hu Ning
College of Computer Science, Beijing University of Technology, Beijing 100024, China
Beijing Key Laboratory of Trusted Computing

Abstract

Cryptography is the cornerstone of trusted computing. At present, classical cryptographic systems face serious challenges from quantum computing. While Post-Quantum Cryptography (PQC) can resist known quantum attacks, the continued development and maturation of quantum computing will inevitably lead to new quantum attack methods. Whether the security of PQC algorithms

can remain effective in the long term remains unknown. Based on this observation, this paper proposes a trusted verification algorithm with information-theoretic security. Designed using modular operations, its security derives directly from mathematical principles without relying on any computational hardness assumptions, providing complete resistance to quantum computing attacks —both known and unknown.

Keywords: Trusted verification; Information-theoretic security; Modular operations; Quantum-resistant

Active immune trusted computing [1-3] is a computing paradigm that performs trusted verification in parallel with computational operations, ensuring that results meet intended objectives. It uses cryptography as a genetic antibody to implement functions such as identity recognition, state measurement, and secure storage, enabling timely identification of "self" and "non-self" components to destroy and reject harmful substances entering the system—effectively cultivating immune capabilities for network information systems. Trusted computing has become the core technology for implementing China' s Cybersecurity Law and the national classified protection system 2.0. In August 2024, the General Office of the CPC Central Committee and the General Office of the State Council issued the "Opinions on Improving the Market Access System," which specifically emphasizes focusing on new business forms and fields such as artificial intelligence, autonomous trusted computing, and information security. As trusted computing becomes one of these new business forms and fields, it will drive new development in China' s trusted computing industry.

Trusted verification is the core mechanism through which trusted computing identifies "self" and "non-self" components. It dynamically verifies the trustworthiness of information system computing environments (including firmware, operating systems, and applications) based on cryptography. Therefore, the security of cryptography directly affects the effectiveness of trusted verification.

Quantum computing, as the core force of a new generation of technological transformation, has profoundly impacted the cybersecurity field. While driving a computational revolution and enabling information-theoretically secure data communication, quantum computing poses severe challenges to classical cryptographic systems. Shor' s algorithm leverages quantum parallelism and superposition to directly break RSA and Elliptic Curve Cryptography (ECC). Grover' s algorithm can halve the security strength of symmetric cryptography and hash functions. Currently, major countries worldwide are actively deploying Post-Quantum Cryptography (PQC) [4,5], researching and formulating PQC migration plans. The U.S. National Institute of Standards and Technology (NIST) has released multiple PQC standards since August 2024, leading the global PQC migration wave. In response to quantum threats, China' s Commercial Cryptography Research Institute launched a call for new-generation commercial cryptographic algorithms in February 2025.

PQC designs rely on mathematical hard problems that existing quantum al-

gorithms cannot solve in polynomial time, and its security remains based on computational complexity theory. With the continuous development and improvement of quantum computing, the emergence of new quantum algorithms will be inevitable. In this context, whether the hardness of the mathematical problems PQC relies on can be maintained, whether PQC algorithm security can remain effective long-term, and whether PQC can resist new quantum attacks all remain unknown. Therefore, PQC' s ability to resist quantum computing attacks has certain limitations.

In 1949, Shannon [6] first provided a security proof for one-time pad in "Communication Theory of Secrecy Systems." Specifically, when the key and plaintext are of equal length, the key is completely random, and the key is used only once, information-theoretically secure data encryption—perfect secrecy—can be achieved. Shannon' s perfect secrecy theory has profoundly influenced modern cryptography, and the confusion and diffusion methods derived from perfect secrecy have become design criteria for contemporary symmetric cryptography. The core of one-time pad' s information-theoretic security is that its security is a direct corollary of mathematical principles, independent of any hardness assumptions. Based on this principle, designing trusted verification algorithms whose security relies solely on mathematical principles rather than any hardness assumptions is the focus of this research.

2.1 Trusted Verification

Trusted verification is the core mechanism of trusted computing, dynamically verifying the trustworthiness of information system computing environments based on cryptography. Verification objects include but are not limited to firmware, operating systems, and applications. Trusted verification consists of two parts: baseline value collection and trust decision. Baseline value collection: During the first system startup, trusted verification algorithms generate verification values for firmware, operating systems, applications, etc., as baseline values. Trust decision: During each subsequent startup or dynamic loading, the system regenerates verification values for firmware, operating systems, applications, etc., and compares them with baseline values. If they match, the system is deemed trustworthy; otherwise, it is deemed untrustworthy. Mainstream trusted verification algorithms primarily use the SM3 algorithm. With the development of quantum computing, the SM3 algorithm faces continuous challenges, making it urgent to research trusted verification algorithms with complete quantum computing attack resistance.

2.2 Physical Trusted Root TPCM

The Physical Trusted Root TPCM [7] is a hardware module integrated on the system motherboard. Throughout the entire lifecycle of system startup and operation, TPCM always maintains active monitoring capabilities for system software and hardware. TPCM is the physical entity of the trusted verifica-

tion mechanism, providing secure data storage and execution environments for trusted verification. TPCM is the trust anchor that dynamically verifies the trustworthiness of the system computing environment through trusted verification mechanisms. During system startup, TPCM performs trusted verification level by level on firmware, operating systems, and applications. If the verification result is trustworthy, the system is allowed to start; otherwise, it is blocked and an alarm is triggered. During system operation, TPCM dynamically verifies the trustworthiness of the runtime environment. For example, during dynamic program loading, TPCM verifies the trustworthiness of the program to be loaded. If the verification result is trustworthy, the program is allowed to load; otherwise, the program to be loaded is controlled. Using TPCM for trusted verification of information system computing environments has become a core requirement of the national classified protection system 2.0.

2.3 Cryptography Security

Cryptography is divided into computational complexity theory-based cryptography and information-theoretically secure cryptography [8]. Computational complexity theory-based cryptography mainly includes asymmetric cryptography (RSA, ECC, and PQC), symmetric cryptography (DES, AES, SM4, ZUC), and hash functions (SHA0, SHA1, MD5, SM3, SHA2, SHA3). The security of such cryptography relies on computational complexity hardness assumptions and may be cracked with the development of computing technology. For example, the cracking of SHA0, SHA1, and MD5 led to the migration of hash functions to SM3 and SHA2. Information-theoretically secure cryptography refers to cryptography whose security relies on information theory principles rather than any computational complexity hardness assumptions—attackers cannot crack it even with unlimited computing resources. Mathematically analyzing, information-theoretically secure cryptography's security is a direct corollary of mathematical principles, independent of any hardness assumptions. Cracking such cryptography is equivalent to overturning mathematical principles! A typical information-theoretically secure cryptography is one-time pad, which attackers cannot decipher from ciphertext even with unlimited computing resources. Therefore, the security of information-theoretically secure trusted verification algorithms should also be a direct corollary of mathematical principles, independent of any hardness assumptions.

3 Trusted Verification Modeling

The trusted verification algorithm is defined as:

$$y = H(k, x) : \{0,1\}^* \rightarrow \{0,1\}^n$$

which maps binary data of arbitrary length to n-bit binary data, where n is typically 256. The length of x should not be less than n. For data shorter than n, direct storage and bit-by-bit comparison can be used for trust verification during validation.

The specific process is as follows:

1) **Baseline Value Collection**: Input the initial value x_0 of data. TPCM randomly selects k_0 and computes $y_0 = H(k_0, x_0)$. TPCM stores the length of x_0 and confidentially stores k_0 and y_0.

2) **Trust Decision**: Input the value x_t of data at time t. TPCM first verifies whether the length of x_t matches the stored length of x_0. If not, data is deemed untrustworthy. Otherwise, TPCM reads k_0 and y_0, computes $y_t = H(k_0, x_t)$, and verifies whether $y_t = y_0$ holds. If it holds, data is deemed trustworthy; otherwise, it is deemed untrustworthy.

This paper designs the trusted verification algorithm $H(k, x)$ based on modular operations. Its security is a direct corollary of mathematical principles, independent of any hardness assumptions, providing complete quantum computing attack resistance.

4.1 Number Theory

The integer set $\mathbb{Z}$ [9] has the following properties:

1) **Divisibility**: For integers a, b, c with $b \neq 0$, if $a = bc$, we say b divides a, denoted as $b|a$, and b is called a factor of a.

2) **Modulo (Remainder)**: For integers a, b, c, r with $b > 0$ and $0 \leq r < b$, if $a = bc + r$, we call r the remainder of a divided by b, denoted as $r = a \bmod b$.

3) **Prime Numbers**: An integer $N \geq 2$ with only 1 and N as factors is called a prime number. Examples: 2, 3, 5, 7, 11, 13, 17, $\cdots$

Prime number theory is the core of number theory. There are two fundamental theorems about primes: the Fundamental Theorem of Arithmetic and the Prime Number Theorem. The Fundamental Theorem of Arithmetic states that every integer greater than 1 can be uniquely factored into a product of prime numbers. The Prime Number Theorem characterizes the asymptotic behavior of the number of primes not less than x as x approaches infinity.

Theorem 4.1 (Fundamental Theorem of Arithmetic): For any integer $N \geq 2$, there exists a factorization:

$$N = p_{1p}2...p_k$$

where $p_1, p_2, ..., p_k$ are primes (possibly equal), called the prime factors of N. This decomposition is unique up to ordering.

From this theorem, for integer $N \geq 2$, we have $N = p_{1p}2...p_k \geq 2^k$, so $k \leq \log_2 N$. Since the number of prime factors of 1 is 0, for any integer $N \geq 1$, the number of prime factors does not exceed $\log_2 N$.

Theorem 4.2 (Prime Number Theorem): Let $x \geq 2$ be a positive integer, and let $\pi(x)$ represent the number of primes not greater than x. Then:

$$\pi(x) \sim \frac{x}{\ln x}$$

A more precise estimate [10] is:

$$\frac{x}{\ln x - 0.5} < \pi(x) < \frac{x}{\ln x - 1.5}, \quad x \geq 67$$

Notably, the optimal estimate of the Prime Number Theorem depends on the resolution of the Riemann Hypothesis, which is one of the ultimate goals of global mathematical research.

4.2 Key Lemmas

Let S be a set, and $|S|$ represent the number of elements in S. Let P_{256} denote the set of all primes in the interval $[2^{255}, 2^{256})$, i.e.:

$$P_{256} = \{p \mid p \in [2^{255}, 2^{256}) \text{ is prime}\}$$

We have the following estimate.

Lemma 4.1: The number of elements in P_{256} satisfies:

$$|P_{256}| \geq \frac{2^{255}}{256 \ln 2 - 0.5}$$

Proof: According to Theorem 4.2:

$$|P_{256}| = \pi(2^{256}) - \pi(2^{255}) \geq \frac{2^{256}}{256 \ln 2 - 0.5} - \frac{2^{255}}{255 \ln 2 - 1.5} \geq \frac{2^{255}}{256 \ln 2 - 0.5}$$

Let $N \geq 1$ be an integer. Let S_N represent the set of all elements in P_{256} that divide N, i.e.:

$$S_N = \{p \mid p \in P_{256} \text{ and } p|N\}$$

We have the following estimate.

Lemma 4.2: Let integer $N \in [1, 2^m]$ $(m \geq 1)$. Then $|S_N| \leq \frac{m}{255}$.

Proof: (i) If S_N is empty, then $|S_N| = 0 \leq \frac{m}{255}$.

(ii) If S_N is not empty, since the number of prime factors of 1 is 0, we have $N \geq 2$. Let $S_N = \{p_1, p_2, ..., p_l\}$ where $p_i|N$ for $i = 1, 2, ..., l$. According to Theorem 4.1:

$$N \geq p_{1p}2...p_l \geq 2^{255l} = 2^{255|S_N|}$$

Since $2^m \geq N \geq 2^{255|S_N|}$, we have $m \geq 255|S_N|$, and thus $|S_N| \leq \frac{m}{255}$.

4.3.1 Data Encoding

Let the input data be m ($m \geq 1$) bits of binary data $data = a_{m-1}a_{m-2}...a_i...a_0$ where $a_i \in \{0, 1\}$ for $0 \leq i \leq m-1$. We encode $data$ as an element N in the integer set $\mathbb{Z}$:

$$N = 2^m + \sum_{i=0}^{m-1} a_{i2}^i$$

Under this encoding scheme, all binary data correspond one-to-one with all positive integers ≥ 2.

4.3.2 Trusted Verification

Trusted verification mainly includes baseline value collection and trust decision.

1) **Baseline Value Collection**: Input the initial value x_0 of data, where the length $m \in [256, 2^{128})$. TPCM encodes x_0 as integer N_0. TPCM uniformly randomly selects a prime $p \in P_{256}$ and computes $y_0 = N_0 \bmod p$. TPCM stores m and confidentially stores p and y_0.

2) **Trust Decision**: Input the value x_t of data at time t. TPCM first verifies whether the length of x_t equals m. If not, data is deemed untrustworthy. Otherwise, TPCM encodes x_t as integer N_t, reads p and y_0, and computes $y_t = N_t \bmod p$. TPCM verifies whether $y_t = y_0$ holds. If it holds, data is deemed trustworthy; otherwise, it is deemed untrustworthy.

In practice, a true random number generator can be used to generate random numbers, and primality testing methods (such as Miller-Rabin tests) can be used for verification, selecting $p \in P_{256}$ through repeated generation and testing.

4.4 Security Proof

Bayesian theory characterizes the probability features of unknown events. In a guessing game, according to Bayesian theory, the answer is a definite value for the puzzle setter but a random variable defined over the space of possible answers for the guesser. Bayesian theory also applies to cryptography. In encryption algorithms, the key is a definite string for the key holder but a random variable defined over the key space for attackers.

Therefore, according to Bayesian theory, p is a definite value for TPCM but a random variable defined over P_{256} for attackers. This is the core idea of the following theorem proof.

Theorem 4.3: Assume an attacker modifies data such that $x_t \neq x_0$. Suppose x_t and x_0 have equal length m. Then the attacker' s success probability is:

$$Pr[y_t = y_0] \leq \frac{m}{255} \cdot \frac{1}{|P_{256}|}$$

Proof: Since $x_t \neq x_0$, we have $N_t \neq N_0$, i.e., $1 \leq N_t - N_0 \leq 2^m - 1$. Therefore, according to Lemma 4.2:

$$|S_{N_t - N_0}| \leq \frac{m}{255}$$

where $S_{N_t - N_0}$ is defined in equation (13).

Since p is uniformly randomly selected from P_{256}, for attackers, p has $|P_{256}|$ possible values with equal probability:

$$Pr[p = p_i] = \frac{1}{|P_{256}|}, \quad \forall p_i \in P_{256}$$

We have:

$$y_t = y_0 \iff N_t \bmod p = N_0 \bmod p \iff (N_t - N_0) \bmod p = 0 \iff p | (N_t - N_0) \iff p \in S_{N_t - N_0}$$

Therefore, combining Lemma 4.1, the attacker' s success probability is:

$$Pr[y_t = y_0] = Pr[p \in S_{N_t - N_0}] = \frac{|S_{N_t - N_0}|}{|P_{256}|} \leq \frac{m}{255} \cdot \frac{256 \ln 2 - 0.5}{2^{255}} \approx \frac{m}{2^{255}}$$

This proof relies only on basic mathematical principles, independent of any hardness assumptions.

Further Analysis: For trusted verification of 1GB of data, the attacker' s success probability is $\frac{2^{33}}{2^{255}} = 2^{-222} \leq 10^{-66}$. IDC predicts that global data volume will reach 213.56ZB in 2025. For trusted verification of this data volume, the attacker's success probability is $\frac{213.56 \times 2^{70}}{2^{255}} \leq 2^{-177} \leq 10^{-53}$. The 10^{-53} figure represents the upper bound of attack success probability when treating 213.56ZB of data as a single integer for trusted verification. In practice, data is distributed, and trusted verification of each data item is performed independently and non-synchronously. In this case, the attacker' s success probability will be far smaller than this value.

5.1 Polynomial Theory

The polynomial ring $GF(2)[x]$ [11] consists of polynomials with coefficients of 0 or 1, where addition and multiplication operations are performed modulo 2. Let $h \in GF(2)[x]$, and $\deg(h)$ represent the degree of polynomial h. Specifically, $\deg(1) = 0$. We define $\deg(0) = -\infty$, under which the formula $\deg(fg) = \deg(f) + \deg(g)$ holds for all $f, g \in GF(2)[x]$.

The polynomial ring $GF(2)[x]$ has the following properties:

1) **Divisibility**: For $h, f, g \in GF(2)[x]$ with $f \neq 0$, if $h = fg$, we say f divides h, denoted as $f|h$, and f is called a factor of h.

2) **Modulo (Remainder)**: For $h, f, g, r \in GF(2)[x]$ with $f \neq 0$ and $\deg(r) < \deg(f)$, if $h = fg + r$, we call r the remainder of h divided by f, denoted as $r = h \bmod f$.

3) **Irreducible Polynomials**: Let $f \in GF(2)[x]$ with $\deg(f) \geq 1$ and whose only factors are 1 and f itself. Then f is called an irreducible polynomial. Examples: $f = x$, $f = x + 1$, $f = x^2 + x + 1$, $\cdots$

Irreducible polynomial theory is the core of polynomial ring theory. There are two fundamental theorems about irreducible polynomials: the Fundamental Theorem of Arithmetic and the Irreducible Polynomial Counting Theorem. The Fundamental Theorem of Arithmetic states that every polynomial of degree ≥ 1 in $GF(2)[x]$ can be uniquely factored into a product of irreducible polynomials. The Irreducible Polynomial Counting Theorem characterizes the total number of irreducible polynomials of a given degree ≥ 1.

Theorem 5.1 (Fundamental Theorem of Arithmetic): For every polynomial h of degree ≥ 1 in $GF(2)[x]$, there exists a factorization:

$$h = f_{1f}2...f_k$$

where $f_1, f_2, ..., f_k$ are irreducible polynomials (possibly equal), called the prime factors of h. This decomposition is unique up to ordering.

From this theorem, for any non-zero polynomial $h \in GF(2)[x]$ with degree ≥ 1:

$$\deg(h) = \deg(f_1) + \deg(f_2) + ... + \deg(f_k) \geq k$$

Since the number of prime factors of 1 is 0, for any non-zero polynomial $h \in GF(2)[x]$, the number of prime factors $k \leq \deg(h)$.

Theorem 5.2 (Irreducible Polynomial Counting Theorem): The number of irreducible polynomials of degree n ($n \geq 1$) is:

$$M = \frac{1}{n} \sum_{d|n} \mu(d) 2^{n/d}$$

where

$$\mu(d) = \begin{cases} (-1)^k, & d = p_{1p}2...p_k \text{ with distinct primes } p_1, p_2, ..., p_k \\ 1, & d = 1 \\ 0, & \text{otherwise} \end{cases}$$

5.2 Key Lemmas

Let I_{256} represent the set of all 256-degree irreducible polynomials in $GF(2)[x]$, i.e.:

$$I_{256} = \{f \mid f \in GF(2)[x], \deg(f) = 256 \text{ and } f \text{ is irreducible}\}$$

We have the following estimate.

Lemma 5.1: The number of elements in I_{256} is:

$$|I_{256}| = \frac{2^{256} - 2^{128}}{256}$$

Proof: Since $\mu(1) = 1$, $\mu(2) = -1$, and $\mu(2^k) = 0$ for $k \geq 2$, according to Theorem 5.2:

$$|I_{256}| = \frac{2^{256} - 2^{128}}{256}$$

Let $N(x) \in GF(2)[x]$ be a non-zero polynomial. Let S_N represent the set of all elements in I_{256} that divide $N(x)$, i.e.:

$$S_N = \{f \mid f \in I_{256} \text{ and } f|N\}$$

We have the following estimate.

Lemma 5.2: Let non-zero polynomial $N(x) \in GF(2)[x]$ satisfy $\deg(N) \in [0, m]$ $(m \geq 1)$. Then $|S_N| \leq \frac{m}{256}$.

Proof: (i) If S_N is empty, then $|S_N| = 0 \leq \frac{m}{256}$.

(ii) If S_N is not empty, since the number of prime factors of 1 is 0, we have $\deg(N) \geq 1$. Let $S_N = \{f_1, f_2, ..., f_l\}$ where $f_i|N$ for $i = 1, 2, ..., l$. According to Theorem 5.1:

$$\deg(N) \geq \deg(f_{1f}2...f_l) = \sum_{i=1}^{|S_N|} \deg(f_i) = 256 \cdot |S_N|$$

Therefore, $|S_N| \leq \frac{\deg(N)}{256} \leq \frac{m}{256}$.

5.3.1 Data Encoding

Let the input data be m $(m \geq 1)$ bits of binary data $data = a_{m-1}a_{m-2}...a_i...a_0$ where $a_i \in \{0, 1\}$ for $0 \leq i \leq m-1$. We encode $data$ as an element $N(x)$ in the polynomial ring $GF(2)[x]$:

$$N(x) = x^m + \sum_{i=0}^{m-1} a_i x^i$$

Under this encoding scheme, all binary data correspond one-to-one with all polynomials of degree ≥ 1 in $GF(2)[x]$.

5.3.2 Trusted Verification

Trusted verification mainly includes baseline value collection and trust decision.

1) **Baseline Value Collection**: Input the initial value x_0 of data, where the length $m \in [256, 2^{128})$. TPCM encodes x_0 as polynomial $N_0(x) \in GF(2)[x]$. TPCM uniformly randomly selects an irreducible polynomial $f \in I_{256}$ and computes $y_0 = N_0 \bmod f$. TPCM stores m and confidentially stores f and y_0.

2) **Trust Decision**: Input the value x_t of data at time t. TPCM first verifies whether the length of x_t equals m. If not, data is deemed untrustworthy. Otherwise, TPCM encodes x_t as polynomial $N_t(x) \in GF(2)[x]$, reads f and y_0, and computes $y_t = N_t \bmod f$. TPCM verifies whether $y_t = y_0$ holds. If it holds, data is deemed trustworthy; otherwise, it is deemed untrustworthy.

In practice, a true random number generator can be used to generate random numbers, and irreducible polynomial testing methods can be used for verification, selecting $f \in I_{256}$ through repeated generation and testing.

5.4 Security Proof

According to Bayesian theory, f is a definite value for TPCM but a random variable defined over I_{256} for attackers.

Theorem 5.3: Assume an attacker modifies data such that $x_t \neq x_0$. Suppose x_t and x_0 have equal length m. Then the attacker' s success probability is:

$$Pr[y_t = y_0] \leq \frac{m-1}{|I_{256}|}$$

Proof: Since $x_t \neq x_0$, we have $N_t - N_0 \neq 0$ and $\deg(N_t - N_0) \leq m-1$. According to Lemma 5.2:

$$|S_{N_t - N_0}| \leq \frac{m-1}{256}$$

where $S_{N_t - N_0}$ is defined in equation (35).

Since f is uniformly randomly selected from I_{256}, for attackers, f has $|I_{256}|$ possible values with equal probability:

$$Pr[f = f_i] = \frac{1}{|I_{256}|}, \quad \forall f_i \in I_{256}$$

We have:

$$y_t = y_0 \iff (N_t - N_0) \bmod f = 0 \iff f | (N_t - N_0) \iff f \in S_{N_t - N_0}$$

Therefore, combining Lemma 5.1, the attacker' s success probability is:

$$Pr[y_t = y_0] = Pr[f \in S_{N_t - N_0}] = \frac{|S_{N_t - N_0}|}{|I_{256}|} \leq \frac{m-1}{2^{256} - 2^{128}} \approx \frac{m}{2^{256}}$$

This proof relies only on basic mathematical principles, independent of any hardness assumptions.

Further Analysis: For trusted verification of 1GB of data, the attacker' s success probability is $\frac{2^{33}}{2^{256}} = 2^{-223} \leq 10^{-67}$. IDC predicts that global data volume will reach 213.56ZB in 2025. For trusted verification of this data volume, the attacker' s success probability is $\frac{213.56\times 2^{70}}{2^{256}} \leq 2^{-178} \leq 10^{-53}$. The 10^{-53} figure represents the upper bound of attack success probability when treating 213.56ZB of data as a single polynomial for trusted verification. In practice, data is distributed, and trusted verification of each data item is performed independently and non-synchronously. In this case, the attacker's success probability will be far smaller than this value.

6 Performance Analysis

The main computation in the trusted verification algorithm is modular operation. By encoding input data as integers (or polynomials) and using randomly selected primes (or irreducible polynomials) as divisors, modular operations are performed.

Let ADD256 represent addition of two 256-bit unsigned integers, and XOR256 represent XOR operation of two 256-bit data blocks. For trusted verification of m ($m \geq 256$) bits of binary data, the main computational costs are:

Modular operations have strong hardware optimization capabilities, especially polynomial modular operations. Polynomial modular operations are also known as Cyclic Redundancy Check (CRC) algorithms [12-14], globally universal data integrity verification algorithms widely used for checking errors in data storage and transmission. CRC algorithms are highly hardware-optimizable, with optimized performance theoretically reaching 10-100 times that of hash algorithms.

One-time pad is also called perfect secrecy, the encryption algorithm with the highest security level. Correspondingly, this paper calls information-theoretically secure trusted verification algorithms **Perfect Trust**. Perfect Trust collaborates in parallel with perfect secrecy, improving and developing information-theoretically secure cryptography systems.

The following table compares Perfect Trust with existing trusted schemes.

Perfect Trust has a natural mathematical structure (modular operations) and natural mathematical proofs (provable using elementary mathematics). It has the same security level as perfect secrecy while avoiding perfect secrecy' s key management drawbacks. It provides complete quantum computing attack resistance, resisting both known and unknown quantum attacks. Even when quantum computing matures, quantum computing + AI matures, or even if the famous mathematical problem P=NP is proven (which would break all computational complexity theory-based cryptography), Perfect Trust remains effective!

Combining Perfect Trust with perfect secrecy constitutes an information-theoretically secure encryption algorithm with unified confidentiality and integrity. This encryption algorithm is detailed in literature [21].

This paper proposes information-theoretically secure trusted verification algorithms, designing them based on both integer theory and polynomial theory. The integer modular operation-based trusted verification algorithm converts input data to integers and performs modulo operations with randomly selected primes. The polynomial modular operation-based trusted verification algorithm converts input data to polynomials and performs modulo operations with randomly selected irreducible polynomials. The security of both trusted verification algorithms is a direct corollary of mathematical principles, independent of any hardness assumptions, providing complete quantum computing attack resistance against both known and unknown quantum attacks. Meanwhile, modular operations are highly hardware-optimizable and have broad application prospects.

References

[1] Shen Changxiang. Building cybersecurity with trusted computing[J]. Qiushi, 2015(20).

[2] Shen Changxiang. Building a cybersecurity defense line with active immune trusted computing 3.0 to create a clean cyberspace[J]. Information Security Research, 2018, 4(04).

[3] Shen Changxiang, Zuo Xiaodong. Introduction to Cyberspace Security[M]. Publishing House of Electronics Industry, 2018 (2024 reprint).

[4] Bernstein D J, Buchmann J, Dahmen E. Post-Quantum Cryptography[M]. Springer, 2008.

[5] Xiang Hong. Research on NIST Post-Quantum Cryptography Analysis and Standardization[M]. Publishing House of Electronics Industry.

[6] Shannon C E. Communication theory of secrecy systems[J]. Bell System Technical Journal, 1949, 28(4): 656–715.

[7] National Information Security Standardization Technical Committee (SAC/TC 260). Information security technology—Trusted computing specification—Trusted platform control module: GB/T 40650-2021[S]. China Standards Press, 2021.

[8] Douglas R S. Cryptography Theory and Practice (Third Edition)[M]. Translated by Feng Dengguo et al. Publishing House of Electronics Industry, 2016.

[9] Pan Chengdong, Pan Chengbiao. Elementary Number Theory (Fourth Edition)[M]. Peking University Press, 2024.

[10] Rosser J B, Schoenfeld L. Approximate formulas for some functions of prime numbers[J]. Illinois Journal of Mathematics, 1962, 6(1): 64–94.

[11] Lin Dongdai. Foundations of Algebra and Finite Fields (Second Edition)[M]. Higher Education Press, 2022.5.

[12] ITU-T. Error-correcting procedures for DCEs using asynchronous-to-synchronous conversion. ITU-T V.42[S]. Geneva: International Telecommunication Union, 1994.

[13] ISO/IEC. Information technology—Telecommunications and information exchange between systems—High-level data link control (HDLC) procedures. ISO/IEC 13239:2002[S]. Geneva: International Organization for Standardization, 2002.

[14] IEEE. IEEE Standard for Ethernet: IEEE Std 802.3-2018[S]. New York: IEEE Standards Association, 2018.

[15] Zyskind G, Nathan O, Pentland A. Decentralizing Privacy: Using Blockchain to Protect Personal Data. IEEE Symposium on Security and Privacy, 2015.

[16] Zheng Z, Xie S, Dai H, Chen X, Wang H. An Overview of Blockchain Technology: Architecture, Consensus and Future Trends. IEEE BigData Congress, 2017.

[17] ElGamal T. A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. IEEE Trans. Inf. Theory, 1985, IT-31(4): 469–472.

[18] Johnson D, Menezes A, Vanstone S. The Elliptic Curve Digital Signature Algorithm (ECDSA). Int. J. Inf. Sec., 2001, 1: 36–63.

[19] Trusted Computing Group. Trusted Platform Module 2.0 Library, Version 184[S]. 2025.

[20] National Information Security Standardization Technical Committee (SAC/TC 260). Information security technology—Trusted computing specification—Trusted software base: GB/T 37935-2019[S]. China Standards Press, 2019.

[21] Ning Z H. Unconditionally secure encryption algorithm with unified confidentiality and integrity. 2025. eprint.iacr.org/2025/1140.

(Email: ningzhenhu@bjut.edu.cn; nzh41034@163.com)

Note: Figure translations are in progress. See original paper for figures.