electronics

MDPI

Article

Research on Secure Communication on In-Vehicle Ethernet Based on Post-Quantum Algorithm NTRUEncrypt

Yuan Zhu [1,2], Yipeng Liu [3], Mingzhi Wu [4], Jinzhao Li [2], Shiyang Liu [5] and Jianning Zhao [1,*]

[1] School of Automotive Studies, Tongji University, Shanghai 201804, China; yuan.zhu@tongji.edu.cn
[2] Sino-German School for Postgraduate Studies, Tongji University, Shanghai 201804, China; 1833383@tongji.edu.cn
[3] College of Electronic and Information Engineering, Tongji University, Shanghai 201804, China; 1931395@tongji.edu.cn
[4] Nanchang Automotive Institute of Intelligence & New Energy, Tongji University (NAIT), Nanchang 330052, China; wumingzhi@naiine.com
[5] Generation Department, East China Electric Power Design Institute, Shanghai 200063, China; liusy3118@ecepdi.com
* Correspondence: 1510810@tongji.edu.cn

Abstract: In the context of the evolution of in-vehicle electronic and electrical architecture as well as the rapid development of quantum computers, post-quantum algorithms, such as NTRUEncrypt, are of great significance for in-vehicle secure communications. In this paper, we propose and evaluate, for the first time, a NTRUEncrypt enhanced session key negotiation for the in-vehicle Ethernet context. Specifically, the time consumption and memory occupation of the NTRUEncrypt Elliptic Curve Diffie–Hellman (ECDH), and Rivest–Shamir–Adleman (RSA) algorithms, which are used for session key negotiation, are measured and compared. The result shows that, besides the NTRUEncrypt's particular attribute of resisting quantum computer attacks, the execution speed of session key negotiation using NTRUEncrypt is 66.06 times faster than ECDH, and 1530.98 times faster than RSA at the 128-bit security level. The memory occupation of the algorithms is at the same order of magnitude. As the transport layer security (TLS) protocol can fulfill most performance requirements of the automotive industry, post-quantum enhanced session key negotiation will probably be widely used for in-vehicle Ethernet communication.

Keywords: secure communication; in-vehicle Ethernet; post-quantum algorithm; NTRUEncrypt

Citation: Zhu, Y.; Liu, Y.; Wu, M.; Li, J.; Liu, S.; Zhao, J. Research on Secure Communication on In-Vehicle Ethernet Based on Post-Quantum Algorithm NTRUEncrypt. *Electronics* **2022**, *11*, 856. https://doi.org/10.3390/electronics11060856

Academic Editor: Jose Eugenio Naranjo

Received: 23 January 2022
Accepted: 7 March 2022
Published: 9 March 2022

1. Introduction

In-vehicle secure communication is a necessary technology for the development of the Internet of Vehicles. Cryptographic algorithms are the basic methods to ensure the security of information. The corresponding relationship between the security demands of the in-vehicle security communication and the cryptographic algorithms is shown in Table 1.

Table 1. Security demands and commonly used basic cryptographic algorithms.

Security Demands	Commonly Used Basic Cryptographic Algorithms
Confidentiality	Symmetric Cryptography, Public Key Cryptography
Integrity	Hash Algorithm
Availability	Hash Algorithm, Public Key Cryptography
Authenticity	Public Key Cryptography
Non-repudiation	Public Key Cryptography

Asymmetric algorithms can guarantee the authenticity and non-repudiation of data for in-vehicle security communication and have an irreplaceable role. The existing commonly

used asymmetric algorithms are RSA and Elliptic Curve Cryptography (ECC) algorithms. However, with the development of quantum computers, the mathematical problems on which RSA or ECC is based will be deciphered. For example, quantum computers, based on the Shor algorithm [1], can easily decipher RSA and ECC algorithms.

One way to eliminate this hidden danger is to use post-quantum cryptographic algorithms. Lattice-Based Public-Key Cryptographic algorithm (LB-PKC), which has the characteristics of simple structure and fast execution, is an important category of post-quantum algorithms. Among LB-PKC algorithms, the yet unbroken post-quantum algorithm, NTRUEncrypt [2], as a variation of the Number Theory Research Unit (NTRU) algorithm [3], is a representative algorithm [4].

The session key negotiation of the handshake protocols is usually achieved by asymmetric algorithms. For example, RSA or ECDH is used to generate session keys in TLS [5]. However, it is unsafe when faced with the future quantum computers that run the Shor algorithm. The NTRUEncrypt-based in-vehicle session key negotiation scheme can resist this kind of quantum computer.

1.1. Related Research

Related research on the use of post-quantum algorithms in the automotive field mainly includes applications of post-quantum algorithms on the PC platforms [6,7], smart card [8], wireless sensor network (WSN) [9], or the vehicle ad-hoc network (VANET) field [10,11]. It focuses on the evaluation of the efficiency of algorithms and protocols or the comparison with other existing public key algorithms.

Hien Ba Nguyen [6] conducted systematic research on the NTRU cryptosystem, including the security of NTRU algorithms, comparison with RSA, ECC, and McEliece in encryption overhead, decryption overhead, and key length. He pointed out that NTRU has the best performance in encryption and decryption speed. The experiment was conducted on a platform based on Pentium 2, Pentium 4, or Intel Core 2 that operated a non-real-time OS, so it does not reflect the execution efficiency in the embedded environment. Aihan Yin [7] applied the NTRU-based NSS signature algorithm to 10G EPON and proposed a new authentication scheme using the NSS signature algorithm. His research proved that the scheme was much better than the ECC signature verification method in terms of registration efficiency and transmission latency. However, it does not take into consideration the limited computing and storage capacity of the in-vehicle environment. Safdar Shaheen [8] used NTRU for election voting and proposed a safe and efficient electronic voting scheme based on NTRU. The NTRU algorithm was modified to be deployed on a smart card to ensure the anonymity and privacy of voters. The article considers the characteristics of low computing ability and limited storage space of the target platform. However, it does not give the related experimental data on an existing platform and it does not carry out comparisons with other algorithms.

Johanna Sepúlveda et al. [9] studied the post-quantum enabled Datagram Transport Layer Security (DTLS) and showed that NTRUEncrypt could feasibly integrate this solution to WSN. However, the connectionless communication feature of DTLS is not suitable for in-vehicle communication. Mi Bo et al. [10,11] used the NTRUEncrypt algorithm in the VANET to realize the protection of privacy information based on location services, which can greatly reduce the computational and communication overhead compared to the original scheme. However, their research is mainly about NTRUEncrypt algorithm-based communication scheme on the inter-vehicle networks and vehicle-to-cloud network. Experimental efficiency is analyzed but experiments are not realized on the vehicle on-board unit.

In summary, although related research has been conducted in the resource-constrained environments and inter-vehicle networks, there is currently no research on the feasibility of post-quantum algorithms in the vehicular on-board communication protocol. Studying the performance of in-vehicle anti-quantum secure communication can provide feasibility

and an efficiency reference for the development of in-vehicle communication with an anti-quantum security attribute.

1.2. The Work in This Paper

In this paper, the feasibility of the NTRUEncrypt algorithm applied to the in-vehicle microcontroller is discussed and proved. NTRUEncrypt is compared to the existing session key negotiation algorithms in terms of the time and memory overhead.

The structure of this paper is as follows. Section 1 shows the significance of the post-quantum NTRU algorithm for in-vehicle communication and related research. Section 2 describes the trend of vehicular Ethernet-based architecture and the application scenario of the NTRUEncrypt algorithm as well as the function of NTRUEncrypt. The mathematical background of the NTRU algorithm is introduced in Section 3 and Appendices A–D. The experiment principle and result analysis are shown in Section 4. In Section 5, conclusions and future work are discussed.

2. In-Vehicle Secure Communication

2.1. Research on In-Vehicle Ethernet Communication

The previous researches on in-vehicle communication are mostly based on Controller Area Network (CAN) [12] and CAN FD [13] secure communication protocol. The in-vehicle network architecture is evolving, and the onboard protocol is transforming from CAN protocol to CAN FD and Ethernet protocol, as shown in Figure 1.

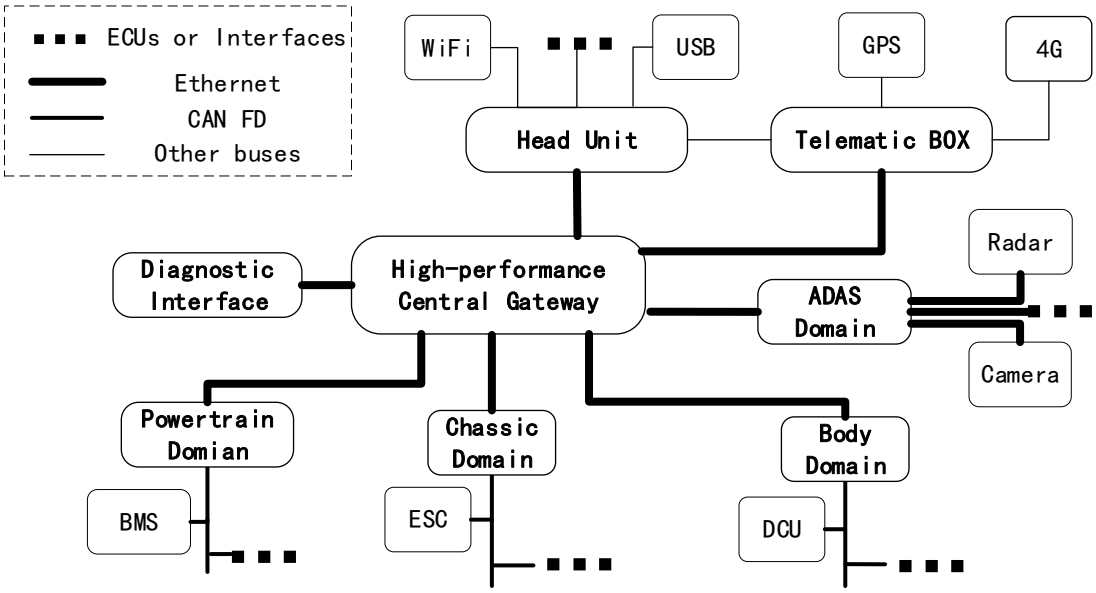

Figure 1. Future trends of in-vehicle electronic and electrical (E/E) architecture.

Zelle et al. [14] conducted a feasibility study of the TLS protocol on the in-vehicle Ethernet. The results showed that the TLS protocol running on the microcontroller could fulfill most performance requirements of the automotive industry. However, as pointed out above, traditional TLS only contains RSA or ECC algorithms, which cannot resist quantum computer attacks.

The introduction of Ethernet to the in-vehicle communication provides the possibility for the application of the post-quantum algorithm NTRU. The ciphertext of NTRU is several hundred bytes long, and for the CAN or CAN FD bus, there will be more communication overheads. The payload of a single Ethernet frame is up to 1500 bytes, and one NTRU ciphertext block can be transmitted at a time.

2.2. Choosing NTRU Variation Algorithm

The premise of encrypted communication on Ethernet is that the session key has been negotiated. The direct encryption mechanism and Diffie–Hellman exchange mechanism, both based on public-key cryptography, are the main methods to realize the key negotiation. The direct encryption mechanism is chosen for NTRU to construct a session key in this paper.

Variations of NTRU encryption algorithms include NTRUEncrypt and NTRU HRSS.

It can be concluded from Table 2 that the NTRUEncrypt algorithm has higher efficiency, as well as an advantage in terms of key length, when tested on 2016 Broadcom BCM2836 (Broadcom, Irvine, CA, USA). The characteristics make NTRUEncrypt more promising in embedded systems.

Table 2. Comparison of various encryption algorithms based on NTRU [15].

Variations of NTRU Algorithm	NIST Level	Security Level	Public Key/(Bit)	CPU Cycles in Key Generation	CPU Cycles in Key Encapsulation/Decapsulation
NTRUEncrypt-443	1	128-bit	611	4,818,993	788,041/1,111,005
NTRUEncrypt-743	1–5	128-bit–512-bit	1023	12,947,474	1,607,275/2,661,836
NTRU HRSS	1	128-bit	1138	112,743,476	3,614,922/10,691,695

2.3. Use Cases of NTRUEncrypt of In-Vehicle Ethernet Communication

As shown in Figure 1, in-vehicle Ethernet will be implemented between a diagnostic tool and a high-performance Central Gateway, between a Telematic Box and a high-performance Central Gateway, and so on.

Currently, the nodes on the Ethernet, including ECUs and MPUs, mainly rely on RSA or ECDH to negotiate and update the session keys. In some cases, the nodes need digital certificates based on the ECDSA algorithm to prove their legitimacy.

A possible attack trace for the vehicular diagnostic system, which uses classical PKC, namely RSA or ECDH algorithm, is illustrated in Figure 2. Providing a quantum computer running the Shor algorithm is available, the above diagnostic system can be deciphered. Firstly, an attacker needs a legitimate diagnostic tool. Then, with the legitimate diagnostic tool, the attacker can eavesdrop the tool's digital certificate and the ciphertext of the Ethernet frames between this tool and the high-performance central gateway with not much effort. At last, the attacker uses the quantum computer to obtain the private key used for signing, as well as the temporary private key, which is used to derivate the session key. At this moment, the attacker can forge diagnostic tools or analyze the diagnostic frames of the OEM to do further attacks. The quantum computer is a disaster for the data asset, which is protected by ECC or RSA algorithm. Post-quantum algorithms such as NTRUEncrypt are an efficient method to resist the future threat of quantum computers.

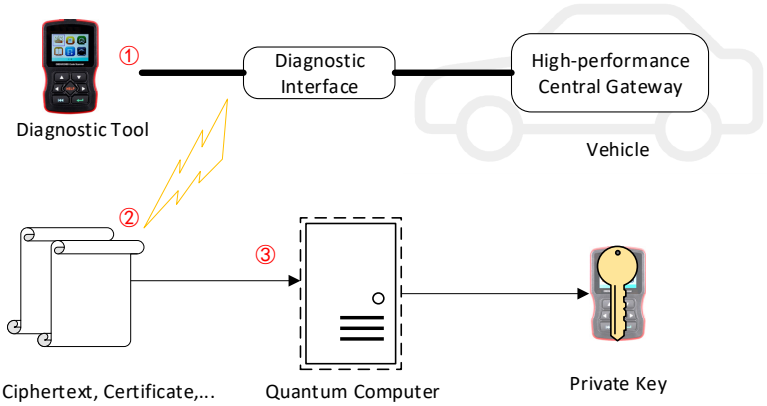

Figure 2. A possible attack trace by using quantum computers in the future.

2.4. Function of NTRUEncrypt Algorithm in TLS

TLS protocol may be used in the diagnostic function for the vehicle. In the TLS protocol, public-key cryptographic algorithms have two main applications: certificate-based authentication and session key negotiation.

This paper focuses on the application of the NTRUEncrypt algorithm in the session key negotiation process, and the following assumptions are made for the identity authentication function.

Assumption 1: The certificate uses a post-quantum cryptographic signature algorithm with good signature verification efficiency.

Taking the Falcon signature algorithm [16], one of the NTRU variations as an example, the efficiency of signature verification is roughly the same order of magnitude as the efficiency of NTRUEncrypt encapsulation or decapsulation operation. The comparison result is displayed in Table 3.

Table 3. Comparison of time overhead between Falcon signature algorithm and NTRUEncrypt algorithm [15].

Algorithms Derived from NTRU	NIST Level	Public Key/(Bit)	CPU Cycles in Verification for the Signature or Key Encapsulation/Decapsulation
Falcon-512	1	897	439,446
Falcon-1024	4–5	1793	882,054
NTRUEncrypt-443	1	611	788,041/1,111,005
NTRUEncrypt-743	1–5	1023	1,607,275/2,661,836

Assumption 2: The certificate can pass the public key of the post-quantum cryptographic algorithm used for the session key negotiation process.

3. Mathematical Background of NTRU

The NTRU algorithm is based on the mathematical problem on the lattice, namely the shortest vector problem (SVP) on the lattice. NTRUEncrypt is a variation of NTRU. NTRU was first proposed by Jeffrey Hoffstein et al. [3], and many variations [2,17,18] of NTRU have appeared since then. The variant algorithms are all based on the same mathematical principle of the original algorithm, so this article takes the original NTRU algorithm as an example to introduce the mathematical background of NTRUEncrypt.

3.1. Mathematical Principle

3.1.1. Public Parameters

The public parameters of the NTRU are (N, p, q, d), where N and p are prime numbers, and the greatest common divisor $\gcd(N,q) = \gcd(p,q) = 1$, and $q > (6d + 1)p$, which enables the realization of decryption.

The NTRU cryptosystem's elements, such as key, plaintext, or ciphertext, are displayed as polynomials. NTRU operations, such as Key generation, Encryption, or Decryption, are based on a polynomial ring R in Appendix A

For the sake of explanation, we assume that the encryption implementer is Bob and the decryption implementer is Alice.

3.1.2. Key Generation

The process of key generation is described as follows [3]:

Alice first selects two polynomials randomly, $f(x) \in R$ and $g(x) \in R$. In this paper, function $f(x)$ and vector f mean the same polynomial.

$$f(x) \in T(d+1, d) \tag{1}$$

$$g(x) \in T(d, d) \tag{2}$$

where T is a ternary polynomial, which is explained in Appendix B. $T(d+1, d)$ indicates that the $d+1$ coefficients of the polynomial are 1, d coefficients are -1, and the remaining coefficients are 0.

Then, we calculate the inverse of the polynomial $f(x)$, which is explained in more detail in Appendix D.

$$F_q(x) = f(x)^{-1} \in R_q \tag{3}$$

$$F_p(x) = f(x)^{-1} \in R_p \tag{4}$$

Equations (3) and (5) are different expressions of the same equation. Mod q means that the integer coefficients of the polynomial are the remainder divided by q.

$$f(x) \otimes F_q(x) \equiv 1 \bmod q \tag{5}$$

$\otimes$, which is explained in Appendix C, means the multiplication operation between polynomials in the ring R_q. If the inverse does not exist, we reselect $f(x)$.

Then, Alice calculates the public key $h(x)$:

$$h(x) = F_q(x) \otimes g(x) \in R_q \tag{6}$$

The corresponding private key is $\left(f(x), F_p(x)\right)$. Alice sends the public key $h(x)$ to Bob.

3.1.3. Encryption

The plaintext that Bob wants to send is a polynomial $m(x)$ whose coefficients are between $-\frac{1}{2}p$ and $\frac{1}{2}p$, that is to say, $m(x)$ is a center promotion of the polynomial in the ring R_p. Bob randomly selects the polynomial $r(x) \in T(d, d)$ and calculates the ciphertext $e(x)$, which is in the ring R_q.

$$e(x) \equiv ph(x) \otimes r(x) + m(x) \bmod q \tag{7}$$

3.1.4. Decryption

Alice receives Bob's ciphertext and starts to decrypt it.

First, we calculate:

$$a(x) \equiv f(x) \otimes e(x) \bmod q \tag{8}$$

Since $q > (6d + 1)p$, $a(x)$ can be center promoted to R_p.

Then, we use the private key to obtain the plaintext $b(x)$:

$$b(x) \equiv F_p(x) \otimes a(x) \bmod p \tag{9}$$

$b(x)$ is the decrypted plaintext, which is equal to the plaintext $m(x)$.

3.2. Choosing Parameters of NTRUEncrypt

Six session key negotiation algorithms with two security levels were selected in our research and experiments, as presented in Table 4. The security levels are presented in bits [19,20].

Table 4. Algorithms studied in this paper, their security level, and length of public key.

Cryptographic Algorithms for Session Key Negotiation	Security Level	Length of Public Key (Bit)
NTRUEncrypt EES443EP1	128-bit	L(Pk) = 611
RSA 3072	128-bit	L(Pk) = 3072
ECDH (ECC256)	128-bit	L(Pk) = 256
NTRUEncrypt EES743EP1	256-bit	L(Pk) = 1023
RSA 15360	256-bit	L(Pk) = 15,360
ECDH (ECC521)	256-bit	L(Pk) = 521

4. Experiment Process and Analysis

4.1. Experiment Principle

The experiment was conducted on the Infineon AURIX TC397 high-performance microcontroller (Infineon, Neubiberg, Germany) [21], which is mainly used for applications such as for in-vehicle domain controllers and advanced gateways.

To implement the TCP-based secure communication, the Lightweight IP (LwIP) protocol stack was transplanted into the projects on the AURIXs. Then negotiation messages were added to the TCP protocol, as presented in Figure 3, to achieve an equivalent experiment of the session key negotiation process of the TLS protocol. Step1 and Step2 are the same as the steps of the session key negotiation based on RSA in TLS protocol version 1.2 [5]. For the convenience of measurement, Step3 and Step4 are added to our experimental principle.

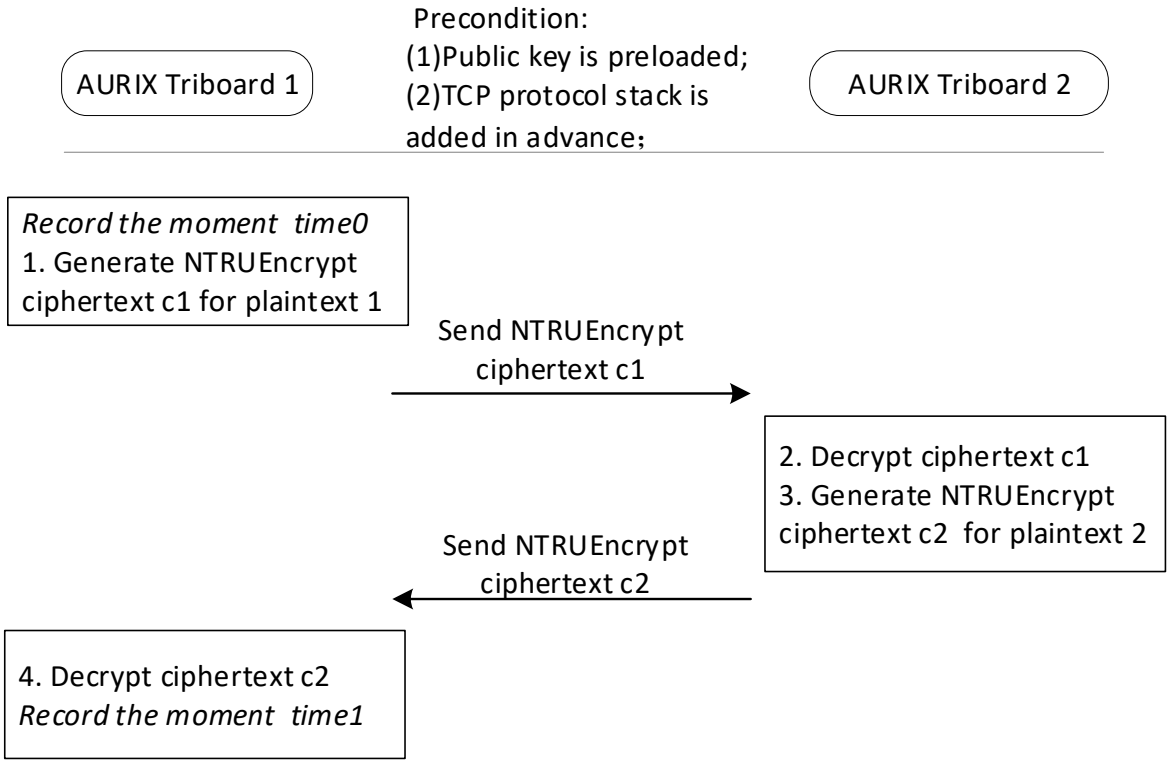

Figure 3. Principle of session key negotiation experiment based on NTRUEncrypt.

If Step1 to Step4 is conducted 1000 times, and the start time *time*0 and end time *time*1 are recorded, the time of session key negotiation t based on NTRUEncrypt can be calculated as follows:

$$t = (time1 - time0)/(1000 \times 2) \quad (10)$$

4.2. Experiment Equipment and Settings

Figure 4 shows the equipment and software used in our experiments. The AURIX Triboard (Infineon, Neubiberg, Germany) [22], which has an AURIX microcontroller and other interfaces, is a development board.

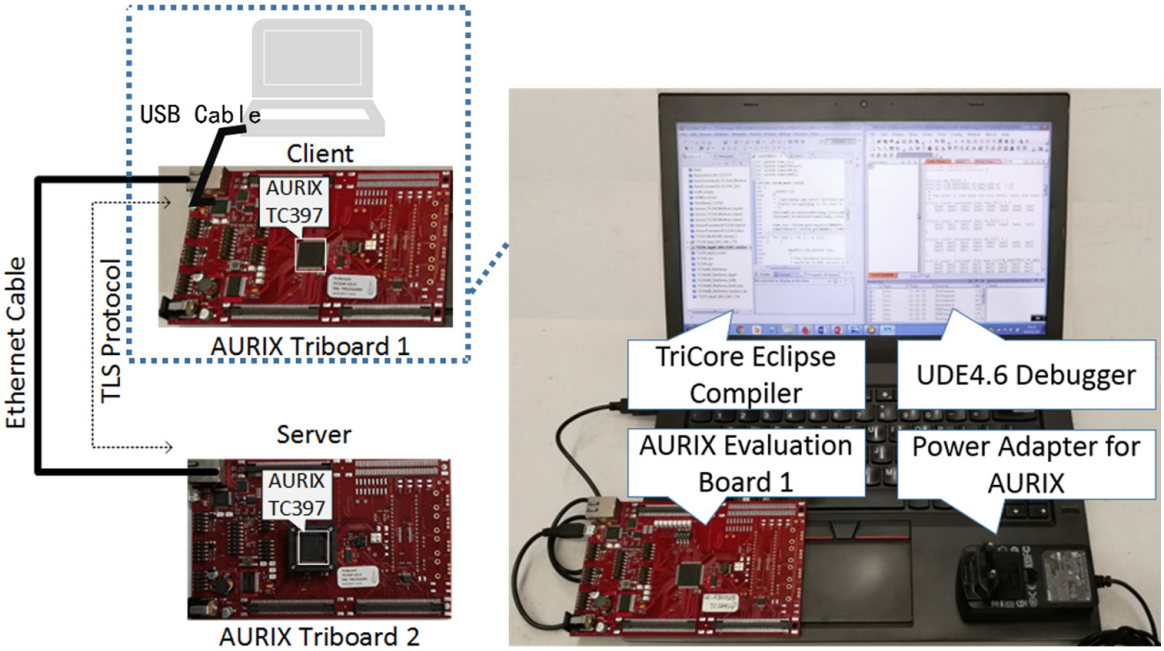

Figure 4. Experiment equipment and software.

The CPU frequency of AURIX TC397 was set to be 300 MHz, and the bandwidth of the Ethernet was 1000 Mb/s. The time and memory overhead of NTRUEncrypt, RSA, and ECDH algorithms running and communicating on the vehicle-mounted microcontroller were measured. The NTRUEncrypt code was derived from the open-source library libntru 0.5 [23], and the RSA and ECDH codes were derived from tls.mbed.org. The code was compiled in Tricore Eclipse, and the results were read in UDE4.6 Debugger or related output files.

4.3. Experiment Results and Analysis

As the in-vehicle communication system is a real-time and embedded system, which has a strict demand for time delay and resource consumption, the execution time and memory occupation of algorithms for session key negotiation were tested based on the above principle and equipment. Former research [24] shows the performance requirements of some in-vehicle applications where Ethernet is implemented. That is, for in-vehicle camera application, the latency should not exceed 45 ms. For in-vehicle audio application, the latency should be less than 150 ms, and for in-vehicle video application, the latency should be no more than 150 ms.

4.3.1. Execution Time of Algorithms

Two sets of data have been acquired and handled. The handling results of the first set of data, shown in Figure 5, present each step's time consumption in the nodes for different session key negotiation algorithms, thus not including the transmission latency. Meanwhile, Figure 6 shows the handling result of the second set of data, namely the whole session key negotiation time, including the transmission latency.

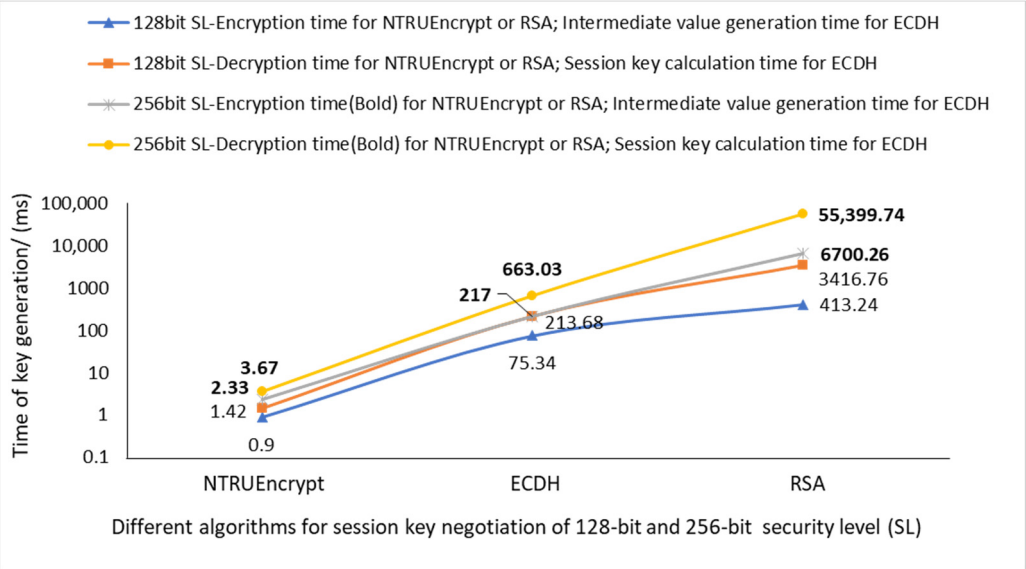

Figure 5. Time consumption comparison of encryption and intermediate value generation, decryption, and session key calculation.

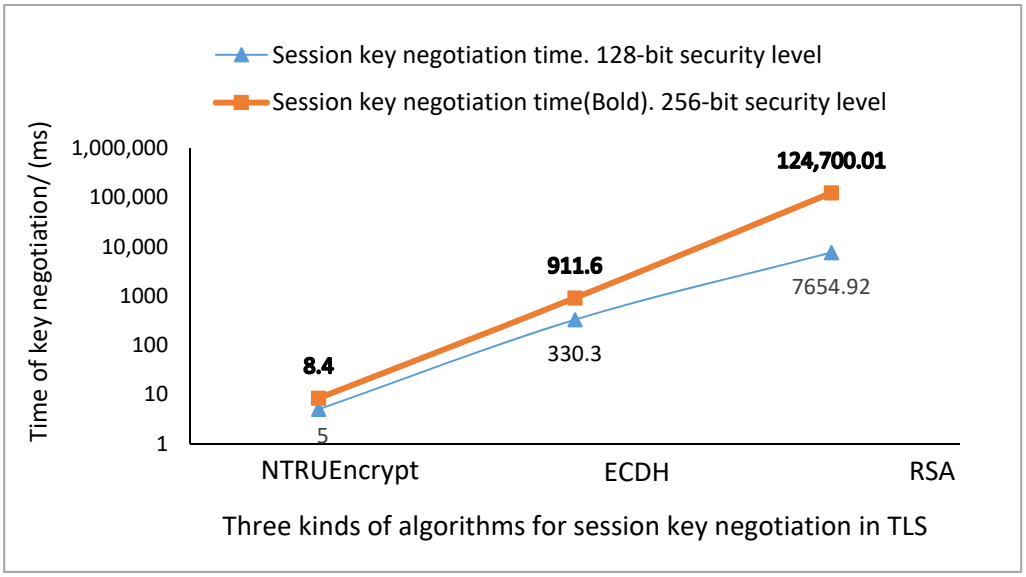

Figure 6. Relationship between key negotiation time and the security strength.

As shown in Figure 5, any key negotiation algorithm has two single steps. For RSA and NTRUEncrypt, the calculation time in the nodes contains encryption time and decryption time. For ECDH, the calculation time in the nodes contains the intermediate value generation time and the time of session key calculation based on the intermediate value.

Figure 5 shows that NTRUEncrypt has the fastest cryptographic operation speed compared to the ECDH and RSA algorithms. Taking the 128-bit security level as an example, the encryption speed of NTRUEncrypt is 83.71 times faster than ECDH's first step and 459.16 times faster than RSA encryption. Additionally, the advantage of decryption speed of NTRUEncrypt is more prominent than RSA decryption or ECDH's second, namely the last step. As to the 256-bit security level, the same conclusion can be drawn.

The second set of data derives from the two AURIXs' communication experiment according to the principle, as presented in Figure 3.

The time of session key negotiation based on six algorithms in the TLS protocol is compared.

It can be concluded from Figure 6 that, on the 128-bit security level, the speed of session key negotiation using NTRUEncrypt is 66.06 times faster than ECDH, and 1530.98 times faster than RSA. The speed advantage of NTRUEncrypt compared to ECDH and RSA is more prominent when the security level becomes higher. It can be concluded that the speed advantage of NTRUEncrypt is more prominent when deploying higher security level cryptographic algorithms in the communication systems.

As 8.4 ms is less than 45 ms which is described in Section 4.3, besides TLS, other secure protocols can also use NTRUEncrypt, which is an enhanced session key negotiation. If the latency requirement is less than 8.4 ms, a possible countermeasure is that the NTRUEncrypt enhanced session key negotiation is only used when the vehicle starts. The session tickets are used when the vehicle is running, which has been described by Zelle [14].

4.3.2. Memory Occupation of Algorithms

The result of memory occupation in Figure 7 is acquired from the microcontroller TC397 in AURIX Triboard 1 as algorithms in both microcontrollers have proximate memory occupation.

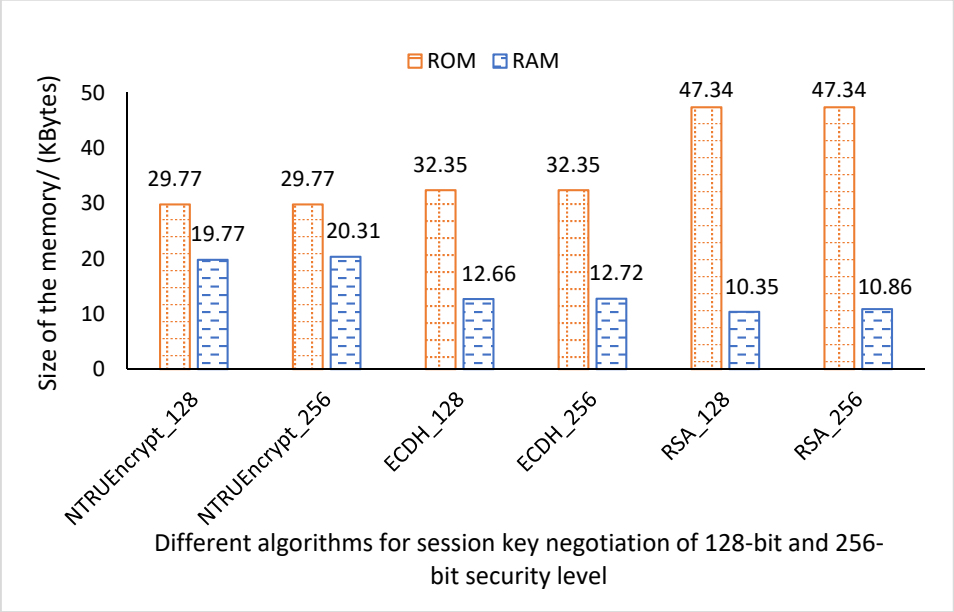

Figure 7. Comparison of memory occupation of different algorithms.

ROM occupation of NTRUEncrypt is the smallest, while the RAM occupation of NTRUEncrypt is the biggest of the three kinds of algorithms. However, what needs to be emphasized is that the ROM and RAM occupation of the three kinds of algorithms are at the same order of magnitude. As there are 16 MB ROM and 6.9 MB RAM in AURIX TC397, the highest ROM and RAM occupancy ratio of the three algorithms is 0.30% and 0.29%.

As the highest ROM and RAM occupation ratio of the algorithms is less than 0.3% of the microcontroller memory, it can be concluded that the memory occupation of the session key negotiation algorithm is not a crucial impact factor for its applicability in an in-vehicle environment.

5. Conclusions

In this paper, the post-quantum enhanced session key negotiation process in the in-vehicle Ethernet was proposed and the negotiation overhead is evaluated for the first time. NTRUEncrypt was chosen as the post-quantum algorithm; meanwhile, ECDH and RSA, as

comparative algorithms, were also implemented in separate processes. The three kinds of algorithms were analyzed in terms of time and memory overhead.

For NTRUEncrypt, compared to ECDH and RSA, besides its property of resisting quantum computers' attacks, three other main conclusions can be drawn in this paper.

1. NTRUEncrypt based session key negotiation for in-vehicle TLS has a prominent speed advantage over ECDH and RSA. The speed advantage of NTRUEncrypt compared to ECDH and RSA is more prominent when the security level becomes higher. On the 128-bit security level, the speed of the session key negotiation using NTRUEncrypt is 66.06 times faster than ECDH, and 1530.98 times faster than RSA. Meanwhile, on the 256-bit security level, the speed of the session key negotiation using NTRUEncrypt is 108.52 times faster than ECDH, and 14,845.24 times faster than RSA.
2. Memory occupation of NTRUEncrypt is at the same order of magnitude compared to that of ECDH and RSA. However, memory occupation is not as crucial an impact factor as the ROM and RAM occupation ratio for a single algorithm is no more than 0.30% and 0.29% of TC397′s ROM and RAM.
3. As TLS can fulfill most performance requirements of the automotive industry, considering the above two conclusions, post-quantum enhanced session key negotiation will probably be widely used for in-vehicle Ethernet communication.

Further research includes the efficiency study of post-quantum algorithms in two functions consisting of identity authentication and session key negotiation in TLS protocol; the study on security mechanisms based on the post-quantum algorithm in other in-vehicle Ethernet protocols such as the network layer-based DoIP and SOME/IP protocol.

Author Contributions: Methodology, Y.Z. and J.Z.; software, Y.L. and J.L.; resources, M.W.; validation, S.L.; writing—original draft, Y.L. and J.Z.; writing—review and editing, Y.L. and J.Z.; supervision, Y.Z and M.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Perspective Study Funding of Nanchang Automotive Institute of Intelligence and New Energy, Tongji University (TPD-TC202110-13) and the APC was funded by TPD-TC202110-13.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors wish to express their many thanks for the support provided by Manager Jianjie Gu and Engineer Chengguo Wang from Shanghai G-Pulse Technology Co., Ltd.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

ECDH	Elliptic Curve Diffie–Hellman
RSA	Rivest–Shamir–Adleman
TLS	Transport Layer Security
ECC	Elliptic Curve Cryptography
LB-PKC	Lattice-Based Public-Key Cryptographic algorithms
NTRU	Number Theory Research Unit
WSN	Wireless Sensor Network
EPON	Ethernet Passive Optical Network
CAN	Controller Area Network
E/E	Electronic and Electrical
DTLS	Datagram Transport Layer Security
ECU	Electronic Control Unit
MPU	Micro Processor Unit
SVP	Shortest Vector Problem
gcd	Greatest Common Divisor
ROM	Read-only Memory

RAM	Random Access Memory
LwIP	lightweight IP
DoIP	Diagnostic over IP
SOME/IP	Scalable Service-Oriented MiddlewarE over IP

Appendix A. Polynomial Ring

NTRU cryptosystem elements such as key, ciphertext, or plaintext are displayed by polynomials.

NTRU operations such as key generation, encryption, and decryption are based on a polynomial ring R, which is a special polynomial set.

$$R = \frac{\mathbb{Z}[x]}{x^N - 1} \tag{A1}$$

$\mathbb{Z}$ is the set of all the polynomials with integer coefficients. The degree of R is $N - 1$, and N is the public parameter in Section 3.1.1.

A polynomial $f \in R$ or $f(x) \in R$ can be noted as a vector:

$$f = \sum_{i=0}^{N-1} f_i x^i = [f_0, f_1, \ldots, f_{N-1}] \tag{A2}$$

Furthermore, if the coefficients of polynomial are limited in the span of a not so big number p or q, the arithmetical operations in NTRU cryptosystems become faster. This can be realized by modulo operation for the coefficients of the polynomial. This kind of polynomial ring is called truncated polynomial ring which is notated as follows

$$R_p = \frac{(\mathbb{Z}/p\mathbb{Z})[x]}{x^N - 1} \tag{A3}$$

$$R_q = \frac{(\mathbb{Z}/q\mathbb{Z})[x]}{x^N - 1} \tag{A4}$$

Appendix B. Ternary Polynomial

$T(a, b)$ is a ternary polynomial, a special polynomial whose coefficients compose of -1, 0 and 1. a means the number of the coefficient which equals 1 is a and b means the number of the coefficient which equals -1 is b. In NTRU cryptosystem, $T(a, b) \in R$ in Equation (A1).

For example, if a polynomial $f(x)$ as Equation (A5) shows can be used in NTRU cryptosystem,

$$f(x) = -1 + x - x^2 + x^3 - x^4 + x^7 - x^8 + x^{10} \tag{A5}$$

Then, $f(x) \in T(4, 4)$ and the parameter N meets $N > 10$.

Appendix C. Operations on *R* and *Rp* or *Rq*

The operations $(+, \otimes)$ on the ring are defined as follows:

$$f + g = (f_0 + g_0) + (f_1 + g_1)x + \ldots + (f_{N-1} + g_{N-1})x^{N-1} \tag{A6}$$

$$f \otimes g = h \tag{A7}$$

where the k-th coefficient

$$h_k = \sum_{i+j \equiv k \bmod N} f_i g_j \tag{A8}$$

The algebraic system formed by $(R, +, \otimes)$ is the basis the NTRU cryptosystem.

Appendix D. The Inverse of a Polynomial

The inverse of a polynomial on the truncated polynomial ring: set an integer $n \geq 2$, for any positive integer q, let R_q notated in Equation (A4) be a truncated polynomial ring modulo q. Take a polynomial g like what's shown in Equation (A2) on this ring

$$g = \sum_{i=0}^{N-1} g_i x^i = [g_0, g_1, \ldots, g_{N-1}] \tag{A9}$$

where each coefficient g_i is an integer and less than q.

If there is another polynomial G which meets that $g \otimes G = 1 \bmod q$, then G is called the inverse of polynomial g on this ring.

References

1. Shor, P.W. Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings of the IEEE 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 20–22 November 1994; pp. 124–134.
2. Zhang, Z.; Chen, C.; Hoffstein, J.; Whyte, W.; Schanck, J.M.; Hulsing, A.; Rijneveld, J.; Schwabe, P.; Danba, O. *NTRUEncrypt. Technical Report*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2019. Available online: https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions (accessed on 22 June 2021).
3. Hoffstein, J.; Pipher, J.; Silverman, J.H. NTRU: A ring-based public key cryptosystem. In *International Algorithmic Number Theory Symposium*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 267–288.
4. Chaudhary, R.; Aujla, G.S.; Kumar, N.; Zeadally, S. Lattice-based public key cryptosystem for internet of things environment: Challenges and solutions. *IEEE Internet Things J.* **2018**, *6*, 4897–4909. [CrossRef]
5. Dierks, T.; Rescorla, E. *The Transport Layer Security (TLS) Protocol, Version 1.2*; RFC 5246: Cincinnati, OH, USA, 2008; pp. 1–104.
6. Nguyen, H.B. *An Overview of the NTRU Cryptographic System*; San Diego State University: San Diego, CA, USA, 2014.
7. Yin, A.; Chen, D.; Ding, Y. An efficient and secure authentication scheme based on NTRU for 10G ethernet passive optical. *Optik* **2014**, *125*, 7207–7210. [CrossRef]
8. Shaheen, S.H.; Yousaf, M.; Jalil, M. A smart card oriented secure electronic voting machine built on NTRU. *Int. Arab J. Inf. Technol.* **2020**, *17*, 386–393. [CrossRef]
9. Sepúlveda, J.; Liu, S.; Mera, J.M.B. Post-quantum enabled cyber physical systems. *IEEE Embed. Syst. Lett.* **2019**, *11*, 106–110. [CrossRef]
10. Bi, B.; Huang, D.; Mi, B.; Deng, Z.; Pan, H.J.W.P.C. Efficient LBS Security-Preserving Based on NTRU Oblivious Transfer. *Wirel. Pers. Commun.* **2019**, *108*, 2663–2674. [CrossRef]
11. Mi, B.; Huang, D.; Wan, S. NTRU implementation of efficient privacy-preserving location-based querying in VANET. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 7823979. [CrossRef]
12. Lin, C.-W.; Sangiovanni-Vincentelli, A. Cyber-Security for the Controller Area Network (CAN) Communication Protocol. In Proceedings of the 2012 International Conference on Cyber Security, Alexandria, VA, USA, 14–16 December 2012; pp. 1–7.
13. Woo, S.; Jo, H.J.; Kim, I.S.; Lee, D.H. A Practical Security Architecture for In-Vehicle CAN-FD. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 2248–2261. [CrossRef]
14. Zelle, D.; Krauß, C.; Strauß, H.; Schmidt, K. On Using TLS to Secure In-Vehicle Networks. In Proceedings of the 12th International Conference on Availability, Reliability and Security, Association for Computing Machinery, Reggio, Calabria, Italy, 29 August–2 September 2017; pp. 1–10.
15. PQC-Forum. Available online: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/Mb5ZKpnO57I/m/S8yaURFYCwAJ (accessed on 20 February 2022).
16. Prest, T.; Fouque, P.-A.; Hoffstein, J.; Kirchner, P.; Lyubashevsky, V.; Pornin, T.; Ricosset, T.; Seiler, G.; Whyte, W.; Zhang, Z. *FALCON. Post-Quantum Cryptography Project of NIST*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2017.
17. Hoffstein, J.; Silverman, J. Optimizations for NTRU. Public-Key Cryptography and Computational Number Theory. In Proceedings of the De Gruyter Proceedings in Mathematics, Vienna, Austria, 18 June–13 July 2001; pp. 77–88.
18. IEEE P1363.1 Draft 10: Draft Standard for Public Key Cryptographic Techniques Based on Hard Problems over Lattices. Available online: https://eprint.iacr.org/eprint-bin/print.pl (accessed on 22 June 2021).
19. Hoffstein, J.; Pipher, J.; Schanck, J.M.; Silverman, J.H.; Whyte, W.; Zhang, Z. Choosing parameters for NTRUEncrypt. In *Cryptographers' Track at the RSA Conference*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 3–18.
20. Paar, C.; Pelzl, J. *Understanding Cryptography: A Textbook for Students and Practitioners*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2009.
21. AURIX Family TC39X. Available online: https://www.infineon.com/cms/en/product/microcontroller/32-bit-tricore-microcontroller/32-bit-tricore-aurix-tc3xx/aurix-family-tc39xxx/ (accessed on 15 November 2020).
22. AURIX Triboard. Available online: https://www.infineon.com/cms/en/product/evaluation-boards/kit_a2g_tc375_5v_trb/ (accessed on 16 November 2020).

23. The NTRU Project. Available online: http://tbuktu.github.io/ntru/ (accessed on 25 May 2020).
24. Lim, H.-T.; Weckemann, K.; Herrscher, D. Performance study of an in-car switched ethernet network without prioritization. In *International Workshop on Communication Technologies for Vehicles*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 165–175.