Implementation of Lattice Theory into the TLS to Ensure Secure Traffic Transmission in IP Networks Based on IP PBX Asterisk

Olga Abramkina[1], Mubarak Yakubova[2], Tansaule Serikov[3], Yenlik Begimbayeva[4], Bakhodyr Yakubov[5]
Department of Cybersecurity, International Information Technology University, Almaty, Kazakhstan[1]
Department of Cybersecurity, Almaty University of Power Engineering and Telecommunications
Name after Gumarbek Daukeev, Almaty, Kazakhstan[2, 4, 5]
Department of Electronics and Telecommunication, S.Seifullin Kazakh AgroTechnical Research University, Astana, Kazakhstan[3]

***Abstract*—This paper presents a novel lattice-based cryptography implementation in the Transport Layer Security (TLS) protocol to enhance the security of traffic transmission in IP networks that use the Asterisk IP PBX platform. Given the growing threat of quantum computing, traditional cryptographic methods are becoming increasingly vulnerable. To address this issue, the study leverages post-quantum cryptography by developing a modified TLS protocol using lattice-based cryptographic algorithms. The performance of the system was evaluated in terms of security, computational efficiency, and real-time communication. The study shows that the proposed lattice-based TLS implementation effectively secures traffic transmission in IP PBX networks, offering a robust solution against both current and future quantum threats.**

Keywords—IP; PBX; Asterisk; TLS; MITM; post-quantum cryptography

I. Introduction

In today's rapidly advancing technological world, the prospect of a quantum computer is becoming increasingly real. Quantum computers, once fully implemented, could potentially break widely used cryptographic algorithms such as RSA and ECC (elliptic curve cryptography) [1-3].

If the intruder's computing capabilities increase by tens, hundreds or thousands of times, this will lead to a sharp need to increase the key length to a critical level, which will become unsuitable for successful operation in real information systems. In addition, if a quantum adversary with enormous computing power appears, there is a possibility of a complete hack of existing cryptosystems by a complete enumeration of keys [4-6]. This problem can be solved by implementing post-quantum cryptography, a relatively new area of cryptography, which is designed to resist quantum computing [7-9]. Unlike asymmetric cryptography, based on conditionally unidirectional mathematical functions, post-quantum cryptography is based on the principles of quantum mechanics and quantum information theory, which guarantee physical unidirectionality. However, there are problems associated with the complexity of implementation and the high cost of equipment [10-12]. With a data transmission channel length of more than 100 km, the transmission speed is significantly reduced (to several bits per second). This fact does not yet allow for the implementation of a full-fledged secure exchange of critical information. In this regard, post-quantum cryptography currently appears more feasible for use in existing systems. The main approaches in post-quantum cryptography include:

1) *Lattice-based cryptography.* This approach is based on mathematical problems involving lattices, such as the shortest vector problem (SVP) and learning with errors (LWE). These problems are considered difficult to solve even for quantum computers, so lattice cryptography has an advantage [13-15].

2) *Code-based cryptography.* This approach is based on the difficulty of decoding a general linear code, a problem that has been studied extensively and is considered difficult for quantum computers [16-18].

3) *Multivariate polynomial cryptography.* This approach is based on the difficulty of solving systems of multivariate quadratic equations over a finite field. The multivariate quadratic (MQ) problem is NP-hard and is considered resistant to quantum attacks [19-21].

4) *Hash-based cryptography.* This approach uses cryptographic hash functions as a basis for constructing secure algorithms, particularly for digital signatures. Hash-based cryptography is widely considered to be secure against quantum attacks, assuming that the underlying hash functions are secure [22-25].

5) *Isogeny-based cryptography.* This approach exploits the mathematical properties of isogenies between elliptic curves. Isogeny-based cryptographic schemes are relatively new and are considered promising due to their resistance to quantum attacks. Small key sizes and robust security based on complex mathematical structures [26-28].

6) *Hyperelliptic curve cryptography* uses the algebraic structures of hyperelliptic curves, which generalize elliptic curves and provide similar security mechanisms at smaller key sizes [29-31]

7) *Symmetric quantum resistance focuses* on making symmetric cryptosystems secure against quantum computing threats. The primary method of improving the resistance is to increase the key length and choose algorithms with greater resistance to Grover-based attacks. For example, using AES-256 instead of AES-128 will provide resistance against

quantum computing attacks at the level of 2128 searches. Quantum attacks on hash functions, such as the Grover-based collision attack, also reduce the complexity of finding a collision from O(2n)) to O(2n/2). Therefore, using hash functions with longer bit lengths (e.g., SHA-512 instead of SHA-256) can be an effective means of protection [32-35].

These approaches form the basis of ongoing research and development in post-quantum cryptography as the world prepares for the potential impact of quantum computing. Each approach has its own strengths and weaknesses, and the future of secure communications will likely involve a combination of these methods.

This paper is a continuation of the research [36] where the publication focused on traditional multi-layered data protection in IP networks based on Asterisk IP PBX using different codecs to reduce latency and improve real-time encryption, while this study proposes the integration of post-quantum lattice-based cryptography into the TLS protocol to protect against threats associated with the development of quantum computers. The [36] solves the existing problems of cryptographic security, and this study extends the research area by proposing protection against future quantum threats, which represents an important step forward in ensuring the security of IP networks.

This article discusses the convergence of lattice theory into the TLS protocol for protecting transmitted traffic in IP networks based on Asterisk PBX. Asterisk is an open source framework for building communications applications such as IP PBX systems, VoIP gateways, and conference servers. Given its widespread use, ensuring the security of communications processed by Asterisk is of paramount importance.

To address these issues, lattice-based cryptography has become a promising approach to post-quantum cryptography, offering strong resistance to quantum attacks.

Over the past decade, the number of published works on this topic, both foreign and domestic, has increased significantly. This fact emphasizes the relevance of this problem and arouses interest in further research in this area.

II. Methodology

In the modern conditions of quantum computing development, traditional cryptographic methods based on the complexity of factorization and discrete logarithm calculation become vulnerable to quantum attacks. Therefore, there is a need to use post-quantum cryptographic algorithms that will ensure security even in the conditions of powerful quantum computing. Lattice-based Cryptography is considered one of the most promising technologies in the field of post-quantum cryptography. The essence of the approach is based on the difficulty of solving problems on lattices, such as the problem of learning with errors (LWE), which remains difficult even for quantum computers [37-40].

These methods were replaced by a lattice algorithm. The LWE problem was proposed as an approximate version of the Shortest Vector Problem (SVP) on lattices. The difficulty of LWE is directly related to solving lattice problems such as the Closest Vector Problem (CVP), since finding a secret vector s in the LWE problem reduces to solving a lattice variation of the CVP problem if the matrix A represents a lattice basis. One can say that LWE translates lattice problems into an algebraic form, where noise (error) is added to a system of linear equations.

The proposed methodology introduces the use of lattice-based cryptographic schemes to provide post-quantum security in the process of key exchange and data encryption. The key exchange scheme is based on LWE or the related Ring-LWE approach, which provides a high level of resistance to both classical and quantum attacks.To integrate such an algorithm into Asterisk, an additional module was created. This module is responsible for generating and verifying keys. Third-party cryptographic libraries such as liboqs (Open Quantum Safe), which supports various post-quantum algorithms, are integrated via AGI. The Asterisk dialplan is then configured to call these modules when establishing a secure SIP session.

Modified TLS (with post-quantum cryptography) has the following implementation stages:

1) The client requests a secure connection.

2) The server sends its certificate based on the post-quantum PQC algorithm.

3) The client and server exchange keys using the PQC algorithm.

4) The symmetric key is used to encrypt data using AES.

5) Messages are protected using HMAC, as in classic TLS, but the key itself was transmitted using PQC.

To generate keys according to the PQC algorithm, a random matrix A and a vector s are used. The matrix A belongs to the set Zq^m: where q is a prime number.

The vector s belongs to the set Zq^m: and is the secret key.

The public key is formed from the matrix A and the error vector e.

The vector b = As + e is calculated, where e is a random noise vector.

b and A make up the public key, and s makes up the secret key.

The encryption process is described by calculating the components of the ciphertext from the vectors u, v, after generating a random vector r from the set Zq^m:

$$u=A^Tr, \tag{1}$$

where, u is the portion of the ciphertext associated with the public key.

$$v=b^Tr+m\cdot\lfloor 2/q\rfloor, \tag{2}$$

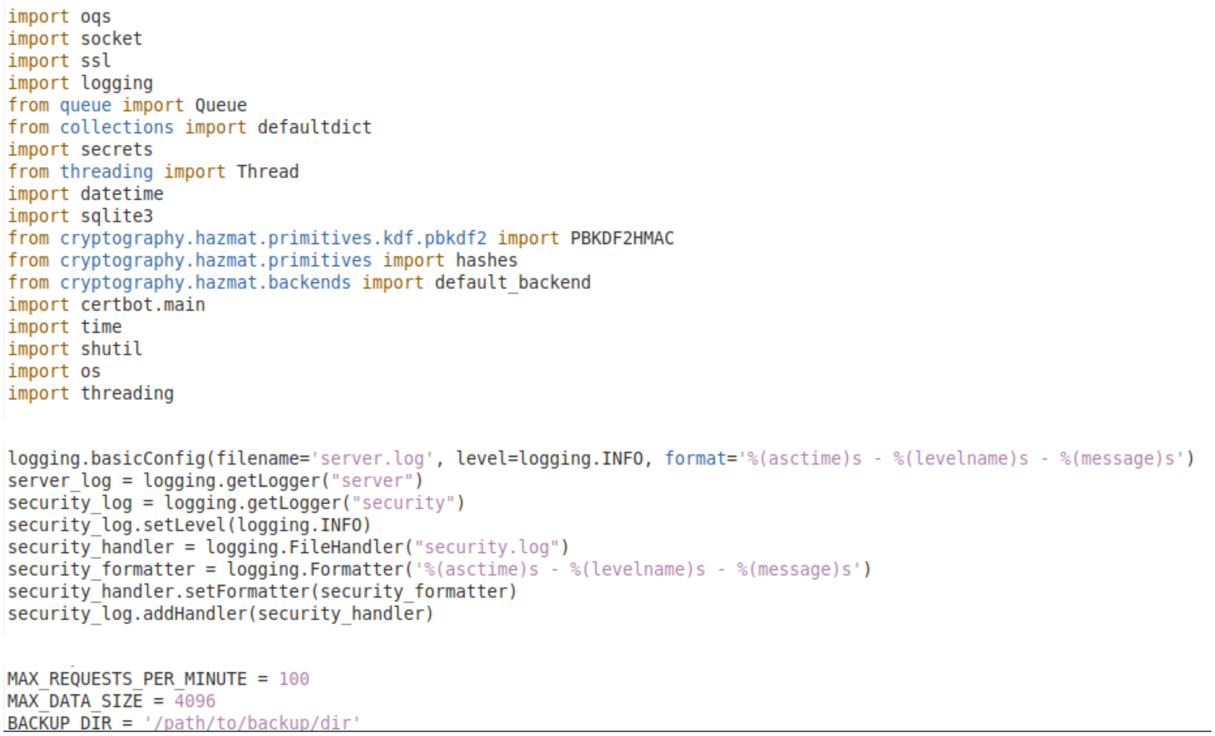
where, v contains the message m itself with added noise.

During decryption, the original message is calculated:

$$m=decode(v-uTs), \tag{3}$$

where, the expression v−uTs allows us to obtain the encrypted message, and then decode it back into the original message.

The module code is shown in Fig. 1.



```
import oqs
import socket
import ssl
import logging
from queue import Queue
from collections import defaultdict
import secrets
from threading import Thread
import datetime
import sqlite3
from cryptography.hazmat.primitives.kdf.pbkdf2 import PBKDF2HMAC
from cryptography.hazmat.primitives import hashes
from cryptography.hazmat.backends import default_backend
import certbot.main
import time
import shutil
import os
import threading


logging.basicConfig(filename='server.log', level=logging.INFO, format='%(asctime)s - %(levelname)s - %(message)s')
server_log = logging.getLogger("server")
security_log = logging.getLogger("security")
security_log.setLevel(logging.INFO)
security_handler = logging.FileHandler("security.log")
security_formatter = logging.Formatter('%(asctime)s - %(levelname)s - %(message)s')
security_handler.setFormatter(security_formatter)
security_log.addHandler(security_handler)


MAX_REQUESTS_PER_MINUTE = 100
MAX_DATA_SIZE = 4096
BACKUP_DIR = '/path/to/backup/dir'
```

Fig. 1. Module code.

The performance and productivity of the developed method were evaluated by conducting a Man-in-the-Middle (MITM) attack, a well-known threat to VoIP systems. In MITM attacks, an attacker intercepts and manipulates real-time data, gaining unauthorized access to voice calls, modifying messages, and retrieving confidential credentials. In the context of VoIP, implementing TLS (Transport Layer Security) effectively protects signaling traffic, significantly reducing the risk of MITM attacks. By integrating a post-quantum key exchange algorithm into TLS, the developed method ensures robust encryption and data integrity, safeguarding communications from both conventional MITM and future quantum-based attacks.

This approach enhances the overall security of VoIP systems against evolving threats.Stages of implementing a MITM attack in a VoIP environment to test TLS:

1) Setting up a test environment includes preparing a virtual machine (VM) with an installed VoIP system based on Asterisk and interaction of virtual clients via SIP.

2) The creation of a MITM attack is based on the use of Ettercap and Bettercap tools, which are capable of intercepting traffic between VoIP clients.

3) Testing without TLS, when protocols transmit messages in clear text, which allows an attacker to easily intercept calls and gain access to data.

4) Implementation *of modified TLS* allows us to evaluate how TLS affects data interception.

5) Analysis of the results shows the results of the MITM attack with and without the use of modified TLS.

III. Result

A. *Installing Asterisk*

Asterisk installed on a virtual machine to emulate a VoIP server was done using Docker.

Docker is an open-source platform that allows you to optimize the management of development, testing and deployment of web applications. Docker is based on the packaging of programs (along with the environment and dependencies) into virtual blocks - containers.

Using Docker for IP PBX Asterisk is an effective way to deploy, manage, and scale an IP telephony system. Asterisk, being one of the most popular platforms for building VoIP (Voice over IP) solutions, can benefit greatly from using containerization via Docker. The PBX AS-IS architecture is shown in Fig. 2.

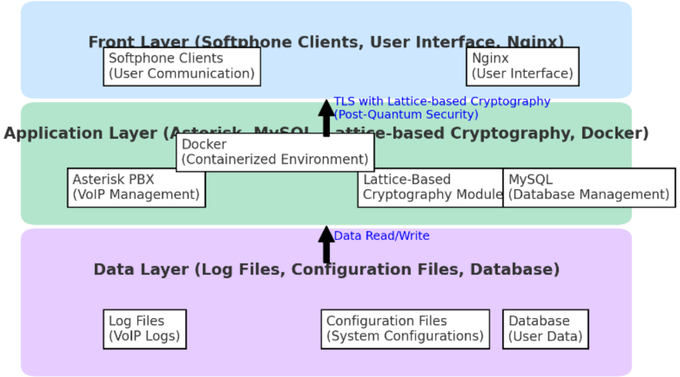


Fig. 2. Architecture of the proposed system.

In application architecture, especially with containerization and technologies like Docker, there are three key layers: Data Layer, Application Layer, and Front Layer. These layers provide separation of duties, simplify support, development, and scaling of the system.

1) Data layer: The data layer is responsible for storing, managing, and processing all the information used in the system. In the context of IP telephony architecture using Asterisk and Docker, this layer is responsible for storing and managing data such as call records, user settings, call sessions, and so on.

The main components of the data layer are:

Databases: Used to store configurations, user data, CDRs (Call Detail Records), logs, session information, and other necessary information.

MySQL, PostgreSQL, MariaDB: Relational databases for storing structured data about users, calls, and other elements of the system.

NoSQL databases (e.g. Redis): Used for caching, storing sessions or other data where high access speed is critical.

File storage: Can be used to record and store conversations, system logs and other files related to IP telephony.

Object Storage (e.g. MinIO or S3): For storing large volumes of audio files, call recordings, data archiving.

File systems (NFS, Ceph): For local or distributed data storage between different containers and system nodes.

Data caching: To optimize access to frequently used data and reduce the load on the main databases.

Redis or Memcached: Used to cache database queries or store session data in real time.

2) Application layer: The application layer is the main functional layer where the business logic of the system is executed. In the case of IP telephony and Asterisk, this is the layer where call processing, SIP request routing, security policy enforcement, etc. take place.

The main components of the application layer are:

Asterisk Server: The main component that handles voice traffic, SIP request routing, session management, and provides telephony functionality. It handles incoming and outgoing calls, controls conferences, organizes IVR systems, and other IP PBX functions.

SIP Proxy (e.g. Kamailio): A component for routing SIP traffic, load balancing between multiple Asterisk servers, and improving system security.

Security Features: Uses TLS to encrypt SIP messages and SRTP to protect voice traffic. These components are integrated into the main application server, providing protection against attacks such as MITM (Man-in-the-Middle).

Application business logic: Applications and services that provide telephone communication functions, such as IVR, call processing automation, integration with CRM systems, and others.

Additional services:

CDR (Call Detail Records): Systems for keeping track of calls and generating reports on the operation of the telephone network.

Call recording functions: For storing call records for security or archiving purposes.

Monitoring and logging: Services for tracking system performance, analyzing logs, and preventing errors.

3) Front layer: The front layer is responsible for user interaction with the system and providing them with interfaces for accessing functionality. These can be web interfaces, mobile applications, client programs for IP telephony, and other user interaction components.

Main components of the front layer:

IP telephony client applications: Programs that users use to work with the IP PBX system.

Softphone clients: For example, Blink, Zoiper, X-Lite are voice communication programs that work via the SIP protocol.

Web clients: Web interfaces for accessing Asterisk functions such as call management, user configuration, statistics analysis, etc.

Web management interfaces: Applications that allow administrators and users to interact with the system via web browsers.

Call control panels: Visual panels for managing calls in real time, viewing line status and interacting with the system.

Mobile applications: Programs for mobile devices that allow users to make and receive calls over a corporate telephone network using SIP.

API: Programming interfaces for integrating Asterisk with external systems (e.g. CRM, ERP). The API allows you to automate some processes and improve the functionality of the system for end users.

RESTful API or AMI (Asterisk Management Interface): Allows you to manage calls,Let's consider the main layers and elements of the architecture:

Installing Docker on Ubuntu distribution consists of 12 steps:

1) Updating the package: sudo apt update

2) Installing the package that is required for the apt package manager to work over HTTPS: sudo apt install apt-transport-https ca-certificates curl software-properties-common

3) Adding the Docker repository GPG key: curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

4) Adding the Docker repository: sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"

5) Updating the package again: sudo apt update

6) Switching to the Docker repository to install it: apt-cache policy docker-ce

Similar information with the Docker version is shown in Fig. 3.

7) Install Docker: sudo apt install docker-ce

8) Check the program's functionality: sudo systemctl status docker

Information that Docker is active is shown in Fig. 4.

9) To use the docker utility, you need to add the user name to the Docker group: sudo usermod -a -G docker user

*10)*Enter the user name: su - user

*11)*Set the user password.

*12)*Check access to Docker images: docker run hello-world

After the information about the successful installation of "Hello from Docker!", you need to install PORTAINER - a graphical panel for managing docker containers.

A data storage for Portainer has been created:

docker volume create portainer_data

The container with Portainer is launched with the command:

docker run -d -p 9000:9000 -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer

After launching in the browser at the server ip address:9000, you need to set the administrator password.

Next, select the location of Docker on the local server (Local) or on a remote one.

The panel is installed, you can launch containers (Fig. 5).

In the control panel, in the "App Template" section, you can find templates with software and run them in containers (Fig. 6).

Portainer is used for comfortable container management.

SIP clients are installed on software SIP phones (softphones).

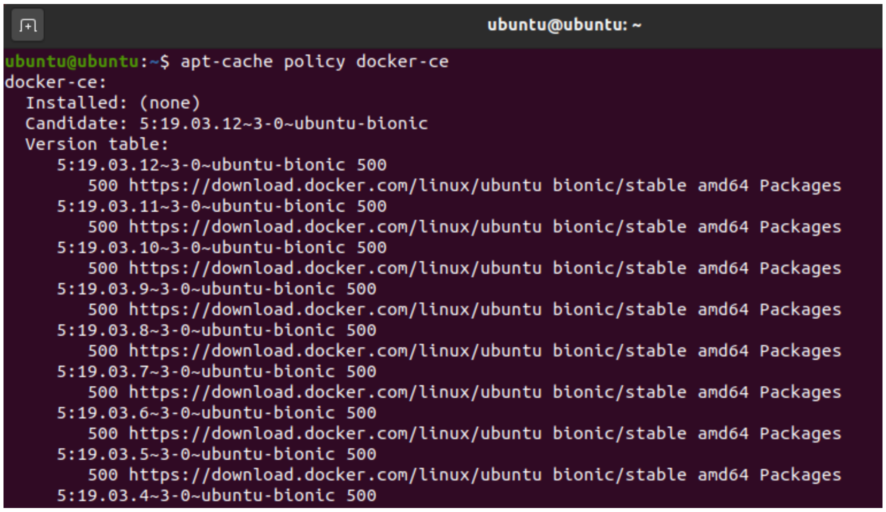
B. Conducting a MITM Attack

Traffic interception is organized by setting up a network bridge and using an ARP spoofing tool (ettercap) so that the attacker's virtual machine can intercept traffic between SIP clients and the VoIP server. The command for an ARP spoofing attack:

ettercap -T -M arp:remote /client-IP-address/ /server-IP-address/

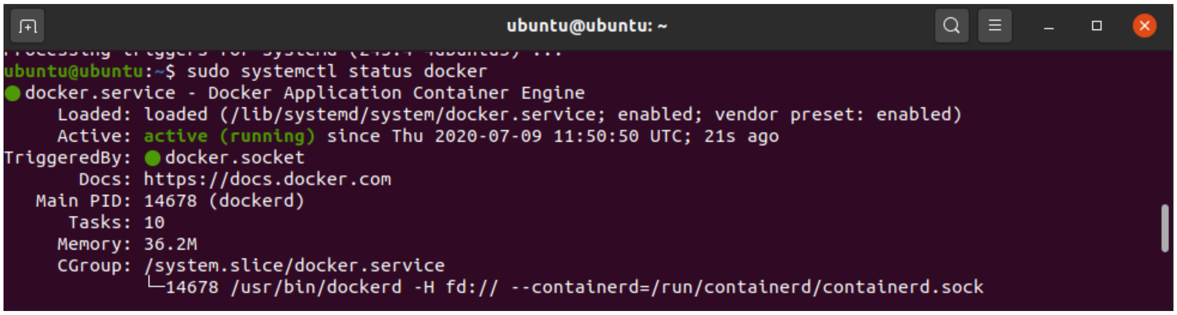
Wireshark was used to analyze traffic. Filters in Wireshark for VoIP:

sip || rtp



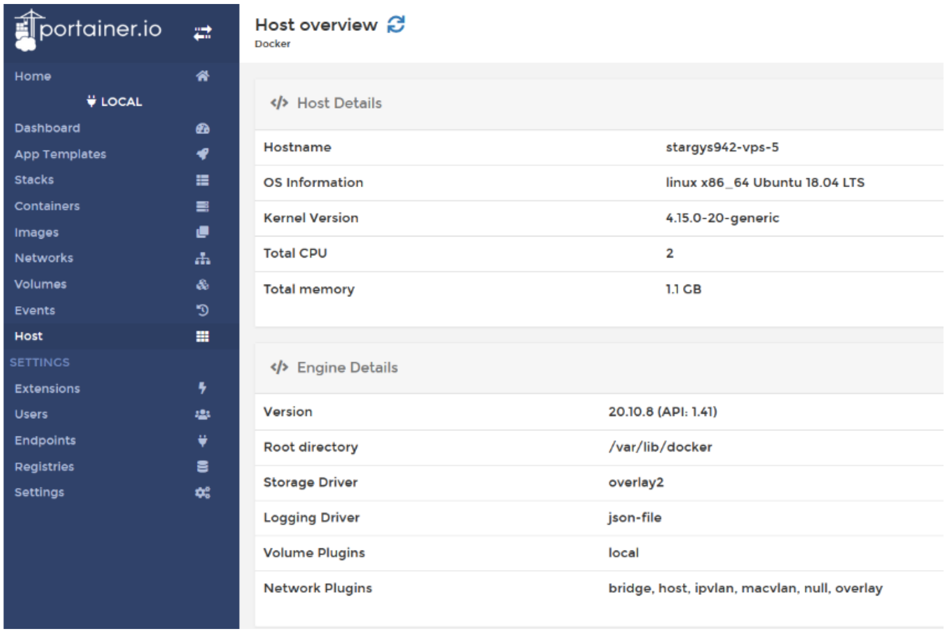
```
ubuntu@ubuntu:~$ apt-cache policy docker-ce
docker-ce:
  Installed: (none)
  Candidate: 5:19.03.12~3-0~ubuntu-bionic
  Version table:
     5:19.03.12~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.11~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.10~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.9~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.8~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.7~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.6~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.5~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.4~3-0~ubuntu-bionic 500
```

Fig. 3. Similar information with docker version.



```
Processing triggers for systemd (245.4-4ubuntu3) ...
ubuntu@ubuntu:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2020-07-09 11:50:50 UTC; 21s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 14678 (dockerd)
      Tasks: 10
     Memory: 36.2M
     CGroup: /system.slice/docker.service
             └─14678 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

Fig. 4. Docker status.



Host overview
Docker

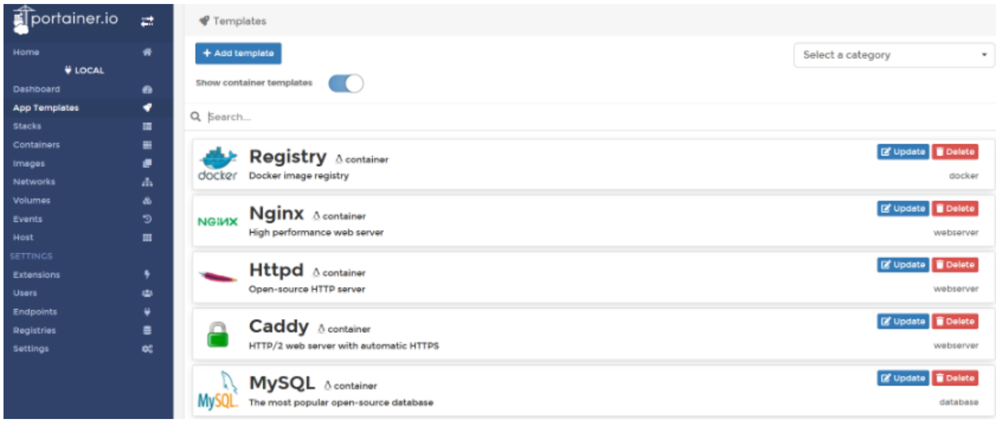
Host Details

Hostname	stargys942-vps-5
OS Information	linux x86_64 Ubuntu 18.04 LTS
Kernel Version	4.15.0-20-generic
Total CPU	2
Total memory	1.1 GB

Engine Details

Version	20.10.8 (API: 1.41)
Root directory	/var/lib/docker
Storage Driver	overlay2
Logging Driver	json-file
Volume Plugins	local
Network Plugins	bridge, host, ipvlan, macvlan, null, overlay

Fig. 5. Running a docker application.



Templates

Show container templates

Registry container — Docker image registry — docker

Nginx container — High performance web server — webserver

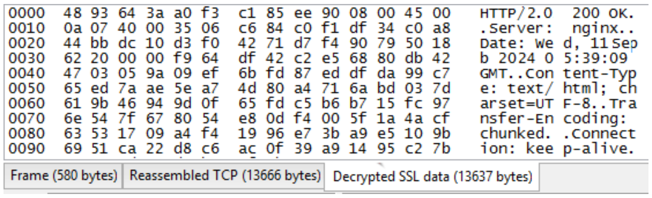
Httpd container — Open-source HTTP server — webserver

Caddy container — HTTP/2 web server with automatic HTTPS — webserver

MySQL container — The most popular open-source database — database

Fig. 6. App template.

In the scenario when we use standard TLS, traffic can be available for analysis and modification; we can see this by switching to the "Decrypted SSL Data" tab. (Fig. 7).

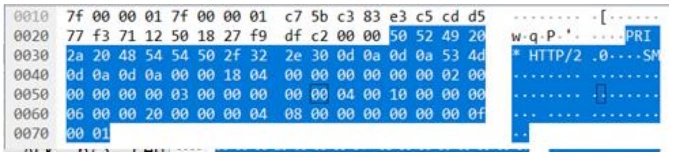


```
0000  48 93 64 3a a0 f3  c1 85 ee 90 08 00 45 00   HTTP/2.0  200 OK.
0010  0a 07 40 00 35 06  c6 84 c0 f1 df 34 c0 a8   .Server:  nginx..
0020  44 bb dc 10 d3 f0  42 71 d7 f4 90 79 50 18   Date: We d, 11Sep
0030  62 20 00 00 f9 64  df 42 c2 e5 68 80 db 42   b 2024 0 5:39:09
0040  47 03 05 9a 09 ef  6b fd 87 ed df da 99 c7   GMT..Con tent-Typ
0050  65 ed 7a ae 5e a7  4d 80 a4 71 6a bd 03 7d   e: text/ html; ch
0060  61 9b 46 94 9d 0f  65 fd c5 b6 b7 15 fc 97   arset=UT F-8..Tra
0070  6e 54 7f 67 80 54  e8 0d f4 00 5f 1a 4a cf   nsfer-En coding:
0080  63 53 17 09 a4 f4  19 96 e7 3b a9 e5 10 9b   chunked. .Connect
0090  69 51 ca 22 d8 c6  ac 0f 39 a9 14 95 c2 7b   ion: kee p-alive.
```

Frame (580 bytes)
Reassembled TCP (13666 bytes)
Decrypted SSL data (13637 bytes)

Fig. 7. Decrypted SSL data.

After setting up modified TLS on the VoIP server, the traffic is encrypted and cannot be analyzed or spoofed (Fig. 8).



```
0010  7f 00 00 01 7f 00 00 01  c7 5b c3 83 e3 c5 cd d5   ········ ·[······
0020  77 f3 71 12 50 18 27 f9  df c2 00 00 50 52 49 20   w·q·P·'· ····PRI
0030  2a 20 48 54 54 50 2f 32  2e 30 0d 0a 0d 0a 53 4d   * HTTP/2 .0····SM
0040  0d 0a 0d 0a 00 00 18 04  00 00 00 00 00 00 02 00   ········ ········
0050  00 00 00 00 03 00 00 00  00 00 04 00 10 00 00 00   ········ ········
0060  06 00 00 20 00 00 00 04  08 00 00 00 00 00 00 0f   ··· ···· ········
0070  00 01                                              ··
```

Fig. 8. Ercrypted SSL data.

IV. Discussion

The integration of lattice-based cryptography into the TLS protocol provides a significant enhancement to the overall security of the IP PBX Asterisk system. Lattice-based algorithms such as Learning With Errors (LWE) offer resistance to quantum computing attacks, which is critical as traditional cryptographic methods like RSA and ECC are vulnerable to quantum threats. By using post-quantum key exchange algorithms, the system ensures that even if a quantum computer were able to break traditional cryptographic schemes, it would not be able to compromise the confidentiality of the communications.

Moreover, the combination of AES for symmetric encryption and HMAC for message integrity ensures a strong layer of defense against eavesdropping and tampering in real-time VoIP communications. This safeguards the system from Man-in-the-Middle (MITM) attacks, where an attacker might attempt to intercept or alter voice and data transmissions. The proposed system's use of lattice-based cryptographic keys during the TLS handshake further secures this process by making key exchanges immune to both classical and quantum-based cracking attempts.

Additionally, the system's architecture, which incorporates Docker for isolated and modular environments, enhances security by reducing the attack surface. Each module operates within a container, minimizing the risk of one compromised module affecting others. The containerized environment also ensures that vulnerabilities in one part of the system do not compromise the whole network, adding a layer of isolation and protection.

However, the system's security also depends on proper management of cryptographic keys and certificates, as failure to securely store or manage these can still leave the system vulnerable to traditional attacks. While lattice-based cryptography provides robust protection against future quantum threats, it is essential to ensure that the deployment and management of the cryptographic infrastructure are secure and well-maintained to prevent other potential security breaches, such as phishing or insider threats.

The proposed system offers several advantages, including quantum-resistant security through lattice-based cryptography, ensuring long-term protection against future quantum attacks. It enhances key exchange protocols using the LWE algorithm, secures real-time communication via post-quantum TLS, and maintains data integrity through HMAC. The system's modular architecture allows for easy integration and scalability. However, it also has some disadvantages, such as increased computational overhead, higher latency in real-time communications, and potential challenges with integration into existing infrastructure. The large key sizes of lattice-based cryptography may also impact storage and bandwidth efficiency. Furthermore, the lack of widespread standardization in post-quantum cryptography adds complexity to its implementation and maintenance.

V. Conclusion

This research makes a significant contribution to the development of cybersecurity systems for IP networks by proposing a practical implementation of post-quantum lattice-based cryptography in the TLS protocol. The main contribution of the research is the integration of lattice cryptographic algorithms into the Asterisk IP PBX infrastructure, which provides reliable protection against threats associated with the development of quantum computers. Unlike existing approaches that focus on the theoretical aspects of post-quantum cryptography, this study demonstrates, the practical application of these methods in real-world conditions using virtual machines and Docker for performance and security testing.

It also contributes to the development and evaluation of a system capable of protecting real-time data transmission with minimal delays, which is especially important for IP telephony and VoIP systems. The implementation of multi-layered data protection using a modified TLS protocol not only increases the security of transmitted information, but also reduces the likelihood of successful man-in-the-middle (MITM) attacks, which is confirmed by experimental data. Furthermore, the work opens up new possibilities for further research and implementation of post-quantum methods in critical communication systems.

Future research in the field of cybersecurity for IP networks and IP PBXs can focus on several directions. First, post-quantum cryptographic algorithms such as lattice systems should be further optimized to reduce their computational complexity and improve their efficiency in real-world conditions, especially for high-load IP telephony. Second, it is worth exploring the possibilities of integrating other post-quantum cryptography methods (based on isogenies or multivariate polynomials) into existing infrastructures to improve security.

Acknowledgment

This article was prepared as part of the government contract as requested by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan on the subject formulated as «Development of a method for improving the security of a telecommunications network based on IP-PBX Asterisk» (project No. AP14871745).

References

[1] V. Mavroeidis, K.Vishi, M.D. Zych, and A, Jøsang, "The Impact of Quantum Computing on Present Cryptography," *Int. J. of Adv. Comput. Sci. and Appl. (IJACSA)*, vol. 9, no.3, 2018. Doi: 10.14569/Ijacsa.2018.090354.

[2] Ch. Majdoubi, S.El Mendili. and Youssef Gahi, "Quantum Cryptology in the Big Data Security Era." *Int. J. of Adv. Comput. Sci. and Appl. (IJACSA)*, vol. 15, no. 7, 2024. Doi:10.14569/Ijacsa.2024.0150761.

[3] Y. Baseri, V. Chouhan, and A.Hafid, "Navigating quantum security risks in networked environments: A comprehensive study of quantum-safe network protocols," *Comput. & Secur.*,vol. 142, 2024, 103883. Doi: 10.1016/j.cose.2024.103883.

[4] A. Zunussov, A. Baikenov, O. Manankova, T. Zheltaev, and T. Zhaksylyk, "Quality of service management in telecommunication network using machine learning technique," *Indonesian J. of Electr. Eng. and Comput. Sci.*, vol. 32, no. 2, pp. 1022–1030, 2023. Doi: 10.11591/ijeecs.v32.i2.pp1022-1030.

[5] S.B. Hegde, A.Jamuar, and R. Kulkarni, "Post Quantum Implications on Private and Public Key Cryptography," In Proceedings of the 2023 International Conference on Smart Systems for applications in Electrical Sciences (ICSSES), Tumakuru, India, 7–8 July 2023; pp. 1–6.

[6] S. V. Singh., D.Kumar, "Enhancing Cyber Security Using Quantum Computing and Artificial Intelligence: A Review," *Int. J. of Adv. Res. in Sc.*, Communication and Technology, pp. 4-11. Doi: 10.48175/ijarsct-18902.

[7] M.Kumar, "Post-quantum cryptography Algorithm's standardization and performance analysis," Array, vol. 15, 2022, 100242, Doi: 10.1016/j.array.2022.100242.

[8] J. Hekkala, M. Muurman, and K. Halunen, "Implementing Post-quantum Cryptography for Developers," *Sn Comput. Sci.*, vol. 4, no. 365, 2023. Doi:10.1007/s42979-023-01724-1.

[9] A. Horpenyuk, I. Opirskyy, and P. Vorobets, "Analysis of Problems and Prospects of Implementation of Post-Quantum Cryptographic Algorithms," *CEUR Workshop Proceedings*, pp.39-49. 2023.

[10] T. Tripathi, A. Awasthi, Sh. Pratap Singh, and A. Chaturvedi, "Post Quantum Cryptography & its Comparison with Classical Cryptography," arXiv:2403.19299v1, 2024.

[11] S. Ricci, P. Dobias, L. Malina, J. Hajny and P. Jedlicka, "Hybrid Keys in Practice: Combining Classical, Quantum and Post-Quantum Cryptography," *in IEEE Access*, vol. 12, pp. 23206-23219, 2024, doi: 10.1109/ACCESS.2024.3364520.

[12] N. Aviram, B. Dowling, I. Komargodski, K. G. Paterson, E. Ronen and E. Yogev, "Practical (post-quantum) key combiners from one-wayness and applications to TLS," *Cryptol. ePrint Arch.*, pp. 1-24, Feb. 2022.

[13] M. R. Albrecht, B. R. Curtis, A. Deo, A. Davidson, R. Player, E. W. Postlethwaite, et al., "Estimate all the LWE NTRU schemes!", Proc. Secur. Cryptogr. Netw. 11th Int. Conf. (SCN), pp. 351-367, Sep. 2018.

[14] V. Dinesh Reddy, P. Ravi, Ashu Abdul, Mahesh Kumar Morampudi and Sriramulu Bojjagani, "Techniques for Solving Shortest Vector Problem" *Int. J. of Adv. Comput. Sci. and Appl. (IJACSA)*, vol.12, no. 5, 2021. Doi: 10.14569/Ijacsa.2021.0120598.

[15] A. Langley, "Real-world measurements of structured-lattices and supersingular isogenies in TLS", 2019

[16] R. Overbeck, and N. Sendrier, "Code-based cryptography," In: Bernstein, D.J., Buchmann, J., Dahmen, E. (eds) Post-Quantum Cryptography. Springer, Berlin, Heidelberg, 2009. Doi: 10.1007/978-3-540-88702-7_4.

[17] D. J. Bernstein, T. Lange, C. Peters, and H. C. A. van Tilborg, "Explicit bounds for generic decoding algorithms for code-based cryptography," In A. Kholosha, E. Rosnes, and M. Parker, editors, Pre-proceedings of WCC 2009, pages 168–180, Bergen, 2009.

[18] K. Preetha Mathew, S. Vasant, and C. Pandu Rangan, "ON PROVABLY SECURE CODE-BASED SIGNATURE AND SIGNCRYPTION SCHEME," *IACR Cryptology ePrint Archive*, 2012:585, 2012.

[19] T. Matsumoto and H. Imai, "Public Quadratic Polynomial-Tuples for Efficient Signature Verification and Message Encryption," *Adv. in Cryptology*, Springer, pp. 419-453, 1988.

[20] J. Ding and A. Petzoldt, "Current State of Multivariate Cryptography," *IEEE Security & Privacy*, vol. 15, no. 4, pp. 28-36, 2017. Doi: 10.1109/MSP.2017.3151328.

[21] C. Tao et al., "Simple Matrix Scheme for Encryption", *Post-Quantum Cryptography* (PQCrypto 13), pp. 231-242, 2013.

[22] A. K, D. S. de Oliveira, J. L´opez, and R, Cabral, "High Performance of Hash-based Signature Schemes" *Int. J. of Adv. Comput. Sci. and Appl. (IJACSA)*, vol. 8, no. 3, 2017. Doi: 10.14569/Ijacsa.2017.080358.

[23] K. Bicakci, K. Ulker, Y.Uzunay, H. T. Şahin, and M. S. Gündoğan, "Quantum-Resistance Meets White-Box Cryptography: How to Implement Hash-Based Signatures against White-Box Attackers?," *IACR Communicat. in Cryptology*, vol. 1, no. 2, Jul 08, 2024. Doi: 10.62056/an59qgxq.

[24] D. A. Cooper, D. C. Apon, Q. H. Dang, M. S. Davidson, M. J. Dworkin, and C. A. Miller, "Recommendation for stateful hash-based signature schemes," *NIST Special Publication*, 800:208, 2020. Doi: 10.6028/NIST.SP.800-208.

[25] L. Li, X. Lu, and K. Wang, "Hash-based signature revisited," *Cybersecurity*, vol. 5, no. 1, pp. 1–26, 2022. Doi: 10.1186/s42400-022-00117-w.

[26] J. Buchmann, E. Dahmen, and M. Szydlo, "Hash-based Digital Signature Schemes. In: Bernstein, D.J., Buchmann, J., Dahmen, E. (eds) Post-Quantum Cryptography. Springer, Berlin, Heidelberg. doi: 10.1007/978-3-540-88702-7_3.

[27] L. De Feo, "Mathematics of Isogeny Based Cryptography," arXiv:1711.04062, 2017. Doi: 10.48550/arXiv.1711.04062.

[28] M. Campagna et al., "Supersingular isogeny key encapsulation," ed, 2019.

[29] F.Tellez, and J. Ortiz, "Comparing AI Algorithms for Optimizing Elliptic Curve Cryptography Parameters in e-Commerce Integrations: A Pre-Quantum Analysis" *Int. J. of Adv. Comput. Sci. and Appl. (IJACSA)*, vol. 15, no. 6, 2024. Doi: 10.14569/Ijacsa.2024.01506153.

[30] V.Rao M. and S. Malladi, "Secure Energy Efficient Attack Resilient Routing Technique for Zone based Wireless Sensor Network" *Int. J. of Adv. Comput. Sci. and Appl. (IJACSA)*, vol. 11, no. 12, 2020. Doi: 10.14569/Ijacsa.2020.0111267.

[31] H. Abroshan, "A Hybrid Encryption Solution to Improve Cloud Computing Security using Symmetric and Asymmetric Cryptography Algorithms" *Int. J. of Adv. Comput. Sci. and Appl. (IJACSA)*, vol. 12, no. 6, 2021. Doi: 10.14569/IJACSA.2021.0120604.

[32] C. V. Manjushree, and A. N. Nandakumar, "A Hybrid Double Encryption Approach for Enhanced Cloud Data Security in Post-Quantum Cryptography," *Int. J. of Adv. Comput. Sci. and Appl. (IJACSA)*, vol. 14, no. 12, 2023. Doi: 10.14569/IJACSA.2023.0141225.

[33] G. Omar, A. and Sh. Kamal, "A Survey on Cryptography Algorithms," *Int. J.of Sci. and Res. Public.* (IJSRP), 2018. Doi:10.29322/IJSRP.8.7.2018.P7978.

[34] M. Yakubova, O. Manankova, A. Mukasheva, A. Baikenov, and T. Serikov, "The Development of a Secure Internet Protocol (IP) Network Based on Asterisk Private Branch Exchange (PBX)," *Appl. Sci.* (Switzerland), vol. 13, no. 19, 2023. Doi: 10.3390/app131910712.

[35] O.A. Manankova, M.Z. Yakubova, M.A. Rakhmatullaev, and A.S.Baikenov, "Simulation of the Rainbow Attack on the SHA-256 Hash function," *J. of Theoret. and Appl. Inf. Tech.*, vol. 101, no. 4, pp. 1594–1603, 2023.

[36] M.Yakubova, T. Serikov, O. Manankova, "Development and Research of a Method for MultiLevel Protection of Transmitted Information in IP Networks Based on Asterisk IP PBX Using Various Codecs," *Int. J. of Adv. Comput. Sci. and Appl. (IJACSA)*, vol. 15, no. 7, pp. 724–731, 2024. Doi: 10.14569/IJACSA.2024.0150771.

[37] M. A. Al-Shabi, "A Survey on Symmetric and Asymmetric Cryptography Algorithms in information Security," *Int. J.of Sci. and Res. Public.*, vol. 9, no. 3, pp.576-589, 2019. Doi: 10.29322/IJSRP.9.03.2019.p8779.

[38] V. Ganeshkar, and M.Kulkarni, "QUANTUM CRYPTOGRAPHY FOR A SECURE COMMUNICATION," *Int. J. of Res. In Comput. Appl. and Inf. Techn.*, vol. 7, no. 1, January-June 2024, pp. 17-29, Article ID: IJRCAIT_07_01_003.

[39] F. Opiłka, M. Niemiec, M. Gagliardi, and M. A. Kourtis, "Performance Analysis of Post-Quantum Cryptography Algorithms for Digital Signature," *Appl. Sci.*, vol.14, no. 12, 2024; Doi: 10.3390/app14124994.

[40] P. Shrivastava, K.K. Soni, and A. Rasool, "Evolution of Quantum Computing Based on Grover's Search Algorithm," In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, pp. 1–6, 6–8 July 2019.