MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE REPUBLIC OF

KAZAKHSTAN

INTERNATIONAL INFORMATION TECHNOLOGY UNIVERSITY

# FACULTY OF COMPUTER TECHNOLOGY AND CYBERSECURITY

**Nuratov A.**

**Ergengali E.**

**Atabaev I.**

## **Development of key exchange application based on homomorphic encryption in IP telephony**

**DIPLOMA PROJECT**

## **Educational program 6B06303 – Network Security**

Almaty 2025

MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE REPUBLIC OF

KAZAKHSTAN

INTERNATIONAL INFORMATION TECHNOLOGY UNIVERSITY

# DEPARTMENT OF CYBERSECURITY

## **Approved**

Head of Department

Cand. tech. sc, assoc, professor

S.T. Amanzholova

«\_\_\_» \_\_\_\_\_\_\_\_\_\_2025

**DIPLOMA PROJECT**

## **Development of key exchange application based on homomorphic encryption in IP telephony**

### Educational program 6B06303 – Network Security

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Done by: | Nuratov A.  «\_\_\_» \_\_\_\_\_2025 |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (signature) |
|  | Ergengali E.  «\_\_\_» \_\_\_\_\_2025 |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (signature) |
|  | Atabaev I.  «\_\_\_» \_\_\_\_\_2025 |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (signature) |
| Research advisor: | «\_\_\_» \_\_\_\_\_2025 |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (signature) |
| Reviewer: | «\_\_\_» \_\_\_\_\_2025            Almaty 2025 |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (signature) |

International Information Technology University Faculty of Computer Technology and Cybersecurity Department of Cybersecurity Educational program 6B06303 – Network Security

### Diploma Project

Students

**Nuratov A., Ergengali E., Atabaev I.**

Diploma project topic

**Development of a key exchange application based on homomorphic encryption in IP telephony**.

Approved by IITU order №\_\_\_ dated «\_\_»\_\_\_\_2025

Diploma project submission date «\_\_»\_\_\_\_2025

Diploma project initial data

Details of computations and explanations (list of issues due to be addressed)

CD containing the digital version of diploma project and attachments

1. The source code and explanatory note;
2. Project presentation;
3. Electronic version of diploma project.

Consultations on Diploma Project (with related Project Chapters named)

|  |  |  |  |
| --- | --- | --- | --- |
| Consultant | Name | Signature, date | |
| Assignment given | Assignment received |
| Consultant on  Economic effectiveness of the project |  |  |  |
| English language consultant |  |  |  |
| Compliance monitor |  |  |  |

Date «\_\_» \_\_\_\_\_\_\_\_2025

|  |  |  |
| --- | --- | --- |
| Research advisor | «\_\_\_» \_\_\_\_\_2025 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (signature) |
| Assignment received by | Nuratov A.  «\_\_\_» \_\_\_\_\_2025 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (signature) |
|  | Ergengali E.  «\_\_\_» \_\_\_\_\_2025 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (signature) |
|  | Atabaev I.  «\_\_\_» \_\_\_\_\_2025 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (signature) |

Diploma Project Writing Schedule

**Nuratov A., Ergengali E., Atabaev I.**

Title: : Development of a key exchange application based on homomorphic encryption in IP telephony.

|  |  |  |
| --- | --- | --- |
| № | Assignment | Submission date |
| 1. | Creation of the graduation project writing schedule | November |
| 2. | Collection, study, processing, analyzing, and generalizing data | November – December |
| 3. | Drafting and submission to the Research advisor (Introduction, Chapter 1, Chapter 2, Chapter 3, Chapter 4, Conclusion) | January – February |
| 4. | Pre-defence | February |
| 5. | Submission of the chapter «Economic effectiveness of the project» to the consultant | February – March |
| 6. | Revision of the graduation project with due consideration of the advisor’s comments | March – April |
| 7. | Submission of the completed diploma project to the Research advisor | April |
| 8. | Submission of the completed diploma project to the English language consultant | April |
| 9. | Submission of the diploma project to the compliance monitor | April – May |
| 10. | Submission of the diploma project for the plagiarism check-up | May |
| 11. | Submission to the reviewer for approval | May |
| 12. | Diploma project defense | May – June |

Student: Nuratov A. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (signature)

Student: Ergengali E. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(signature)

Student: Atabaev I. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(signature)

Research advisor: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(signature)

Date «\_\_\_» \_\_\_\_\_\_2025

# АҢДАТПА

**Жоба атауы:** Гомомофтық шифрлау негізінде IP-телефонияда кілт алмасу қосымшасын əзірлеу

Бұл жоба IP телефониядағы байланыс қауіпсіздігін жақсарту үшін гомоморфты шифрлауды пайдаланатын кілт алмасу қосымшасын әзірлеуге және енгізуге арналған. Voice over IP (VoIP) және бейнеконференция технологияларына артып келе жатқан сеніммен осы платформалар арқылы берілетін құпия деректерді қорғау маңызды мәселеге айналды. Дәстүрлі кілт алмасу хаттамалары, олардың тиімділігіне қарамастан, жиі әртүрлі киберқауіптерге осал болып табылады, бұл неғұрлым қауіпсіз баламаларды іздеуді талап етеді. Жоба ZRTP және TLS-SRTP сияқты қолданыстағы кілт алмасу хаттамаларын жан-жақты талдаудан, олардың қауіпсіздік, өнімділік және ыңғайлылық тұрғысынан күшті және әлсіз жақтарын бағалаудан басталады. Содан кейін ол гомоморфты шифрлаудың принциптері мен артықшылықтарын зерттейді, бұл шифрланған деректерде шифрды ашпай-ақ есептеулерді орындауға мүмкіндік береді, бұл деректер қауіпсіздігін айтарлықтай жақсартады. Осы талдау негізінде операциялық тиімділікті сақтай отырып, қауіпсіз байланыстарды қамтамасыз ететін, кілт алмасу процесіне гомоморфты шифрлауды біріктіретін қосымшаның прототипі әзірленді. Қолданба кідіріс, есептеу жүктемесі және өткізу қабілеттілігі талаптары сияқты негізгі өнімділік көрсеткіштеріне назар аудара отырып, нақты әлемдегі сценарийлердегі өнімділігін бағалау үшін әртүрлі жағдайларда кеңінен сыналады.

Сайып келгенде, бұл жоба IP телефонияда кілт алмасудың сенімді шешімін ұсына отырып, қауіпсіз байланыс саласына үлес қосуды көздейді. Күтілетін нәтижелерге қолданбаның функционалды прототипі, эмпирикалық өнімділік деректері және болашақ енгізу және зерттеу бағыттары бойынша практикалық ұсыныстар кіреді. Бұл жұмыс қазіргі қауіпсіздік мәселелерін қарастырып қана қоймайды, сонымен қатар коммуникациялық технологияларға озық криптографиялық әдістерді біріктіру үшін прецедент орнатуға бағытталған.

АННОТАЦИЯ

**Название проекта:** Разработка приложения для обмена ключами на основе гомоморфного шифрования в IP-телефонии

Данный проект посвящен разработке и внедрению приложения для обмена ключами, использующего гомоморфное шифрование для повышения безопасности коммуникаций в IP-телефонии. С ростом зависимости от технологий передачи голоса по IP (VoIP) и видеоконференций защита конфиденциальных данных, передаваемых по этим платформам, стала критически важной задачей. Традиционные протоколы обмена ключами, несмотря на свою эффективность, часто оказываются уязвимыми перед различными киберугрозами, что требует поиска более безопасных альтернатив.

Проект начинается со всестороннего анализа существующих протоколов обмена ключами, таких как ZRTP и TLS-SRTP, оценки их сильных и слабых сторон с точки зрения безопасности, производительности и удобства использования. Затем исследуются принципы и преимущества гомоморфного шифрования, которое позволяет выполнять вычисления над зашифрованными данными без их расшифровки, что значительно повышает безопасность данных.

На основе этого анализа разрабатывается прототип приложения, которое интегрирует гомоморфное шифрование в процесс обмена ключами, обеспечивая безопасную связь при сохранении эффективности работы. Приложение тщательно тестируется в различных условиях, чтобы оценить его эффективность в реальных сценариях, уделяя особое внимание ключевым показателям производительности, таким как задержка, вычислительная нагрузка и требования к пропускной способности.

В конечном итоге данный проект призван внести вклад в область безопасных коммуникаций, предоставив надежное решение для обмена ключами в IP-телефонии. Ожидаемые результаты включают в себя функциональный прототип приложения, эмпирические данные о производительности, а также практические рекомендации по будущей реализации и направлениям исследований. Эта работа не только решает текущие проблемы безопасности, но и призвана создать прецедент для интеграции передовых криптографических методов в коммуникационные технологии.

ABSTRACT

**Project Title**: Development of a key exchange application based on homomorphic encryption in IP telephony.

This project focuses on the design and implementation of a key exchange application that leverages homomorphic encryption to enhance the security of communications in IP telephony. With the increasing reliance on voice over IP (VoIP) and video conferencing technologies, the protection of sensitive data transmitted over these platforms has become a critical concern. Traditional key exchange protocols, while effective, often exhibit vulnerabilities to various cyber threats, necessitating the exploration of more secure alternatives.

The project begins with a comprehensive analysis of existing key exchange protocols, such as ZRTP and TLS-SRTP, evaluating their strengths and weaknesses in terms of security, performance, and usability. It subsequently investigates the principles and advantages of homomorphic encryption, which allows for computations on encrypted data without decryption, thereby significantly enhancing data security.

Following this analysis, a prototype application is developed that integrates homomorphic encryption into the key exchange process, ensuring secure communication while maintaining performance efficiency. The application is rigorously tested under varying conditions to assess its effectiveness in real-world scenarios, focusing on key performance metrics such as latency, computational load, and bandwidth requirements.

Ultimately, this project aims to contribute to the field of secure communications by providing a robust solution for key exchange in IP telephony. The anticipated outcomes include a functional application prototype, empirical performance data, and practical recommendations for future implementation and research directions. This work not only addresses current security challenges but also aims to set a precedent for integrating advanced cryptographic techniques into communication technologies.

**CONTENTS**

INTRODUCTION

1. ANALYTICAL PART  
   1.1 Market Analysis  
   1.2 Comparison of Existing Solutions  
   1.3 Key Product Features  
   1.4 Project Description
2. THEORETICAL PART  
   2.1 General architecture of the project  
   2.2 Structural description of the security system.  
   2.3 ER- diagram of the database

**List of Terms and Abbreviations**

1. **VoIP** – Voice over Internet Protocol
2. **RSA** - Rivest–Shamir–Adleman
3. **IP** - Internet Protocols
4. **SME** - Small and medium-sized enterprise
5. **HIPAA**  - Health Insurance Portability and Accountability Act
6. **GDPR** - General Data Protection Regulation
7. **ZKP** - Zero-knowledge proofs
8. **PFS** - Personal Finance Society
9. **ECC** - Error correction code
10. **TLS** - Transport Layer Security

INTRODUCTION

In the modern digital era, communication technologies are rapidly evolving, with voice communication shifting from traditional telephony to Voice over Internet Protocol (VoIP) systems. This transition has significantly improved accessibility, cost-efficiency, and communication quality. However, it has also introduced new cybersecurity challenges. VoIP communications are increasingly targeted by cyber threats such as data breaches, eavesdropping, and man-in-the-middle (MITM) attacks, making secure key exchange a critical aspect of VoIP security.

To ensure confidentiality, VoIP communications rely on encryption mechanisms that protect transmitted data. However, encryption is only as secure as the key exchange protocol that establishes shared encryption keys between parties. Traditional key exchange protocols, such as Diffie-Hellman (DH), RSA, and Elliptic Curve Diffie-Hellman (ECDH), provide strong security but are increasingly vulnerable to attacks, particularly with advancements in computing power and the emergence of quantum computing. These concerns highlight the need for more resilient key exchange mechanisms in VoIP systems.

Asterisk, an open-source platform for implementing IP telephony, provides a flexible and scalable foundation for VoIP communications. It supports various security mechanisms but primarily relies on conventional cryptographic methods for key exchange. Given the increasing security risks, there is a growing demand for innovative solutions that enhance encryption key protection without compromising real-time performance requirements.

Homomorphic encryption presents a promising solution in this context. Unlike traditional encryption methods, homomorphic encryption allows computations to be performed directly on encrypted data without requiring decryption. This property can enhance security in various cryptographic applications. In this study, however, homomorphic encryption is not used for computation but rather to protect the transmission of public keys. Specifically, Paillier homomorphic encryption is employed to encrypt public keys exchanged during key agreement, preventing their exposure to potential attackers. This approach addresses the vulnerability of plaintext public key transmission in traditional key exchange protocols, offering an additional layer of security without significantly increasing computational overhead.

The increasing demand for secure communication in VoIP systems makes this research particularly relevant. VoIP is widely used in enterprises, government institutions, and personal communications, making it an attractive target for cybercriminals. While traditional encryption methods remain effective, their limitations necessitate new approaches to security. The integration of homomorphic encryption into key exchange mechanisms aligns with the global trend of enhancing privacy and compliance in digital communication systems.

Thus, the goal of this research is to develop a key exchange protocol for Asterisk-based VoIP communication that utilizes Paillier homomorphic encryption for securing public key transmission while employing ECDH for key agreement and SRTP for secure voice encryption. The primary objective is to enhance the security of VoIP key exchange without revealing confidential information at any stage of the process.

The object of this study is the key exchange process in IP telephony systems based on Asterisk. The subject of the study is the application of homomorphic encryption techniques to develop a secure key exchange protocol within the Asterisk platform.

The novelty of this research lies in the integration of homomorphic encryption into Asterisk’s key exchange process—an area traditionally reliant on conventional cryptographic methods. Unlike standard key exchange protocols, where public keys are transmitted in plaintext and can be intercepted, this study proposes a method where Paillier encryption protects the transmission of public keys while maintaining efficiency.

Additionally, this research explores the practical feasibility of homomorphic encryption in real-world VoIP applications, analyzing key challenges such as computational overhead, latency, and compatibility with existing VoIP infrastructure. The proof-of-concept implementation in Asterisk demonstrates the viability of the proposed approach and lays the groundwork for further advancements in secure communication technologies.

To achieve its main goal, this study conducts a thorough analysis of existing key exchange protocols, identifying their security and performance limitations. It then examines the principles and properties of homomorphic encryption, with a specific focus on its applicability to key exchange. Based on these findings, a secure key exchange protocol is developed, integrating Paillier encryption for public key protection, ECDH for key agreement, and SRTP for call encryption, ensuring end-to-end security in VoIP communications.

**Literature Review**

Today IP-telephony systems require secure data protection in the context of key exchange. Existing traditional cryptographic protocols such as Diffie-Hellman and TLS-SRTP provide basic protection but unfortunately remain vulnerable to man-in-the-middle (MITM) and replay attacks. Faced with this problem scientists and engineers are considering an alternative in the form of homomorphic encryption. According to Akar et al. [1], homomorphic encryption schemes are classified into partially homomorphic encryption, somewhat homomorphic encryption and fully homomorphic encryption. Homomorphic encryption perform computations on encrypted data without the need for prior decryption which raises and provides a level of security[1].

Homomorphic encryption is subdivided into several types depending on the number of operations supported. This encryption allows only one arithmetic operation, addition or multiplication. Homomorphic encryption allows a limited number of operations per session and fully homomorphic encryption allows unlimited arithmetic calculations. Research on key exchange protocols like Diffie-Hellman protocol and authenticated variants provide secure communication over insecure channels as analysed by Mishra and Kar [2]. Encryption methods have been performed using various mathematical frameworks including lattice based cryptography, bilinear pairings and integer factorisation. And also a study by Warhovski and Shokurova [3] highlights the algebraic structures supporting homomorphic encryption operations and their computational complexity.But there is a disadvantage of this scheme is its high complexity which makes it inefficient for use in VoIP systems [3].

The main challenge of implementing fully homomorphic encryption in IP telephony at the moment is the high computational load affecting the latency of data transmission. Studies on this topic suggest optimisation of fully homomorphic encryption aim to reduce data processing time and reduce key size. The study by Alanavar et al. [4] studies partially homomorphic encryption for privacy-preserving set-based evaluation. In addition, computational acceleration has been considered in particular . Advances in fully homomorphic encryption (FHE) [5] have extended applications in secure multi-party computation, making it applicable to encrypted key negotiation. Other research focuses on authentication techniques in key exchange protocols. Homomorphic encryption, as defined in Halevy [6], is a cryptographic scheme that allows computation over ciphertexts while maintaining encryption security enhancing protection against MITM attacks. There are also mathematical models supporting homomorphic computation example as cryptosystem and elliptic curve cryptography which are considered as promising directions in the field of homomorphic encryption.

Another study by Pratibha Chaudhary et. all [7] investigates that current FHE schemes provide a high level of security but their performance is limited by high computational cost. For computation, the use of partially homomorphic encryption is investigated to find the golden mean between security and computational burden. In another study, researcher Jing-Li Han et al. [8] investigates the implementation of FHE library on microcontrollers, extending its applicability to mobile and IoT devices. The use of partial and full homomorphic encryption has various advantages such as real-time voice data transmission, demonstrating that homomorphic encryption is suitable for tasks where a high level of privacy is required in processing. There are no available FHE libraries that implement all arithmetic operations, and this requires further research as stated in the paper by Janerke Temirbekova et al. [9].

There are certain directions to make homomorphic encryption more suitable for inclusion in VoIP systems. A study in [10] shows that the combination of multi-key FHE (MKFHE) and Identity Based Encryption (IBE) reduces the overhead of public key management and improves security in multi-user environments. But computational complexity is still an issue. The paper [11] shows that Paye encryption is efficient in its homomorphic properties and hence can be a suitable candidate for application in scenarios where additive homomorphic encryption is required. The paper [12] also demonstrates the impact of key size on encryption performance and how efficient key management strategies are required for VoIP calls. More efficient implementations of the Gentry scheme are also proposed as one solution for IP telephony services [13]. In addition, a fully homomorphic floating-point encryption method is being investigated to improve computational accuracy and avoid round-off errors in data computation [14].

Although tremendous progress has been made in homomorphic cryptography, there is still much work to be done in this direction. Perhaps the most pressing work remains, however, its high computational complexity, which prevents its use in VoIP networks where low latency is required [15]. As yet, there is no hybrid solution that can combine fully homomorphic cryptography with traditional data protection techniques to reduce CPU overhead [16]. There is another challenge in the form of lack of integration of homomorphic encryption into existing VoIP protocols such as TLS-SRTP and ZRTP [17]. The trade-off between security and computational performance is also being investigated [18]. Homomorphic encryption has made tremendous progress, but there are challenges that need to be addressed. Future work should focus on developing new hybrid encryption schemes that can utilise the functionality of fully homomorphic and traditional data protection schemes [19].

Another important area is the integration of homomorphic encryption with traditional VoIP protocols and the development of adaptive security models that support dynamic security customisation according to system needs [20]. Homomorphic encryption is an area with high potential for securing key exchange in IP telephony, supporting high security without decrypting data during processing. But its application remains limited due to its extremely high computational complexity, and further research is needed in the areas of algorithm optimisation, use of hardware accelerators and deployment of hybrid encryption algorithms [21]. Despite these challenges, homomorphic encryption holds tremendous promise and may become a fundamental component of VoIP security in the future [22]. Lee and Shin [23], Yaji, Banger, and Nilima [24], and Sinha, Majumder, and Ghosh [25] mention future improvements to fully homomorphic floating-point encryption, which are expected to be more accurate and less computationally expensive.

**CHAPTER 1.**

**1.1 Market Analysis**

Modern IP telephony is evolving, but cybersecurity threats are growing along with it. One of the main problems is the protection of cryptographic keys that are used in encrypted communications. Traditional methods of key encryption such as RSA and Diffie-Hellman are becoming obsolete, gradually giving way to modern methods.

In this analysis we examine the main market segments for which the security of IP communications is a critical criterion. We investigate which companies and individuals are interested in adopting homomorphic encryption for secure key exchange and what benefits it offers them.

These include large corporations, financial institutions, healthcare, and government agencies dealing in sensitive information. Many of these have gradually started migrating to IP telephony in a view to modernizing their communications infrastructure and cutting down on the costs. This transition opens a growing need for security and strict regulatory standards. This makes it very important to ensure data security and privacy of information traversing their networks. This shifts the balance in their favor by enhancing data protection against unauthorized access and cyber-attacks, since homomorphic encryption has been used for a key exchange application.

Equally enthusiastic about the increasing security of their services are some telecom companies and VoIP service providers. They want to assure customers of an additional layer of security in view of growing privacy and data protection concerns. The incorporation of key exchange applications grounded in modern cryptographic techniques would therefore provide an acute competitive edge to them in terms of improving customer trust and giving proof of conformance to standards like General Data Protection Regulation.

SMEs are also supposed to protect their data, while the resources, too, are limited. Many of them work with sensitive information or are highly regulated. Thus, a practical and cost-effective way for such enterprises to protect their data—without investing in extensive and expensive infrastructure—is to integrate secure key exchange into their IP telephony solutions.. As a result, enable the SME to follow regulations and ensure their communication is kept secured within budgetary limits.

The cybersecurity specialists and system integrators will also be able to take advantage of it, providing service protection for communications infrastructure. These companies are constantly in search of advanced means of countering absolutely new threats in cyberspace that are equally serious in their effect. Thanks to homomorphic encryption, they get the opportunity to offer their customers state-of-the-art and workable solutions, and thus achieve high credibility for their services, building a decent reputation for themselves in this market.

Another important category includes private users who prioritize privacy, such as journalists, lawyers, and high-ranking officials. Certainly, this category represents a more narrow part of the market; nevertheless, their requirements to safeguard private data and communications are rather high. They are ready to use solutions based on advanced cryptographic technologies provided that these solutions are easy to use and affordable.

The demand for cryptographic solutions keeps growing very fast, mainly under the impetus of increased cybersecurity threats, increased data protection requirements, and increased privacy awareness. Solutions like homomorphic encryption are key to ensuring secure communications in the context of IP telephony, where the protection of personal data is increasingly a concern for users.

While RSA and Diffie-Hellman traditional methods of cryptography are still applied, their usage has already been largely replaced by modern technologies. It includes elliptic curve cryptography and hybrid encryption models designed to comply with modern standards and endure computational loads arising in order to protect communications in distributed systems.

It gives special attention to protocols with advanced cryptographic features, such as Perfect Forward Secrecy and Zero-Knowledge Proof. These technologies provide reliable protection against sophisticated attacks and help build trust in secure communication systems. They form a sound basis for the development of secure communications that can meet modern requirements for data protection.

Also, blockchain and distributed ledger technologies are highly contributing to the development of cryptography, finding their applications in many industries: finance, healthcare, and even vote systems. This shows that strong cryptographic tools are very demanding and applied in many fields.

In conclusion the integration of advanced encryption methods, such as homomorphic encryption, into IP telephony aligns with the increasing demand for secure communication solutions. As cybersecurity threats continue to evolve, businesses and service providers seek robust security mechanisms to protect sensitive voice data. The adoption of homomorphic encryption and hybrid key exchange methods can serve as a key differentiator in the competitive VoIP market, offering enhanced privacy and compliance with data protection regulations. This trend highlights a growing market opportunity for secure communication solutions, positioning companies that invest in these technologies at the forefront of innovation.

1.2 Comparative analysis of existing analogues

One of the most important characteristics of secure network communication is cryptographic key exchange. There are several typical key exchange methods, each with its own characteristics, advantages, and disadvantages. Table 1 provides the main methods, their description, areas of application, degree of security, and speed.

Table 1. Key Exchange Methods and Their Characteristics.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **What it does** | **Where is it used?** | **Security** | **Speed** |
| **Diffie-Hellman (DH)** | A way for two people to create a shared secret key over an open network. | IPsec, TLS 1.2 | Medium (can be hacked without extra protection) | Fast (~5-10 ms) |
| **Elliptic Curve Diffie-Hellman (ECDH)** | A better and faster version of DH using math with curves. | TLS 1.3, Signal, ZRTP | High (if done correctly) | Very fast (~1-5 ms) |
| **RSA Key Exchange** | Uses RSA encryption to send a key securely. | Older versions of TLS | Strong, but slow because it needs big keys | Slow (~100-500 ms) |
| **Hybrid RSA + AES** | Uses RSA to send a key, then AES to encrypt data (faster than RSA alone). | TLS, VPN | High | Faster than RSA (~50-300 ms) |
| **ZRTP (Zimmermann Real-Time Transport Protocol)** | Secure key exchange for voice and video calls, no central server needed. | VoIP (Jitsi, Signal) | High (users can check if the connection is safe) | Very fast (~10-50 ms) |
| **Paillier + ECDH (Our Project)** | Uses ECDH for fast key exchange and Paillier encryption to keep the key secret. Even if someone intercepts it, they can’t read it. | Used for secure VoIP calls (Asterisk-based systems). | Very High (extra protection to keep keys hidden). | Moderate (~100-300 ms, due to extra encryption work). |

Diffie-Hellman (DH) is an older algorithm based on exponential computation, allowing two entities to share a secret key over an insecure channel. It finds most use in IPsec and TLS 1.2. As efficient as DH is, however, it remains only moderately secure because it's vulnerable to man-in-the-middle (MITM) attacks in the absence of authentication. In spite of that, with its speed ranging between 5 to 10 milliseconds, it is still feasible. Elliptic Curve Diffie-Hellman (ECDH) is a more efficient implementation of DH, utilizing elliptic curves to gain better security using smaller key sizes. This is optimized to make it more secure and quicker compared to regular DH. It is popularly utilized in TLS 1.3, Signal, and ZRTP. ECDH, if implemented correctly, offers great security with the benefit of speed, with functions consuming only 1 to 5 milliseconds. RSA Key Exchange utilizes public key cryptography where one party will encrypt a key and the second will decrypt with their private key. This was widely utilized within older versions of TLS but has been considered inefficient since it relies on long lengths of keys, which significantly retard computations. RSA is still immensely secure, yet its processing at 100 to 500 milliseconds makes it highly impractical when compared to other newer alternatives. The RSA + AES hybrid technique combines symmetric and asymmetric encryption strengths. RSA is applied in the technique to safely encrypt the symmetric AES key, which will thereafter be used to encrypt actual data. The technique provides high efficiency because AES encryption is faster than RSA by a large margin. Used in TLS and VPN protocols, it provides good security with reasonable performance with 50 to 300 milliseconds of processing time—far better than if plain RSA were to be used individually. Zimmermann Real-Time Transport Protocol (ZRTP) is a key exchange method specifically designed for VoIP communications without a centralized server. It supports either DH or ECDH and possesses a Short Authentication String (SAS) process, which allows users to authenticate the connection by hand in order to prevent MITM attacks. ZRTP is implemented in applications like Jitsi and Signal. Due to its decentralization and its speed (10 to 50 milliseconds), it is the best choice for secure VoIP calls.   Dependent upon the level of security required, speed, and environment within which the system will be deployed is selection of the method used to exchange the keys.   
 One of the most highly regarded algorithms is ECDH, owing to security and the imperceptible processing overhead. Hybrid methods such as RSA + AES remain relevant in protocols such as TLS and VPN, where security and performance must be traded off. ZRTP is ideal for VoIP since it is decentralized and incorporates voice authentication. Meanwhile, legacy DH remains used but requires additional authentication measures to prevent MITM attacks. Pure RSA due to its enormous computational expense is being used with more effective solutions. Our project suggests an advanced approach by the integration of ECDH with Paillier homomorphic encryption.  The method supports key exchange in a secure way with no leakage of key contents even if they get intercepted.   
 Our approach contrasts with traditional methods involving the transfer of public keys in plaintext, but our technique involves encrypting them using Paillier before their transfer. This additional level of security is especially valuable for implementation in VoIP applications, since real-time security is required. In an Asterisk-based implementation of VoIP encryption, this solution significantly enhances the privacy of voice communications. Even when attackers attempt to intercept key exchanges, no useful information is yielded as a consequence of the homomorphic encryption. Although Paillier encryption does introduce some computational overhead, our system remains quite robust in terms of performance, with key exchange times ranging from 100 to 300 milliseconds. This is a reasonable trade-off given the dramatic security improvements.   Key exchange mechanisms continue to evolve based on factors like security requirements, operational speed, and technological advancements.  Although RSA is becoming obsolete, newer algorithms like ECDH dominate because they are efficient. Hybrid methods such as RSA + AES are still used in VPNs and TLS where a balance between security and performance is required. ZRTP remains a suitable solution in VoIP communication because of its performance and decentralization. With the advent of cryptographic attacks such as quantum computing, new encryption techniques are necessary. Our solution bringing Paillier and ECDH together is a novel approach that optimizes key exchange security with reasonable performance. The continuous innovation of cryptographic methods is essential to the security of digital communications within an ever-changing technological landscape.

In cryptography, the choice of encryption method is a major step in data security. Each technique has strengths in terms of speed and vulnerability to hacking. Some algorithms are highly secure but computationally intensive, while others are quicker but less safe from attacks. The following graphs compare six crypto methods: DH (2048-bit), ECDH (256-bit), RSA (2048-bit), Hybrid RSA+AES, ZRTP, and Paillier + ECDH. The first graph in Figure 1 shows the number of years required to break each of them with brute force attacks, and the second shows their efficiency in operations per second. Comparing these characteristics makes you understand which methods provide the best security-performance tradeoff.

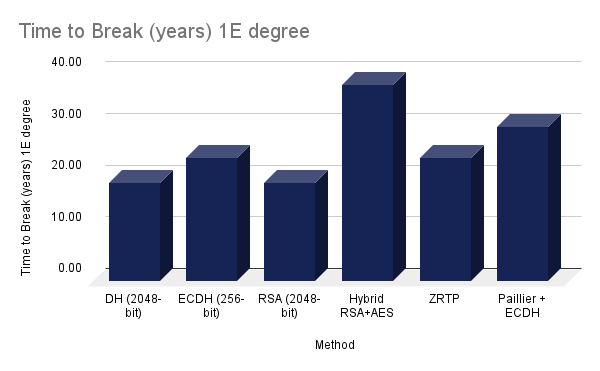


Figure 1.Time to Break  (years) to the power of 10.

If you take a look at this graph, you can observe that different cryptographic methods have spectacular differences in resistance to hacking. The most secure is the hybrid RSA + AES method, and it has a staggering value of about 10³⁸ years to break. This is due to the fact that it is a merger of the strength of asymmetric and symmetric encryption, which provides optimum security for information. Other methods also have great resistance. For example, Paillier + ECDH costs 10³⁰ years, which puts it among the safest solutions. On the other hand, the old algorithms DH (2048-bit), ECDH (256-bit) and RSA (2048-bit) have approximately the same resilience on the order of 10¹⁹–10²⁴ years. It remains an extremely high value, yet lower than hybrid methods. ZRTP, utilized in VoIP communication, likewise demonstrates effective safeguarding with similar figures.  
 On the whole, it can be stated that today's cryptographic measures are very hackable, yet hybrid solutions guarantee maximum protection.

In addition to safety against being hacked, another essential requirement for the choice of a cryptographic algorithm is performance. You can see  in Figure 2 that there is quite some difference between the algorithms in terms of performance. ECDH (256-bit) has the best performance with about 300,000 operations per second. It is thus a suitable candidate to use where data must be processed extremely fast, such as in key exchange protocol.

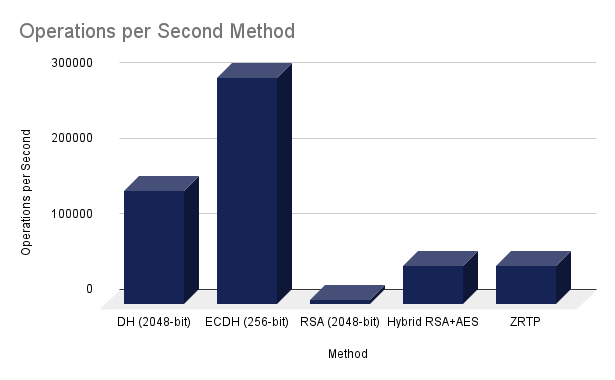


Figure 2 .Number of operations per second for different cryptographic methods.

DH (2048-bit) is not too bad at 150,000 operations per second, therefore adequate for most applications. RSA (2048-bit) is, however, much slower at 5,000 operations per second. This is due to the nature of its computations that render it unsuitable for continuous use but still necessary for secure key exchange. Hybrid RSA+AES, are mid-range performers. They perform more operations per second than vanilla RSA (2048-bit), but lag behind less complex algorithms such as ECDH (256-bit). Hybrid schemes leverage symmetric encryption (AES) to process data quicker after the initial key exchange using asymmetric encryption. It is intriguing to discover that ZRTP, although neither the most efficient algorithm, works at the level of hybrid methods and possesses a compromise between performance and security. Looking at this graph, we can state that if the most important issue is the speed, then ECDH (256-bit) would be the best option. However, if the highest priority is maximum security, then one should be cautious enough to take hybrid methods into consideration.  
 Both graphs show that the choice of cryptographic algorithm depends on specific needs. If security from being hacked is your primary concern, then Hybrid RSA+AES and Paillier + ECDH are the best options since their brute force resistance is far greater than other solutions. However, these algorithms might require more computer resources. Otherwise, if high-level speed is the issue at hand, then ECDH (256-bit) is the optimal choice as it performs an order of magnitude more operations per second than the standard RSA (2048-bit) and DH (2048-bit) algorithms. Meanwhile, it is extremely secure. Thus, we can say that there is no one cryptographic method - the choice depends on the specific application. In some situations, maximum security at the expense of performance is acceptable, and in others, high speed with an acceptable level of security is acceptable.

1.3 Features of existing analogues

The difference between large exchange protocols speaks to the merits and limitations of existing methods, in terms of performance, security, and appropriateness for real-time application in Table 2. Traditional methods like Diffie-Hellman (DH) and RSA were thoroughly used but are not the best. DH is sufficiently fast but lacks inherent immunity to man-in-the-middle (MITM) attack in the absence of an extra authentication step. RSA is encryptable with but requires long keys that are computationally expensive and hence less desirable for real-time systems. Hybrid methods such as RSA + AES provide better efficiency where asymmetric encryption is only used for key exchange purposes and efficient symmetric encryption for data transport.

Table 2. Features of analogues.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Feature** | **Diffie-Hellman (DH)** | **ECDH** | **RSA Key Exchange** | **Hybrid RSA + AES** | **ZRTP** | **Paillier + ECDH (Our Project)** |
| **Encryption Type** | Asymmetric (− less efficient than ECDH) | Asymmetric (Elliptic Curve, + more efficient) | Asymmetric (− requires long keys) | Hybrid (+ balance of security and speed) | Asymmetric (DH/ECDH, + adaptive) | Hybrid (+ combines elliptic curves and homomorphic encryption) |
| **Performance** | + | ++ | − | + | + | + |
| **Security Level** | − | ++ | + | + | + | ++ |
| **Confidentiality** | − | + | − | + | + | ++ |
| **Compatibility** | + | + | + | + | + | + |
| **Data Integrity** | − | + | + | + | + | ++ |

ECDH is among the favorites and offers excellent security with enhanced efficiency with the use of elliptic curves. The deployment in modern protocols like TLS 1.3 and Signal shows its usage in secure communication. ZRTP takes it a notch higher with offering secure VoIP communications without using a key infrastructure central server, with the use of SAS authentication to ensure that interception cannot be achieved. In our project  Paillier homomorphic encryption-ECDH hybrid project sets the new gold standard for key exchange security. By combining ECDH's speed and efficiency with Paillier encryption's next-generation security features, it enables key exchanges to be safeguarded without releasing sensitive information. In contrast with other solutions where central authorities need to be trusted or secondary verification mechanisms are occasionally used, the approach maintains confidentiality by ensuring not even partial access to keys is exposed. It is especially relevant in VoIP environments where security and performance are equally matched.  
 In environments where security and computation cost are not concessions to the lack of either in providing both, Paillier + ECDH is a robust solution for high-level encrypted communication. Being part of secure VoIP calls made in the Asterisk environment, it stands among newer solutions moving away from existing processes while addressing issues within dominant strategy.

Information security is crucial in all domains in the modern world, either in private communications, business data or financial transactions. With technology development, several various encryption methods as you can see in Table 3  that are utilized to protect users' information in products and services by companies have been developed. Different organizations choose different cryptographic algorithms depending on their needs: some need speed, others maximum security, and others a compromise between these two. In this paper, we will explain what protection mechanisms big companies use, why they chose them, and also explain what is new in our project, which combines modern encryption technologies to ensure secure data transfer.

Table 3. Encryption Methods Used by Companies and Their Products.

|  |  |  |  |
| --- | --- | --- | --- |
| **Company** | **Product** | **Encryption Method** | **Reason for Choice** |
| Dell | PowerScale | Diffie-Hellman (DH) | Secure key exchange over an open channel |
| Google | Chrome (TLS 1.3) | Elliptic Curve Diffie-Hellman (ECDH) | Fast and secure key exchange |
| Meta (Facebook) | WhatsApp | Signal Protocol (ECDH + Double Ratchet) | Ensures end-to-end message encryption |
| MIT | CryptDB | Homomorphic Encryption | Allows computation on encrypted data without decryption |
| Jitsi | Jitsi Meet | ZRTP (Zimmermann Real-Time Transport Protocol) | Voice and video calls without a central server |
| OpenVPN | OpenVPN | Hybrid RSA + AES | Combines RSA security with AES speed |
| Cisco | Cisco AnyConnect VPN | Hybrid RSA + AES | Fast and secure key transmission for VPN |

Among the firms using cryptography is Google. For Chrome, starting from TLS 1.3, they use Elliptic Curve Diffie-Hellman (ECDH) to provide a secure communication channel between the sites and the users. With this method, encryption keys can be shared quickly and securely and in the process provide maximum data protection. Google chose to use ECDH because of its performance as opposed to the standard Diffie-Hellman algorithm, but without being less secure. It was a fix that made surfing the Internet safer without any lag in page loads. The Meta (Facebook) owned WhatsApp messenger uses the Signal Protocol, which is a combination of ECDH and Double Ratchet algorithm. This technology ensures that encryption keys are replaced with each new message, and it is effectively unbreakable to break correspondence. WhatsApp adopted this method to guarantee complete confidentiality of user data even if an attacker manages to obtain one of the keys. As a result of the use of the Signal Protocol, WhatsApp is now among the most secure messengers in the world. Hybrid encryption methods are used by certain businesses. For example, Cisco and OpenVPN use a hybrid RSA + AES strategy in their VPN software. RSA is used to safely transmit a symmetric AES key, and AES itself performs a fast data encryption routine. This option allows you to combine the security of asymmetric encryption and the performance of symmetric encryption, creating a balance between security and speed. Due to this, Cisco and OpenVPN technologies ensure a stable connection even when the load is high. A great example is Jitsi, which uses ZRTP (Zimmermann Real-Time Transport Protocol) to make secure video calls. Unlike other solutions, ZRTP doesn't require a central server to store encryption keys - they are generated and transmitted directly between the communicators. This solution precludes data compromise in the event of server hacking and makes ZRTP an ideal solution for decentralized communication systems.  
 So, firms select encryption strategies according to the problems they must address. Google emphasizes connection speed, Apple and WhatsApp emphasize secure messaging, Cisco and OpenVPN emphasize reliability-performance balance, and Jitsi emphasizes decentralization.

1.4 Project Description

The SRTP protocol is used for encrypting voice messages, using the symmetric AES algorithm, which provides a high degree of encryption for voice data. The organisation's system should temporarily store encrypted session keys for ongoing connections, securely delete them immediately after session termination and log the key exchange process without saving sensitive data. The project does not provide additional protection or optimisation of the Asterisk server. The main focus is on protecting key exchange and encrypting voice communications. However, VoIP can be eavesdropped and data can be intercepted by attackers. To prevent this, encryption techniques are used to protect the contents of verification messages. Two main security mechanisms are used in this assignment: homomorphic Paillier encryption, which allows computations to be performed on encrypted data without decrypting it, and SRTP (Secure Real-Time Transport Protocol), which is designed to protect real-time output data. Combining the technologies creates a secure system that protects not only the data, but also the keys used to encrypt it. Setting up a secure voice connection requires several steps. The SIP client establishes a connection to the Asterisk server. To ensure SRTP encryption, session keys must be exchanged. The session key is encrypted using the Paye algorithm and passed to the client, and then transmitted to the server in encrypted form. Due to the nature of homomorphic encryption, some operations can be performed by the server on the key without decrypting it. The server retains the encrypted key for as long as the active session remains active, and automatically deletes it when the call ends. SIP clients begin receiving SRTP-encrypted voice data after the key exchange is used. Each step of the process is recorded in system logs, although they do not contain data revealing the content of the calls or the keys used. This implementation enhances the security of VoIP communications by accounting for encryption keys and voice data to protect sensitive information from unauthorised use and eavesdropping, and may find application in networked networks as well as other applications where high security communications are required. One of the unique features of the solution is the use of homomorphic encryption for key exchange, which minimises the risk of data compromise even if the servers are compromised.  
 In conclusion, within the framework of this thesis we develop a system that will provide reliable protection of VoIP-calls on the Asterisk platform. The use of Paillier homomorphic encryption for key exchange and SRTP for audio stream encryption allows to ensure confidentiality of service data and prevent its unauthorised interception. Implementation of these solutions will be useful for any organisation that requires secure voice communication.

Our project novelty and application

Our diploma project offers a new approach to secure key exchange for VoIP technology. Our project incorporates Paillier homomorphic encryption along with ECDH, thereby enhancing the security of the encryption scheme without any loss in the transfer rate of the data. In contrast to other traditional practices, wherein keys are transmitted in plaintext form (though under mathematical protection), our system utilizes Paillier homomorphic encryption to protect even the process of key exchange. This implies that even if the intruder intercepts the transmitted data, he is unable to get the actual key.  
 Our solution is designed for VoIP solutions, with a focus on communication with Asterisk, which is one of the most widespread IP telephony solutions. Corporate customers, government and organizations conducting secret negotiations will be the main recipients of the technology. Due to our solution, it is possible to rule out the risk of interception of encryption keys, which is especially important in the context of growing cyber attacks. One of the most significant advantages of our system is that it never loses high speed even with additional protection. As opposed to traditional schemes based on full homomorphic encryption, which requires colossal computing power, we use it merely to protect the keys, and data exchange is not slowed down like in ECDH. Because of this, our solution has the best balance between the speed and the security, securing at the top level of the best solutions.   
 Cryptography algorithms are evolving every day, and companies strive to implement the best data security measures based on their needs. Google, Apple, WhatsApp and other giants utilize ECDH and its derivatives as they are secure and fast. Business networks and VPN use hybrid networks on asymmetric and symmetric cryptography. VoIP software such as Jitsi try to remove intermediaries from data exchange, creating decentralized systems.  
 Our project assists in the development of cryptographic algorithms for IP telephony with a new solution that is a blend of the efficiency of ECDH and the security of Paillier homomorphic encryption. Using the solution, organizations running VoIP will be able to gain complete security with minimal computational burden. In case cybersecurity threats increase in the future, our technology will be a major step towards developing a completely secure communication system.

.

**CHAPTER 2.**

### **2.1 General architecture of the project**

This project implements a mechanism of secure key exchange using homomorphic encryption in the environment of Asterisk and SIP-protocol.

The main purpose of the system is to protect encryption keys that are used to establish a secure VoIP connection. The use of homomorphic encryption allows computations to be performed on encrypted data without decrypting it, which makes the process more secure. Figure 5 presents a structural diagram of the system and its main components.

The server platform is Asterisk - an open source framework designed for building IP-telephony systems. It works on the basis of SIP protocol, which is one of the most common protocols for call management in VoIP networks. The client part of the project is implemented through the linphone application, which allows to implement an interface through which the user will perform authorization, registration and free call. Figure 3 shows that the project also uses a database to store service information about users, encryption module to realize the functions of key generation and storage.

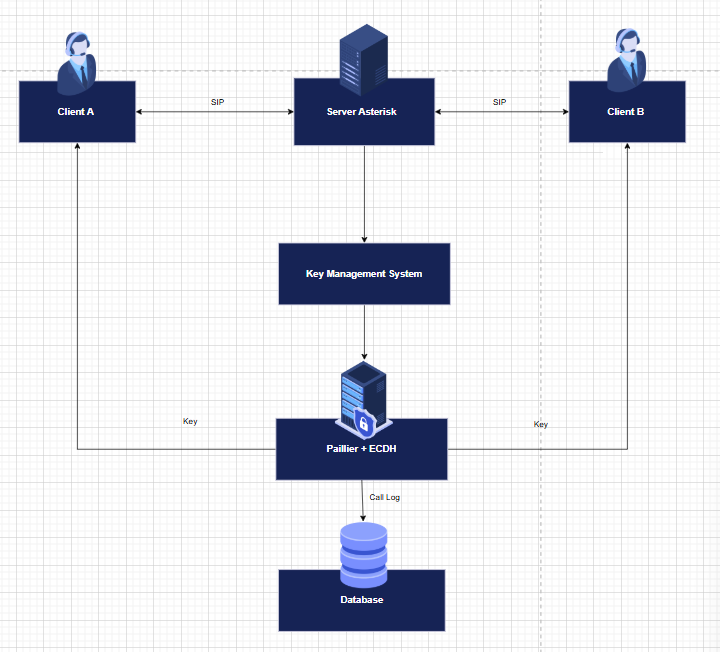


Figure 3. System Architecture Diagram.

This is the system architecture diagram for the Secure IP Telephony project; it shows how the components interact to secure communications. The Asterisk server is central and regulates user registration, call routing, and authentication to secure communications with all other components, thus becoming the core of the system. Linphone is the application that creates the client interface for the user to perform registration, authentication and initiate a SIP call, and handles all call management and data transfer issues. The key management system provides encryption key Paillier key pair and facilitates secure public key encryption during the key exchange process. This system ensures that the public keys remain protected throughout transmission, minimizing exposure to attackers. The Database component is a database that stores critical data such as user accounts, call records and all configurations retrieved by the server when required. The arrows in the diagram indicate data flows such as call control between the Linphone Client and the server, key exchange with the key management system, and database operations to store or retrieve data. All of these components work together to create a secure and efficient IP telephony system in which the Asterisk server plays a major role in coordinating communication and ensuring data security.

The Asterisk server is the core of the project architecture, responsible for user registration, call management, and data routing. This module performs the following:

Core of the architecture, responsible for:

○ User Registration & Authentication: Handles SIP-based authentication for Linphone clients and verifies credentials against the database.

○ Call Processing: Establishes, maintains, and terminates communication sessions, routing media data between users.

○ Encryption Key Management: Generates and facilitates secure key exchange between clients to protect transmitted data.

○ Security Enforcement: Ensures encrypted data transmission using homomorphic encryption (Paillier) for protecting ECDH public keys and deriving a secure session key for SRTP.

The Asterisk Gateway Interface (AGI) allows for running external scripts, enabling encryption logic and scalability.

Asterisk has an integrated Asterisk Gateway Interface module that can run external scripts and applications for further functions like call logging or implementing business logic for encryption. This brings flexibility and the ability to further scale the system.

Linphone is a client application that enables users to interact with the server. It allows you to:

A SIP-based VoIP client responsible for:

● User Registration & Authentication: Connects to the Asterisk server using SIP credentials.

● Call Management: Initiates and receives calls over a secure channel.

● Key Generation & Encryption: Generates an ECDH key pair for secure session key agreement and encrypts the public key using Paillier before transmitting it to the Asterisk server.

Linphone connects to the Asterisk server and sends data in real-time. The client layer also generates and manages encryption keys, which are negotiated with the server for data confidentiality using ECDH.

This is a relational database to store the user information, call record. The major role that should be played in the database are:

● User Accounts: SIP credentials, authentication tokens.

● Call Records: Time, duration, participants, and connection status.

● Temporary Encryption Keys: Only stored during an active call and deleted upon termination.

Encryption keys are used exclusively during an active call. To minimize the possibility of information leakage, the key information is permanently deleted once the communication session is over.

An integral part of this project is homomorphic encryption, which allows computations to be performed on encrypted data without decryption. This property ensures that no sensitive information is exposed, even if intercepted. The following steps outline the system’s operation:

●Provides encryption key generation, storage (during active calls), and secure exchange.

● Uses Paillier Homomorphic Encryption (HE) to encrypt ECDH public keys before transmission, ensuring that public keys are never exposed in plaintext.

● Uses ECDH to derive a shared session key for encrypting VoIP communication with SRTP.

When a user starts a call, the system starts the communication process between its components. All users are pre-registered on the Asterisk server, and before establishing a connection, the system checks their authenticity (authentication). If authentication is successful, the server creates a secure communication channel between the participants. For this purpose, ECDH is used to securely derive a symmetric session key, and Paillier encryption protects the exchange of public keys.

During the conversation, all media data is encrypted on the client side using SRTP with the derived session key and transmitted through the server. It is important to note that the server does not decrypt or process this data, which guarantees its confidentiality.

After the call is completed, the server closes the session, deletes all temporary data and permanently erases the encryption keys. This prevents them from being reused or compromised.

### **2.2 Structural description of the security system.**

The developed project security system is designed in a multilevel model, with the aim of ensuring confidentiality, integrity, and availability during data transmission and processing processes. Based on three cornerstones, the system is closely interconnected and works as one mechanism that allows minimizing the risks of unauthorized access and information leakage.

**Data encryption**

The core of the security system in this solution is a hybrid encryption approach that ensures both the confidentiality of transmitted data and the protection of encryption keys. The system integrates Paillier homomorphic encryption and Elliptic Curve Diffie-Hellman (ECDH) to secure key exchange, as shown in Figure 4. Paillier encryption enables mathematical operations on encrypted keys without requiring decryption, ensuring that sensitive cryptographic data remains protected throughout the entire lifecycle—during transmission, storage, and processing on the server. This approach was chosen because it guarantees end-to-end confidentiality while allowing secure and efficient key exchange.

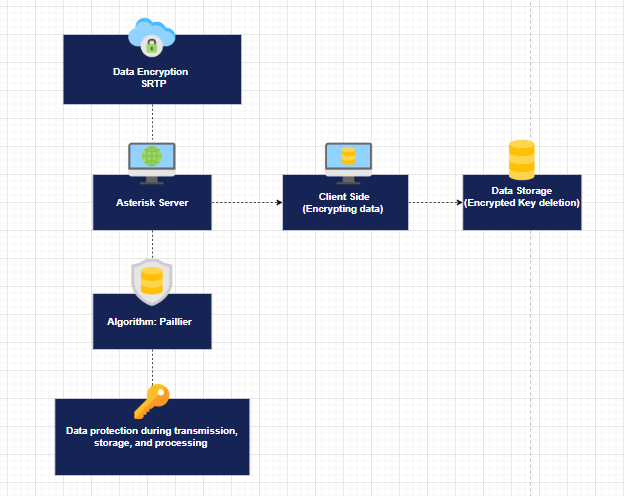


Figure 4. Data Encryption Diagram

Paillier is an asymmetric cryptographic algorithm belonging to the class of homomorphic ciphers. Its key feature is that it supports additive homomorphism, that is, it allows you to perform addition operations on encrypted data without decrypting it.

1. Key generation:

Two large prime numbers are selected 𝑝 and 𝑞, then the module 𝑛 = 𝑝 ⋅ 𝑞 is calculated.

Public key: (𝑛,𝑔), where 𝑔 is an auxiliary number.

The private key: ( 𝑝 , 𝑞 ).

2.Encryption:

The public key is used to encrypt the number 𝑚 using the formula:



where 𝑟 is a random number.

3.Homomorphic properties:

If there are two ciphertexts 𝑐1 and 𝑐2 , then their product corresponds to the sum of their initial values :



This property allows you to perform some calculations without decrypting the data.

4.Decryption: The private key is used to restore the original value.

In our project, Paillier is used to encrypt symmetric keys that are used to protect voice traffic. This ensures their confidentiality during transmission.

ECDH is a cryptographic protocol that allows two parties to securely exchange secret keys through an unsecured channel. It is based on elliptic curve arithmetic, which makes it more efficient and secure compared to the classic Diffie-Hellman (DH).

How does ECDH work?

1.Key generation:

Participants choose an elliptical curve and a generator point 𝐺.

Each party creates a private key 𝑑 (a random number) and calculates the public key 𝑃 according to the formula: *P*= *d* ⋅ *G*

2.Public key exchange:

The client and the server exchange their public keys 𝑃𝐴 and 𝑃𝐵.

3.Calculating the shared secret:

The client calculates: 𝑆 = 𝑑 𝐴⋅ 𝑃 𝐵

The server calculates: 𝑆 = 𝑑 𝐵 ⋅ 𝑃 𝐴

Both receive the same shared secret key.

In our project, ECDH is used to set a shared session key, which is then encrypted using Paillier and transmitted to the participants in the call. This ensures the secure exchange of keys without the need to disclose them in the clear.

**Key Management**

The most important securities are the usage of reliable encryption key management. The project dynamically will implement a key management in Figure 5 mechanism in such a way that it ensures key generation, exchange, and deletion after termination of the communication session.

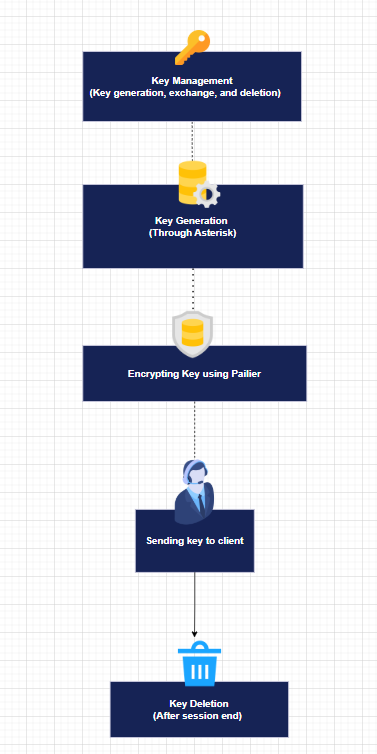


Figure 5. Key Management Diagram

The key exchange diagram shows the secure generation, exchange, and deletion of encryption keys. Encryption keys are generated by the Asterisk server when a connection is established between call participants. If two participants connect, Asterisk creates a key pair. The keys are transmitted over a secure channel using Paillier asymmetric encryption, which prevents them from being intercepted by intruders. Once the keys are exchanged, they are used for encryption during the session. At the end of the session, all keys will be immediately deleted on both the client and server side. This process minimizes risks as old keys cannot be reused and even in case of interception, the data itself remains safe. Key management allows the system to follow modern data security standards..

**FlowChart**

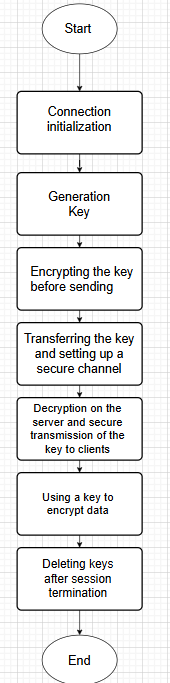
****

Figure 6. Flow chart.

This is the Flowchart of our process(Figure 6).It consists of several steps such as:

1. Connection initialization

At this stage, users initiate a connection to the Asterisk server by authenticating and negotiating communication parameters. This process includes verifying credentials, allocating resources, and preparing the system for secure data transfer

* Client A sends a request to establish a connection with client B.
* The Asterisk server accepts the request and starts the key generation process.

2. Generating an ECDH key pair

Each client (A and B) generates its secret key pair using Elliptic Curve Diffie-Hellman (ECDH).

* A suitable elliptic curve E over a finite field 𝔽ₚ is chosen.
* A generator point G on this curve is selected.
* Client A generates a random private key 𝑎 and computes the public key: PA =a ⋅ G
* Client B generates a random private key 𝑏 and computes the public key: PB = b ⋅ G

3. Exchange of Public Keys via Asterisk

* Client A sends its P\_A to the Asterisk server.
* Client B sends its P\_B to the Asterisk server.
* Asterisk forwards the public keys to the respective clients.

4. Computation of the Shared Secret (ECDH)

Each client computes the shared secret **S** using the received public key:

* A computes:



* **B** computes:



5. Encrypting the Key Before Transmission

Now, the Asterisk server must securely transmit S to Client B, ensuring it cannot be intercepted. This is achieved using Paillier homomorphic encryption.

Paillier Key Generation

* Two large prime numbers *p* and *q* are chosen.
* The modulus *n* is computed: *n* = *p⋅ q*
* Compute λ (least common multiple): λ=lcm( *p*−1, *q*−1)
* Define *g*: *g* = *n* + 1
* Compute μ:  , where *L*(*x*) = (*x*−1)/ *n*.
* Public key: (*n*, *g*), Private key: (λ, μ).

Now, Asterisk encrypts the shared secret S before sending it to Client B:



where *r* is a random number such that gcd(*r*, *n*) = 1.

6. Decrypting the Key on the Client-Side

Client B receives the encrypted key c and decrypts it using its private key (λ, μ):  . Now, Client B has the same shared key S as Client A.

7. Using the Key for Secure Voice Encryption (SRTP)

Once the shared key S is established, it is used to secure the voice traffic via Secure Real-time Transport Protocol (SRTP).

SRTP Encryption Methods

* Encryption Algorithm: AES-256-GCM
* Message Authentication: HMAC-SHA1
* Replay Protection: Sequence number verification

8. Call Termination and Key Deletion

When the call ends:

* Asterisk sends a SIP BYE message to both clients.
* Both clients immediately delete key S from memory.
* Asterisk also deletes key S from RAM.

**Threat Resistance**

The developed security system takes into account modern cyber threats and provides multi-layered protection. The use of strong encryption with secure key management combined with server infrastructure protection creates a stable and attack-resistant platform.

One of the key advantages of the system is its modularity. As a result, each component can be upgraded without major changes to the architecture, allowing it to adapt to new threats and security requirements. Thus, the proposed structure provides a high degree of data and infrastructure protection in accordance with modern cybersecurity standards.

### **2.3 ER diagram**

The ER diagram in Figure represents the structure of our database system, which is designed to securely manage VoIP calls with encrypted key transmission. This model ensures secure communication, tracks call activity, and logs events.

In this section, we will describe the key entities, their attributes, and the relationships between them to illustrate how data flows within the system.

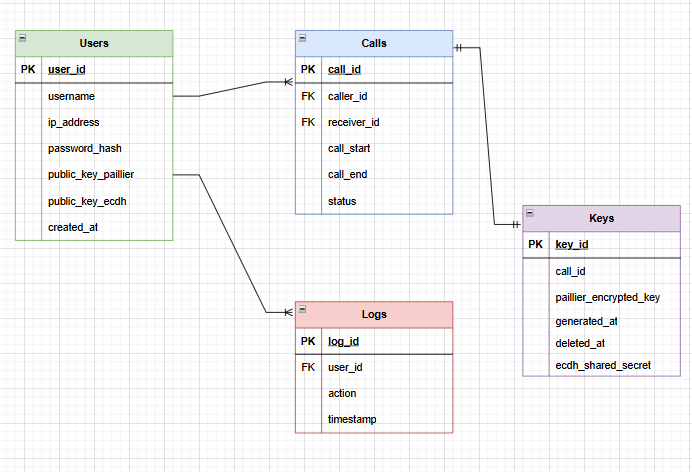


Figure7. ER-diagram

The database consists of four main entities(Figure 7): Users, Calls, Keys, and Logs. Each entity plays a crucial role in maintaining the integrity and security our system

1. **Users**Stores user information, including authentication credentials and public keys for secure key exchange.

* user\_id (PK)– Unique identifier for each user.
* username – SIP login of the client.
* ip\_address – IP address of the client.
* password\_hash - Securely stored password hash.
* public\_key\_paillier - Paillier public key, used for encrypting session keys.
* ecdh\_public\_key - ECDH public key for secure key exchange.
* status – Current status (active, inactive).
* created\_at – Date when the client was registered.

2.**Calls**This table records details about VoIP calls, including their participants and status.

* call\_id (PK) – Unique identifier of the call
* caller\_id (FK) – ID of the caller
* receiver\_id (FK) – ID of the receiver
* call\_start– Timestamp of when the call started
* call\_end– Timestamp of when the call ended
* status – Status of the call.

3. **Keys**This table holds encryption keys used for securing VoIP sessions.

* key\_id (PK) – Unique identifier for each key.
* call\_id (FK) – ID of the associated call
* paillier\_encrypted\_key – Session key encrypted using Paillier homomorphic encryption.
* ecdh\_shared\_secret - The shared secret established using ECDH (deleted after session ends).
* generated\_at – Timestamp when the key was generated

4. **Logs (Security and System Logs)**To enhance security, the system records logs related to calls, authentication, and potential security breaches.

* log\_id (PK) – Unique identifier of the log entry
* user\_id (FK) – ID of the user involved in the log event
* action – Type of event (e.g., call\_started, call\_ended, intrusion, error)
* timestamp – Timestamp of the logged event

Reference List

[1] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A Survey on Homomorphic Encryption Schemes: Theory and Implementation," *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–35, Jul. 2018.

[2] N. Li, "Research on Diffie-Hellman key exchange protocol," in *2010 2nd International Conference on Computer Engineering and Technology*, Chengdu, China, 2010, pp. V4-634–V4-637, doi: 10.1109/ICCET.2010.5485276.

[3] N. P. Varnovsky and A. V. Shokurov, "Homomorphic Encryption," *Proceedings of ISP RAS*, 2007. [Online]. Available:<https://cyberleninka.ru/article/n/gomomorfnoe-shifrovanie>. Accessed: Feb. 22, 2025.

[4] A. Alanwar et al., "Privacy-preserving set-based estimation using partially homomorphic encryption," *European Journal of Control*, vol. 68, no. 1, 2023, doi: 10.1016/j.ejcon.2023.100786.

[5] X. Yi, R. Paulet, and E. Bertino, *Homomorphic Encryption and Applications*, SpringerBriefs in Computer Science. Springer, Cham, 2014, doi: 10.1007/978-3-319-12229-8\_2.

[6] M. Abdalla, M. Bellare, and P. Rogaway, "DHAES: An Encryption Scheme Based on the Diffie-Hellman Problem," *IEEE P1363a*, Sep. 1998.

[7] P. Chaudhary, R. Gupta, A. Singh, and P. Majumder, "Analysis and Comparison of Various Fully Homomorphic Encryption Techniques," in *2019 International Conference on Computing, Power and Communication Technologies (GUCON)*, New Delhi, India, 2019, pp. 58–62.

[8] J.-L. Han, M. Yang, C.-L. Wang, and S.-S. Xu, "The Implementation and Application of Fully Homomorphic Encryption Scheme," in *2012 Second International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC)*, Harbin, China, 2012, pp. 714–717, doi: 10.1109/IMCCC.2012.173.

[9] Z. Temirbekova, A. Pyrkova, Z. Abdiakhmetova, and A. Berdaly, "Library of Fully Homomorphic Encryption on a Microcontroller," in *2022 International Conference on Smart Information Systems and Technologies (SIST)*, Nur-Sultan, Kazakhstan, 2022, pp. 1–5, doi: 10.1109/SIST54437.2022.9945722.

[10] G. Tu, W. Liu, T. Zhou, X. Yang, and F. Zhang, "Concise and Efficient Multi-Identity Fully Homomorphic Encryption Scheme," *IEEE Access*, vol. 12, pp. 49640–49652, 2024, doi: 10.1109/ACCESS.2024.3384247.

[11] Ç. K. Koç, F. Özdemir, and Z. Ödemiş Özger, *Partially Homomorphic Encryption*, Book, 2021.

[12] S. J. Mohammed and D. B. Taha, "Performance Evaluation of RSA, ElGamal, and Paillier Partial Homomorphic Encryption Algorithms," in *2022 International Conference on Computer Science and Software Engineering (CSASE)*, Duhok, Iraq, 2022, pp. 89–94, doi: 10.1109/CSASE51777.2022.9759825.

[13] D. Wang, B. Guo, Y. Shen, S.-J. Cheng, and Y.-H. Lin, "A faster fully homomorphic encryption scheme in big data," in *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, Beijing, China, 2017, pp. 345–349, doi: 10.1109/ICBDA.2017.8078836.

[14] K. Rohloff, D. B. Cousins, and D. Sumorok, "Scalable, Practical VoIP Teleconferencing With End-to-End Homomorphic Encryption," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1031–1041, May 2017, doi: 10.1109/TIFS.2016.2639340.

[15] L. Jiefeng and L. Sheng, "An Efficient and More Secure Searchable Encryption Algorithm Based On Fully Homomorphic Encryption," in *2023 International Conference on Automation, Control and Electronics Engineering (CACEE)*, Chongqing, China, 2023, pp. 83–92, doi: 10.1109/CACEE61121.2023.00025.

[16] A. N. Gerasimov, A. V. Epishkina, and K. G. Kogos, "Research of homomorphic encryption algorithms over integers," in *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, St. Petersburg and Moscow, Russia, 2017, pp. 398–403, doi: 10.1109/EIConRus.2017.7910576.

[17] M. Beunardeau, A. Connolly, R. Geraud, and D. Naccache, "Fully Homomorphic Encryption: Computations with a Blindfold," *IEEE Security & Privacy*, vol. 14, no. 1, pp. 63–67, Jan.–Feb. 2016, doi: 10.1109/MSP.2016.8.

[18] S. Mittal and S. Singh, "Retrospective Study on Classical Homomorphic Encryption Algorithms," in *2023 International Conference on IoT, Communication and Automation Technology (ICICAT)*, Gorakhpur, India, 2023, pp. 1–6, doi: 10.1109/ICICAT57735.2023.10263650.

[19] J. Kim and A. Yun, "Secure Fully Homomorphic Authenticated Encryption," *IEEE Access*, vol. 9, pp. 107279–107297, 2021, doi: 10.1109/ACCESS.2021.3100852.

[20] *Evaluating Encryption Schemes for Use in Secure Multi-Party VoIP Conferencing*.

[21] M. Boyu and C. Hong, "Adaptive Secure Homomorphic Encryption Scheme," in *2024 IEEE 2nd International Conference on Control, Electronics and Computer Technology (ICCECT)*, Jilin, China, 2024, pp. 1–5, doi: 10.1109/ICCECT60629.2024.10546172.

[22] P. Stobbe, T. Keijzer, and R. M. G. Ferrari, "A Fully Homomorphic Encryption Scheme for Real-Time Safe Control," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, Cancun, Mexico, 2022, pp. 2911–2916, doi: 10.1109/CDC51059.2022.9993055.

[23] S. Lee and D.-J. Shin, "Overflow-Detectable Floating-Point Fully Homomorphic Encryption," *IEEE Access*, vol. 12, pp. 6160–6180, 2024, doi: 10.1109/ACCESS.2024.3351738.

[24] S. Yaji, K. Bangera, and B. Neelima, "Privacy Preserving in Blockchain Based on Partial Homomorphic Encryption System for AI Applications," in *2018 IEEE 25th International Conference on High Performance Computing Workshops (HiPCW)*, Bengaluru, India, 2018, pp. 81–85, doi: 10.1109/HiPCW.2018.8634280.

[25] K. Sinha, P. Majumder, and S. K. Ghosh, "Fully Homomorphic Encryption based Privacy-Preserving Data Acquisition and Computation for Contact Tracing," in *2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, New Delhi, India, 2020, pp. 1–6, doi: 10.1109/ANTS50601.2020.9342834.