ELEC5616 - Project Report
Defeating SkyNet – Security Essentials

Harry (Tianze) Zeng
SID: 510215695

Kevin (Junkai) Mei
SID: 500210899

Contents

1 Introduction

In this project, we devoted our efforts to constructing a secure robot network system and engaged in an in-depth discussion of the key technologies involved and the primary security challenges faced. The project encompassed security measures for peer-to-peer file transfer and a comparison of the pros and cons of a centralized server distribution mechanism. Furthermore, we explored strategies to defend against external threats, such as replay attacks and tampering attacks.

2 Key exchange

- **Security:** The employment of a 2048-bit prime modulus enhances security robustness, effectively countering the most sophisticated publicly known attack algorithms.
- **Performance and Compatibility:** The adoption of longer numerical sequences invariably extends processing duration. A 2048-bit key length has been selected to optimize the balance between performance efficiency and security integrity.
- **Industry Standards and Intended Application:** Organizations such as NIST and IETF advocate for a minimum key length of 2048 bits for the assurance of secure key exchange processes [1]. This recommendation is considered an optimal equilibrium in safeguarding the confidentiality and integrity of information systems and data.

3 Confidentiality

In this project, Cipher Block Chaining (CBC) mode of AES encryption was chosen for the following reasons:

- **Enhanced Security:** CBC mode employs a technique where it Xor the previous ciphertext block with the current plaintext block before encryption [2]. This ensures that identical plaintext blocks produce different ciphertexts in different contexts, reducing the risk of pattern leakage. This method improves the security of the encryption process, especially important for large datasets
- **Error Propagation:** CBC mode features error propagation, meaning that an error in one ciphertext block affects the decryption of all subsequent blocks. This can help in detecting data tampering or corruption.

- **Wide Support and Recognition:** CBC mode is recommended by multiple standards and protocols and is recognized as a secure and practical choice for encryption. Its wide support allows for easy implementation using existing cryptographic libraries.

Finally, this is a model that does not have a built-in message authentication code. It meets the project requirements.

4 Integrity

In order to prevent attackers from tampering with messages during message transmission, we have adopted comprehensive security measures, covering encryption and message authentication codes (MAC).
First, we encrypt all transmitted messages to ensure the confidentiality of message content. The encryption process uses advanced encryption standards, such as AES, and secure modes, such as CBC, so that even if the data is intercepted during transmission, it cannot be decrypted without the correct key, thereby protecting the security of the data.
Next, we further ensure the integrity and authenticity of the message by appending a shared key-based MAC to each encrypted message. The generation of a MAC involves the combination of a hash function, such as SHA-256, with a shared secret key and the content of the message. This ensures that only the sender and receiver can check if the message has been altered. At the receiving end, the MAC is first recalculated using the same method for the received message, and then compared with the MAC received during transmission. If the two are consistent, it means that the message has not been changed since it was sent, verifying the integrity of the message; if they are inconsistent, it means that the message may have been tampered with during transmission, and the receiver should reject the message.

5 Replay Prevention

In order to prevent replay attacks, we have taken a series of comprehensive measures to ensure that the security and integrity of communications are maintained during information transmission. First, by embedding a timestamp in each message, we are able to verify the timeliness of the message. This mechanism compares the timestamp in the message with the current time when it is received. If the time difference exceeds a preset threshold (for example, 30 seconds), it is considered that the message may have been replayed, and therefore it is rejected. This way, even if an attacker intercepts a message and attempts to resend it at a later time, the system can prevent such replay attempts by checking the validity of the

timestamp.
Secondly, the introduction of the sequence number mechanism further enhances message security. Each message sent contains a uniquely increasing sequence number against which the receiver verifies the sequence of the message. If the sequence number of a received message does not match what is expected, indicating a possible replay attack or message loss, the system will reject the message. In this way, we ensure coherence and order in the message flow, preventing potential security issues due to replay attacks.
In addition, adding a random salt value and HMAC (hash-based message authentication code) to each message is also an important means to defend against replay attacks. The random salt value introduces additional randomness into the encryption process. Even for two identical messages, the encryption results will be different due to different salt values, which makes it difficult for an attacker to re-encrypt by analyzing the encrypted data. Let attack. At the same time, HMAC ensures that the message has not been tampered with during transmission, making the attack more difficult.
In summary, by implementing multiple security measures such as timestamps, sequence numbers, random salt values, and HMAC verification, we can effectively prevent replay attacks and ensure the security and reliability of data transmission. Together, these measures build a robust defense mechanism against potential security threats and protect the communication process from being exploited by malicious attackers.

6 Authentication

6.1 Advantages and Disadvantages of Peer-to-Peer transfer and Central Web Server

Peer-to-peer transfers avoid reliance on a central web server, reducing the risk of single points of failure and enhancing the system's stability and reliability. As the number of participants increases, so do the network's bandwidth and storage resources, improving the efficiency of file transfers. Moreover, it facilitates direct exchanges between two users, minimizing the risk of data diversion and enhancing the privacy of communication.

Utilizing a central web server enables more convenient management and distribution of files, facilitating maintenance and updates. It also allows for the implementation of unified access control policies to effectively manage file access permissions. However, it can become a system vulnerability; in the event of an attack or malfunction, the entire system's stability and security could be compromised. With the expansion of the system, the server's load may increase, necessi-

tating additional hardware to meet demands and thus increasing deployment costs. Furthermore, as all communications must pass through the server, they could be subject to monitoring, posing a risk of leakage.

6.2 Securing Botnet Communications Against External Control

In the current template implementation, a significant vulnerability lies in the lack of a robust authentication mechanism and insufficient security in encrypted communications, potentially allowing the botnet to be easily manipulated by other hackers or government entities. For instance, if communications are not encrypted or employ weak encryption algorithms, attackers could intercept and alter messages through man-in-the-middle attacks, thereby gaining control over the botnet. Additionally, the absence of a verification process for the identities of both the server and bots within communications enables attackers to masquerade as legitimate servers or bot members and issue malicious commands. To mitigate these risks, it is recommended to employ stronger encryption technologies, such as TLS/SSL, to safeguard data transmission, and to introduce certificate-based authentication mechanisms to verify the identities of the communicating parties, ensuring that only authorized servers and bots can engage in network communications. Furthermore, regularly updating encryption algorithms and keys, along with adopting dynamic IP addressing and port randomization strategies, can complicate attack efforts and enhance the overall security of the network.

7 References

References

[1] Barker, E., & Roginsky, A. (2018). Transitioning the use of cryptographic algorithms and key lengths (No. NIST Special Publication (SP) 800-131A Rev. 2 (Draft)). National Institute of Standards and Technology.

[2] Shetty, V. S., Anusha, R., MJ, D. K., & Hegde, P. (2020, February). A survey on performance analysis of block cipher algorithms. In 2020 International Conference on Inventive Computation Technologies (ICICT) (pp. 167-174). IEEE.