Securing Data Communication in Networks by Combining Error Correction and Data Encryption Techniques.

Aparna Singh, Sailada Sowjanya, Shreyas Udaya

Department of Computer Science and Engineering
National Institute of Technology Karnataka
Surathkal, Mangalore, India

+91 91134 92554,+91 6304381921,+91 7022053251

aparnasingh.211cs107@nitk.edu.in,
sailadasowjanya.211cs149@nitk.edu.in,
shreyasudaya.211cs152@nitk.edu.in

ABSTRACT

A. Background of the problem statement

These days data communication through networks is an important element in use that affects many aspects of our daily lives, ranging from watching YouTube videos at home for entertainment to receiving confidential data for telecommunications from satellites. Everywhere we look, data communication is somewhat involved. Thus, the security of data is a prioritized concept in today's digital world. [1]

B. Challenges and issues of the problem statement

Corrupted data and tapping into networks can lead to serious damage and repercussions for many people. For example, important security details of an Organisation could land in the wrong hands who have no right to access them and may cause a loss in unimaginable ways.

C. Existing approaches or methods and their issues

Many methods exist for implementing data security [2] and error correction. Techniques such as Hamming codes, Low-Density-Parity-Check(LDPC) [3]and Reed-Solomon codes, and Symmetric-key encryption algorithms and functions such as Data-Encryption-Standard(DES) and Advanced-Encryption-Standard(AES) are all used for improving data security. However, there are issues such as increased complexity causing latency, key management [4], compatibility, and trade-offs.

D. Your problem statement

These issues must be considered when implementing such measures. That brings us to our problem statement: "Securing data communication in networks using error correction and data encryption techniques".

E. Objectives of the proposed work

[5] In this project, we will be exploring modern ways to enhance security and also acknowledging algorithms used to date for improving security measures. The objectives of this project are to provide data integrity and enhance data security by developing a MATLAB program during data transmission.

INTRODUCTION

There are various methods of securing data and its integrity. In these, two important ways are error correction and data encryption. The former is used to detect and correct errors that may occur during transmission, such as those caused by noise or interference in the communication channel [6]. The latter, on the other hand, are used to scramble data so that it cannot be read by unauthorized parties [7]. The combination of these two techniques can provide a powerful solution for securing data communication in networks [8]. By first applying error correction techniques to ensure the completeness and accuracy of the transmitted data, and then encrypting the data to protect it from unauthorized access, the data can be securely transmitted over the network. However, the effectiveness of this approach depends on the specific error correction and data encryption techniques used, as well as the implementation details. In addition, there may be trade-offs between security, performance, and complexity that need to be carefully considered when designing such systems. Hence, Our project is intended to find out the combination of which data encryption algorithm and error correction technique provides the most optimum security model. The security model will be built on MATLAB and judged on a variety of factors, such as threat model, data sensitivity, performance requirement, and regulation compliance, along with the best balance of security, cost, and utilization.

Challenges:

There are several challenges and issues associated with the problem statement. Some are:

- Selection of error correction and data encryption techniques: The choice of error correction and data encryption technique we are using is critical to the effectiveness of the security model. There are many different algorithms available, each with its own strengths and weaknesses. Selecting the most appropriate combination of techniques can be challenging and requires careful consideration of the specific requirements of the system.
- Regulatory compliance: There may be regulatory requirements that need to be taken into account. Compliance with regulations such as HIPAA or GDPR can add additional complexity to the system design.
- Performance impact: The use of error correction and data encryption techniques can impact the working of the system. It is important to carefully balance security requirements with performance needs.
- Complexity: The implementation of the security model can have high complexity, especially if too many error corrections and data encryption algorithms are used. This can increase the cost and time, along with the potential for errors. However, using too less of it makes the model more susceptible to attacks.
- Attack mitigation: Encryption algorithms can be vulnerable to attacks, such as brute-force attacks. Regular updates and testing must be conducted to ensure the system is protected against these types of attacks.

Existing methods and issues

Usually, models deploy just a single error correction and encryption algorithm. For example, many times AES(Advanced Encryption Standard) is used alone and often combined with just one, such as ldpc(Low-density-parity Codes) as stated in [9] [10]. However, it has many issues.

- This may not provide enough security against sophisticated attacks. For example, attackers may focus on exploiting a single algorithm, making it more vulnerable to such algorithm-specific attacks.
- Inefficient use of resources, along with compatibility issues with some networks, as different networks may require different error correction and encryption techniques.
- A single algorithm may not be flexible enough to adapt to changing network requirements or security threats, leading to reduced effectiveness over time.

However, we must not use too many either, as it increases costs and risks.

Objectives

Objectives of the proposed work:
Problem Statement: Securing data communication in networks using error correction and data encryption techniques by determining the optimal combination of encryption algorithms and error correction techniques.

- To evaluate and compare the effectiveness of different combinations of encryption algorithms and error correction techniques in terms of security, efficiency, and compatibility with various networks.
- To develop a secure model using the optimal combination of encryption algorithms and error correction techniques, while taking into account factors such as data sensitivity, performance requirements, and regulatory compliance.

LITERATURE REVIEW

There are several prevailing methods of error correction and data encryption. As we said earlier, most models deploy just a single error correction and encryption algorithm. This leaves it vulnerable to attacks and is not safe against sophisticated attacks. Given below are some algorithms and techniques used today in various networks spanning different applications:

Error Correction Techniques:

- Forward Error Correction (FEC) which will add redundant data to the information, which is being transmitted. A type of this is Reed-Solomon codes. They provide high reliability in the presence of noise and other disturbances in the transmission medium. Some others are Convolution codes and LDPC [11].
- Automatic Repeat Request (ARQ) which re-transmits the corrupted data during transmission. ARQ can be used with other error correction techniques, such as Forward Error Correction (FEC), to enhance network security.
- Checksum in which the receiver add up the total bytes in the message together with the checksum value and matches it to check if an error has occurred.
- Cyclic Redundancy Check (CRC) is another complex version of checksum in which the message is divided into blocks, each of which generates a checksum and the receiver matches the calculated received message with the checksum.

The main problem with combining data encryption with error correction, is that data encryption techniques are susceptible to the methods used to attack error correction techniques. For e.g., in combining say, Blowfish and Hamming algorithms, if attackers attack Hamming then the Blowfish is also susceptible.

Some Data Encryption Algorithms are:

- Advanced Encryption Standard (AES) which is more secure than Data Encryption Standard (DES) since it can support key sizes of 128,192, or 256 bits.
- Rivest-Shamir-Adleman (RSA) in which the public key is used for encryption and anyone can encrypt data but decryption can only done by the private key which is kept by the owner.
- Elliptic Curve Cryptography (ECC) provide the same security level as RSA but the difference comes in small key sizes which makes it more efficient in memory usage as well as computation.
- ChaCha20: a cipher algorithm is used for rapid and secure symmetric encryption.
- Blowfish: Blowfish is a symmetric encryption algorithm that provides strong security and is suitable for

use in applications that require high performance and low memory requirements.

Many combinations also have their pros and cons.

- Encryption followed by error correction: The encrypted data is protected from unauthorized users before error correction is applied to it. The encryption can neglect some error patterns and hence turns out to be less effective in error detection.
- Error correction followed by encryption: The error detection and correction can be improved as error correction is performed on the original data. The compromised encryption key can lead to the vulnerability of the original data to unauthorized access.
- Encryption and error correction simultaneously: It is an optimized process of error correction and data encryption working together which leads to increased performance rate. Moreover, it requires more processing resources and its' implementation is complex.
- Hybrid encryption and error correction: It provides protection against errors as well as unauthorized access but its' implementation is complex and requires special hardware. An example of this combination is a block cipher for encryption and Reed-Solomon code for error correction.

Overall, these are just a few examples and the pros and cons of different combinations will depend on the techniques used and the context of the application. the optimal combination of the encryption algorithm and error correction techniques will depend on the factors as mentioned earlier.

It is troublesome to combine these techniques as most of the time it results in increased Complexity, increased bandwidth usage, and also reduced efficiency.

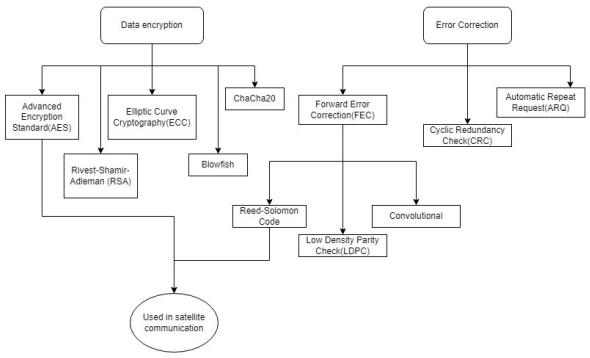

Fig. 1. Flowchart showing various methods

PROPOSED METHODOLOGY

Our proposed methodology consists of a combination of error correction and data encryption techniques to secure data communication in satellite networks. The methodology can be divided into two main phases: Overall we will have n steps:

1) Select error correction techniques to implement: This involves researching and selecting error correction techniques that best suit the specific encryption algorithms and network. For example, if an encryption algorithm is vulnerable to attacks that can recover the encryption key, then using an error correction technique that relies on the same key could also be vulnerable. In this case, it may be better to use an error correction technique that is based on a different type of key or that uses a different type of algorithm altogether.
2) Select an encryption technique: Similarly, research and select an encryption technique that suits the former. Factors to consider when selecting an encryption technique include the level of security required, the type of data being transmitted, and the available processing resources.
3) Integrate the techniques to form a combined approach: This involves implementing the selected error correction and encryption techniques together in a way that minimizes the vulnerabilities and limitations of each technique. For example, the encryption algorithm can be applied first, followed by the error correction technique, or the two techniques can be applied simultaneously. The goal is to achieve a balance.
4) Test the network and simulate it on MATLAB to ensure it works with the factors and parameters in mind: It is important to test the combined approach thoroughly to ensure that it works as expected and is secure. This can be done through simulations on MATLAB or other similar software. Factors to consider during testing include the complexity, the threat model, and also checking for regulatory compliance.
5) Continuously monitor: Finally, it is essential to continuously monitor the network to detect any issues or vulnerabilities that may arise. This can involve using monitoring tools or software to detect and correct errors in real-time, as well as updating the encryption and error correction techniques as needed to stay ahead of potential threats. Regular testing and updates can help ensure that the network remains secure and reliable over time.

More specifically the approach is to first encrypt the data using an optimal technique with a secret/private key and then encode the encrypted data using a profound algorithm when the receiver receives the encoded data, they first decode to correct any errors that may have occurred during data transmission then decryption of the data using the same private key that was used for encryption.

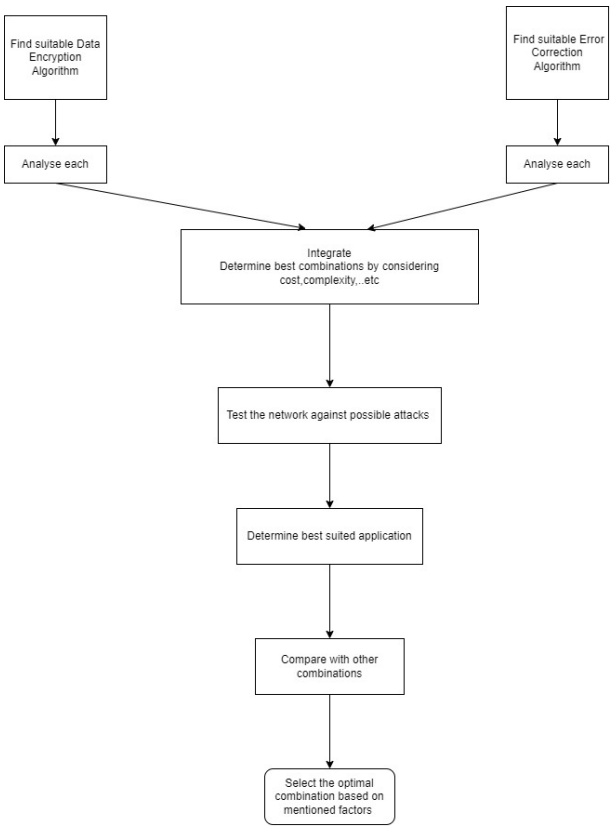

Fig. 2. Flowchart showing various methods

Justification: The proposed methodology combines error correction and data encryption techniques to provide a secure data communication system. Error correction techniques ensure that the data transmitted is error-free, thereby lessening breaches and vulnerability, while data encryption techniques ensure that the data is secure and can only be accessed by authorized users. This methodology will provide a robust and reliable communication system resistant to common types of errors and attacks.

IMPLEMENTATION

For the implementation of the above we used the language Python to run the codes. As our focus is more on the error correction codes rather than the encryption algorithms which occur at higher levels of networks, we mainly focused on finding optimal combinations of error correction codes which have best accuracy and make up for the errors the other ones have missed. First let us analyse the type of errors and which ones are suited for them. We will be measuring four parameters:

1) Bit Error Rate (BER): BER represents the probability of a single bit being received incorrectly. All three error correction codes mentioned above can help reduce the BER.
2) Frame Error Rate (FER): FER represents the probability of an entire frame (or block) being received with errors. Reed-Solomon and Turbo codes perform better than Hamming codes in reducing FER.
3) Overhead: Overhead refers to the additional bits introduced by the error correction code. Hamming codes have the lowest overhead, followed by Reed-Solomon codes, and Turbo codes have the highest overhead.
4) Latency: Latency refers to the time delay introduced by the error correction code during encoding and decoding. Hamming codes have the lowest latency, followed by Reed-Solomon codes, and Turbo codes have the highest latency due to their more complex encoding and decoding algorithms.
We prefer to look for an optimal solution where combination has low latency, with low overhead and reduced BER and FER. We will find out a combination that does so.

OBSERVATION

1) Reed Solomon+AES+RSA: The combination of Reed-Solomon error correction, RSA encryption, and AES encryption provides a robust and secure mechanism for data transmission. Reed-Solomon (RS) error correction is an algorithm used to detect and correct errors in data transmission. It operates on blocks of data and adds redundant information, allowing the receiver to identify and correct errors introduced during transmission. In the provided code, RS error correction is used to add error correction bytes to the plain text before encryption. These error correction bytes help in recovering the original data even if some errors occur during transmission.
Advantages: RS detects errors in bit and frame levels more precisely. Reed-Solomon , a powerful technique as mentioned above This hybrid encryption benefits both symmetric and asymmetric encryption ,where AES is for bulk encryption and RSA handles key exchange safely
Limitations: BER , FER require additional signalling in the data stream in order to determine error rate accurately RS codes raises complexity Hybrid approach may add additional computational cost
Here's a general overview of the performance characteristics:
   a) Bit Error Rate (BER): The combined Reed-Solomon and Hybrid encryption coding scheme can achieve a lower BER compared to using either code alone. Depending on the specific implementation, channel conditions, and signal-to-noise ratio (SNR), the BER can be significantly reduced. However, it's challenging to provide an exact BER value without specific details of the system.
   b) Frame Error Rate (FER): The FER of the combined coding scheme will also be improved compared to using individual codes. By leveraging the error correction capabilities of Reed-Solomon , the FER can be reduced. The exact FER value depends on various factors, including the coding parameters, modulation scheme, channel conditions, and SNR.
   c) Latency: The latency introduced by combining these codes can be influenced by the encoding and decoding complexity of the AES and RSA codes. These hybrid encryption codes generally have higher encoding and decoding complexity

compared to Reed-Solomon codes. As a result, the latency may increase when using the combined coding scheme. The specific latency values will depend on the implementation and system configuration.

d) Overhead: This combination introduces additional overhead due to the redundancy introduced by each code. Reed-Solomon codes add extra check symbols.

It's important to note that the actual values for BER, FER, latency, and overhead in a combined system can vary based on the specific implementation, channel conditions, and other factors. These values need to be evaluated through simulations or real-world testing to assess the performance and trade-offs of the combined coding scheme in a particular system or application.

2) Checksum,Reed-Solomon and AES-GCM: The checksum is used in data communication to ensure data integrity and detect errors.AES and GCM encryption technique is an efficient combination for encryption. Let's discuss their advantages , disadvantages and how these techniques work and their implications regarding BER (Bit Error Rate), FER (Frame Error Rate), overhead, and latency.
Advantages: Reed-Solomon codes are well-established and widely used for error correction. They can effectively recover data even in the presence of a significant number of errors. AES-GCM is a strong symmetric encryption algorithm that provides both confidentiality and authentication. It is widely adopted and considered secure.
Disadvantages: The use of a checksum alone may not provide as fine-grained error detection. AES-GCM requires a shared symmetric key between the sender and receiver, which may introduce additional key management challenges.
Now, let's consider the implications of this combination in terms of BER, FER, overhead, and latency:

a) Bit Error Rate (BER): The checksum can help reduce the BER. By using this combination, errors are detected at different levels, increasing the chances of error detection. Therefore, the overall BER is likely to be lower compared to using either technique individually.

b) Frame Error Rate (FER): The FER is influenced by the BER. By reducing the BER, the FER is also likely to decrease. However, it's important to note that the FER can be affected by other factors, such as the quality of the communication channel and the error correction mechanisms in place.

c) Overhead: Checksum introduce additional bits into the transmitted data, which increases the overhead.It adds a fixed-size checksum value. The overhead is directly proportional to the size of the data being transmitted. The larger the data, the more significant the overhead. However, the benefits of enhanced error detection usually outweigh the overhead cost.

d) Latency: The detection technique does not directly impact latency since latency refers to the time delay in data transmission. However, the inclusion of additional bits in the transmitted data increases the overall transmission time. This increase in transmission time can indirectly affect the overall latency, especially in real-time or time-sensitive applications.

In summary, the combination checksum provides improved error detection capabilities, resulting in a lower BER and FER. However, it introduces additional overhead due to the inclusion of extra bits. While it does not directly affect latency, the increased transmission time due to the added bits can have an indirect impact on overall latency.

RESULTS AND CONCLUSION

Checksum,Reed-Solomon and AES-GCM offers straightforward error detection ,strong correction ,whereas Reed Solomon, AES with RSA offers fine-grained detection and efficient correction but with a bit higher complexity .

So,finally if a system prioritizes simplicity and efficiency with a good level of security and when real-time processing is required ,first combination is preferable due to its fast processing and low latency. For instance, case of Video streaming, message applications, large file transfers.Whereas if the system requires detailed error analysis ,secure key exchange ,second combination is preferable. For instance , satellite communications, financial transactions , transferring medical records and broadcasting systems.

Another good combination would be Reed Solomon+AES+RSA+Turbo.

This combination finds applications in various domains, such as wireless communication systems, satellite communication, secure file transfer protocols, and digital rights management (DRM). In wireless communication systems, the combination is used to ensure secure and error-free transmission of data over unreliable channels. Satellite communication systems leverage this combination to protect sensitive data during transmission and prevent unauthorized access.

However, we must note that this is only for extremely precise corrections as both error correction codes together have a very high overhead and latency. And it may not be worth the time to use. Thus it is only used where data is crucial,for example: satellite communication.

In conclusion, the combination of error correction and encryption techniques forms a comprehensive solution for securing data communication. By integrating these two approaches, we establish a robust framework that ensures the accuracy of transmitted information while safeguarding its confidentiality. This powerful synergy strengthens the trustworthiness and reliability of data networks, offering enhanced protection against errors and unauthorized access. As we navigate the evolving landscape of digital communication, leveraging this optimal combination empowers us to maintain the integrity and security of our data in various domains.

REFERENCES

[1] B. A. Forouzan, Data communications and networking (mcgraw-hill) (2007).

[2] N. C, Survey on network security with cryptography, https://www.ijser.org/researchpaper/Survey-on-Network-Security-with-Cryptography.pdf (2019).

[3] A. M. Abdelaziz, E. Abdelwanees, A. D. Elbayoumy, Securing the space data link communication protocol of earth observation satellites (2019). doi:10.1109/ICICIS46948.2019.9014846.

[4] K. W. R. James F Kurose, Computer Networking, A top-down approach (2000).

[5] E. Bertino, Data security and privacy concepts (2016). doi:10.1109/COMPSAC.2016.89.
URL https://ieeexplore.ieee.org/document/7552042

[6] A. M. Abukari, An enhanced error detection and correction scheme for enterprise resource planning (erp) data storage, Journal of Advances in Computer science and Maths 36 (2021). doi:10.9734/jamcs/2021/v36i930405.
URL https://journaljamcs.com/index.php/JAMCS/article/view/1602

[7] M. Dener, O. F. Bay, Teenysec: a new data link layer security protocol for wsns, Security and Communication Networks 9 (18) (2016) 5882–5891.

[8] N. Islam, Z. Shahid, W. Puech, Denoising and error correction in noisy aes-encrypted images using statistical measures (2016). doi:https://doi.org/10.1016/j.image.2015.11.003.
URL https://www.sciencedirect.com/science/article/pii/S0923596515002003

[9] L. Ning, L. Kanfeng, L. Wenliang, D. Zhongliang, A joint encryption and error correction method used in satellite communications (2014). doi:10.1109/CC.2014.6825260.

[10] T. Rao, Joint encryption and error correction schemes, ACM SIGARCH Computer Architecture News 12 (3) (1984) 240–241.

[11] W. Stallings, Data and Computer Communications (2007).

[12] I. B. Djordjevic, Physical-layer security and quantum key distribution, Springer, 2019.

[13] W. H. Jeong, B.-G. Yeo, K.-H. Kim, S.-H. Park, S.-W. Yang, J.-S. Lim, K.-S. Kim, Performance analysis of the encryption algorithms in a satellite communication network based on h-arq, The Journal of The Institute of Internet, Broadcasting and Communication 15 (1) (2015) 45–52.

[14] J. M. Hamamreh, M. Yusuf, T. Baykas, H. Arslan, Cross mac/phy layer security design using arq with mrc and adaptive modulation, in: 2016 IEEE Wireless Communications and Networking Conference, IEEE, 2016, pp. 1–7.

[15] O. S. Younes, Securing arp and dhcp for mitigating link layer attacks, Sādhanā 42 (2017) 2041–2053.

[16] Z. Chen, L. Yin, Y. Pei, J. Lu, Codehop: physical layer error correction and encryption with ldpc-based code hopping, Science China Information Sciences 59 (2016) 1–15.

[17] S. Mahmood, S. M. Mohsin, S. M. A. Akber, Network security issues of data link layer: An overview, in: 2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2020, pp. 1–6. doi:10.1109/iCoMET48670.2020.9073825.

[18] G. Yang, L. Dai, Z. Wei, Challenges, threats, security issues and new trends of underwater wireless sensor networks, Sensors 18 (11) (2018) 3907.

[19] D. Purwanto, Optimization of data security system control with crc (cyclic redundancy check) algorithm, Budapest International Research and Critics Institute-Journal (BIRCI-Journal) 4 (3) 4635–4642.

[20] A. Couvreur, M. Lequesne, On the security of subspace subcodes of reed–solomon codes for public key encryption, IEEE Transactions on Information Theory 68 (1) (2021) 632–648.

[21] O. Aitsab, R. Pyndiah, Performance of reed-solomon block turbo code, in: Proceedings of GLOBECOM'96. 1996 IEEE Global Telecommunications Conference, Vol. 1, 1996, pp. 121–125 vol.1. doi:10.1109/GLOCOM.1996.594345.

[22] B. Tahir, S. Schwarz, M. Rupp, Ber comparison between convolutional, turbo, ldpc, and polar codes, in: 2017 24th International Conference on Telecommunications (ICT), 2017, pp. 1–7. doi:10.1109/ICT.2017.7998249.

****** END ******

- [1]–[22]