Implementing Java Encryption for File Transfer

ICTU Student 152
Thai Nguyen University of Information and Communication Technology
Thai Nguyen University, Thai Nguyen, Vietnam
student152@ictu.edu.vn

***Abstract*—The manuscript examines information security and the use of cryptography in Java-based file transfer. It reviews cryptographic requirements, RSA and DES, Java security facilities, socket communication, and the NetBeans development environment. A socket example illustrates a server that receives text from clients and returns the reversed text; this example is separate from encrypted file transmission. The available source ends at the heading for the chapter on constructing the encryption program. It therefore documents the theoretical basis and development tools, but does not provide implementation results, security tests, or evidence that an encrypted file-transfer application was completed.**

***Index Terms*—Java, cryptographic, file transmission, RSA, DES, NetBeans**

I. Introduction

File exchange over a network raises questions about confidentiality and the choice of encryption methods. The thesis frames this problem through an introductory review of information security and classical cryptographic criteria, including key distribution, algorithmic complexity, and the speed of encryption and decryption. It then turns to Java as a possible development environment for a file-transfer application. The scope of the supplied manuscript is important: its readable text contains the literature and platform review, while the next chapter begins with a program-construction heading and supplies no corresponding implementation report [1].

II. Methods

The author surveys the mathematical and security background of cryptography and describes RSA and DES at a conceptual level. A separate chapter reviews Java, its security mechanisms, network sockets, and NetBeans as an integrated development environment. The text includes a client-server socket exercise in which a server accepts connections, receives a string, reverses it, and sends the result back. That exercise demonstrates socket communication; it is not presented as an encrypted file-transfer test. The source does not provide a reproducible experimental protocol for measuring confidentiality, throughput, or file integrity [1].

III. Results

The available manuscript produces a structured explanation of security concepts, RSA, DES, Java facilities, socket programming, and the NetBeans workflow. It includes sample server and client code and instructions for running a simple text-exchange exercise. These examples show how a client and server communicate but do not establish that files were encrypted and transferred. The last readable material is the heading of a chapter on construction of the proposed program. No screenshots, test records, benchmark values, or completed program outcomes for encrypted file transfer appear in the available text [1].

IV. Discussion

The socket demonstration and the proposed encrypted file-transfer application must be kept distinct. Returning a reversed string is useful for illustrating client-server communication, but it provides no evidence of cryptographic protection. Likewise, discussing RSA, DES, and Java security features does not by itself show that those components were correctly integrated. The available text supports a review of design ingredients and tools, not an assessment of security, performance, or deployment. A future implementation report would need to document the encryption flow, the file-transfer workflow, and observable tests of both [1]. Earlier ICTU undergraduate work by Đinh Thị Thúy addressed encryption algorithms for protected file transfer [2].

V. Conclusion

The source establishes the conceptual and programming background for an encryption application intended for file transfer. Its documented contribution is an account of information security, RSA and DES, Java development facilities, and socket communication, together with a simple networking exercise. Because the supplied manuscript stops as the program-construction chapter begins, the summary does not infer a finished secure-transfer system or successful testing. This distinction preserves what the document actually shows and identifies the missing implementation evidence that would be required to evaluate the proposed application [1].

Supervisor: ICTU Teacher 95, Thai Nguyen University of Information and Communication Technology.

References

[1] ICTU Student 152, ”Implementing Java Encryption for File Transfer,” internal undergraduate thesis (original in Vietnamese), Thai Nguyen University of Information and Communication Technology, 2010.

[2] Đinh Thị Thúy, ”Study of Cryptographic Algorithms,” internal undergraduate thesis, Thai Nguyen University of Information and Communication Technology, 2008.