A

SYNOPSIS

of

MINOR PROJECT

on

Digital Signature verification system

Submitted by

DASHARATH KUMAR

Project Guide
Dr. Ajay Kumar Sharma

Head of Department
Dr. Mayank Patel

Geetanjali Institute of Technical Studies, Dabok , Udaipur (Raj.)
Department of Computer Science and Engineering
October,2023

Problem Statement: The project aims to address the need for a reliable system to verify digital signatures using Python. Digital signatures are crucial for ensuring the authenticity and integrity of digital communications and documents.

Brief Description: This project involves creating a tool to generate and verify digital signatures using RSA keys and the PKCS1_v1_5 standard. The project utilizes Python libraries like argparse for command-line argument parsing and PyCryptodome for cryptographic operations.

Objective and Scope:

- Develop a Python-based tool that can generate and verify digital signatures.
- Implement RSA key generation, digital signature creation, and verification processes. The tool should handle different hashing algorithms (SHA256, SHA384, SHA512).

Methodology:

1. Key Generation: Generate RSA private and public keys using the Crypto.PublicKey module.
2. Hashing Data: Hash data using the specified algorithm (SHA256, SHA384, SHA512) with the Crypto.Hash module.
3. Generating Signature: Use the RSA private key to sign the hashed data.
4. Verifying Signature: Use the RSA public key to verify the digital signature against the original data.

Hardware and Software Requirements:

- Hardware: Standard computer with Python environment setup.
- Software: Python 3.x, PyCryptodome library, and standard Python libraries (argparse, sys).

Technologies:

- Programming Language: Python
- Libraries: PyCryptodome for cryptographic functions, argparse for command-line argument parsing.

Testing Techniques:

- Unit Testing: Test individual functions for hashing, key generation, signature generation, and verification.
- Integration Testing: Ensure the complete workflow from key generation to signature verification works seamlessly.
- Edge Case Testing: Validate the tool's behavior with various input sizes and types, and handle errors gracefully.

Project Contribution: The project provides a practical tool for verifying digital signatures, which can be used to enhance the security of digital communications. It demonstrates the application of cryptographic principles using Python, contributing to educational and practical knowledge in the field of cybersecurity.