**Course – Cryptography and System Security (CSS)**

|  |  |
| --- | --- |
| **UID** | 2021300038 |
| **Name** | Atharva Gite |
| **Class and Batch** | BE Computer Engineering - Batch VIII |
| **Date** | 10/10/2024 |
| **Lab #** | 5 |
| **Aim** | To generate and calculate hashes and checksum files. |
| **Problem Definition** | This experiment has two parts: a) Simulation of SHA-1 on Virtual Lab - https://cse29-iiith.vlabs.ac.in/exp/hash-functions/index.html and b) Implementation of SHA-512 Function for four cases given on page no. 354 Fig. 11.2 by using any symmetric and any public-key cryptosystem. |
| **Theory** | SHA-1 (Secure Hash Algorithm 1) is a cryptographic hash function that produces a 160-bit hash value from input data of any size. Developed by the National Security Agency (NSA) and released by the National Institute of Standards and Technology (NIST) in 1995, SHA-1 was widely used in various security applications, including digital signatures, certificates, and data integrity checks.  The algorithm processes data in blocks and uses a series of logical operations to generate a hash. This hash is intended to be a unique fingerprint of the input data, meaning even small changes in the input should produce drastically different hashes.  SHA-1 (Secure Hash Algorithm 1) operates through a series of mathematical transformations and logical operations on input data to produce a 160-bit hash value. Its mathematical structure:   1. **Input Padding**: The input data is padded to ensure its length is congruent to 448 modulo 512. This is followed by appending a 64-bit representation of the original message length. 2. **Message Parsing**: The padded message is divided into 512-bit blocks, which are processed sequentially. 3. **Initialization**: SHA-1 uses five 32-bit words (A, B, C, D, E) initialized to specific constants. 4. **Processing**: Each 512-bit block undergoes 80 rounds of processing. The algorithm uses logical functions (such as AND, OR, NOT, and XOR) and modular arithmetic. The main operations involve:    * **Word Expansion**: Expanding the 16 initial 32-bit words into 80 words.    * **Mixing**: Each round updates the five words using a combination of the current words, the expanded word, and a constant derived from the round number. 5. **Final Hash**: After processing all blocks, the final hash is obtained by concatenating the five 32-bit words, resulting in a 160-bit output.   SHA-1's design emphasizes diffusion and avalanche effects, ensuring small changes in the input produce significant changes in the output.  SHA-512 (Secure Hash Algorithm 512) is a cryptographic hash function that belongs to the SHA-2 family, designed by the National Security Agency (NSA) and published by the National Institute of Standards and Technology (NIST) in 2001. It generates a 512-bit hash value, commonly represented as a 128-digit hexadecimal number.  SHA-512 operates on input data by processing it in 1024-bit blocks. The algorithm begins with padding the input message to ensure its length is congruent to 896 modulo 1024, followed by appending a 128-bit representation of the original message length. This ensures that the input is processed correctly.  The core of SHA-512 consists of initializing eight 64-bit hash values, derived from the fractional parts of the square roots of the first 64 prime numbers. The algorithm then expands the input into 80 words and performs 80 rounds of processing. Each round utilizes logical functions and modular arithmetic to update the hash values, promoting diffusion and complexity.  Its mathematical structure:   1. **Input Padding**: The input message is padded to ensure its length is congruent to 448 modulo 512. This is followed by appending a 128-bit representation of the original message length. 2. **Message Parsing**: The padded message is divided into 1024-bit blocks, which are processed one at a time. 3. **Initialization**: SHA-512 initializes eight 64-bit hash values (H0 through H7) using specific constants derived from the fractional parts of the square roots of the first 64 prime numbers. 4. **Processing**: Each 1024-bit block undergoes 80 rounds of processing. Key operations include:    * **Word Expansion**: The initial 16 64-bit words are expanded into 80 words using bitwise operations and modular arithmetic.    * **Mixing**: Each round updates the hash values using a combination of the current values, the expanded words, and constants derived from the first 80 prime numbers. 5. **Final Hash**: After processing all blocks, the final hash is obtained by concatenating the eight 64-bit hash values, resulting in a 512-bit output.   SHA-512 is designed to be secure against collision and preimage attacks, making it suitable for applications requiring high security, such as digital signatures, certificates, and data integrity verification. Its larger output size provides an added layer of security compared to shorter hash functions, ensuring robustness against brute-force attacks. |
| **Output** | 1. **Simulation of SHA-1 on Virtual Lab** 2. **SHA-512 examples from Cryptography and Network Security**   **a)**      **b)** |