**Rivest Shamir Adleman………RSA**

The RSA algorithm is a public-key signature algorithm developed by Ron Rivest, Adi Shamir, and Leonard Adleman. Their paper was first published in 1977. M - Plaintext H - Hash function h - Hash digest ‘+’ - Bundle both plaintext and digest E – Encryption D – Decryption RSA can also encrypt and decrypt general information to securely exchange data along with handling digital signature verification. The image above shows the entire procedure of the RSA algorithm. You will understand more about it in the next section.RSA in Data Encryption When using RSA for encryption and decryption of general data, it reverses the key set usage. Unlike signature verification, it uses the receiver’s public key to encrypt the data, and it uses the receiver’s private key in decrypting the data. Thus, there is no need to exchange any keys in this scenario. There are two broad components when it comes to RSA cryptography,

**•** **Key Generation:** Generating the keys to be used for encrypting and decrypting the data to be exchanged.

**• Encryption/Decryption Function:** The steps that need to be run when scrambling and recovering the data. Key Generation You need to generate public and private keys before running the functions to generate your ciphertext and plaintext. They use certain variables and parameters, all of which are explained below: Choose two large prime numbers (p and q) Calculate n = p\*q and z = (p-1)(q-1) Choose a number e where 1 < e < z Calculate d = e-1mod(p-1)(q-1) You can bundle private key pair as (n,d) You can bundle public key pair as (n,e) Encryption/Decryption Function Once you generate the keys, you pass the parameters to the functions that calculate your ciphertext and plaintext using the respective key. If the plaintext is m, ciphertext = me mod n. If the ciphertext is c, plaintext = cd mod n To understand the above steps better, you can take an example where p = 17 and q=13. Value of e can be 5 as it satisfies the condition 1 < e < (p-1)(q-1). N = p \* q = 221 D = e-1mod(p-1)(q-1) = 29 Public Key pair = (221,5) • Private Key pair = (221,29) • If the plaintext(m) value is 10, you can encrypt it using the formula me mod n = 82. • To decrypt this ciphertext(c) back to original data, you must use the formula cd mod n = 29.

Merkle Puzzles The Merkle puzzle is a cryptographic construction used in information security to securely establish a shared secret key between two parties over an insecure communication channel. It was introduced by Ralph Merkle in 1974 and provides a solution to the key exchange problem. 1. Puzzle Generation: The sender (Alice) generates a large number of encrypted puzzle pieces. Each puzzle piece is encrypted with a different candidate key. 2. Puzzle Transmission: Alice sends the encrypted puzzle pieces to the receiver (Bob) over the insecure channel. 3. Puzzle Solving: Bob selects a puzzle piece from the received set and attempts to decrypt it using a candidate key. He repeats this process with different puzzle pieces until he successfully decrypts one of them. 4. Key Exchange: Once Bob successfully decrypts a puzzle piece, he informs Alice by sending the index or identifier of the puzzle piece. 5. Key Confirmation: Alice checks the index or identifier received from Bob and reveals the corresponding candidate key used to encrypt the puzzle piece. 6. Shared Key: Both Alice and Bob now possess the shared secret key used for secure communication. The Merkle puzzle provides security by making it computationally infeasible for an eavesdropper to discover the shared secret key by decrypting multiple puzzle pieces. Additionally, the puzzle complexity can be adjusted by varying the number of puzzle pieces generated, making it resistant to brute-force attacks. It's worth noting that the Merkle puzzle is a conceptual construction and may not be the most practical solution for key exchange in modern cryptographic protocols. It serves as an important milestone in the development of cryptographic primitives and has paved the way formore efficient and secure key exchange mechanisms like Diffie-Hellman key exchange and public-key cryptography.