**Trapdoor Permutations and RSA**

A trapdoor permutation is a cryptographic primitive that allows for efficient computation in one direction, while making the inverse computation computationally difficult without specific knowledge called the "trapdoor." It forms the basis for various cryptographic schemes, including public key encryption and digital signatures. Here's an explanation of how trapdoor permutations work and an example:

1. **Definition:** A trapdoor permutation is a function that is easy to compute in one direction but hard to invert without knowledge of additional information (the trapdoor). Given the output of the function, it should be computationally difficult to determine the input without the trapdoor information.

2. **Key Generation:** To create a trapdoor permutation, a specific algorithm is used to generate a pair of keys: a public key and a private key. The public key is made available to everyone, while the private key is kept secret.

3. **Encryption:** Given a message M, the trapdoor permutation can be applied with the public key to generate a cipher-text C. The encryption process is efficient and can be performed by anyone who has access to the public key.

4. **Decryption:** The decryption process is where the trapdoor comes into play. With the private key (which includes the trapdoor information), the recipient can efficiently compute the inverse of the trapdoor permutation and obtain the original message M from the cipher-text C.