Connect what we now understand about cryptography with few areas of current and future application.

This study block connects what we now understand about cryptography with a few areas of current and future application as a little potpourri of subject matter.

**Stream cyphers**

We have only looked at the thing to be encrypted as a block of text, but what if the data is not in a block, but a stream, i.e. time is also a factor? For example, when a video provider wants to encrypt a movie as they send it to you, how might that change the possibilities?

**Quantum computing and cryptography**

We have no doubt all heard that quantum computers will some day be able to break the cryptographic algorithms we have studied and that we use. Can we get some perspective on how this can be, how much of a threat it is, and what we will do when that day comes closer?

**Cryptography in distribute ledgers**

It is nice to know that security is not just about locking things down, but can be about opening up new possibilities. Distributed ledgers are very useful things that are a recently invented application of cryptographic principles. What are they and what promises do they potentially hold?

video

Collection of data with beginning and end, all put together in order.

Pile of data which looks like a file.

English text, I need to pass quickly to someone in Swedish Secret Service.

Put message in translation machine.

Compress message.

Put it into the encrypt box and then process it.

Consider it to be a data stream.

Might not be able to see the end.

Need a different kind of machine. Machine still translates for me. Feed it through a translator. Then need a similar compression machine. Ultimately

then the encryption.

Stream cipher

Same keystream used for encryption and decryption.

Recipient who knows secret key, generates a keystream and combines it with ciphertext to get the original text.

Stream ciphers are faster for certain applications, especially when dealing with streaming data.

Stream ciphers are a type of symmetric key cipher where plaintext is combined with a pseudorandom cipher digit stream, or keystream. In simpler terms, a stream cipher encrypts data one bit or byte at a time, as opposed to handling larger blocks of data at once like block ciphers. Here's an overview of how stream ciphers work and their characteristics:

1. **Sequential Processing**:
   * Stream ciphers encrypt data one bit or byte at a time, making them suitable for environments where data comes in a continuous stream and the total size of the data may not be known in advance.
2. **Keystream Generation**:
   * The core of a stream cipher is the generation of a keystream, which is a sequence of bits used for encryption. This keystream is produced using a secret key.
   * The keystream should ideally be as long as the plaintext message that needs to be encrypted.
3. **Encryption and Decryption**:
   * **Encryption**: In the encryption process, each bit of the plaintext is encrypted by combining it with the corresponding bit from the keystream. The combination is typically done using the XOR (exclusive or) operation.
   * **Decryption**: The same keystream used for encryption is used for decryption. The recipient, who knows the secret key, generates the identical keystream and combines it with the ciphertext using XOR, recovering the original plaintext.
4. **Types of Stream Ciphers**:
   * **Synchronous Stream Ciphers**: In these ciphers, the keystream is generated independently of the plaintext and ciphertext messages. Encryption and decryption must be perfectly synchronized.
   * **Asynchronous (Self-Synchronizing) Stream Ciphers**: These use a few of the previous bits of the ciphertext to generate the next bit of the keystream, offering some resilience to lost or altered bits in the ciphertext.
5. **Advantages**:
   * **Speed and Efficiency**: Stream ciphers are generally faster and more efficient than block ciphers for certain applications, especially when dealing with streaming data.
   * **Low Latency**: Ideal for real-time applications where data needs to be encrypted and decrypted with minimal delay.
6. **Disadvantages**:
   * **Security Concerns**: The security of a stream cipher depends heavily on the randomness and unpredictability of the keystream. If the keystream can be predicted or is not properly random, the cipher can be vulnerable to attacks.
   * **Key Management**: Securely managing the keys and ensuring they are not reused is crucial for maintaining the cipher's security.
7. **Applications**:
   * Stream ciphers are often used in scenarios where data is transmitted over a network, particularly in real-time applications like voice and video streaming, where the efficiency and low latency of stream ciphers are beneficial.

In summary, stream ciphers are an efficient and fast method for encrypting data in a streaming manner, bit by bit or byte by byte. They are particularly useful for real-time encryption scenarios but require careful handling of the keystream and key management to ensure security.

High diffusion – information from the plaintext is diffused into several ciphertext symbols. One ciphertext block may depend on several plaintext

letters.

|  |  |  |
| --- | --- | --- |
|  | **Stream** | **Block** |
| **Advantages** | * *Speed of transformation.* Because each symbol is encrypted without regard for any other plaintext symbols, each symbol can be encrypted as soon as it is read. Thus, the time to encrypt a symbol depends only on the encryption algorithm itself, not on the time it takes to receive more plaintext. * *Low error propagation.* Because each symbol is separately encoded, an error in the encryption process affects only that character. | * *High diffusion.* Information from the plaintext is diffused into several ciphertext symbols. One ciphertext block may depend on several plaintext letters. * *Immunity to insertion of symbol.* Because blocks of symbols are enciphered, it is impossible to insert a single symbol into one block. The length of the block would then be incorrect, and the decipherment would quickly reveal the insertion. |
| **Disadvantages** | * *Low diffusion.* Each symbol is separately enciphered. Therefore, all the information of that symbol is contained in one symbol of ciphertext. * *Susceptibility to malicious insertions and modifications.* Because each symbol is separately enciphered, an active interceptor who has broken the code can splice pieces of previous messages and transmit a spurious new message that may look authentic. | * *Slowness of encryption.* The person or machine doing the block ciphering must wait until an entire block of plaintext symbols has been received before starting the encryption process. * *Padding.* A final short block must be filled with irrelevant data to make a full-sized block. * *Error propagation.* An error will affect the transformation of all other characters in the same block. |

In the context of information security, "diffusion" refers to a principle where the effect of a single change in the input or key of a cryptographic algorithm spreads out and affects many parts of the output.

The idea is rooted in Claude Shannon's concept of "confusion and diffusion" as strategies for secure communication. Here's a more detailed look at diffusion in information security:

1. **Spreading Changes**: Diffusion ensures that a change in any single bit of the plaintext or key results in a significant and unpredictable change in many bits of the ciphertext. This makes it more difficult for attackers to analyze and predict the relationship between the plaintext and the ciphertext.
2. **Enhancing Security**: By ensuring that the output (ciphertext) doesn't reveal any obvious patterns related to the input (plaintext), diffusion strengthens the security of the encryption. It reduces the chances of successful cryptanalysis, where attackers attempt to derive the plaintext or the key from the ciphertext.
3. **Algorithm Design**: In designing cryptographic algorithms, diffusion is often achieved through various techniques like substitution, transposition, and more complex operations. For example, in block ciphers, the use of S-boxes (substitution boxes) and permutation steps contribute to achieving diffusion.
4. **Against Statistical Attacks**: Diffusion is particularly important for defending against statistical attacks. If the ciphertext is too similar to the plaintext, statistical methods can be used to deduce patterns and potentially break the encryption.
5. **Complement to Confusion**: While diffusion deals with the dispersion of the effects of a single input change over the entire output, confusion, its counterpart, focuses on making the relationship between the ciphertext and the encryption key as complex and as hidden as possible. Both principles are vital for a robust encryption system.

In essence, diffusion in information security is about ensuring that the encrypted output (ciphertext) is as unrelated and as unpredictable as possible in relation to the input (plaintext) to thwart efforts at decryption without the correct key.

Top of Form

**CBC Mode Explained**

CBC (Cipher Block Chaining) mode is a method of encrypting data used in conjunction with block ciphers.

Here's how it works:

* **Initial Vector (IV)**: CBC mode starts with an initial vector (IV), which is a block of data of the same size as the block cipher's block size. The IV is typically random and unique for each encryption session.
* **Encryption Process**: Each block of plaintext is XORed with the previous ciphertext block before being encrypted. For the first block, the IV is used as the "previous ciphertext block."
* **Chain Reaction**: This chaining ensures that identical blocks of plaintext result in different ciphertext blocks, enhancing security. It also means that an error in one ciphertext block affects the decryption of all subsequent blocks, leading to error propagation.
* **Decryption Process**: During decryption, each block of ciphertext is decrypted, and the result is XORed with the previous ciphertext block to recover the plaintext. For the first block, the IV is used for the XOR operation.

CBC mode adds an additional layer of security to block cipher encryption by ensuring that the same plaintext block doesn't result in the same ciphertext block, thus preventing certain types of cryptographic attacks.

Initial vector – block of data same size as block cipher’s block size.