Feistel Cipher Modes

Assignment #6

Gabriel Lee

A00884904
COMP 7402

INTRODUCTION

The Feistel architecture is widely used in modern ciphers. It has various different modes that vastly affect the confusion and diffusion properties. Most commonly the ECB mode is used to introduce modes as it is the most straightforward. But ECB mode has its weaknesses. Thus other modes were developed to improve it. The two other widely used modes are CBC and CTR. Analyzing these modes' confusion and diffusion properties can help identify their benefits and its drawbacks.

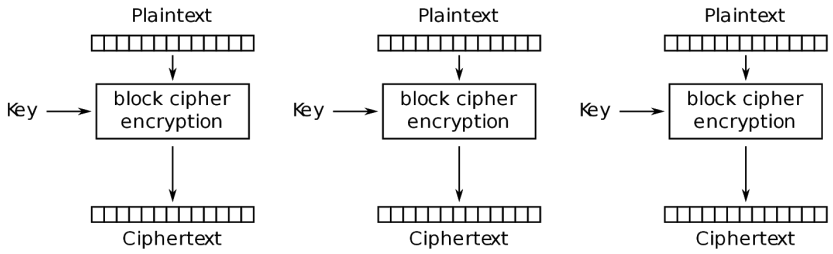
ANALYSIS

There are numerous modes of operation, but some of the most popular modes are: ECB, CBC, and CTR. These are all block ciphers which use a symmetrical key to encrypt and decrypt a block of plaintext.

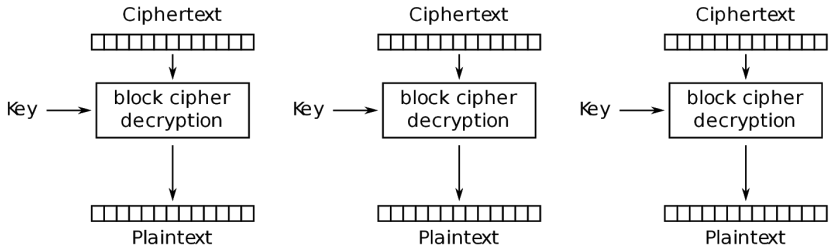
The strength of these modes can be visualized using images. The image above will serve as a base for comparing the different modes confusion and diffusion properties. In addition to images, the Avalanche effect of each mode will be analyzed for both Strict Plaintext Avalanche Criterion (SPAC) and Strict Key Avalanche Criterion (SKAC). After analyzing the strength, the speed of the modes will help understand the pros and cons.

Electronic Codebook (ECB)

The ECB mode is one of the simplest modes. It divides up the plaintext into a determined block size and performs encryption on them using the provided key. The decryption is the same process: perform decryption on the ciphertext block with the encryption key.



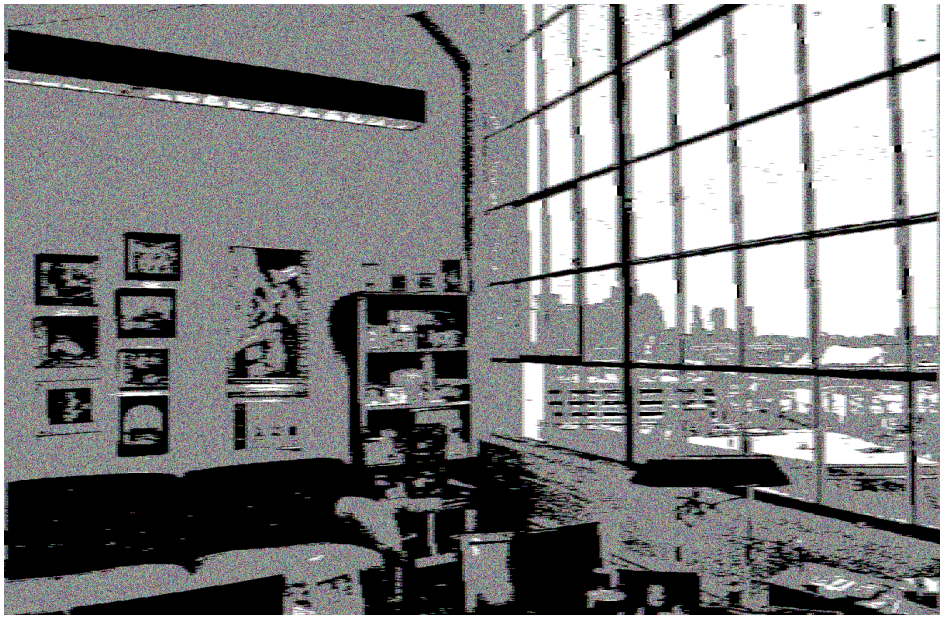
Electronic Codebook (ECB) mode encryption



Electronic Codebook (ECB) mode decryption

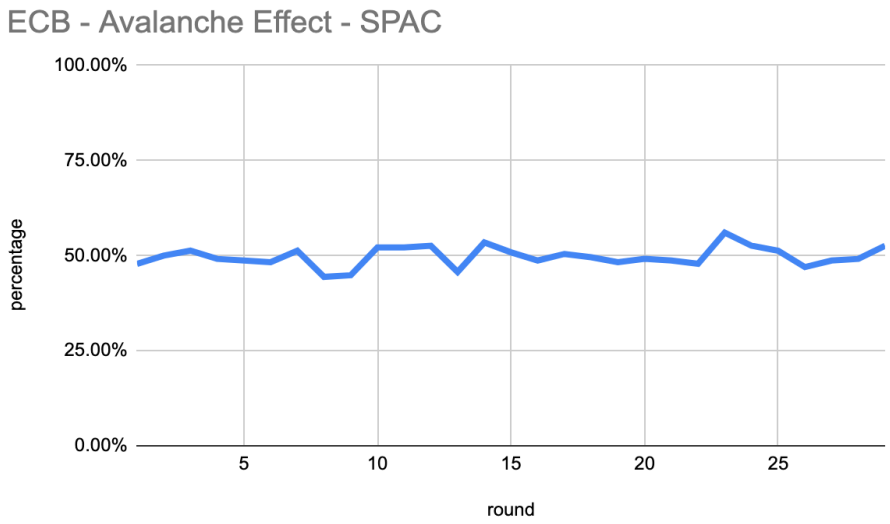
Visualization

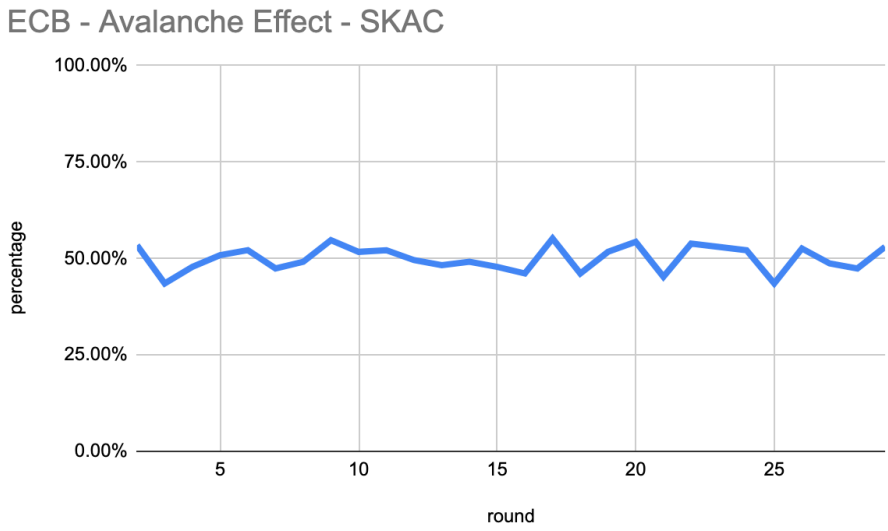
The ECB mode is a big weakness when it comes to diffusion. Because the plaintext block is encrypted to the same ciphertext block, the patterns are not very well hidden. The following images are encrypted using different block sizes: 256, 32, and 4 bytes. The smaller block used, the more coherent the pattern.



Avalanche Effect

The Avalanche Effect of the ECB mode remains steady around 50%. Two different criterias of the Avalanche effect have been tested. The SPAC has been tested by modifying the plaintext by 1 bit each round and keeping the key constant. Whereas the SKAC, has been tested by keeping the plaintext constant and modifying the key by 1 bit each round. Through these tests, it's clear that the ECB mode has a great Avalanche Effect.



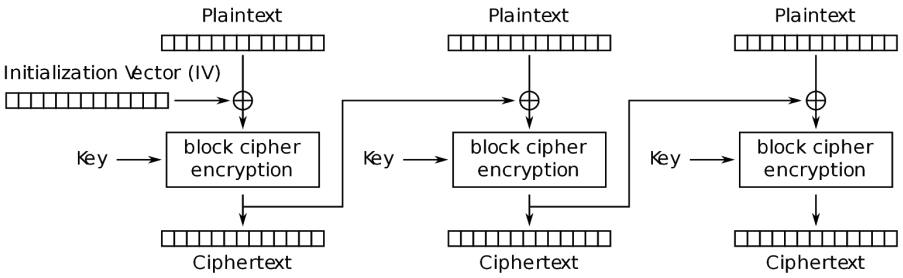


Execution Duration

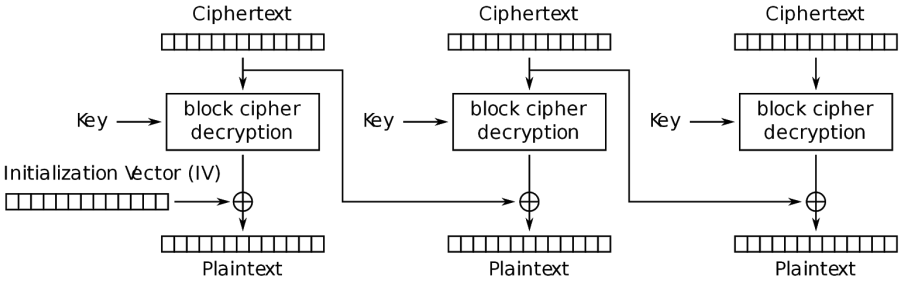
The duration for both encryption and decryption of the ECB mode is 7.88s and 8.31s respectively. This is quite fast in comparison to CBC (which will be discussed later). The benefit of the ECB mode is that each plaintext/ciphertext block during their operation is independent from each other. Thus, blocks can be encrypted/decrypted in parallel which greatly reduces the execution time.

Cipher Block Chaining (CBC)

The CBC mode is, in a way, an extension of the ECB mode. Similar to ECB, it divides the plaintext into blocks and encrypts them with the key. However, before the encryption is performed, the plaintext block is XOR'd with the ciphertext of the previous block. This adds confusion and diffusion which was a weakness of ECB mode. A problem that arises with this is at the first plaintext block where there's no previous ciphertext. For this, an Initialization Vector (IV) is used. Which is a random fixed-size input that is generated and included in the ciphertext for decryption. For decryption, IV is retrieved from the ciphertext. The decryption is simply the reverse of the encryption. And since the ciphertext is already readily available for XOR, this helps improve the speed by using parallelization.



Cipher Block Chaining (CBC) mode encryption

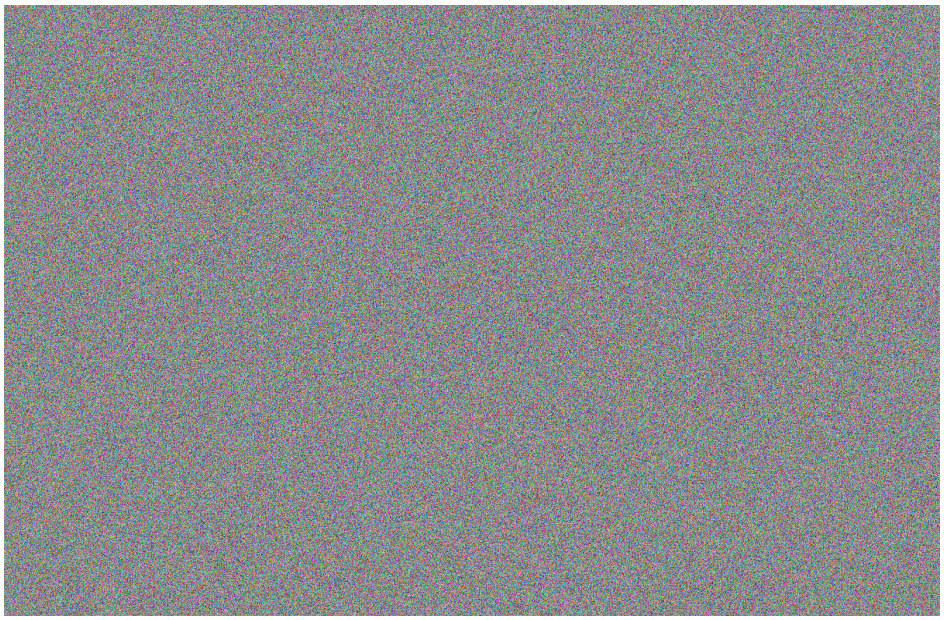


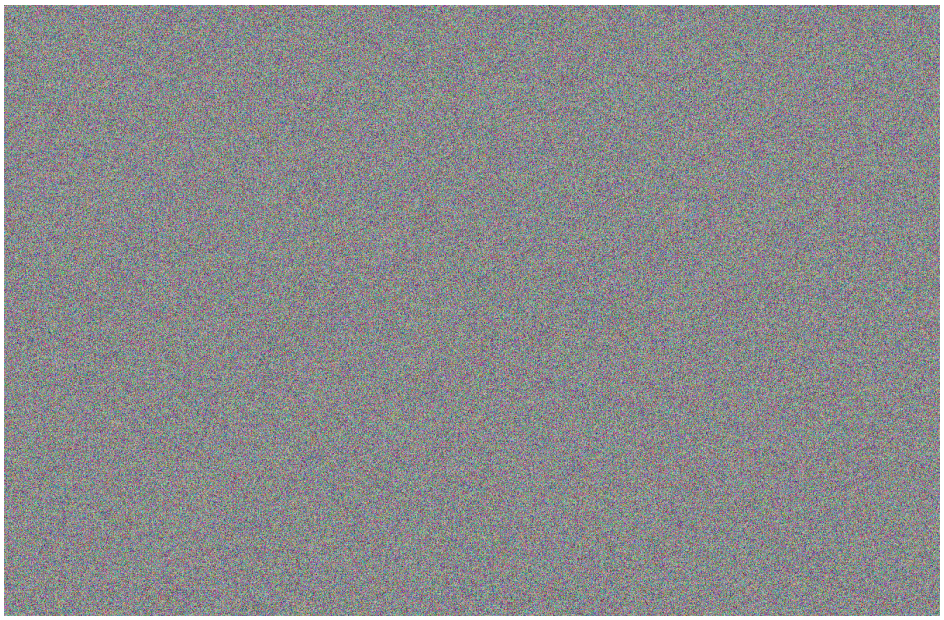
Cipher Block Chaining (CBC) mode decryption

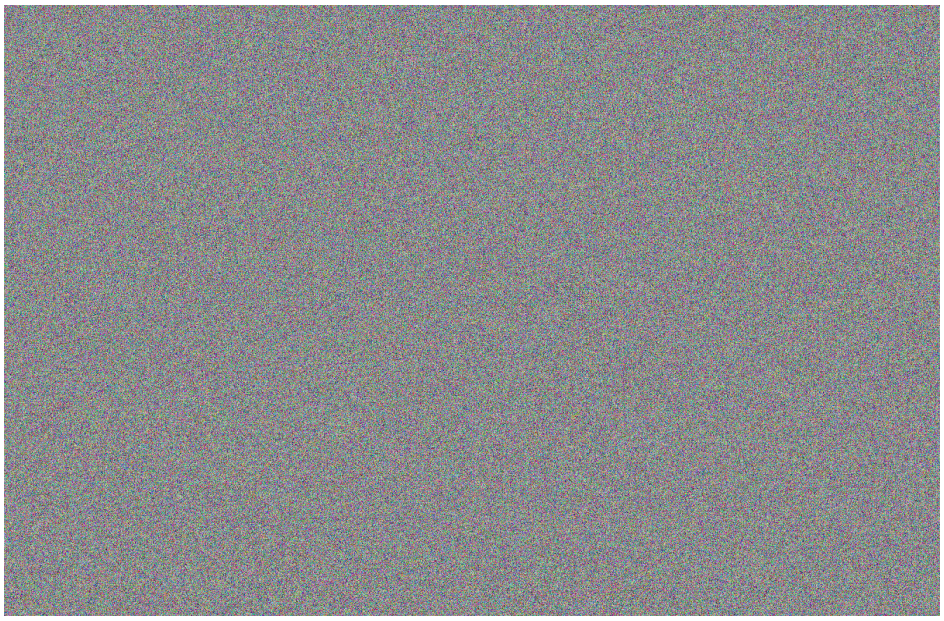
Visualization

The confusion and diffusion property of the CBC mode is much better than ECB. The XOR operation with the previous ciphertext block adds confusion and diffusion properties. Since it uses a "pseudorandom" IV for the first block and the first block propagates through the entire operation, it greatly improves on confusion. The reliance of the previous ciphertext also improves diffusion as the plaintext is spread throughout the ciphertext. It is clear looking at the encrypted images that, unlike ECB, it is extremely

difficult to find patterns no matter the block size. Even at 256, 32, and 4 byte size blocks below (respectively), the images seem identical as all the images are just static thanks to the strong confusion and diffusion properties.

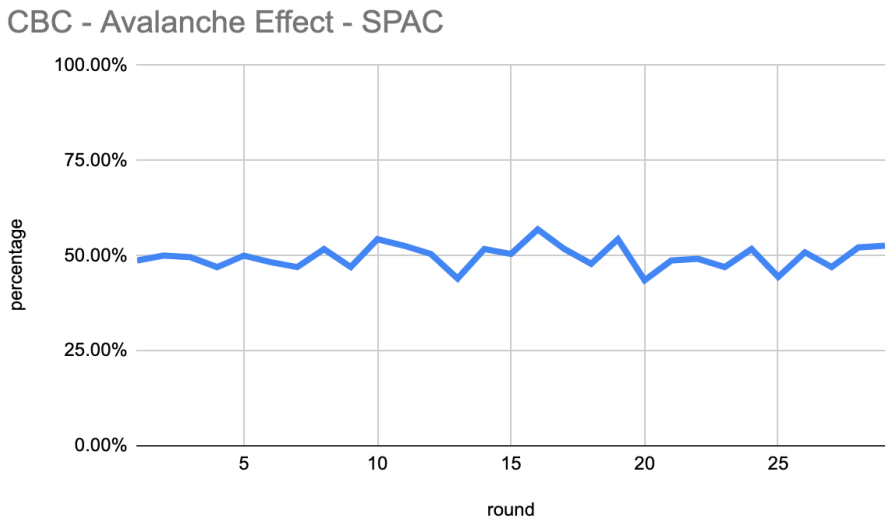


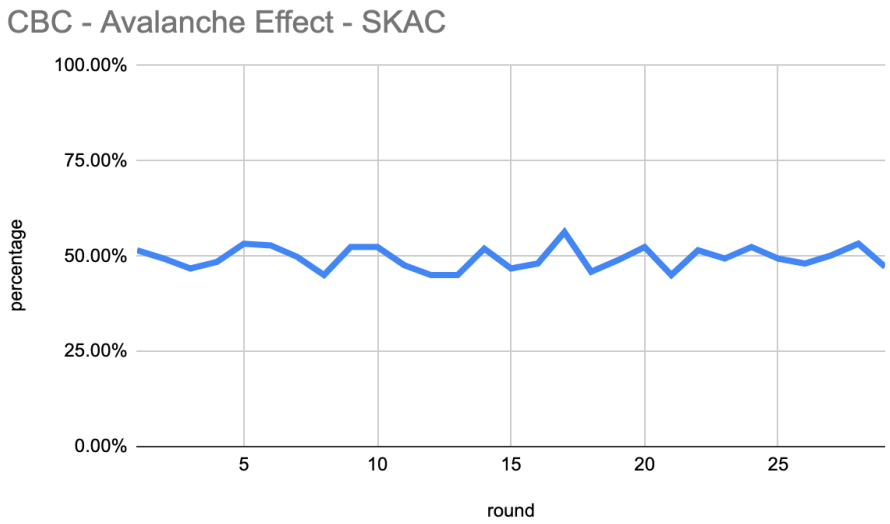




Avalanche Effect

Using the identical Avalanche Effect testing criterias as before, it it clear that the Avalanche Effect of the CBC mode is quite good. The SPAC and SKAC both remain steady at 50% throughout each round.



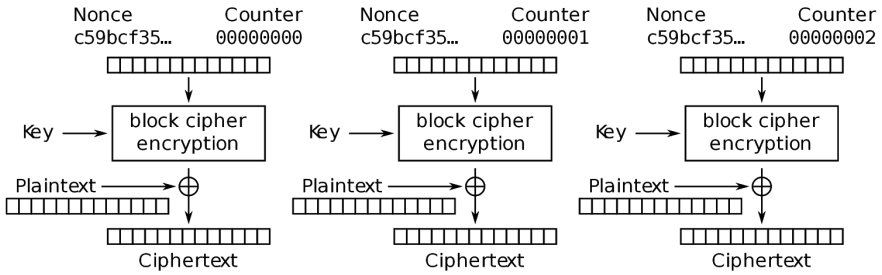


Execution Duration

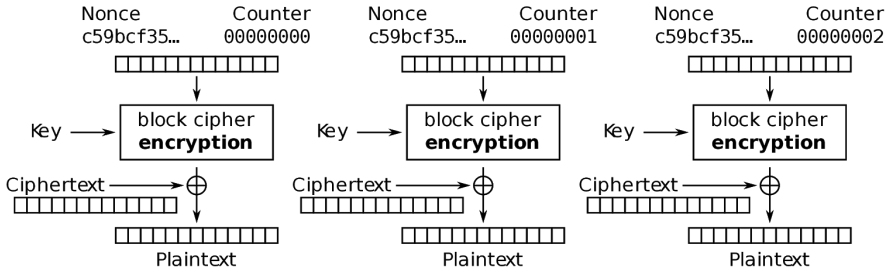
The CBC mode is not perfect by any means. The area where it suffers is the execution time during encryption. The encryption takes 22.16s. The cause of this is, unlike ECB, the encryption cannot be parallelized. Due to the dependency of the previous ciphertext, the consequent plaintext blocks cannot be encrypted until the previous plaintext is encrypted and the ciphertext is provided. Fortunately, unlike encryption, the CBC mode decryption can be parallelized. Because the ciphertext is already readily available during decryption, the ciphertext blocks can be parallelized. Resulting in faster execution of 14.46s.

Counter (CTR)

The Counter mode is quite literal in its name. The mode has a counter that increments for every block which ensures that a sequence will not repeat. This improves the confusion property as the plaintext is substituted with different values every time. The mode is very simple as both encryption and decryption goes through the same process. Unlike the ECB and CBC, the plaintext doesn't go through encryption. The encryption is done with a nonce, counter, and key. The encrypted result of that is then XOR'd with the plaintext to produce the ciphertext. In a way, this mode is similar to the One Time Pad cipher where the plaintext is XOR'd with a "pseudorandom" value. The decryption is nearly exactly the same where it uses encryption of nonce, counter, and key to generate the same value then XOR the result with the ciphertext to produce the plaintext.



Counter (CTR) mode encryption

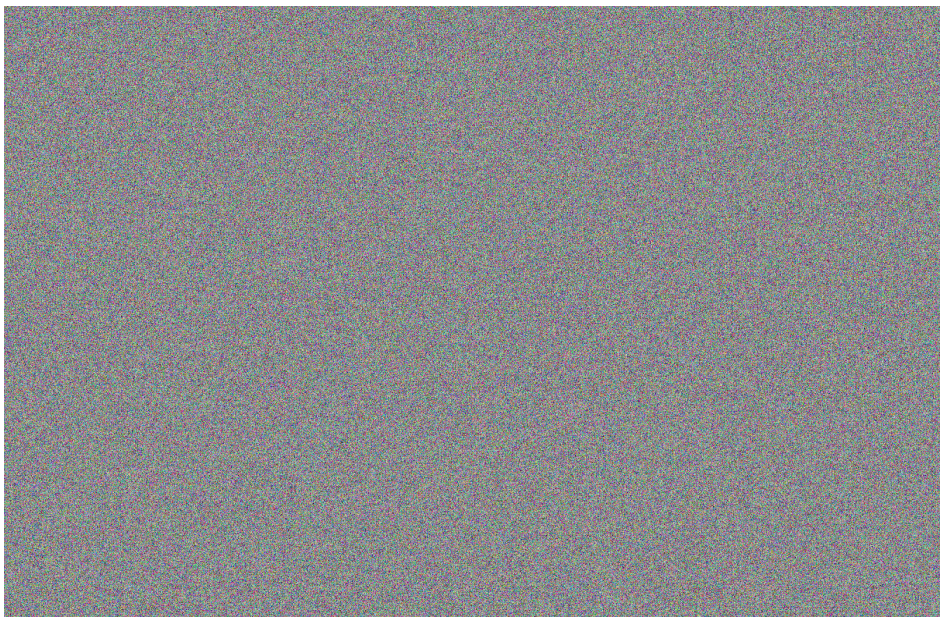


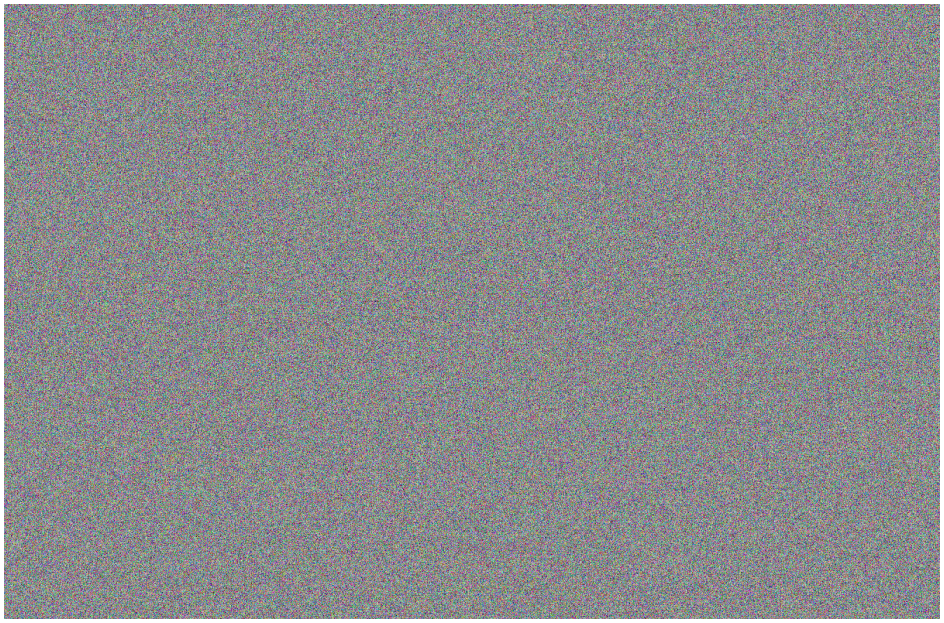
Counter (CTR) mode decryption

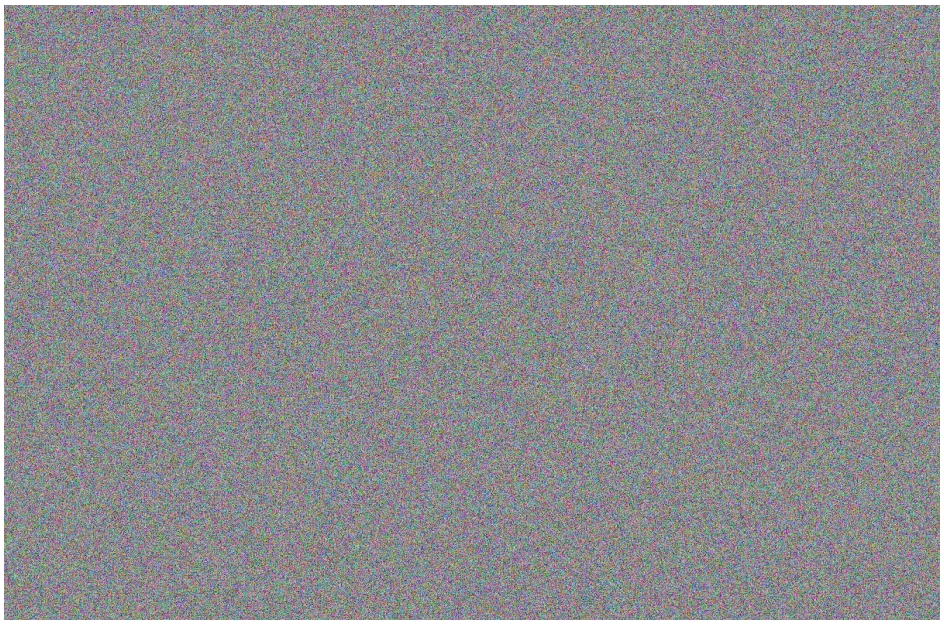
Visualization

The visualization of the CTR looks similar to the CBC mode; the 3 images encrypted in block sizes (256, 32, and 4 respectively) look identical because it's all static. The aspect that makes CTR differ from CBC is the diffusion property. The chaining aspect of CBC greatly added to the diffusion. The CTR mode doesn't have chaining, but the visualization looks identical. The confusion property makes up for CTR. If CTR XOR operation is compared to the OTP cipher, then each plaintext block is "encrypted" using a different

“key”, which greatly improves the confusion property.

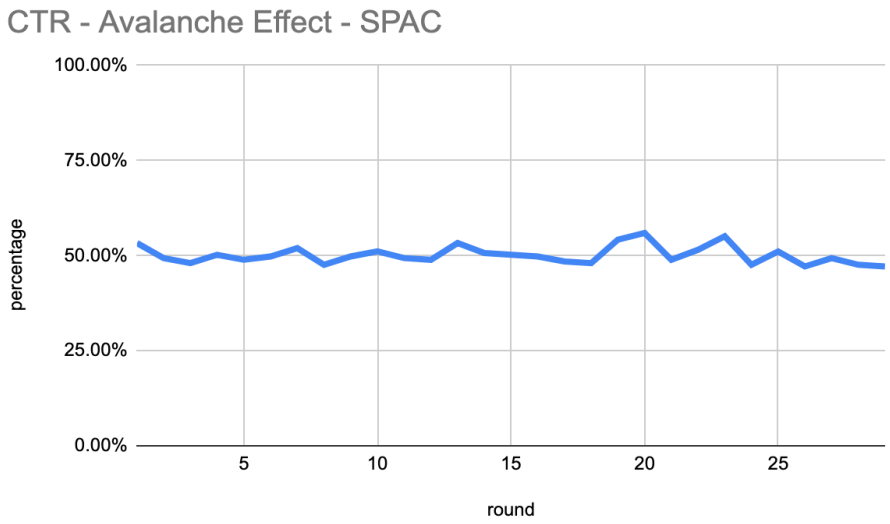


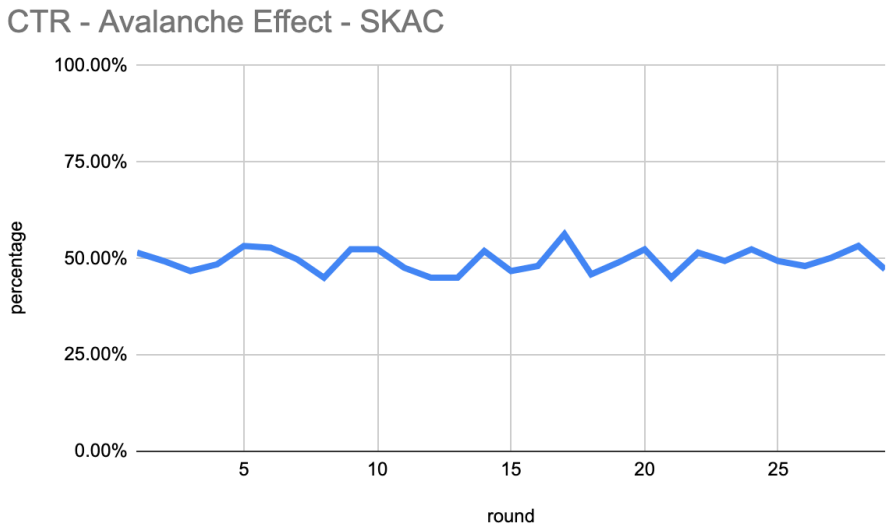




Avalanche Effect

As with ECB and CBC, the Avalanche Effect CTR holds steady at 50% throughout the rounds for both SPAC and SKAC.





Execution Duration

Both process, encryption and decryption, can be parallelized for CTR mode. The nonce and the key remains constant throughout the operation and the counter is persistently incremented thus it can be easily calculated. Since there is no reliance in other blocks, the encryption/decryption can be done in parallel. The execution time for encryption/decryption is 14.27s and 14.21s respectively.

SUBKEY GENERATION

Subkeys add another layer of security to ciphers. A subkey is a key derived from the original key. Based on how the subkey is generated, it can add confusion and diffusion properties to the cipher. If an attacker is able to get a subkey, if the implementation relies on other blocks, the rest of the data is not compromised. For my implementation of the subkey generation, the key is shifted by 1 bit every block. This adds to the diffusion property of the cipher and is easy to generate for quick executions.

CONCLUSION

The ECB mode lacked diffusion and visualizing the result, the patterns of the original plaintext were very coherent. But due to the straightforwardness, it was the quickest mode which is beneficial. Whereas the CBC mode clearly had better confusion and diffusion, it suffered slow executions due to the dependency of block forcing it to be sequential only. While ECB was fast and CBC was stronger, CTR mode stood in the middle. It wasn't the fastest and wasn't as strong as CBC as it lacked diffusion. However, it was quicker than CBC because of its ability to be parallelized and stronger than ECB as it didn't use the same codebook for every encryption block. None of these modes are particularly better than the other, but they all have their benefit. Using the appropriate mode of operation is important to providing security and adequate speed.