Cryptography 2 Report - SAC and BIC for Hash Functions + Collisions

Adam Korba

30/10/2024

1 Introduction

This report presents the results of experiments conducted to analyze the properties of hash functions, specifically SHA-256 and a toy hash function, in terms of their Strict Avalanche Criterion (SAC) and Bit Independence Criterion (BIC). Additionally, we explore collision resistance through random collision searches and hill climbing techniques.

2 Toy Hash Function

The toy hash function was implemented to see how a simple hash function fails to meet the SAC and BIC properties. Below is the python code for the toy hash:

```
def toy_hash(data: bytes) -> np.ndarray:
    x = np.frombuffer(data, dtype=np.uint8).astype(np.uint16)
    mix = (x ^ np.roll(x, 1) + 17 * np.roll(x, 2)) % 256
    digest = np.tile(mix.sum() % 256, 32).astype(np.uint8)
    return np.unpackbits(digest)
```

3 Strict Avalanche Criterion (SAC)

The SAC is evaluated by flipping each input bit and measuring the change in output bits. The results are visualized in heatmaps for both SHA-256 and the toy hash function. Input and output lengths are fixed and equal to 256 bits for both functions. Every bit is checked 2000 times to ensure statistical significance.

The Figure 1 shows the SAC heatmaps for both hash functions, as you can see the sha-256 Heatmap is completely uniform while the toy hash function has visible patterns indicating poor avalanche properties. The average flip probabilities are as follows:

- SHA-256: 0.500
- Toy Hash: 0.340

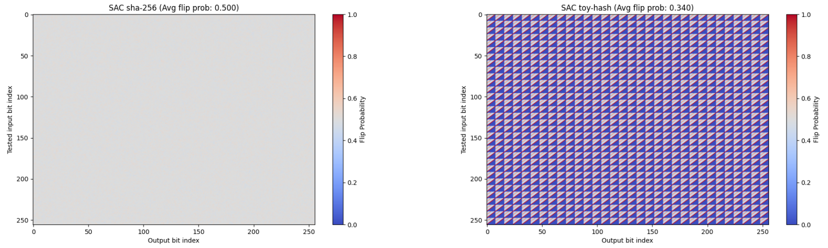
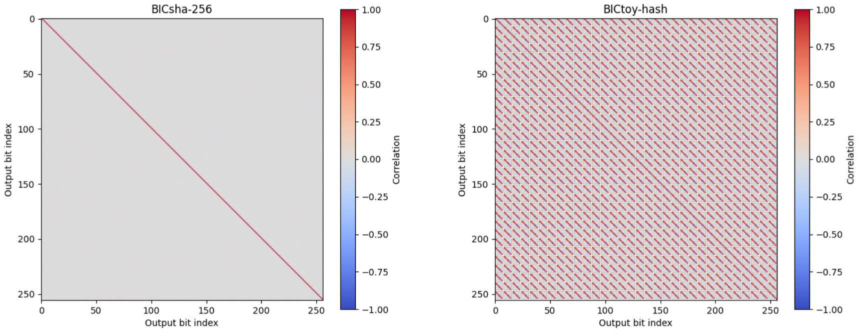


Figure 1: Figure 1: SAC Heatmaps - Left: SHA-256, Right: Toy Hash Function

4 Bit Independence Criterion (BIC)

The second property, BIC, was tested by flipping a single input bit and measuring the independence of output bits.



BIC Heatmap - SHA-256

BIC Heatmap - Toy Hash Function

Figure 2: Figure 2: BIC Heatmaps - Left: SHA-256, Right: Toy Hash Function

The Figure 2 shows the BIC heatmaps for both hash functions, as you can see the sha-256 Heatmap is uniform (except for the obvious diagonal) while the toy hash function has visible patterns indicating poor bit independence.

5 Collision Resistance

Random collision searches were performed for both hash functions truncated to 32 bits. The results are as follows:

- truncated SHA-256: Found a collision after 85,286 attempts.
- truncated Toy Hash: Found a collision after 19 attempts.

I attempted to see how close I could get to a near-collision in sha-256 using different step counts. The results are as follows:

- 1,000 steps: Best hamming distance 7 bits.
- 10,000 steps: Best hamming distance 5 bits.
- 100,000 steps: Best hamming distance 5 bits.
- 1,000,000 steps: Best hamming distance 3 bits.

Hill climbing was also used to find collisions:

- truncated SHA-256: Did not find a collision after 1,000,000 attempts.
- truncated Toy Hash: Found a collision after 5 attempts.

6 Conclusion

The experiments demonstrate that SHA-256 meets the SAC and BIC properties effectively, while the toy hash function fails to do so. Additionally, SHA-256 shows strong collision resistance compared to the toy hash function.