Introduction to Cryptology — Lab

Iskander Gaba

April 30, 2020

1 Part One

1. In this question, we will consider a generic brute-force attack where given an initialization vector, a plain-text, and its encryption, the adversary tries to query the block cipher to decrypt the cipher-text using all possible keys up to a certain limit. We consider a single-user setting and a goal success probability of 2^{-10} mounted by an adversary with a certain budget.

   (a) \$1 000: At this budget level, the best hardware we can buy is Bitmain Antminer S9i with an approximate computational capability of 14 TH/s. We assume that one hash operation is equivalent to one query to the block cipher.

   i. **SPECK 32/64:** We will need to try 2^{54} guesses to reach a success probability 2^{-10}. The amount of time in seconds needed to exhaust all 2^{54} keys:

   $$\frac{2^{54}}{14 \times 10^{12}} \approx 1287s$$

   This is equivalent to about 20 minutes. It is therefore possible to effectively attack the block cipher encryption and guess the key with a success probability of 2^{-10} with this budget.

   ii. **SPECK 48/96:**: We will need to try 2^{86} guesses to reach a success probability 2^{-10}. The amount of time in seconds needed to exhaust all 2^{86} keys:

   $$\frac{2^{86}}{14 \times 10^{12}} \approx 5 \times 10^{12}s$$

   This is equivalent to about 175244 years. It is therefore impossible to effectively attack the block cipher encryption and guess the key with a success probability of 2^{-10} with this budget.

   iii. **SPECK 64/128:** We will need to try 2^{118} guesses to reach a success probability 2^{-10}. Given the impossibility observed to do such a thing for SPECK 48/96, we can conclude that SPECK 64/128 is immune to such an attack with this level of budget.

   (b) \$100 000: At this budget level, the best hardware we can buy is cluster of 100 Antminer S19 an approximate cumulative computational capability of 10^6 TH/s. We assume that one hash operation is equivalent to one query to the block cipher.

   i. **SPECK 32/64:** Feasible.

   ii. **SPECK 48/96:** We will need to try 2^{86} guesses to reach a success probability 2^{-10}. The amount of time in seconds needed to exhaust all 2^{86} keys:

   $$\frac{2^{86}}{10^6 \times 10^{12}} \approx 7.7 \times 10^7 s$$

This is equivalent to about 2.5 years. It is therefore impossible to effectively attack this block cipher encryption and guess the key with a success probability of 2^{-10} with this budget.

iii. **SPECK 64/128:** We will need to try 2^{118} guesses to reach a success probability 2^{-10}. Given the impossibility observed to do such a thing for SPECK 48/96, we can conclude that SPECK 64/128 is immune to such an attack with this level of budget.

(c) \$1 000 000: At this budget level, the best hardware we can buy is cluster of 1000 Antminer S19 an approximate cumulative computational capability of 10^7 TH/s. We assume that one hash operation is equivalent to one query to the block cipher.

i. **SPECK 32/64:** Feasible.

ii. **SPECK 48/96:** We will need to try 2^{86} guesses to reach a success probability 2^{-10}. The amount of time in seconds needed to exhaust all 2^{86} keys:

$$\frac{2^{86}}{10^7 \times 10^{12}} \approx 7.7 \times 10^6 s$$

This is equivalent to about 90 days. Depending of how valuable the information to be retrieved over time, this could be acceptable.

iii. **SPECK 64/128:** We will need to try 2^{118} guesses to reach a success probability 2^{-10}. The amount of time in seconds needed to exhaust all 2^{118} keys:

$$\frac{2^{118}}{10^7 \times 10^{12}} \approx 3.3 \times 10^{16} s$$

This is equivalent to more than 1 billion years. It is therefore impossible to effectively attack this block cipher encryption and guess the key with a success probability of 2^{-10} with this budget.

(d) \$100 000 000: At this budget level, the best hardware we can buy is cluster of 100000 Antminer S19 an approximate cumulative computational capability of 10^9 TH/s. We assume that one hash operation is equivalent to one query to the block cipher.

i. **SPECK 32/64:** Feasible.

ii. **SPECK 48/96:** We will need to try 2^{86} guesses to reach a success probability 2^{-10}. The amount of time in seconds needed to exhaust all 2^{86} keys:

$$\frac{2^{86}}{10^9 \times 10^{12}} \approx 77371 s$$

This is equivalent to about 22 minutes. It is therefore possible to effectively attack this block cipher encryption and guess the key with a success probability of 2^{-10} with this budget.

iii. **SPECK 64/128:** We will need to try 2^{118} guesses to reach a success probability 2^{-10}. The amount of time in seconds needed to exhaust all 2^{118} keys:

$$\frac{2^{118}}{10^9 \times 10^{12}} \approx 3.3 \times 10^{14} s$$

This is equivalent to more than 10.5 million years. It is therefore impossible to effectively attack this block cipher encryption and guess the key with a success probability of 2^{-10} with this budget.

2. Implemented in the code. Check `cbc_enc_s32_64(uint16_t key[4], uint8_t *ct, uint8_t *pt, size_t ctlen)` as well as `test_nondeterminism()` functions

3. Implemented in the code. Check `cbc_dec_s48_96(uint32_t key[4], uint8_t *ct, uint8_t *pt, size_t ctlen)` as well as `test_enc_dec()` functions.

   Note: CBC mode Encryption and decryption for SPECK 48/96 and SPECK 64/128 were also implemented in the code.

2 Part Two

1. (a) **The attack:** Explore the fact that after encrypting enough blocks with the same block cipher key in CBC mode, collision is very likely to happen. If c_i and c_j are two identical cipher text blocks, then $c_{i-1} \oplus c_{j-1} = m_{i-1} \oplus m_{j-1}$.

   (b) **Data complexity:** Considering that n is our block size, a collision occurs with a high probability after observing $2^{\frac{n}{2}}$ cipher text blocks encrypted with the same key. Therefore, the data complexity is approximately $2^{\frac{n}{2}} \times n$ bits as we need to store about $2^{\frac{n}{2}}$ cipher text blocks with length of n each.

2. (a) To handle and find colliding cipher text blocks efficiently, one can use a hash table data structure where the keys of entries are the cipher text blocks and values are the indices.

   (b) For SPECK 32/64, on average, the amount of data needed for a collision to be almost certainly found is between 2^{17} and 2^{18} cipher text blocks. This is consistent with the theoretical expectations of this attack as it proved to be very hard to find a collision for a cipher text consisting of less than 2^{16} cipher text blocks.

3. For SPECK 48/96, on average, the amount of data needed for a collision to be almost certainly found is between 2^{25} and 2^{26} cipher text blocks. This is consistent with the theoretical expectations of this attack as it proved to be very hard to find a collision for a cipher text consisting of less than 2^{24} cipher text blocks.

4. Implemented the function: `uint32_t *attack_s64_128(uint8_t *ct, size_t ctlen)`.

5. No it would not be feasible on a decent desktop for this will require a crazy amount of about 11 Terabyte of memory for it to succeed, i.e. (80×2^{40}) bits. With that being said, it is obviously not possible to succeed with this attack on a 128-bit long block cipher using a decent desktop if we keep in mind that in nowadays "decent" desktops there is little reason for anyone to have more than 64 GB of RAM.

6. (a) We know that for CBC mode, for q blocks observed and $q \leq 2^{\frac{n}{2}}$, the success probability of finding a collision is equal to $\frac{q^2}{2^n}$. Therefore, we can conclude that if we want a security margin of 2^{-32}, then:

      i. For SPECK 32/64, we have $\frac{q^2}{2^{32}} \leq 2^{-32}$ and therefore $q \leq 1$. We can only encrypt one block (or 32 bits of data) with the same key at most if we want to keep that safety margin.

      ii. For SPECK 48/96, we have $\frac{q^2}{2^{48}} \leq 2^{-32}$ and therefore $q^2 \leq 2^{16}$ which means that $q \leq 2^8$. We can only encrypt 2^8 blocks (or 48×2^8 bits of data) with the same key at most if we want to keep that safety margin.

iii. For SPECK 64/128, we have $\frac{q^2}{2^{64}} \leq 2^{-32}$ and therefore $q^2 \leq 2^{32}$ which means that $q \leq 2^{16}$. We can only encrypt 2^{16} blocks (or 2^{22} bits of data) with the same key at most if we want to keep that safety margin.

(b) CTR mode has approximately the same advantage function and thus cannot increase the number of bits encrypted with the same key under the same safety margin. However, CENC mode of operation is claimed to be able to do slightly better according to a workshop presentation published by the International Association for Cryptologic Research. According to them, the success probability of finding a collision is equal to $\frac{wq}{2^n}$ with w being a constant ($w = 2^8$ in the case of AES). It is therefore, apparently, possible to encode more bits with the same key while keeping the 2^{-32} safety margin using the CENC mode of operation. For example, if we assume that $w = 2^8$ just like in the case of AES, then one can encrypt up to 2^{24} blocks of data with the same key using SPECK 64/128 while keeping the 2^{-32} safety margin.