1 DES

1.1 Introduction

The Data Encryption Standard (DES) is an encription algorithm that was published in 1975 and standartized in 1977. It was published under a non-exclusive and royalty free license. Therefore it was available for adoption in varoius branches of industry as well as for extensive academical studies of its security. It was considered secure for over ten years, however there had been concerns over the key size. The hardware was continuously getting better and it was getting easier to produce specifical DES cracking chips. In 1993 the cost of a brute force machine that at most needs 7 hours to crack the key was about 1 million US dollars. [2] The online DES cracker found at https://crack.sh/ boasts the capability to exhaust the key-space of DES in 26 hours for the build price of about hundred thousand US dollars.

DES is a Feistel cipher, hence the encryption and decryption is very similar and the performance measures can be used interchangeably. The only difference betweeen encryption and decryption lies in the order of the subkeys (they are reversed for decryption). [1], [2]

1.2 Our implementation of the algorithm

Hier kommt dann die Berschreibung der Implementation, evlt. auch genauere Erklärungen über DES

Hier könnte man auch vergleichen die geschwindigleit von unserem Algorithmus und vorimplementierten, denn ein vorimplementiertes wurde empfohlen. Reine Verschüsselungsgeschwindigkeit ist gemeint.

1.3 The key-space of DES

DES keys consist of 64 bits, however every eight bit is a parity bit. This means, that DES effectively has the key size of 56 bits. Keys are usually given in hexadecimal, conseqently every four bits use a single symbol, e.g. $0001 = 1, 1111 = F$. The parity bit for DES is even. This means that if the first seven bits would have an odd amount of 1 then the eight (parity) bit is set to 0 and vice versa. [2]

Thus, the size of key-space is 2^{56}. To calculate the time for key-space exhaustion the number of DES operations per second x is needed. Then the time is equal to $2^{56}/x$.

unsere x herausfinden

1.4 Known key bits

If we make the assumption that some of key bits are known then the size of remaining key-space is almost always consistently reduced. For the brute force approach it generally does not matter if the known bit is a parity bit, see below.

1. Lets assume that one regular bit is known. Then still 55 other bits have to be found, thus the key-space is 2^{55}.

2. Lets assume that one parity bit is known. Then the last bit of the seven bits referring to the known parity bit does not need to be tested, as its value is known form parity check. About the remaining 49 bits there is no additional information. This means that 55 bits have to be found and the key-space is 2^{55}.

This shows that there is generally no difference between knowing a regular and a parity bit. The only situation where the knowledge of a parity bit does not decrease the unknown key-space is if the other seven regular bits for the particular parity bit are known.

Therefore the time for brute force attack with some known key bits can be calculated and is shown in ref to figure. Hier spaeter wenn Geschwindigkeit bekant, Tabelle oder Diagramm mit Anzahl bekannte bits vs Zeit. Ab 8 bekannten bits Unterscheid zeigen zwischen 8 mit paritats bit oder ohne. (ich mache das)

1.5 Compelement property

DES keys have a complement property that reduces the number of needed encryptions for a chosen plaintext brute force attack.

Lets define the complement $\bar{x}$ of a bit-sequence x as the bit-wise replacement of 0s with 1s and 1s with 0s.

Complement property. Lets assume that plaintext P is encryptedd to cyphertext C with the key k. Then the complement property of DES means that plaintext $\bar{P}$ will be encrypted to cyphertext $\bar{C}$ with the key $\bar{k}$, put differently: $DES_k(P) = \overline{DES_{\bar{k}}(\bar{P})}$. [2]

Proof. werde ich noch schreiben. Waere gut die Bezeichnungen aus 1.2 davor zu wissen

This property can be used to half the relevant keyspace. Lets assume that encryptions for the plaintext P and its complement $\bar{P}$ are known, in other words, $DES_k(P) = C_1$ and $DES_k(\bar{P}) = C_2$. We consider a brute force attack over $k_i \in \mathbb{Z}_2^{56}$. If the unknown key k is k_i, then $DES_{k_i}(P) = DES_k(P) = C_1$ holds. However if it is $\bar{k_i}$ then $DES_{k_i}(P) = \overline{DES_{\bar{k_i}}(\bar{P})} = \overline{DES_k(\bar{P})} = \overline{C_2}$. Therefore one encryption has tested two keys: k_i and $\bar{k_i}$. This means that 2^{55} keys have to be checked.

1.6 Weak keys

Not all DES keys are equaly secure. DES has 4 weak keys, 12 semiweak keys and 48 possibly weak keys.

Weak keys have identical bits in each half of the actual key. This means that permutations and rotations do essentialy nothing and all 16 subkeys are the same.

Semiweak keys come in pairs. Here one key can decrypt the text that has been encrypted with the other key. Only two different subkeys are generated for these keys. For possibly weak keys only four different subkeys are generated instead of sixteen.

However the effect of weak keys on DES is negligible due to the total size of keyspace. [2]

1.7 Other attacks

das werde ich schreiben Im Prinzip DIFFERENTIAL AND LINEAR CRYPTOANALYSIS, aber funktioniert bei DES nicht unbedingt viel besser als einfach brute force. Ich weiß nicht ob bekannte Teil von Schlüssel da was ändert oder ob es eine bessere Angriff unter diesen Voraussetzungen gibt. Bis jetzt habe ich nichts gefunden.

1.8 Our results

Hier koennte man dann beschreiben wie unsere experimente gelaufen sind

1.9 Conclusion

Das werde ich schreiben, wenn wir mit Rest von DES fertig sind

Literatur

[1] Christian Karpfinger, Hubert Kiechle, Kryptologie: Algebraische Methoden und Algorithmen. Vieweg +Teubner GWV Fachverlage GmbH, Wiesbaden, 2010.

[2] Bruce Schneier, Applied Cryptography: Protocols, Algorithms and Source Code in C Preparation System. John Wiley & Sons, Inc., Indianapolis, 2015.

2 Anhang

Quellcode?