Balloon Search engine is a new generation Web 3 search engine by taking Scientific examples from History for the first time in 2022.

The examples inspired by Kriptex and Da brown in the book Davinci's code have been significantly redesigned as code and integrated with the search engine.

Structure: The cryptex can be compared to a bicycle lock.
There are 5 rings fixed on top of a cylinder and a ring on top of each ring with the entire alphabet lined up.

The key has gaps where they match the full rings, and protrusions elsewhere.
When the letters are placed in the correct
order, the rings at the bottom create a space for the key to come out, and thus the key comes out.

The keys produced by Balloon ensure that the securely received and transmitted data is locked and locked when the keying is initiated, and then analyzes. Both the computer user and the internet environment have a more secure data flow.

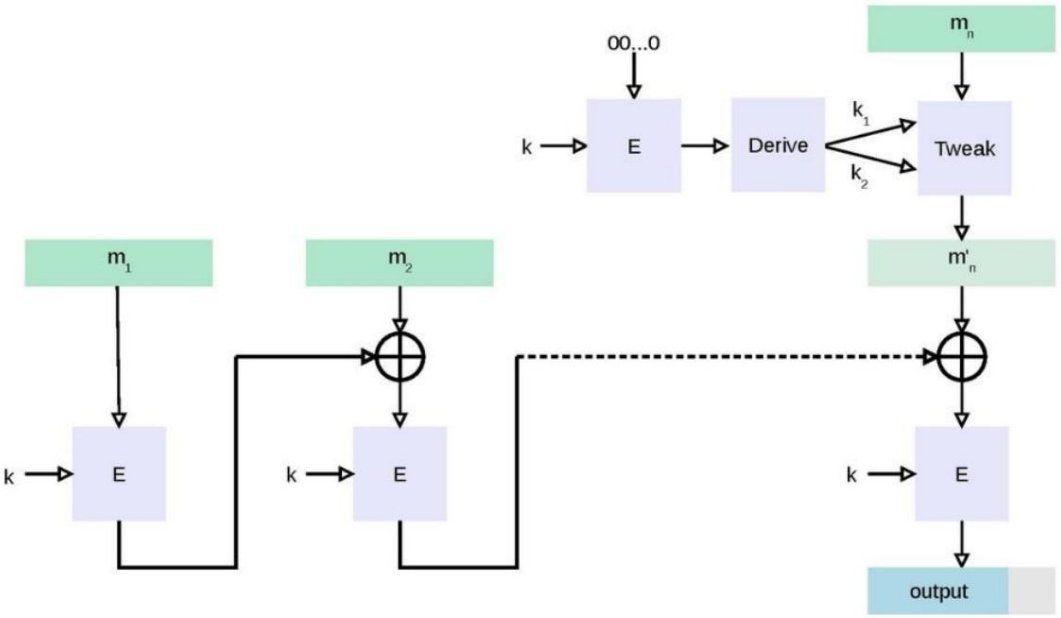
According to Da Brown:

According to Dan Brown, for security reasons, the secret document written on papyrus is wrapped in a bottle of vinegar and put inside the cryptex.
The purpose of this is to enable someone who does not know the password and tries to open it by force, to break the bottle and melt the papyrus.
In addition, each part can rotate around its own axis independently from other parts.
In some specially made cryptex you have only one right to enter the password.

Schemas that were encoded when building the Balloon search engine.

When calculating CBC-MAC, m is encrypted with the initialization vector in CBC mode and the last block is kept. The figure below

Picture:



In cryptography, CBC- , to generate message authentication code with a block cipher MAC is used.

The message is encrypted with a block cipher algorithm in CBC mode to form a blockchain, with each block dependent on the previous block being properly encrypted. This dependence prevents the change made at any end of the plaintext, the last encrypted block, from being guessed or neutralized without knowing the block cipher key. When calculating the CBC-MAC for the message, m is encrypted with the initialization vector in CBC mode and the last block is kept. The figure below shows the CBC-MAC calculation using secret key k and a block cipher E.

Security in fixed and variable length Messages

If the block cipher used is secure (pseudorandom is a permutation), then CBC-MAC is secure for fixed-length messages.[1]
. However, by itself, it is not safe for variable length messages. .
Indeed, any single key should only be used for messages of fixed and known length.
The reason for this is that when the attacker knows the correct message-tag pairs for the two messages, while calculating the MAC for his message, the Mac for m is first calculated in the classical way as in t, but this value is chained to a further computational step,

The special or operation is performed with the value generated from the mac in the first message. The presence of the tag in the new message means that the first m messages generated from the plaintext blocks will be canceled with no impact to the mac:

This problem cannot be solved by adding a message-size block to the end.[2]
There are three main ways CBC-MAC is modified to make it safe for variable length messages; 1) Allocate an input-sized key; 2) adding length to the head; 3) encrypting the last block. In such a case, it may be advisable to use a different block cipher operating mode, for example CMAC or HMAC to preserve the integrity of the variable size message.

Recline Length: As a
solution method, it includes the message length in the first block. [3]; CBC-MAC
has proven safe as long as no two prefixed messages are used, and prefixing the length is a special case. [4] Problems may occur if the length of the message is unknown when the process starts.

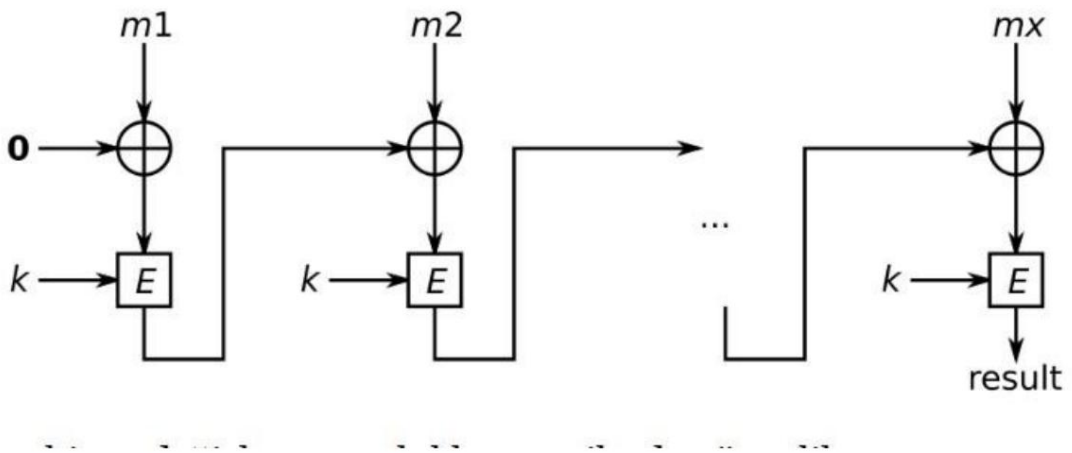
Encrypting the Last Blog:

Encrypting the last block in CBC-MAC (ECBC-MAC) [5] is defined as CBC-MAC-ELB(m, (k1, k2)) = E(k2, CBC-MAC(k1, m))[2] The advantage of encrypting the last block over extending the CBC-MAC for variable length messages is that the message length does not need to be known until the computation is finished.

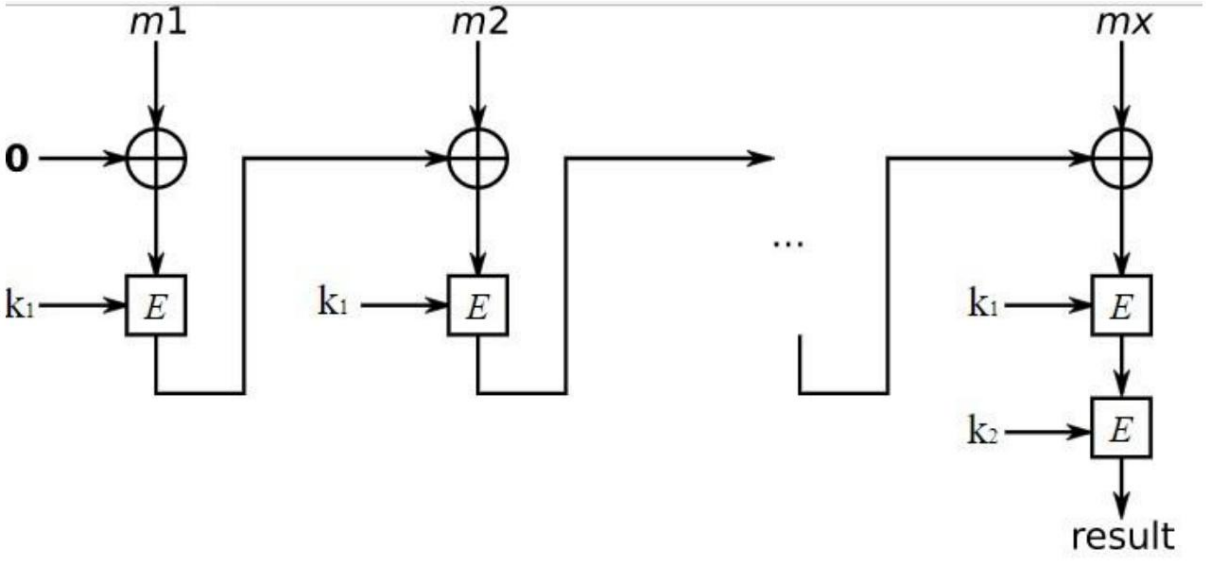
Attack Methods: As
with many cryptographic schemes, the
naive use of passwords and protocols can
lower the effectiveness of cryptographic protection (or even render it useless) and make attacks possible.

We will talk about attacks caused by misuse of CBC-MAC.[6]

Picture



Using the same key for Encryption and Authentication:



Do not allow the value of the Initial Vector to Change:

When encrypting data using a block cipher in the cipher block chaining (or other) mode, it is common to add an initialization vector to the first stage of the cipher operation.

Typically this vector should be chosen randomly (one time key) and should not be repeated for any secret key where the block cipher works.
This ensures that the same plaintext is encrypted to the same ciphertext, allowing the attacker to create a relationship providing semantic security.

Using an initialization vector when calculating a message authentication code such as CBC-MAC is a possible attack vector.

In the operation of a cipher block chaining cipher, the first block of the plaintext is mixed with the initialization vector using the special OR P1-O-IV.
The result of this operation is the input of the block cipher for the encryption operation. However, when performing encryption and decryption operations, we are obliged to send the initialization vector in plaintext—typically the block before the first block of ciphertext—so that the first block of plaintext can be decrypted. and can be successfully recovered. If you are calculating a MAC, verify that the label in the message matches the calculated value to know. we will need to pass the initialization vector to the other party in clear text.

If the initialization vector is allowed to be chosen arbitrarily, the first block of plaintext can be changed (by passing a different message) while generating the same message tag.
Consider a message M1=P1/P2. In particular, let's assume that when calculating the message tag for the CBC-MAC, we choose an IV initialization vector such that the calculation of the MAC starts with EK(IV 1 OP 1).
This generates a message tag.(m1-,T1)
Now generate message M2=p1/P2. For each modified bit in P1, the corresponding bits in the initialization vector are translated and the IV initialization vector is generated. When calculating the MAC for this message, we start with the process: EK(p1 O IV1). Since both the plaintext and initialization vector bits are changed in the same place, the change is canceled in the first step, ie the block cipher input is the same as for M1.
If no further changes are made to the plaintext, the same tag is generated even if a different message is being transmitted.

If the freedom to choose an initialization vector is removed, and all CBC-MAC implementations if it is fixed to an initialization vector.

(usually zero vector but can be anything) this attack cannot continue. To summarize, the attacker can specify the IV to be used for MAC authentication, make arbitrary changes to the first data block without invalidating the MAC.

Using a predictable initialization vector:

Sometimes IV is used as a counter to prevent message replay attacks.
However, if the attacker can guess which IV will be used for MAC authentication, he or she can replay the previously observed message by changing the first data block to compensate for the change in the IV to be used for authentication. For example, with the aggressor IV-1
M1=P1-P2 ...

observed the message sse IV-2
M'1-=(P1 O IV1 O IV 2)P2 can produce . With IV2 it passes mac verification .

The simplest countermeasure is to encrypt (ie prepare the IV for data) before using the IV. Alternatively, the MAC in CFB mode can be used because in CFB mode the IV is encrypted before being XORed with the data.

As another solution (unless repetition protection is required) always use the zero vector IV

For the above formula M1 becomes M1 =(P1- O -OO) P2 ...=P1-P2 M1 becomes M1 Since m1 and m1 are the same message, by definition they have the same label. This is not a forgery, it is the intended use of CBC-MAC.

Standards Using the Algorithm:

FIPS PUB 113 Computer Data Authentication is a US government standard (no longer used) that specifies the CBC-MAC algorithm using DES as the block cipher.
The CBC-MAC algorithm is equivalent to the ISO / IEC 9797-1 MAC 1 Algorithm.

And web 3 is the first example of Search engine. It does not scale with any Entrepreneur. Web 3 Search engine is built with Kriptex codes. It is the layer of safe transfer of secure data to the computer user of healthy inbound and outbound connection through deadlock.