CSE 470

CRYPTOGRAPHY AND COMPUTER SECURITY

Homework 1

Romulus Project Report

Muhammed Bedir ULUÇAY

1901042697

Romulus is a lightweight block cipher that was designed by researchers at the University of Maryland in 2018. It was designed to be fast and efficient, while also offering a high level of security. In this report, we will delve into the details of the Romulus algorithm and how it works.

The Romulus cipher uses a 128-bit block size and a 128-bit key. It is a 16-round block cipher, with each round consisting of several different operations. The main operations in each round of Romulus are a key-dependent substitution layer, a linear layer, and a key-dependent permutation layer.

The key-dependent substitution layer in Romulus is implemented using a set of S-boxes. These S-boxes are designed to be highly non-linear and resistant to linear and differential attacks. The S-boxes in Romulus are constructed using a combination of look-up tables and Boolean functions.

The linear layer in Romulus is implemented using a set of linear transformations. These transformations are designed to be highly non-linear and resistant to linear and differential attacks. The linear transformations in Romulus are constructed using a combination of bitwise operations and modular arithmetic.

The key-dependent permutation layer in Romulus is implemented using a set of P-boxes. These P-boxes are designed to be highly non-linear and resistant to linear and differential attacks. The P-boxes in Romulus are constructed using a combination of look-up tables and Boolean functions.

In addition to the main operations in each round, Romulus also includes a key schedule that is used to generate the keys for each round. The key schedule in Romulus is designed to be highly non-linear and resistant to attacks. It is constructed using a combination of bitwise operations and modular arithmetic.

Overall, the Romulus cipher is a highly secure and efficient block cipher that is well-suited for a wide range of applications. It has a number of desirable properties, including a high level of security, fast performance, and low hardware complexity. As a result, it has the potential to be widely

The Romulus cipher uses a simple and efficient design, which makes it well-suited for a wide range of applications. It is implemented using a combination of look-up tables and Boolean functions, which allows it to be implemented efficiently in hardware.

One of the key features of the Romulus cipher is its use of S-boxes and P-boxes, which are designed to be highly non-linear and resistant to linear and differential attacks. These S-boxes and P-boxes are constructed using a combination of look-up tables and Boolean functions, which allows them to be implemented efficiently in hardware.

The Romulus cipher also includes a key schedule, which is used to generate the keys for each round of processing. The key schedule in Romulus is designed to be highly non-linear and resistant to attacks, and it is constructed using a combination of bitwise operations and modular arithmetic.

Overall, the Romulus cipher is a highly efficient and secure block cipher that is well-suited for a wide range of applications. It has a number of desirable properties, including fast performance, low hardware complexity, and a high level of security.

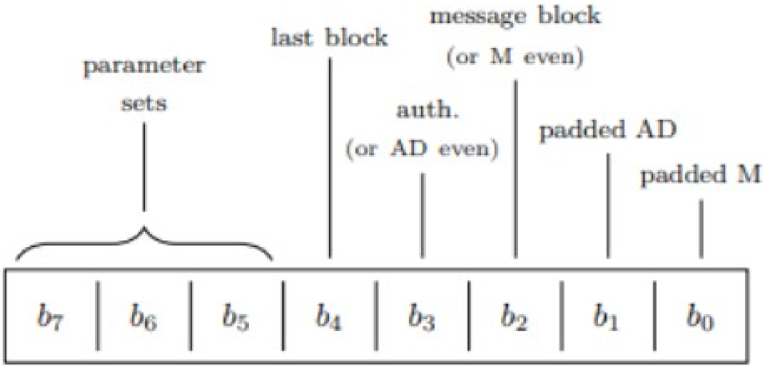
Specification of ROMULUS:

Let {0, 1} ∗ be the set of all finite bit strings, including the empty string ε. For X ∈ {0, 1} ∗ , let |X| denote its bit length. Here |ε| = 0. For integer n ≥ 0, let {0, 1} n be the set of n-bit strings, and let {0, 1} ≤n = S i=0,...,n{0, 1} i , where {0, 1} 0 = {ε}. Let JnK = {1, . . . , n} and JnK0 = {0, 1, . . . , n − 1}.

Notation used on ROMULUS:

- Nonce length $nl \in \{96, 128\}$.
- Key length $k = 128$.
- Message block length $n = 128$.
- Counter bit length $d \in \{24, 56, 48\}$.
- AD block length $n + t$, where $t \in \{96, 128\}$.
- Tag length $\tau = 128$.
- A TBC $\widetilde{E} : \mathcal{K} \times \overline{\mathcal{T}} \times \mathcal{M} \to \mathcal{M}$, where $\mathcal{K} = \{0,1\}^k$, $\mathcal{M} = \{0,1\}^n$, and $\overline{\mathcal{T}} = \mathcal{T} \times \mathcal{B} \times \mathcal{D}$. Here, $\mathcal{T} = \{0,1\}^t$, $\mathcal{D} = [\![2^d - 1]\!]_0$, and $\mathcal{B} = [\![256]\!]_0$ for parameters t and d, and $\mathcal{B}$ is also represented

Authentication with Romulus:



Domain separation. Romulus will use a domain separation byte B to ensure appropriate independence between the tweakable block cipher calls and the various versions of Romulus. Let B = (b7kb6kb5kb4kb3kb2kb1kb0) be the bitwise representation of this byte, where b7 is the MSB and b0 is the LSB Then, romulus have the following:

- $b_7b_6b_5$ will specify the parameter sets. They are fixed to:
  - 000 for Romulus-N1
  - 001 for Romulus-M1
  - 010 for Romulus-N2
  - 011 for Romulus-M2
  - 100 for Romulus-N3
  - 101 for Romulus-M3

  Note that all nonce-respecting modes have $b_5 = 0$ and all nonce-misuse resistant modes have $b_5 = 1$.
- b_4 is set to 1 once we have handled the last block of data (AD and message chains are treated separately), to 0 otherwise.
- b_3 is set to 1 when we are performing the authentication phase of the operating mode (*i.e.*, when no ciphertext data is produced), to 0 otherwise. In the special case where $b_5 = 1$ and $b_4 = 1$ (*i.e.*, last block for the nonce-misuse mode), b_3 will instead denote if the number of message blocks is even ($b_5 = 1$ if that is the case, 0 otherwise).
- b_2 is set to 1 when we are handling a message block, to 0 otherwise. Note that in the case of the misuse-resistant modes, the message blocks will be used during authentication phase (in which case we will have $b_3 = 1$ and $b_2 = 1$). In the special case where $b_5 = 1$ and $b_4 = 1$ (*i.e.*, last block for the nonce-misuse mode), b_3 will instead denote if the number of message blocks is even ($b_5 = 1$ if that is the case, 0 otherwise).
- b_1 is set to 1 when we are handling a padded AD block, to 0 otherwise.
- b_0 is set to 1 when we are handling a padded message block, to 0 otherwise.

Security of Romulus:

Romulus consider the standard security notions for nonce-based AE [6, 7, 37]. Let Π denote an NAE scheme
consisting of an encryption procedure Π.EK and a decryption procedure Π.DK, for secret key K uniform over set K
(denoted as K $← K). For plaintext M with nonce N and associated data A, Π.EK takes (N, A, M) and returns
ciphertext C (typically |C| = |M|) and tag T. For decryption, Π.DK takes (N, A, C, T) and returns a decrypted plaintext
M if authentication check is successful, and otherwise an error symbol, ⊥.
For A ∈ {0, 1} ∗ , Romulus say A has a AD blocks if it is parsed as (A[1], . . . , A[a]) n,t ←−− A. Let a˜ = ba/2c+1 which is
a bound of actual number of primitive calls for AD. Similarly for plaintext M ∈ {0, 1} ∗ , we say M has m message
blocks if |M|n def = d|M|/ne = m. The same applies to ciphertext C. For encryption query (N, A, M) or decryption
query (N, A, C, T) of a AD blocks and m message blocks, the number of total TBC calls is at most a˜ + m, which is
called the number of effective blocks of a query.

Family	NR-Priv	NR-Auth	NM-Priv	NM-Auth
Romulus-N	128	128	–	–
Romulus-M	128	128	64 ∼ 128	64 ∼ 128

Romulus-m and Romulus-n has both 128bit key recovery as well.
l as the maximum effective block length of all the encryption and decryption queries:

$$\mathbf{Adv}^{\mathsf{auth}}_{\mathsf{Romulus\text{-}M}}(\mathcal{B}) \leq \mathbf{Adv}^{\mathsf{tprp}}_{\widetilde{E}}(\mathcal{B}') + \frac{2\ell q_d}{2^n} + \frac{2q_d}{2^n}.$$

ALGORITHM:

Algorithm $\textsf{Romulus-N.Enc}_K(N, A, M)$

$S \leftarrow 0^n$
$(A[1], \ldots, A[a]) \xleftarrow{n} A$
if $|A[a]| < n$ **then** $w_A \leftarrow 26$ **else** 24
$A[a] \leftarrow \mathtt{pad}_n(A[a])$
for $i = 1$ **to** $\lfloor a/2 \rfloor$
    $(S, \eta) \leftarrow \rho(S, A[2i-1])$
    $S \leftarrow \widetilde{E}_K^{(A[2i],8,\overline{2i-1})}(S)$
end for
if $a \bmod 2 = 0$ **then** $V \leftarrow 0^n$ **else** $A[a]$
$(S, \eta) \leftarrow \rho(S, V)$
$S \leftarrow \widetilde{E}_K^{(N,w_A,\overline{a})}(S)$
$(M[1], \ldots, M[m]) \xleftarrow{n} M$
if $|M[m]| < n$ **then** $w_M \leftarrow 21$ **else** 20
for $i = 1$ **to** $m - 1$
    $(S, C[i]) \leftarrow \rho(S, M[i])$
    $S \leftarrow \widetilde{E}_K^{(N,4,\overline{i})}(S)$
end for
$M'[m] \leftarrow \mathtt{pad}_n(M[m])$
$(S, C'[m]) \leftarrow \rho(S, M'[m])$
$C[m] \leftarrow \mathtt{lsb}_{|M[m]|}(C'[m])$
$S \leftarrow \widetilde{E}_K^{(N,w_M,\overline{m})}(S)$
$(\eta, T) \leftarrow \rho(S, 0^n)$
$C \leftarrow C[1] \,\|\, \ldots \,\|\, C[m-1] \,\|\, C[m]$
return (C, T)

Algorithm $\textsf{Romulus-N.Dec}_K(N, A, C, T)$

$S \leftarrow 0^n$
$(A[1], \ldots, A[a]) \xleftarrow{n} A$
if $|A[a]| < n$ **then** $w_A \leftarrow 26$ **else** 24
$A[a] \leftarrow \mathtt{pad}_n(A[a])$
for $i = 1$ **to** $\lfloor a/2 \rfloor$
    $(S, \eta) \leftarrow \rho(S, A[2i-1])$
    $S \leftarrow \widetilde{E}_K^{(A[2i],8,\overline{2i-1})}(S)$
end for
if $a \bmod 2 = 0$ **then** $V \leftarrow 0^n$ **else** $A[a]$
$(S, \eta) \leftarrow \rho(S, V)$
$S \leftarrow \widetilde{E}_K^{(N,w_A,\overline{a})}(S)$
$(C[1], \ldots, C[m]) \xleftarrow{n} C$
if $|C[m]| < n$ **then** $w_C \leftarrow 21$ **else** 20
for $i = 1$ **to** $m - 1$
    $(S, M[i]) \leftarrow \rho^{-1}(S, C[i])$
    $S \leftarrow \widetilde{E}_K^{(N,4,\overline{i})}(S)$
end for
$\widetilde{S} \leftarrow (0^{|C[m]|} \,\|\, \mathtt{msb}_{n-|C[m]|}(G(S)))$
$C'[m] \leftarrow \mathtt{pad}_n(C[m]) \oplus \widetilde{S}$
$(S, M'[m]) \leftarrow \rho^{-1}(S, C'[m])$
$M[m] \leftarrow \mathtt{lsb}_{|C[m]|}(M'[m])$
$S \leftarrow \widetilde{E}_K^{(N,w_C,\overline{m})}(S)$
$(\eta, T^*) \leftarrow \rho(S, 0^n)$
$M \leftarrow M[1] \,\|\, \ldots \,\|\, M[m-1] \,\|\, M[m]$
if $T^* = T$ **then return** M **else** $\perp$

Algorithm $\rho(S, M)$

$C \leftarrow M \oplus G(S)$
$S' \leftarrow S \oplus M$
return (S', C)

Algorithm $\rho^{-1}(S, C)$

$M \leftarrow C \oplus G(S)$
$S' \leftarrow S \oplus M$
return (S', M)

Figure 2.5: The Romulus-N nonce-based AE mode. Lines of [**if (statement) then** $X \leftarrow x$ **else** x'] are shorthand for [**if (statement) then** $X \leftarrow x$ **else** $X \leftarrow x'$]. The dummy variable η is always discarded.

```
=========== EXAMPLE 1 ===========
 Plain text: dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text

 Encrypted text :  b'\xfd\x94\xf4;\xac\xe8:"\x03f\x18\xc5T\xb0@GB\x92\xe9\xf6\xf6D\x94\x0e\x83\xf3\xec\x01x\xac\xfc\x02\xb6\xe9\x84\x95\x82\xad\x1aojL_`i\xba?V\x8d\x80\xab\xed\xd2\x06V%D\x8c\x95\x87_\xdd8#\xebn
 \x12\xa3\xaf\xe0\xd7Vo\x9dm\xe7\x1e\x9d2\xa8H\xa3\xc9\x11"\xcbQ-N\x13B\x9c<\x7f\x86teZ\x1c\x9f\xa6:\xe4\x8ff\x00\x05\x81.k\xa9'
Decrpyted text:  b"dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text"

=========== EXAMPLE 2 ===========
 Plain text: dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text

 Encrypted text :  b'u0\xed6\xb1\xac\xcb\rl"L`\xf6\x97V\xcd7\xd35\xf9l0\xa7>\xb5\x91P\xb6\x8b\xcf\xc4\t+\x93\xaecz\x06\xfd\xd0f8\xdaP\xeb\'q\xb9\x8b\x12\xd6\x94\xd7\x9e4\xe6D\x0e\xbb\x14\xf6\x0c\xc9\x12\x92+7\x
a5\xa8\xceem\x9b\xdc9\x1c\x9f\x17\x82\x1b[\xec\x18\x1e\x03\x1a\xee\xfd\xc2\xc8c\xd6\xc2*\xd5]3J\rz\xa71\x01\x0e\x80\xa4\xf6=\x89\xca\xea\x8c'
Decrpyted text:  b"dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text"

=========== EXAMPLE 3 ===========
 Plain text: It was popularised in the 1960s with the release of Letraset sheets containing

 Encrypted text :  b"\xbd\xc4\xcb\xe2\xba\x03,\x1ayR\xa7\xdbA\xda*\x8a:~\xf2s\xc9|\x1b\xbc\xfdbp`s?\xd2\x9d\xf6\xbcDH\x8bV\x9b\xd2\xf7W\xd9\x03\xe3\x80d\x82\xc9\xd9\xaeg\xfbc\x1a\xaa\x90\xa6\xfd\x13N\xc1\xc1\x9
bj\xdb\xbe\xd5\xb1\xa1\n\xe1k\x8e\xc9T'\x18\x0c\xcc"
Decrpyted text:  b'It was popularised in the 1960s with the release of Letraset sheets containing'

=========== EXAMPLE 4 ===========
 Plain text: It was popularised in the 1960s with the release of Letraset sheets containing

 Encrypted text :  b'\x10dr+\xb8\x10\x98\xe1,\x82\x0cU\xf0M\x812z\xa7\x10\xf5t\x8e\x8e\xb7\x96*\xfc\x00\xd0]I\xcb\xa0\x80\xd9p\xfa\x92\x01\x16\xe0\xdev\xbb\xc6W8\xf8\xf2hP\xfe<(\xac\xec\x1b\x949\xbf\x1b\xfb\xe6
\xcb\x04\xde\xbfY\xaf$\xa4\xf5V"h\xc60B\x1fk'
Decrpyted text:  b'It was popularised in the 1960s with the release of Letraset sheets containing'

++++++++ OFB MODE +++++++
=========== EXAMPLE 1 ===========
 Plain text: dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text

 Encrypted text :  b'Mj4\xf9\xbe\x0c\xa8W\xee\xe8d:\xb1\x19\xed\x98\x99\xa6\xf93O\x83\xd8\x1b\xa8@\xa8\x95\x17\x9f\n\xach[\x85^\x87\xd9%G^`\xe4\xec\x025-\x9b\xe9\xf2t1\x1e\x810\xf5\xa84E\xbfY<\x98\xb2\xd0X\xe5%
\xb2\x06\x89I\x83)@\xd2\xca\x8f\x9a\xa7\xf94\xd2+\x1bPoF\x888\x85\xa0pq\xd5\xf0\xb5\x80V\xd48\x03%u\xff\x1ah\xd9'
Decrpyted text:  b"dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text"

=========== EXAMPLE 2 ===========
 Plain text: dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text

 Encrypted text :  b"\xf2\x9e\xe8\x11k\x99\x0f\x8e\x87\xd0K\xccV{\xcd\x1a'f\xdf\x84\xa0W\x18\xfe\xcb\xd5&u\x87\x04\x86_\xbf\xe5\xd6\x86\xcbv\x9483\xaf\x81\x82Q\xb1\xe9\xb8w\xab\x02>\x89\x1e#\x84\x12-\xe4&\xfeP\
xca%\x9e\xfcj\xceF\xde\xdeP\x8f\x11H\x89\xdb\x13\x17\xdd\x9d\xc7A\t\x911t\x18\x19;\x80y\xd7\xaeW\xe6\xff\xa1O\xdbg\x9d\xe2Y\x84}\xa9\xbd"
Decrpyted text:  b"dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text"

=========== EXAMPLE 3 ===========
 Plain text: It was popularised in the 1960s with the release of Letraset sheets containing

 Encrypted text :  b'`ky\xe3\xa6_\xfcB\xf9\xec19\xb6K\xf0\x83\x99\xe2\xa9(H\xcd\xd8\x1a\xa3\x07\xb9\xcdO\xcbY\xf8fB\x94E\xc2\xd99K\x10u\xa1\xe9\t0+\x8d\xbd\xefk?r\xa8+\xf5\xac*\x00\x82\t<\x85\xba\x95D\xf7v\xf1\
x0b\x82X\x8c`Z\xd3\xc1\xc8'
Decrpyted text:  b'It was popularised in the 1960s with the release of Letraset sheets containing'

=========== EXAMPLE 4 ===========
 Plain text: It was popularised in the 1960s with the release of Letraset sheets containing

 Encrypted text :  b'\xdf\x9f\xa5\x0bs\xca[\x9b\x90\xd4\x1e\xcfQ)\xd0\x01\'"\x8f\x9f\xa7\x19\x18\xff\xc0\x927-\xdfP\xd5\x0b\xb1\xfc\xc7\x9d\x8ev\x884}\xba\xc4\x87Z\xb4\xef\xae#\xb6\x1d0\xe578\x84\x163\xa1\x1b\x
aeP\xd7-\xdb\xe0x\x9d\x05\xd3\xd5A\x80XR\x88\xd0T'
Decrpyted text:  b'It was popularised in the 1960s with the release of Letraset sheets containing'

mbulucay@mbulucay:~/Desktop/Crypto/1901042697/romulus$ cd
```