Final Project

Smart Internz

Technology Stack: Cybersecurity with IBM QRadar

Project Title: SHL: A Secure Homomorphic Lightweight Cryptography Algorithm for IoT

Team ID: LTVIP2024TMID13132

Team Members:

1. Shaik Sohail
2. Peyyeti Chushma
3. Harshitha Toomu
4. Nallamothu Sivamani

Name of the Department: Information Technology.

Name of the College with Address: Kallam Haranadhareddy Institute of Technology, Guntur.

INDEX

INTRODUCTION

Lightweight cryptography algorithms are designed to provide security for constrained environments, such as Internet of Things (IoT) devices, RFID tags, wireless sensors, and other devices with limited computational power and memory. These algorithms prioritize efficiency, requiring less processing power, memory, and energy while still providing adequate security.

A few Examples of Lightweight Cryptography Algorithms:

1. Block Ciphers

- PRESENT
- HIGHT
- SIMON & SPECK

2. Steam Ciphers

- GRAIN
- ChaCha
- HC-128/HC-256

Characteristics of Lightweight Cryptography

Lightweight cryptography algorithms are designed with several key characteristics in mind:

1. **Efficiency:** They prioritize efficiency in terms of computation, memory usage, and energy consumption. This efficiency allows them to operate on devices with limited resources without compromising performance.
2. **Low Latency:** Many lightweight algorithms are optimized for low latency, ensuring that cryptographic operations can be performed quickly, which is crucial for real-time applications.
3. **Small Code Size:** These algorithms have compact code size, making them suitable for implementation in environments where program memory is scarce.
4. **Low Power Consumption:** Lightweight algorithms are designed to minimize power consumption, extending the battery life of devices such as sensors and IoT devices.

Applications of Lightweight Cryptography

The adoption of lightweight cryptography extends to various domains:

1. IoT Security: IoT devices, from smart thermostats to wearable fitness trackers, rely on lightweight cryptography to secure communication and data transmission. These algorithms ensure that sensitive information remains confidential and integral.

2. Wireless Sensor Networks: Lightweight cryptography is vital for securing wireless sensor networks used in environmental monitoring, healthcare, and industrial automation. Algorithms like PRESENT and LEO are well-suited for these applications.
3. Mobile Devices: Smartphones and tablets benefit from lightweight cryptography for securing user data, authentication, and communication over wireless networks.
4. RFID Systems: Radio-frequency identification (RFID) tags often have limited computational capabilities, making lightweight cryptography essential for ensuring their security and preventing unauthorized access.

ABSTRACT

Secure communications in the Internet of Things (IoT) ecosystem are crucial for preserving the confidentiality, integrity, and privacy of data exchanged between IoT devices. As the IoT continues to proliferate across various sectors, ranging from smart homes to industrial automation, ensuring robust security measures is paramount to mitigate potential threats. The communications in IoT have opened up many possibilities for launching Cyber-attacks. These Cyber-attacks compromise the Security of an IoT device. The main objective of this project is to propose A Secure Homomorphic Lightweight Cryptography Algorithm for secure data exchange in IoT. This proposed algorithm uses lightweight computation to provide Security to information exchanged between IoT devices. This proposed algorithm consumes less battery power and hardware resources of IoT. This proposed algorithm optimizes the IoT resources and enhances the security of the IoT devices.

Keywords: Secure Communications, IoT, Cyber-attack, Lightweight Algorithm, IoT Security.

EXISTING SYSTEM

LWC Algorithm	Fiestal/ Non-Fiestal	Cipher Type	Key Type	Key Size(in bits)	Block Size(in bits)
AES	Non-Fiestal	Block	Symmetric	128/192/256	128bit
Novel	Non-Fiestal	Block	Symmetric	32	N*8
PRESENT	Non-Fiestal	Block	Symmetric	80	64
GIFT	Non-Fiestal	Block	Symmetric	128/192/256	64
ICEBERG	Non-Fiestal	Block	Symmetric	80/120	64
PUFFIN-2	Fiestal	Block	Symmetric	128	128
BEST-1	Non-Fiestal	Block	Symmetric	128	64
LEA	Non-Fiestal	Block	Symmetric	128/192/256	64

Drawbacks of Existing System

The Major Drawbacks of the above-mentioned algorithms are:

1. In the AES Algorithm, The S-Box which was used to Substitute the value of the cipher block was very vast and contains 16*16 entries.
2. In the Novel, The Level of Security which includes Confusion and Diffusion is very low.
3. Other Algorithms like PRESENT, GIFT, ICEBERG, PUFFIN, LEA, etc., The Cipher block is very less when compared to others.

Proposed System

The main idea of this project is taken from our observation that there were many Lightweight Cryptography Algorithms for IoT but very few of them were using the resources of the IoT effectively. So, Under this project, A Secure Homomorphic Lightweight Cryptography Algorithm for secure communication in IoT.

In this project, a new Lightweight Cryptography algorithm makes use of resources effectively and increases the performance of the system and better security too.

SYSTEM ARCHITECTURE

1. System Architecture

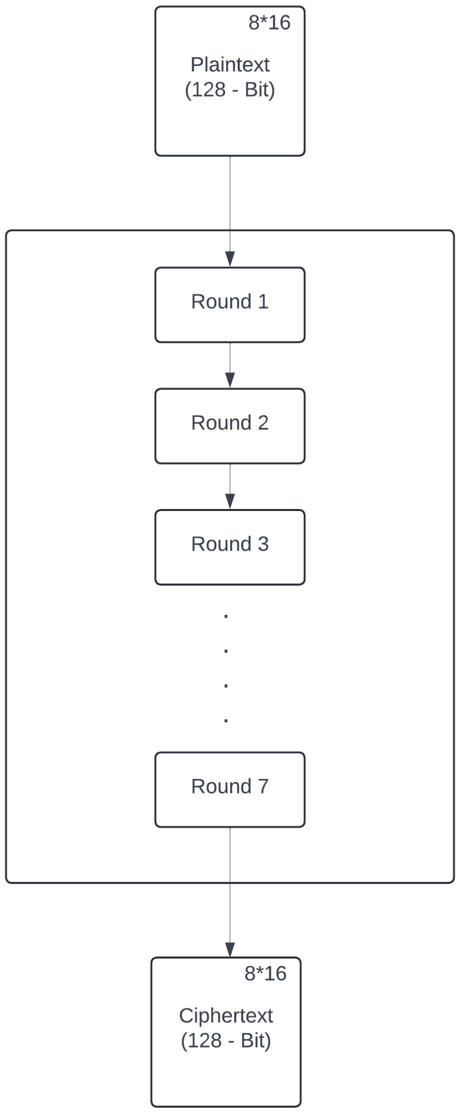


Fig 1: The Main Architecture of Proposed Algorithm

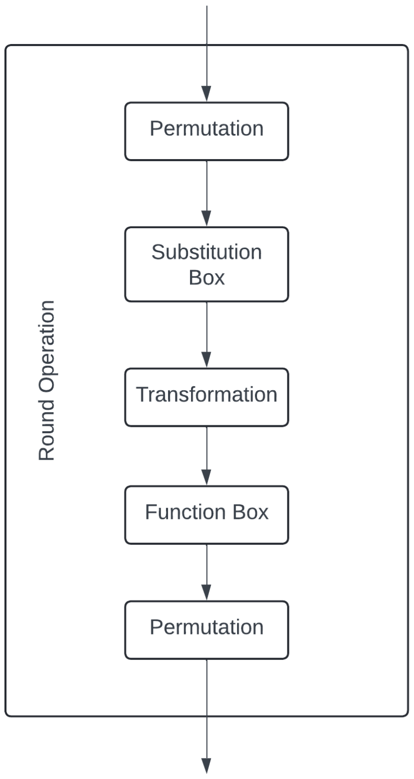


Fig 2: Architecture of Round Operations in Proposed Algorithm

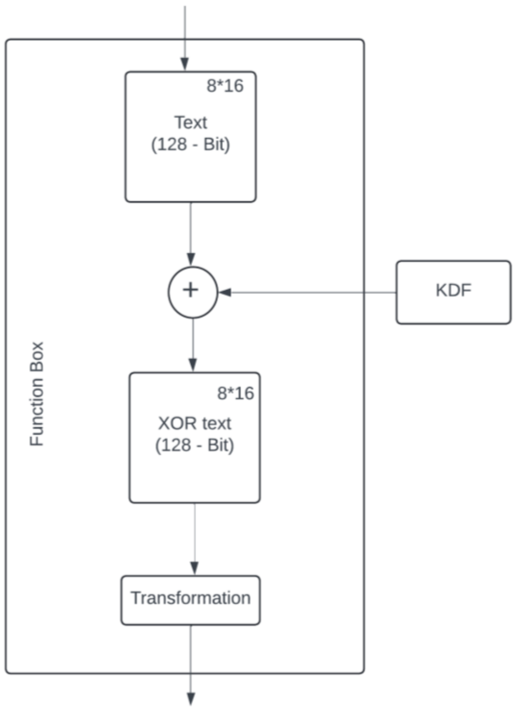


Fig 3: Architecture of the Function Box

1.1 Plaintext

Plaintext is nothing but, the input given by the user or the input that is to convert to the ciphertext. Under this project, we have restricted the size of the input of the Lightweight Cryptographic Algorithm (LWC) because of resource constraints. The proposed algorithm will take the input of 128 – bits, the input which was given by the user can be of any length but the input block size of the SHL is 128 – bits. If the size of the input is less than 128 then we apply padding for the input to full fill the block up to its mark. After the selection of the plaintext which was to be encrypted, now we arrange that plaintext into a block which has the size of 8 rows and 16 columns (8*16) matrix. The plaintext is to be fitted from the Top to Bottom method for a single column at a time.

1.2 Round Operations

Round Operations are the operations that we can perform on the plaintext to convert them into unreadable text which is also known as Ciphertext or Encrypted Text. The Round operations to be performed in SHL are:

1) Permutation Box
2) Substitution Box
3) Transformations
4) Function Box
5) Permutation Box

1.2.1 Permutation Box

Permutation Box is well known for the interchange of the positions of the bits in the matrix. This Permutation Box will help us to generate confusion over the bits. Permutation Box is categorized into 5 types:

1) **Initial Permutation:** This Initial Permutation operates interchange at the start of the block or the start of the round operation.

2) **Final Permutation:** Final Permutation operates interchange at the ending of the block or end of the round operation.
3) **Key Permutation:** Key Permutation boxes, also known as PC1 and PC2, are used in key scheduling of block ciphers like DES.
4) **Expansion Permutation: The** Expansion Permutation box is used in Feistel Cipher, it expands the input half-block to match the size of the round key for XOR operation.
5) **Permutation:** The Permutation box is used in the middle rounds of Feistel Ciphers, it shuffles the bits of the input half-block according to a fixed permutation table.
6) **Algorithm Specific Permutation:** This permutation is also known as free hand permutation which gives the flexibility for the vendor of the cryptography algorithm to generate or to create a new Permutation box according to the algorithm.

This Project has adapted the Algorithm Specific Permutation. So, under this permutation, we perform three operations which are known as:

1) **Shift Up:** Under this operation, we rotate on the matrix in the direction of Up.
2) **Shift Left:** After performing Shift Up on a matrix, now we perform Shift Left on the resultant matrix of the Shift Up. Under Shift Left the rotation of the matrix is performed in the direction of Left.
3) **Shift Up:** This Shift Up operation is the same as the above one, which performs a rotation operation on the matrix in the upward direction.

This Permutation operation is performed at the start and ending in the Round operation.

1..2.2 Substitution Box

The main idea of this substitution box is to convert the text into another value. Under this project, we concat 4 bit as a string and the s-box is to convert the values.

1.2.3 Transformation Box

The transformation box will perform 1’s complement on the matrix

1.2.4 Function Box

Function box will perform the XOR operation between the Key and Plaintext and perform Transformation.

IMPLEMENTATION

The implementation process for Encryption

Step 1: As Input, Take the plaintext, Secret Message, and Salt from the user.

Step 2: Derive the Key by using the Key Derivative Function(KDF) with the help of a Secret message and salt.

Step 3: Convert the key derived from KDF and Plaintext into Binary format

Step 4: Arrange the plaintext into an 8*16 matrix by partitioning the plaintext into 128-bit blocks. Perform padding if required.

Step 5: Now, perform Round operation 7 times on each block. In each round, the following operations would be performed commonly. Those are Permutation, Substitution, Transformation, Function Box, and again Permutation.

Step 6: The result is added up and converted from binary to text.

The implementation process for Decryption

Step 1: As Input, Take the Ciphertext, Secret Message, and Salt from the user.

Step 2: Derive the Key by using the Key Derivative Function(KDF) with the help of a Secret message and salt.

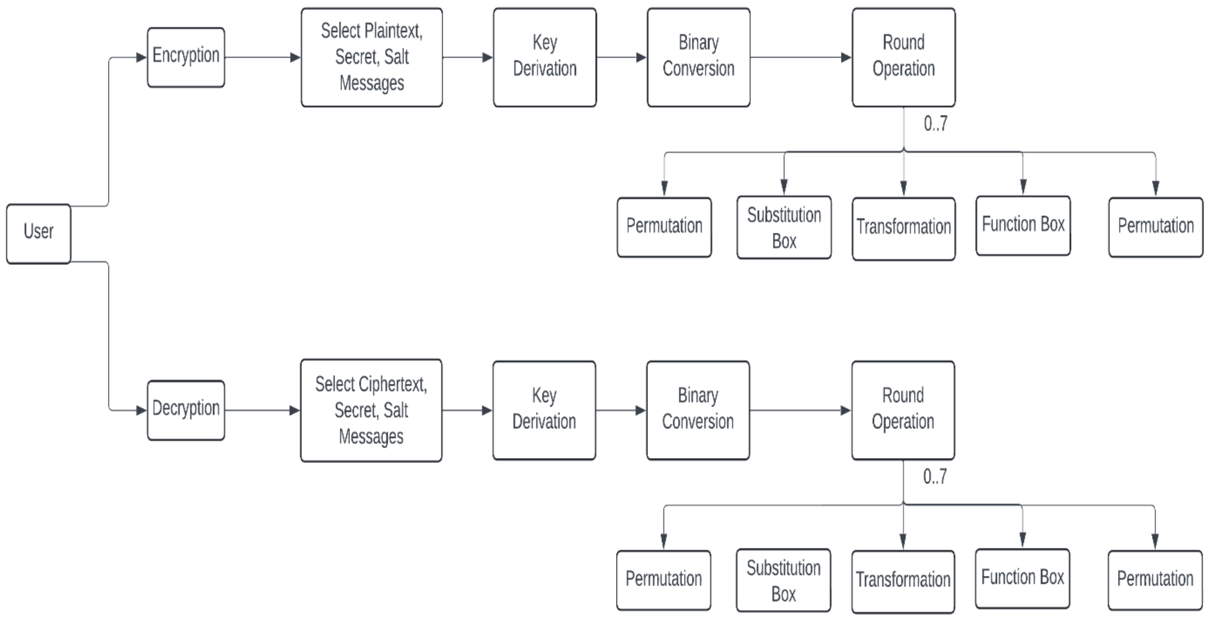
Step 3: Convert the key derived from KDF and Ciphertext into Binary format

Step 4: Arrange the plaintext into an 8*16 matrix by partitioning the Ciphertext into 128-bit blocks. Perform padding if required.

Step 5: Now, perform the Round operation 7 times on each block. In each round, the following operations would be performed commonly. Those are Permutation, Substitution, Transformation, Function Box, and again Permutation.

Step 6: The result is added up and converted from binary to Plaintext.

Process Flow

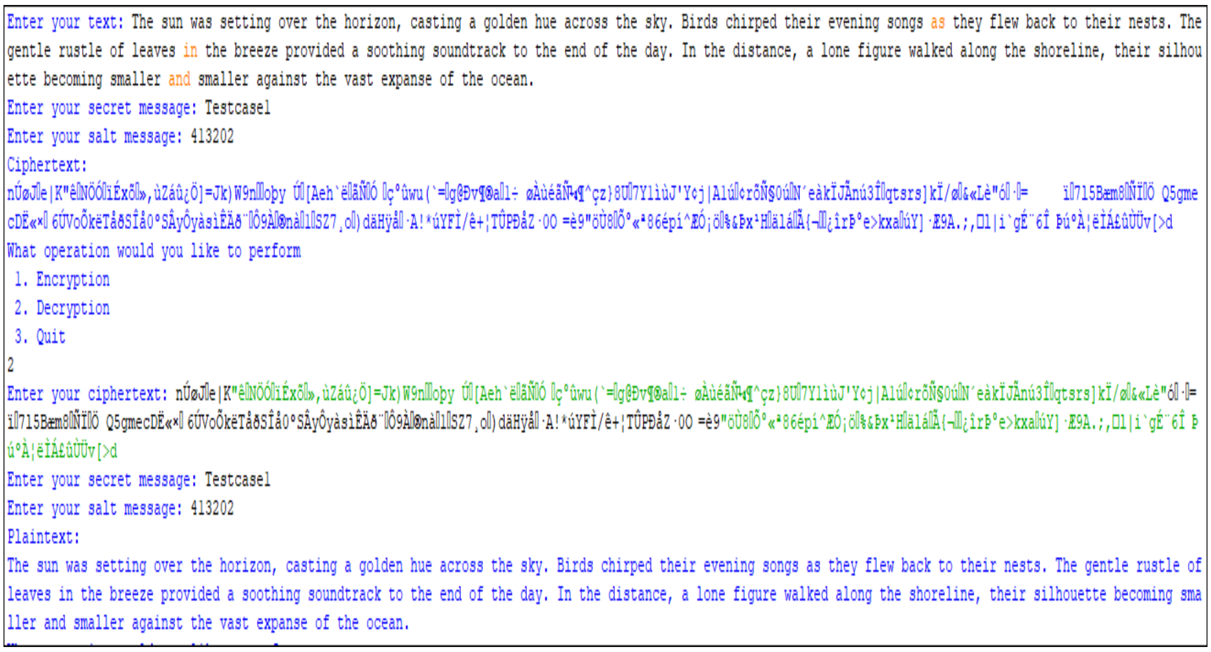


TEST CASES

S. No	No. Of Test Case	Type of the Test Case	Size of the Input	Status
1	Test Case 1	Encryption	34Kb	Success/Encrypted
2	Test Case 2	Decryption	34Kb	Success/Decrypted
3	Test Case 3	Encryption	220Kb	Success/Encrypted
4	Test Case 4	Decryption	220Kb	Success/Decrypted
5	Test Case 5	Encryption	1Mb	Success/Encrypted
6	Test Case 6	Decryption	1Mb	Success/Decrypted

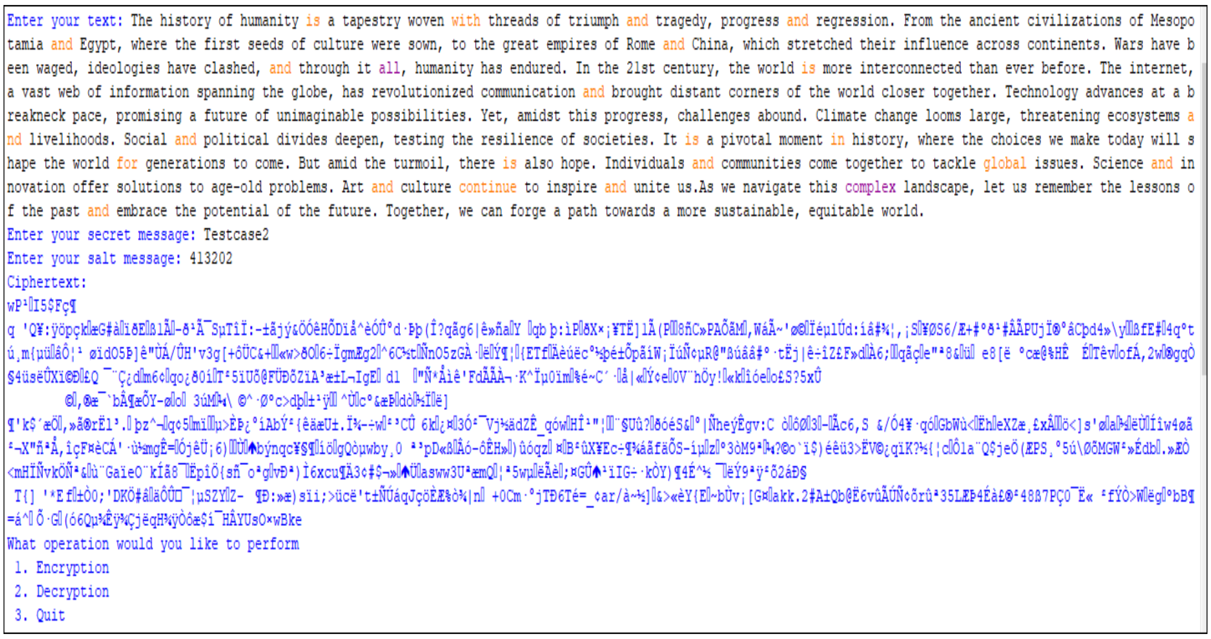
Output Screen:

Test Case 1 & 2:



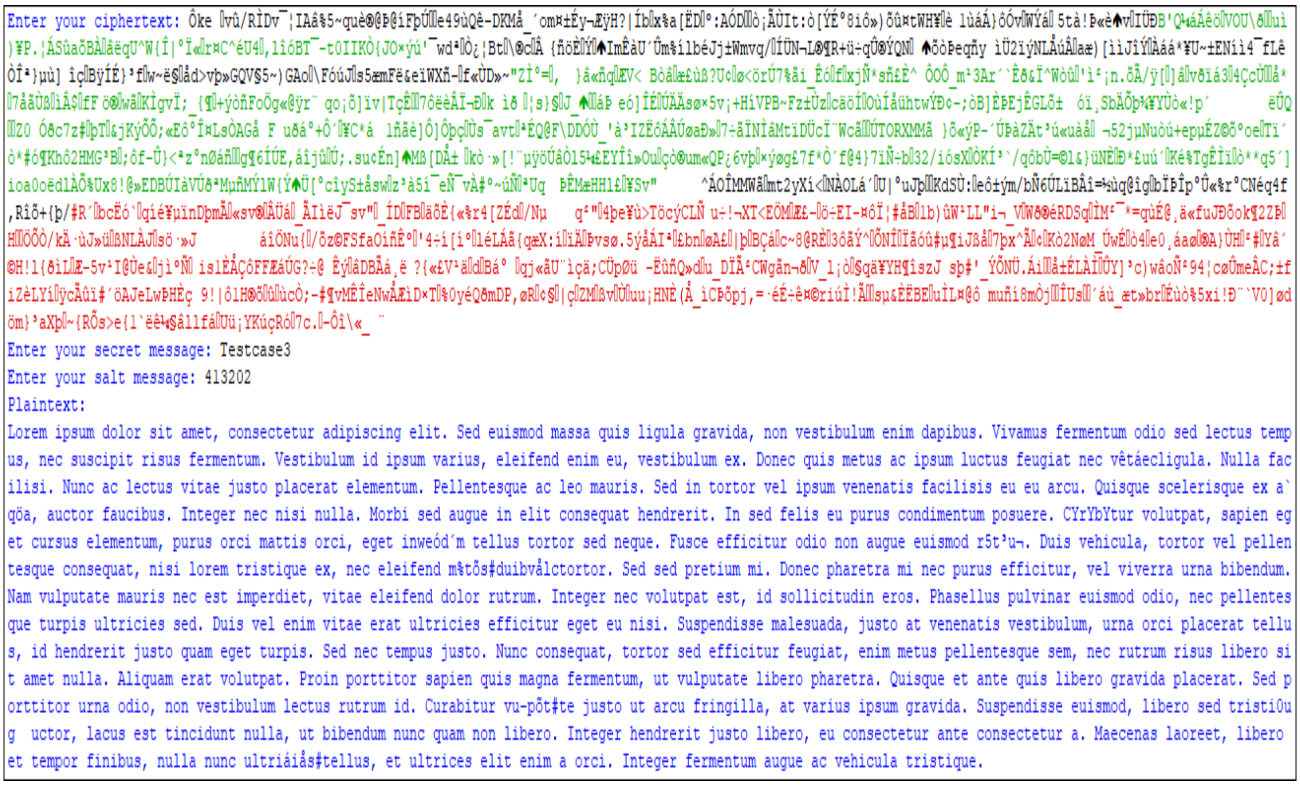
```
Enter your text: The sun was setting over the horizon, casting a golden hue across the sky. Birds chirped their evening songs as they flew back to their nests. The
gentle rustle of leaves in the breeze provided a soothing soundtrack to the end of the day. In the distance, a lone figure walked along the shoreline, their silhou
ette becoming smaller and smaller against the vast expanse of the ocean.
Enter your secret message: Testcasel
Enter your salt message: 413202
Ciphertext:
nÚøJ▯e|K"ê▯NÖÓ▯iÉxõ▯»,ùZáû¿Ö]=Jk)W9n▯▯oþy Ú▯[Aeh`ë▯ãÑ▯Ó ▯ç°ûwu(`=▯g@Ðv¶®a▯1÷ øÀùéãÑ¼¶^çz}8U▯7Y1ìùJ'Y¢j|Alú▯¢rõÑ§0ú▯N´eàkÏJÃnú3Î▯qtsrs]kÏ/ø▯&«Lè"ó▯·▯=     ï▯715Bæm8▯ÑÏ▯Ö Q5gme
cDË«×▯ 6ÚVoÕkëTåðSÎå0°SÂyÔyàsìÊÄð¨▯Ô9À▯®nà▯1▯SZ7¸o▯)däHÿå▯·A!*úYFÌ/ê+¦TÛPÐåZ·0O =è9"öÙ8▯Õ°«ª86épí^ÆÓ¡ö▯%&Þx¹H▯äláÃ{¬▯▯¿îrÞ°e>kxa▯úY]·Æ9A.;,□1|i`gÉ¨6Î Þú°À¦ëÌÁ£ûÙÜv[>d
What operation would you like to perform
 1. Encryption
 2. Decryption
 3. Quit
2
Enter your ciphertext: nÚøJ▯e|K"ê▯NÖÓ▯iÉxõ▯»,ùZáû¿Ö]=Jk)W9n▯▯oþy Ú▯[Aeh`ë▯ãÑ▯Ó ▯ç°ûwu(`=▯g@Ðv¶®a▯1÷ øÀùéãÑ¼¶^çz}8U▯7Y1ìùJ'Y¢j|Alú▯¢rõÑ§0ú▯N´eàkÏJÃnú3Î▯qtsrs]kÏ/ø▯&«Lè"ó▯·▯=
ï▯715Bæm8▯ÑÏ▯Ö Q5gmecDË«×▯ 6ÚVoÕkëTåðSÎå0°SÂyÔyàsìÊÄð¨▯Ô9À▯®nà▯1▯SZ7¸o▯)däHÿå▯·A!*úYFÌ/ê+¦TÛPÐåZ·0O =è9"öÙ8▯Õ°«ª86épí^ÆÓ¡ö▯%&Þx¹H▯äláÃ{¬▯▯¿îrÞ°e>kxa▯úY]·Æ9A.;,□1|i`gÉ¨6Î Þ
ú°À¦ëÌÁ£ûÙÜv[>d
Enter your secret message: Testcasel
Enter your salt message: 413202
Plaintext:
The sun was setting over the horizon, casting a golden hue across the sky. Birds chirped their evening songs as they flew back to their nests. The gentle rustle of
leaves in the breeze provided a soothing soundtrack to the end of the day. In the distance, a lone figure walked along the shoreline, their silhouette becoming sma
ller and smaller against the vast expanse of the ocean.
```

Test Case 2:



```
Enter your text: The history of humanity is a tapestry woven with threads of triumph and tragedy, progress and regression. From the ancient civilizations of Mesopo
tamia and Egypt, where the first seeds of culture were sown, to the great empires of Rome and China, which stretched their influence across continents. Wars have b
een waged, ideologies have clashed, and through it all, humanity has endured. In the 21st century, the world is more interconnected than ever before. The internet,
a vast web of information spanning the globe, has revolutionized communication and brought distant corners of the world closer together. Technology advances at a b
reakneck pace, promising a future of unimaginable possibilities. Yet, amidst this progress, challenges abound. Climate change looms large, threatening ecosystems a
nd livelihoods. Social and political divides deepen, testing the resilience of societies. It is a pivotal moment in history, where the choices we make today will s
hape the world for generations to come. But amid the turmoil, there is also hope. Individuals and communities come together to tackle global issues. Science and in
novation offer solutions to age-old problems. Art and culture continue to inspire and unite us.As we navigate this complex landscape, let us remember the lessons o
f the past and embrace the potential of the future. Together, we can forge a path towards a more sustainable, equitable world.
Enter your secret message: Testcase2
Enter your salt message: 413202
Ciphertext:
wP¹I5$Fç¶
q 'Q¥:ÿöpçkæG#àïöEß1Ã-ð¹Ã¯SµTîÏ:-±ãjý&ÖÓêHÕDïå^èÓÛ°d·Þþ(Î?qãg6|ê»ñaY qb þ:iPðX×;¥TË]1Ã(P8ñC»PAÕãM,WáÃ~'øÏéµ1Úd:íâ#¾¦,;S¥ØS6/Æ+#°ð¹#ÂÃPUjÏ®°âCþd4»\yßfE#4q°t
ú,m{µüâÔ¦¹ øïdO5Þ]ê"ÜÁ/ÛH'v3g[+õÜC&+«w>ðO6÷ÏgmÆg2^6C¾tÑn05zGA·ðÝ¶¦[ETfÄèüéc°¼þé±Öp&iW;ÏüÑ¢µR@"ßúâå#°·tÈj|ê÷ìZ£F»dÃ6;qãçle"ª8&ü e8[ë °cæ@%HÊ ÉTèvofÅ,2w®gqÔ
§4üsëÜXí©Ð£Q ¯¨Ç¿dm6¢qo¿ðOíT²5iÚð@FÜÐõZiA³æ±L¬IgE dl "Ñ*Àiê'FdÃÂÀ¬·K^ÌµOïm%é~C´·ã|«Ý¢eOV¨hÖy!«kióeo£S?5xÛ
        ©,®æ¯`bÅ¶æÕY-øo 3úM¼\ ©^·Ø°c>dþ±¹ÿ ^Üc°&æÞdò¼íë]
¶'k$´æÓ,»â®rÉ1³.þz^-q¢5mïµ>Èþ¿°iAbÝ²{ëäæU±.Ï¾-÷wl²³CÜ 6k¿¤3Ó²¯Vj¼ádZÊ_qówHÎ¹"¦¨§Uû?ßóé$&°|ÑheýÊgv:C òlõØ3-Ãc6,S &/Ö4¥·qóGbWü<ÈheXZæ¸£xÂò<]s'øaÞèÜÏîw4øâ
²¬X"ñ*Å,îçF¤éCÀ'·ü¬mgÊ=Öjêü;6)ÜÙþýnqc¥§ïóqÒòwby,O ª³pD«ßÀó-ôÉH»)úóqzl ¤B¹ûX¥Ec÷¶¼ãäfãÕS-iµlz°3óM9ª¼?®o`î$)éêü3>ËV®¿qìK?¼{¦clÔla¨Q$jeÖ(ÆPS¸°5ú\ØõMGW²»Édb.»ÆÒ
<mHÏÑvkÕÑª&ù¨GaïeO¨kÎã8¯ÊpïÖ{sñ¯o°gvÐª)Ï6xcu¶Ã3¢#§¬»ÙIasww3Uªæm¦ª5wµëÃè;¤GÛ¹iIG÷·kÒY)¶4É^½ ¯ëY9ª ÿ²õ2âÐ§
 T{] '*Ef±Òò;'DKÖ#âäÖÛ¯¦µSZYZ- ¶Ð:»æ)sïì;>ücë't±ÑüâgJçöÉÆ%ò¼|n +OCm·°jTÐ6Té=_¢ar/à¬½]&>«èY{E¬bÜv¦[G«akk.2#A±Qb@Ê6vüÁÚÑ¢õrû³35LÆÞ4Êá£®²4ß7PÇO¯Ë« ²fÝÒ>Wëg°bB¶
=á^Õ·G(ó6Qµ¾Êý¾Çj€qH¾ÿÒ6æ$i¯HÄYUsO×wBke
What operation would you like to perform
 1. Encryption
 2. Decryption
 3. Quit
```

Test Case 3:



```
Enter your ciphertext: Ôke vû/RÌDv¯¦IAâ%5~què@Þ@íFþÚe49üQê-DKMå´om¤Êý¬ÆÿH?|Ìbx%a[ÉD°:AÓDò;ÂÙIt:ò[ÝÉ°8iô»)ôû¤tWH¥è 1úáÁ}ôÓvWÝá 5tà!Þ«è♠vÎÜÐB'Q¼áÂêôVOU\ðui
)¥P.¦ÂSûaôBÃäëqU^W{Î|°Ï«r¤C^éU4,1íóBT¯-tOIIKÖ{JO×yú'¨wdªÖ¿¦Btl\®cA {ñòÈÝ♠ImÊàU´Ûm%ilbéJj±Wmvq/ÍÜN-L®¶R+ü÷qÛ@ÝQN ♠õòÞeqñy ìÜ2iýNLÂúÂæ)[ììJiÝÂáá*¥U~±ENíi4¯fLê
ÔÎ²}µù] içBÿÍÉ}³fw~ë§d>vþ»GQV§5~)GAo\FóúJs5æmFë&eïWXñ-f«ÜD»"Zî°=, }â«ñqÆV< Bóálæ£úß?Uo<örÜ7%äi_Êólf xjÑ*sñ£È^ ÔOÔ_m¹3Ar´´Ê&Î^WóÜ'i²¡n.ðÃ/ÿ[]âlvðïá3¼ÇcÜÂ*
7åãÜßiÂ$ïfF ö®wäKÌgvÎ; {¶+ýòñFoÖg«@ÿr¨ qo¡ð]iv|TçÉ7óëâÏ-Ðk ìð ¦s}§J ♠äÞ eó]ÎÉÜÄÄsø×5v¦+HiVPB~Fz±Uzcäö0ûíåühtwÝÐ¢-;öB}ÉÞEjÊGLô± óï¸SbÃÔþ¾¥YÜò«!p´ ëÛQ
Z0 Ó8c7z#pTl&jKýÔÕ;«Eó°Ì¤LsÖAGâ F uðâ°+Ô´Ì¥C*á 1ñãè]Ô]ÔþçÜs¯avtlªÉQ@F\DDÓÙ_'à³IZÊÖÂÂÜøaÐ»7÷âÏNÌâMtiDÜcÏ¨WcäÜTORXMMã }õ«ýP-´ÚÞàZÄt³ú«uàå ¬52µNuòú+epµÉZ©õ°oeTi´
ò*#ó¶Khô2HMG³B;ôf-Û}<ªz°nØáñg¶éÍÚE,âîjûÜ;.su¢Ên]♠Mß[DÁ± kò·»[!¨µÿóÚáÒ15¼ËEYÎì»OuçõømQP¿6vþ×ýøg£7f*Ò´f@4}7iÑ-b32/íósXÔKÍ³`/qôbÜ=©1&}üNÈÞ*£uú´Ké%TgÊÎìò**q5´]
ìoaOoëdlÃÕ%Ux8!@»EDBÚIàVÜðªMµñMÝ1W{Ý♠Ü[°ciyS±åswz³à5í¯eÑ¯vÄ#°~úÑªUq ÞÉM¤HH1£¥5v" ^ÀÕÌMMWâmt2yXi<NÁÓLá´Ü|°uJþKdSÜ:eó±ým/bÑ6ÜLiBÂí=¼úq@íqbÏÞÎp°Û«%r°CNéq4f
,Rìõ+{þ/#R´bcËó¨qíé¥uìnDþmÃ«sv®ÂÜá ÃIìëJ¯sv" ÏDFBàõÈ{«%r4[ZÉd/Nµ q²"4þe¥ü>TöcýCLÑ u÷!-XT<EÓMÆ£-ö-EI-¤ôÏ¦#àB1b)üW¹LL"i¬_VW8®éRDSqÎM²¯*=qüÉ@¸ä«fuJÐõok¶2ZÞ
HÕÖÒ/kÄ·ùJ»ülßNLÀJsö·»J áîÖNu{/õz®FSfaÓiñÊ°'4÷í[i°leLÁã{qæX:iÏiÄÞvsø.5ýâAÏ*£bnøA£|þBÇálc~8@RÈ3óâÝ^ÕNÎÏÏãóû#µ¶iJßå7þx^ä¢Kó2NøM_ÜwÊò4e0¸äaø®A}ÜH²#ÝÝá´
©H!1{ðìLÆ-5v¹I@Ûe&jì°Ñ isLÈÃÇöFFÆÁÚG?÷@ ÊýiâDBÂä¸ë ?{«£V¹ädBá° qj«ãU¨içä;CÜpØü -ÈûñQ»du_DÏÂ²CWgãn-ðV_1¦ö§qä¥YH¶iszJ sþ#'_ÝÕNÜ.AiÂ±ÉLÀÎÛY]³c)wâoÑ²94¦cøÛmeÀC;±f
iZèLYìýçÃüí#´öAJeLwÞHÈç 9!|ö1H®ðüc0;-#¶vMÊÎeNwÂÆiD×Tì%OyéQ®mDP,øR¢§|çlZMßvÜuu;HNÉ(Â_ìCÞöpj,=·éÉ÷ê¤®riúÎ!ÂsµÈËBÊuÎL¤@ó muñí8mÖjÎÏUs´áù æt»brÏÉùò%5xi!Ð¨`V0]ød
öm}³aXþ~{RÕs>e{1`ëê¼§âllfáÜü;YKúçRó7c.-Ôî\«_ ¨
Enter your secret message: Testcase3
Enter your salt message: 413202
Plaintext:
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed euismod massa quis ligula gravida, non vestibulum enim dapibus. Vivamus fermentum odio sed lectus temp
us, nec suscipit risus fermentum. Vestibulum id ipsum varius, eleifend enim eu, vestibulum ex. Donec quis metus ac ipsum luctus feugiat nec vêtáecligula. Nulla fac
ilisi. Nunc ac lectus vitae justo placerat elementum. Pellentesque ac leo mauris. Sed in tortor vel ipsum venenatis facilisis eu eu arcu. Quisque scelerisque ex a`
qöa, auctor faucibus. Integer nec nisi nulla. Morbi sed augue in elit consequat hendrerit. In sed felis eu purus condimentum posuere. CYrYbYtur volutpat, sapien eg
et cursus elementum, purus orci mattis orci, eget inweód´m tellus tortor sed neque. Fusce efficitur odio non augue euismod r5t³u¬. Duis vehicula, tortor vel pellen
tesque consequat, nisi lorem tristique ex, nec eleifend m%tõs#duibvålctortor. Sed sed pretium mi. Donec pharetra mi nec purus efficitur, vel viverra urna bibendum.
Nam vulputate mauris nec est imperdiet, vitae eleifend dolor rutrum. Integer nec volutpat est, id sollicitudin eros. Phasellus pulvinar euismod odio, nec pellentes
que turpis ultricies sed. Duis vel enim vitae erat ultricies efficitur eget eu nisi. Suspendisse malesuada, justo at venenatis vestibulum, urna orci placerat tellu
s, id hendrerit justo quam eget turpis. Sed nec tempus justo. Nunc consequat, tortor sed efficitur feugiat, enim metus pellentesque sem, nec rutrum risus libero si
t amet nulla. Aliquam erat volutpat. Proin porttitor sapien quis magna fermentum, ut vulputate libero pharetra. Quisque et ante quis libero gravida placerat. Sed p
orttitor urna odio, non vestibulum lectus rutrum id. Curabitur vu-põt#te justo ut arcu fringilla, at varius ipsum gravida. Suspendisse euismod, libero sed tristi0u
g  uctor, lacus est tincidunt nulla, ut bibendum nunc quam non libero. Integer hendrerit justo libero, eu consectetur ante consectetur a. Maecenas laoreet, libero
et tempor finibus, nulla nunc ultriáiås#tellus, et ultrices elit enim a orci. Integer fermentum augue ac vehicula tristique.
```

CONCLUSION

In conclusion, the provided code implements a custom lightweight cryptographic algorithm for both encryption and decryption processes. This algorithm is designed to secure data transmission by transforming plaintext into ciphertext and vice versa using a series of matrix operations, substitution, permutation, and key derivation techniques. Here are the key points highlighted by the code:

Encryption Process

- The EncryptionProcess class takes user input for text, a secret message (IK), and a salt message.
- It converts the input text to binary and generates a Pseudorandom Key (PRK) using HKDF (HMAC-based Key Derivation Function) with the salt and IK.
- The algorithm then expands the PRK into multiple output keys for encryption.
- The plaintext binary is divided into matrices, and a series of operations are performed on each matrix:
    1. Permutation: Shifting rows and columns of the matrix.
    2. Substitution: Replacing matrix elements using a substitution box (s_box).
    3. Transformation: Inverting matrix elements.
    4. Core Function (function_box): XORing the matrix with an output key and applying a transformation.
- The resulting ciphertext is then converted back to a readable format and displayed to the user.

Decryption Process

- The DecryptionProcess class takes user input for ciphertext, a secret message (IK), and a salt message.
- It follows a similar process as encryption but in reverse:
- Generates the same PRK using HKDF with the salt and IK.
- Expands the PRK into output keys for decryption.
- Divides the ciphertext binary into matrices and performs reverse operations:
- Reverse Permutation: Shifting rows and columns in the opposite direction.
- Reverse Substitution: Replacing matrix elements using the reverse substitution box.
- Reverse Transformation: Inverting matrix elements back to their original state.
- The resulting binary plaintext is converted to readable text and displayed.

References

1. Ch. Rupa, Greeshmanth and Mohd Asif Shah: A Study on Lightweight Symmetric Encryption Algorithm for data protection. https://doi.org/10.1186/s13677-023-00425-7
2. Mohammad Abdulla: A Detailed study on the Advanced Encryption Algorithm. https://www.researchgate.net/publication/317615794
3. Vishal A. Thakor, Mohammad Abdur Razzaque (Member, IEEE), and Muhammad R. A. Khandaker, (Senior Member, IEEE): A Study on the Key challenges and resources of IoT. Different approaches and formulas to calculate the performance of an Algorithm. 10.1109/ACCESS.2021.3052867
4. A.Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin, and C. Vikkelsoe, "Present: An ultra-lightweight block cipher," in Proc. 9th Int. Workshop Cryptograph. Hardw. Embedded Syst. Berlin, Germany: Springer, Sep. 2007, pp. 450-466. [Online]. Available: https://link.springer.com/chapter/10. 1007/978-3-540-74735-2_31
5. S. Banik, S. K. Pandey, T. Peyrin, Y. Sasaki, S. Sim, and Y. Todo. (2007). Gift: A Small Present Towards Reaching the Limit of Lightweight Encryption (Full Version). [Online]. Available: https://infoscience.epfl.ch/record
6. F.-X. Standaert, G. Piret, G. Rouvroy, J.-J. Quisquater, and J.-D. Legat, "ICEBERG: An involutional cipher efficient for block encryption in reconfigurable hardware," in Proc. Int. Workshop Fast Softw. Encryption. Berlin, Germany: Springer, 2004, pp. 279-298.
7. H. Cheng, H. M. Heys, and C. Wang, "PUFFIN: A novel compact block cipher targeted to embedded digital systems," in Proc. 11th EUROMI- CRO C Digit. Syst. Design Architect., Methods Tools, 2008, pp. 383-390.
8. J. Borgho et al., "PRINCE-A low-latency block cipher for per-pervasive computing applications," in Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secure. (ASIACRYPT), Adv. Cryptol. Springer, 2012, pp. 208-225. [Online]. Available: https://link.springer.com/chapter/10. 1007/978-3-642-34961-4 14
9. D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, and B. Koo, "Hight: A new block cipher suitable for the low-resource device," in Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst. Berlin, Germany: Springer, 2006, pp. 46-59.
10. J. John, "BEST-1: A lightweight block cipher," IOSR J. Compute. Eng.. vol. 16, no. 2, pp. 91-95, 2014.
11. D. Hong, J.-K. Lee, D.-C. Kim, D. Kwon, K. H. Ryu, and D.-G. Lee, "LEA: A 128-bit block cipher for fast encryption on common processors," in Proc. Int. Workshop Inf. Secure. Appl. Cham, Switzerland: Springer, 2013, pp. 3-27.