Quantum
Smart information technology

SDC Safe Note (SDCSN)

Steganography

The art of injecting invisible File Content into an image pixel’s noise and color

SDCSN Sophisticated Technique
In Data Security for Secure transmission

SDC Safe Note (SDCSN)
Noise Data

Noise data is Data injected into an Image Pixel’s Noise. Noise in an image is the presence of artifacts that do not originate from the original scene content. Generally speaking, noise is a statistical variation of a measurement created by a random process. In imaging, noise emerges as an artifact in the image that appears as a grainy structure covering the image. The main types of image noise are random noise, fixed pattern noise, and banding noise. Random noise is shown by the fluctuation of the colors above the actual intensity of the image.

Confidential Data injected inside the image Pixel’s Noise
Can be sent thru unsecured internet public connection
without the need to install any complex infrastructure.

SDC Safe Note (SDCSN)

The difference between Steganography and Cryptography

The steganography and cryptography are the two sides of a coin where the steganography hides the traces of communication while cryptography uses encryption to make the message unreadable. On the other hand, the cryptography alters the standard secret message structure when transferred across the network.

Injecting Confidential Data inside the Image
Pixel’s noise is completely secure
while transmitting

SDC Safe Note (SDCSN)

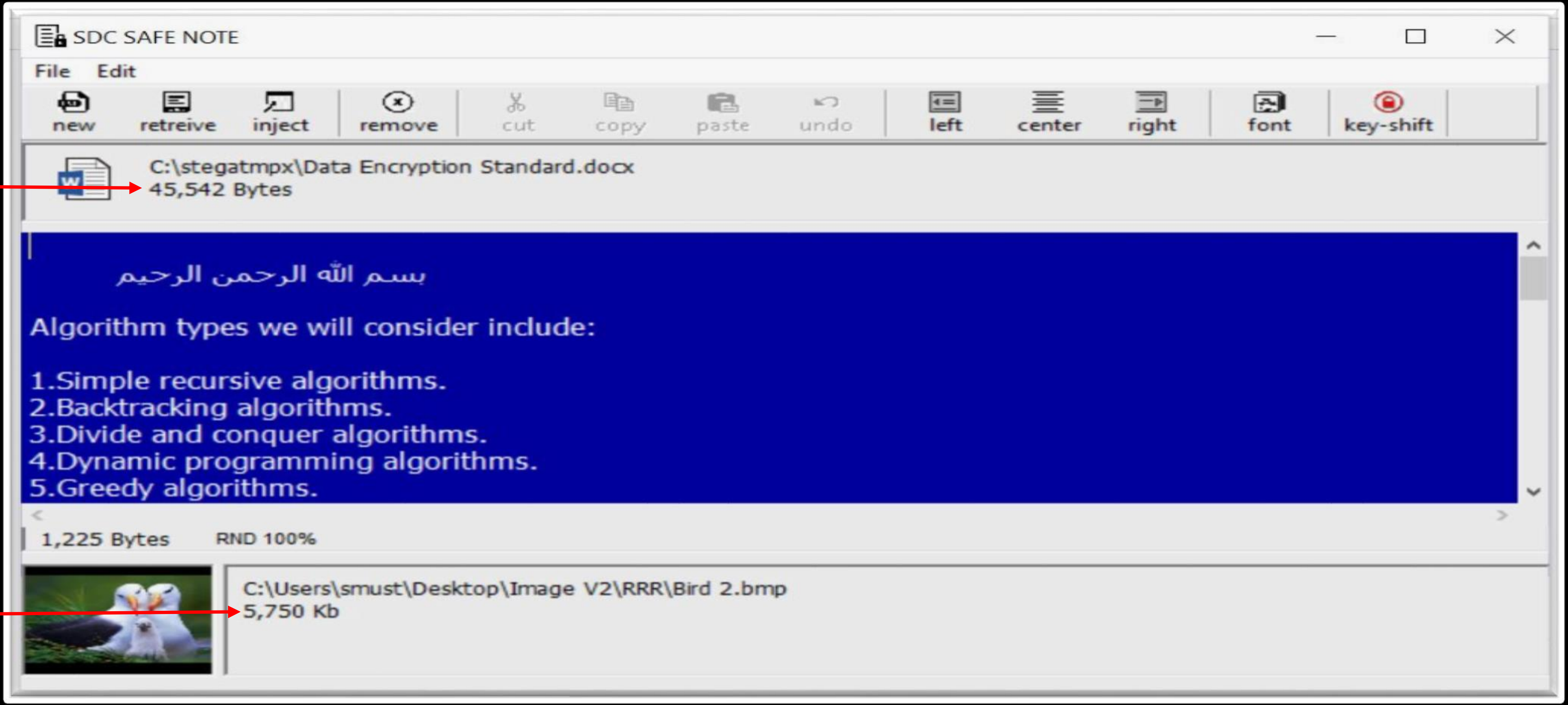
SDC Safe Note is the solution to protect and transmit sensitive information inside an image in a hidden and secure way without any change in the image size and color.

SDCSN Technique

- **Crypto-Secure Steganography (CSS)** is to inject compressed and encrypted Data inside an image Pixel's noise making it totally invisible for human eyes due to the noise data manipulation process.
- **Random Pixel Positioning (RPP)** ensures that even when it is known that the image is a data carrier, it is impossible to retrieve and access the original injected data.

Chest X-ray (radiographic) image contains Sensitive Data

SDC Safe Note (SDCSN)



C:\stegatmpx\Data Encryption Standard.docx
45,542 Bytes

بسم الله الرحمن الرحيم

Algorithm types we will consider include:

1.Simple recursive algorithms.
2.Backtracking algorithms.
3.Divide and conquer algorithms.
4.Dynamic programming algorithms.
5.Greedy algorithms.

1,225 Bytes RND 100%

C:\Users\smust\Desktop\Image V2\RRR\Bird 2.bmp
5,750 Kb

Data injected into the image

Actual image Size

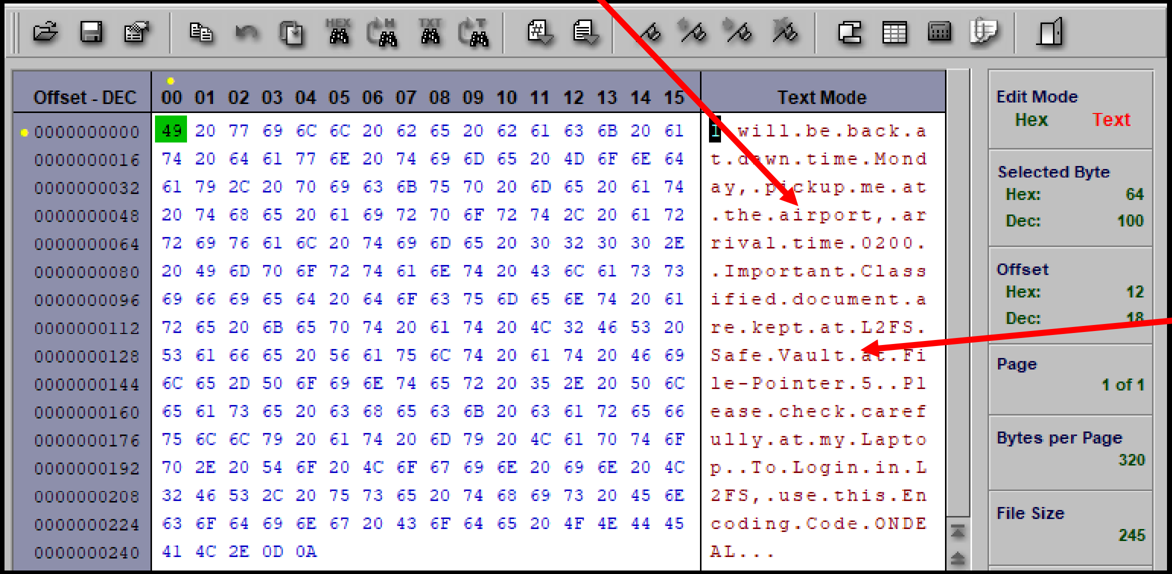
The file content is injected into the image, bit by bit approach, and the size of the image will not change

SDC SAFE NOTE (Steganography Technique)

Injecting Data inside the Image Pixels

Below is the sample information to be injected into the image pixels at the right.

I will be back at dawn time Monday, pickup me at the airport, arrival time 0200. Important Classified documents are kept at L2FS Safe Vault at File-Pointer 5. Please check carefully at my Laptop. To Login in L2FS, use this Encoding Code **ONDEAL**.



```
Offset - DEC  00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15   Text Mode
0000000000    49 20 77 69 6C 6C 20 62 65 20 62 61 63 6B 20 61   I.will.be.back.a
0000000016    74 20 64 61 77 6E 20 74 69 6D 65 20 4D 6F 6E 64   t.dawn.time.Mond
0000000032    61 79 2C 20 70 69 63 6B 75 70 20 6D 65 20 61 74   ay,.pickup.me.at
0000000048    20 74 68 65 20 61 69 72 70 6F 72 74 2C 20 61 72   .the.airport,.ar
0000000064    72 69 76 61 6C 20 74 69 6D 65 20 30 32 30 30 2E   rival.time.0200.
0000000080    20 49 6D 70 6F 72 74 61 6E 74 20 43 6C 61 73 73   .Important.Class
0000000096    69 66 69 65 64 20 64 6F 63 75 6D 65 6E 74 20 61   ified.document.a
0000000112    72 65 20 6B 65 70 74 20 61 74 20 4C 32 46 53 20   re.kept.at.L2FS.
0000000128    53 61 66 65 20 56 61 75 6C 74 20 61 74 20 46 69   Safe.Vault.at.Fi
0000000144    6C 65 2D 50 6F 69 6E 74 65 72 20 35 2E 20 50 6C   le-Pointer.5..Pl
0000000160    65 61 73 65 20 63 68 65 63 6B 20 63 61 72 65 66   ease.check.caref
0000000176    75 6C 6C 79 20 61 74 20 6D 79 20 4C 61 70 74 6F   ully.at.my.Lapto
0000000192    70 2E 20 54 6F 20 4C 6F 67 69 6E 20 69 6E 20 4C   p..To.Login.in.L
0000000208    32 46 53 2C 20 75 73 65 20 74 68 69 73 20 45 6E   2FS,.use.this.En
0000000224    63 6F 64 69 6E 67 20 43 6F 64 65 20 4F 4E 44 45   coding.Code.ONDE
0000000240    41 4C 2E 0D 0A                                    AL...
```

Edit Mode: Hex Text

Selected Byte
Hex: 64
Dec: 100

Offset
Hex: 12
Dec: 18

Page: 1 of 1

Bytes per Page: 320

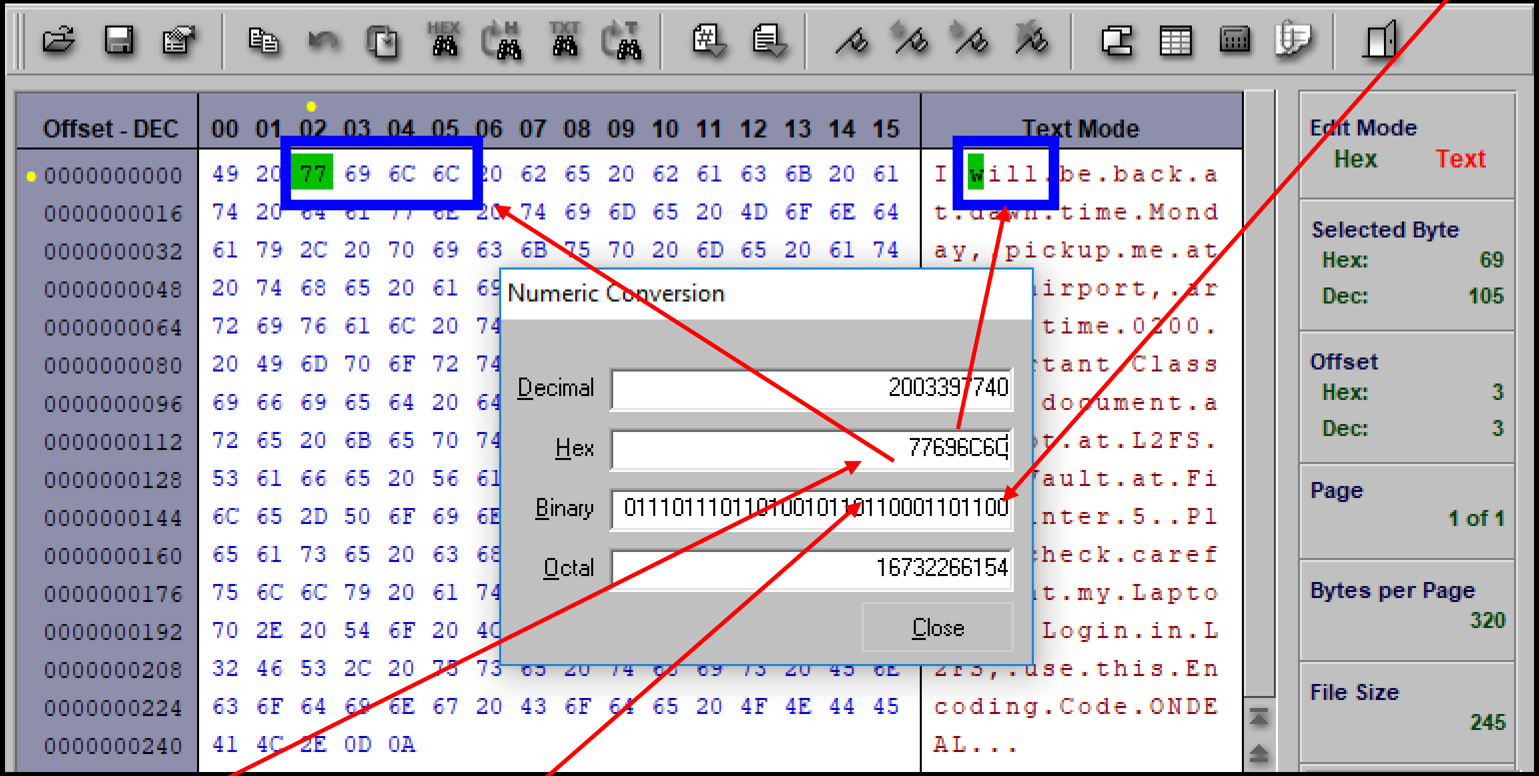
File Size: 245

You noticed that the Data is visible in the Hexadecimal Software Utility because it is not Encrypted. You will see in the next slides after injecting the Data into the Image Pixels.

SDC SAFE NOTE (Steganography Technique)

Getting Characters Binary Value.

Below image is the conversion of Data or Information into Low-Level format which **is Binary numbers (0 and 1)**. Binary is a language of Computers to communicate to Programs or Software.



```
Offset - DEC  00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15   Text Mode
0000000000    49 20 77 69 6C 6C 20 62 65 20 62 61 63 6B 20 61   I.will.be.back.a
0000000016    74 20 64 61 77 6E 20 74 69 6D 65 20 4D 6F 6E 64   t.dawn.time.Mond
0000000032    61 79 2C 20 70 69 63 6B 75 70 20 6D 65 20 61 74   ay,.pickup.me.at
0000000048    20 74 68 65 20 61 69                                   irport,..ar
0000000064    72 69 76 61 6C 20 74                                   time.0200.
0000000080    20 49 6D 70 6F 72 74                                   tant.Class
0000000096    69 66 69 65 64 20 64                                   document.a
0000000112    72 65 20 6B 65 70 74                                   t.at.L2FS.
0000000128    53 61 66 65 20 56 61                                   ault.at.Fi
0000000144    6C 65 2D 50 6F 69 6E                                   nter.5..Pl
0000000160    65 61 73 65 20 63 68                                   heck.caref
0000000176    75 6C 6C 79 20 61 74                                   t.my.Lapto
0000000192    70 2E 20 54 6F 20 4C                                   Login.in.L
0000000208    32 46 53 2C 20 75 73 65 20 74 68 69 73 20 45 6E   2FS,.use.this.En
0000000224    63 6F 64 69 6E 67 20 43 6F 64 65 20 4F 4E 44 45   coding.Code.ONDE
0000000240    41 4C 2E 0D 0A                                    AL...
```

Numeric Conversion

Decimal: 2003397740

Hex: 77696C6C

Binary: 01110111011010010110110001101100

Octal: 16732266154

Edit Mode: Hex Text

Selected Byte
Hex: 69
Dec: 105

Offset
Hex: 3
Dec: 3

Page: 1 of 1

Bytes per Page: 320

File Size: 245

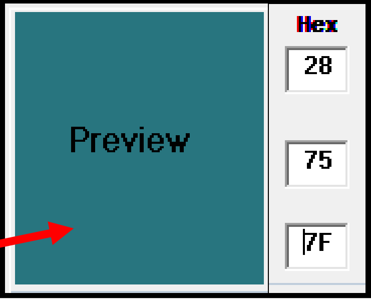
This is only to give you some understanding on how SDC manipulates Data into Low-Level processing which is Binary Processing. Let us see on the next slides on how to process data.

77696c6c Hexadecimal value equivalent to English word **Will in Binary numbers it is 01110111011010010110110001101100.**

SDC SAFE NOTE (Steganography Technique)

Getting Pixels Binary Value.

Below image is the conversion of Data or Information into Low-Level format which **is Binary numbers (0 and 1**).
Binary is a language of Computers to communicate to Programs or Software.



Hex

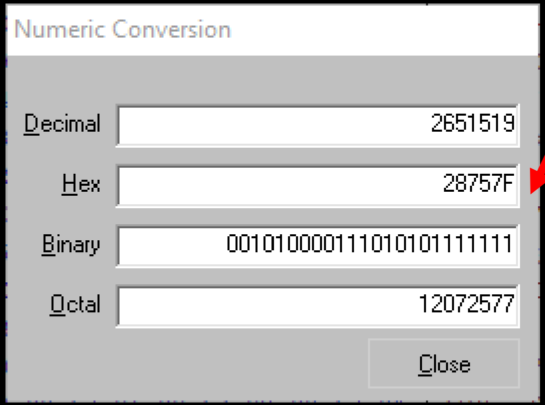
28

75

7F

Pixel is smaller than a DOT. An image has Million Pixels.

This colour above is coming from Image Pixel from the Left. 28757F is an Hexadecimal value of Pixel Colour. In Binary it is 001010000111010101111111.



Numeric Conversion

Decimal 2651519

Hex 28757F

Binary 001010000111010101111111

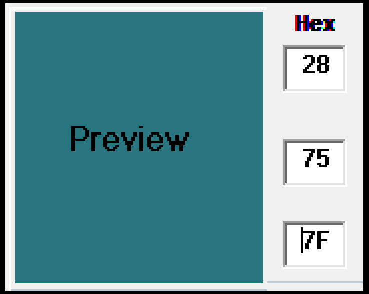
Octal 12072577

Let us see on the next Slide, the Injection Manipulation Process.

SDC SAFE NOTE (Steganography Technique)

Injecting Binary Data value into an Image Pixels Binary value

Below image is the conversion of Data or Information into Low-Level format which **is Binary numbers (0 and 1**)
Binary is a language of Computers to communicate to Programs or Software.

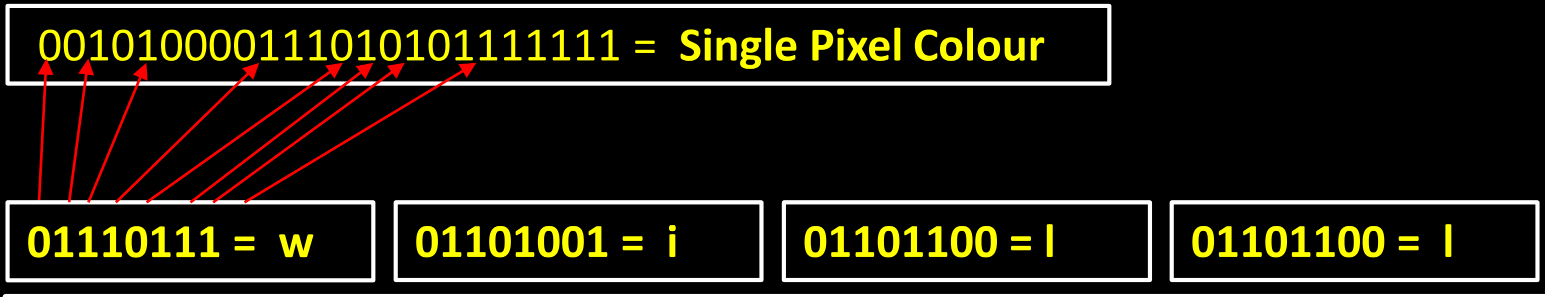

Preview

Hex
28
75
7F

001010000111010101111111 = **Pixel Colour**

01110111011010010110110001101100 = **Binary value of will**

This data security process is totally unbreakable, impossible to retrieve the data from the Image.


001010000111010101111111 = **Single Pixel Colour**

01110111 = w

01101001 = i

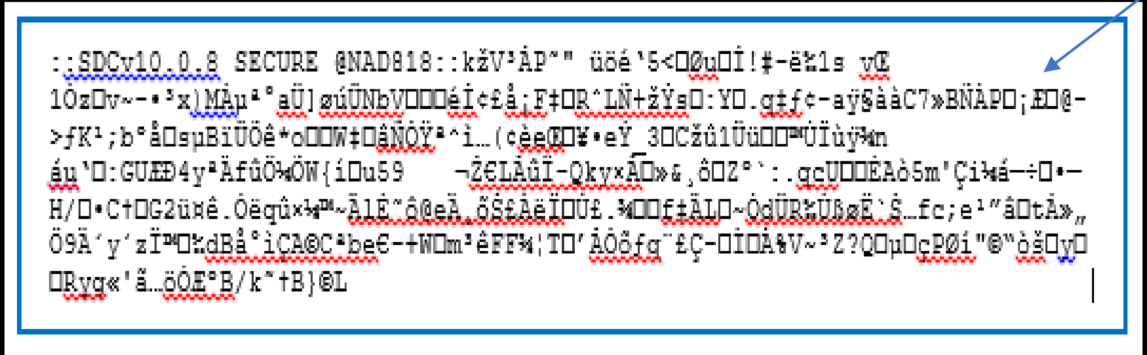
01101100 = l

01101100 = l

w is already allocated to a single-pixel color, the next letter or character will do the same process until all characters are injected into an image. SDC will look for the next pixel color in order to allocate the next characters.

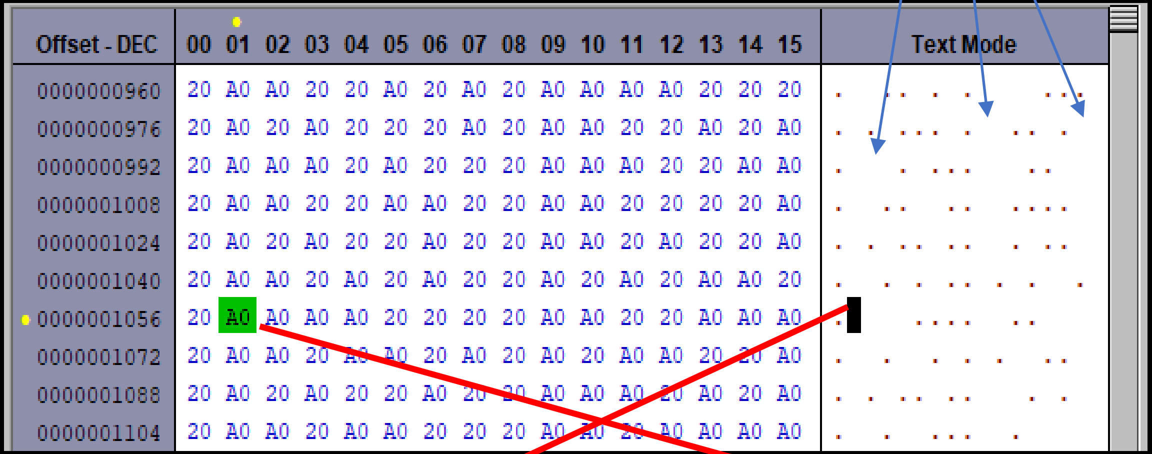
BIT HIDING (Invisible Ciphertext)

SDC Encrypted output or Ciphertext before the Bit Hiding process.



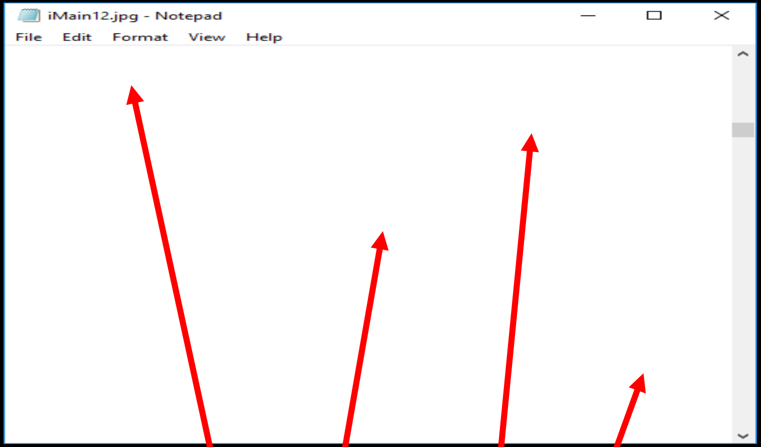
```
::SDCv10.0.8 SECURE @NAD818::kžV³ÅP˜" üöé'5<□ØuΠİ!‡-ë‰1s vŒ
1Òz□v~-•³x)MÀµª°aÜ]øúÜNbV□□□éÌ¢£å;F‡□R^LÑ+žÝs□:Y□.q‡f¢-aÿ§ààC7»BÑÄP□;Æ□@-
>fK¹;b°å□spBïÜÖê*o□□W‡□âÑÖŸª^ì…(çèe®□¥•eŸ 3□Cžû1Üü□□™ÜÏùý¾n
áu'□:GUÆÐ4yªÄfûÖ¼ÖW{í□u59    ¬Ž€LÅûÏ-QkyxÃ□»&¸ô□Z°`:.qcU□□ÈAò5m'Çi¼á—÷□•—
H/□•C†□G2ü¤ê.Öëqûx¼™~ÃlÈ˜ô@eÄ‚õŠ£ÄëÏ□Ü£.¾□□f‡ÃL□~ÒdÜR‰ÜßøË`Ŝ…fc;e¹"â□tÃ»„
Ö9Ã´y´zÏ™□‰dBå°ìÇA©C*be€-+W□m³êFF¾¦T□'ÀÖõfq¨£Ç-□Ì□Å‰V~³Z?Q□µ□çPØí"®"òš□y□
□Ryg«'ã…öÒÆ°B/k"†B}©L
```

Below image is a representation of Invisible characters of the Ciphertext in a low-level byte area. Invisible even in Low-level area.



```
Offset - DEC  00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15   Text Mode
0000000960    20 A0 A0 20 20 A0 20 A0 20 A0 A0 A0 A0 20 20 20   .  .. . .    ...
0000000976    20 A0 20 A0 20 20 20 A0 20 A0 A0 20 20 A0 20 A0   . . ... .  .. . 
0000000992    20 A0 A0 A0 20 A0 20 20 20 A0 A0 A0 20 20 A0 A0   .   . ...   ..  
0000001008    20 A0 A0 20 20 A0 A0 20 20 A0 A0 20 20 20 20 A0   .  ..  ..  .... 
0000001024    20 A0 20 A0 20 20 A0 20 20 A0 A0 20 A0 20 20 A0   . . .. ..  . .. 
0000001040    20 A0 A0 20 A0 20 A0 20 20 A0 20 A0 20 A0 A0 20   .  . . .. . .  .
0000001056    20 A0 A0 A0 A0 20 20 20 20 A0 A0 20 20 A0 A0 A0   .    ....  ..   
0000001072    20 A0 A0 20 A0 A0 20 A0 20 A0 20 A0 A0 20 20 A0   .  .  . . .  .. 
0000001088    20 A0 20 A0 20 20 A0 20 20 A0 A0 A0 20 A0 20 A0   . . .. ..   . . 
0000001104    20 A0 A0 20 A0 A0 20 20 20 A0 A0 20 A0 A0 A0 A0   .  .  ...  .    
```

Above image is the Hexadecimal Conversion Utility. **A0** is the Hexadecimal value **of Invisible Character in Ciphertext.**



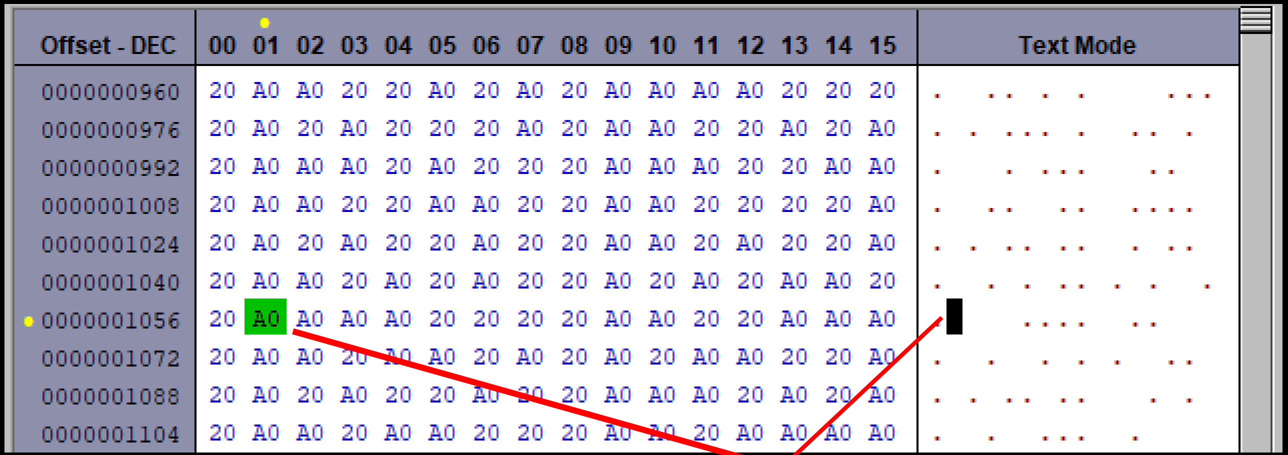
After the Bit Hiding process, the file was opened by Notepad. There is NO Data at all, totally Invisible. This part is a High-Level representation of Invisible Ciphertext in Windows Notepad.
See above image.

Conclusion:
Impossible to recover the Data after Bit Hiding Process.

BIT HIDING (Invisible Ciphertext)

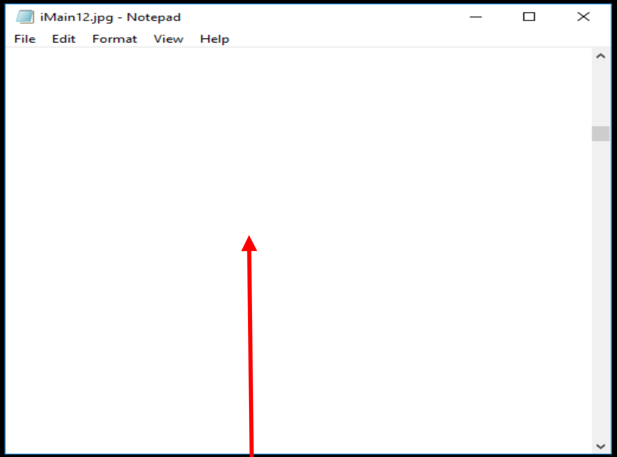
Encryption is the Common Data Protection in the Market today to any business around the world.

SDC Bit Hiding is a NEW TECHNOLOGY in the Market today. Capable of hiding any type of file content and any type of file either known or unknown type of file. The File Content after injection it becomes totally invisible content, cannot be seen by human and any Hexadecimal Software Utility in the market. See image below.



```
Offset - DEC  00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15   Text Mode
0000000960    20 A0 A0 20 20 A0 20 A0 20 A0 A0 A0 A0 20 20 20   .  .. . .    ...
0000000976    20 A0 20 A0 20 20 20 A0 20 A0 A0 20 20 A0 20 A0   . . ... .  .. . 
0000000992    20 A0 A0 A0 20 A0 20 20 20 A0 A0 A0 20 20 A0 A0   .   . ...   ..  
0000001008    20 A0 A0 20 20 A0 A0 20 20 A0 A0 20 20 20 20 A0   .  ..  ..  .... 
0000001024    20 A0 20 A0 20 20 A0 20 20 A0 A0 20 A0 20 20 A0   . . .. ..  . .. 
0000001040    20 A0 A0 20 A0 20 A0 20 20 A0 20 A0 20 A0 A0 20   .  . . .. . .  .
0000001056    20 A0 A0 A0 A0 20 20 20 20 A0 A0 20 20 A0 A0 A0        ....  ..   
0000001072    20 A0 A0 20 A0 A0 20 A0 20 A0 20 A0 A0 20 20 A0   .  .  . . .  .. 
0000001088    20 A0 20 A0 20 20 A0 20 20 A0 A0 A0 20 A0 20 A0   . . .. ..   . . 
0000001104    20 A0 A0 20 A0 A0 20 20 20 A0 A0 20 A0 A0 A0 A0   .  .  ...  .    
```

Above image is the Hexadecimal Conversion Utility. **A0** is the Hexadecimal value of ***Invisible Ciphertext*.**

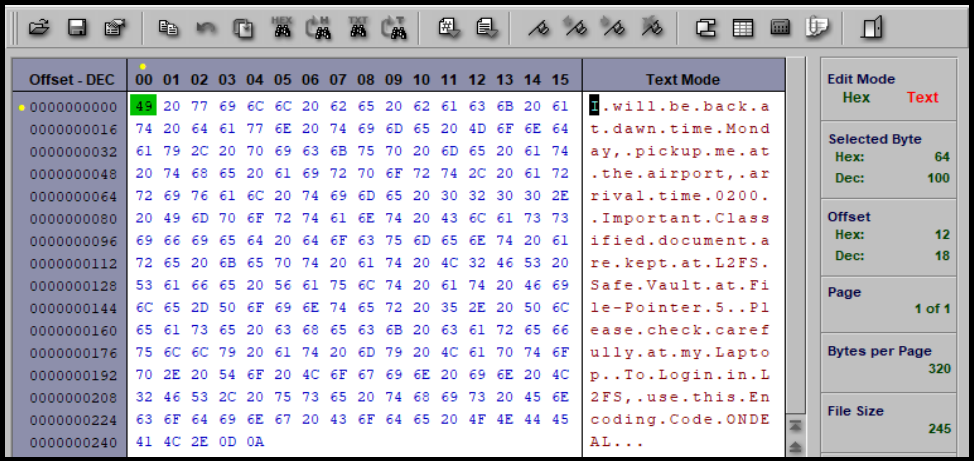


Above is the Display of Notepad for ***Invisible File Content.***

Before and after data injection

Unsuspicious type of data security.

Before Data Injection



```
Offset - DEC  00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15   Text Mode
0000000000    49 20 77 69 6C 6C 20 62 65 20 62 61 63 6B 20 61   I.will.be.back.a
0000000016    74 20 64 61 77 6E 20 74 69 6D 65 20 4D 6F 6E 64   t.dawn.time.Mond
0000000032    61 79 2C 20 70 69 63 6B 75 70 20 6D 65 20 61 74   ay,.pickup.me.at
0000000048    20 74 68 65 20 61 69 72 70 6F 72 74 2C 20 61 72   .the.airport,.ar
0000000064    72 69 76 61 6C 20 74 69 6D 65 20 30 32 30 30 2E   rival.time.0200.
0000000080    20 49 6D 70 6F 72 74 61 6E 74 20 43 6C 61 73 73   .Important.Class
0000000096    69 66 69 65 64 20 64 6F 63 75 6D 65 6E 74 20 61   ified.document.a
0000000112    72 65 20 6B 65 70 74 20 61 74 20 4C 32 46 53 20   re.kept.at.L2FS.
0000000128    53 61 66 65 20 56 61 75 6C 74 20 61 74 20 46 69   Safe.Vault.at.Fi
0000000144    6C 65 2D 50 6F 69 6E 74 65 72 20 35 2E 20 50 6C   le-Pointer.5..Pl
0000000160    65 61 73 65 20 63 68 65 63 6B 20 63 61 72 65 66   ease.check.caref
0000000176    75 6C 6C 79 20 61 74 20 6D 79 20 4C 61 70 74 6F   ully.at.my.Lapto
0000000192    70 2E 20 54 6F 20 4C 6F 67 69 6E 20 69 6E 20 4C   p..To.Login.in.L
0000000208    32 46 53 2C 20 75 73 65 20 74 68 69 73 20 45 6E   2FS,.use.this.En
0000000224    63 6F 64 69 6E 67 20 43 6F 64 65 20 4F 4E 44 45   coding.Code.ONDE
0000000240    41 4C 2E 0D 0A                                     AL...
```

Edit Mode
Hex Text

Selected Byte
Hex: 64
Dec: 100

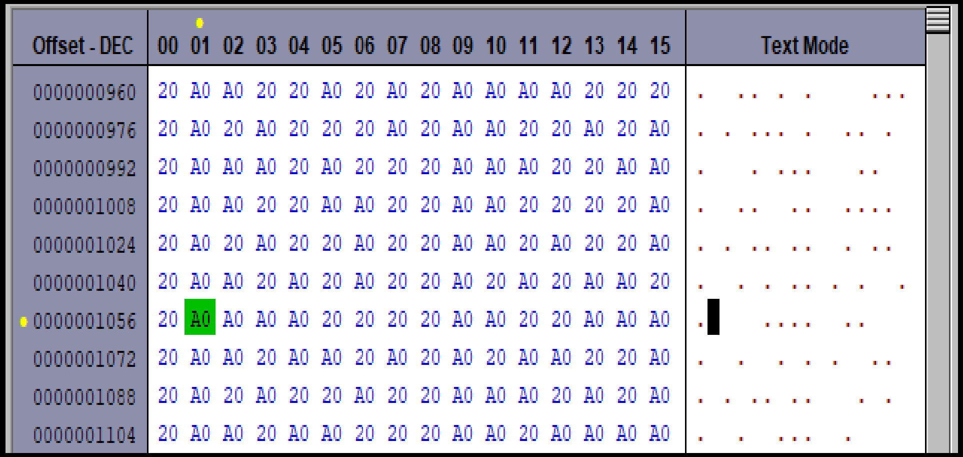
Offset
Hex: 12
Dec: 18

Page
1 of 1

Bytes per Page
320

File Size
245

After Data Injection



```
Offset - DEC  00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15   Text Mode
0000000960    20 A0 A0 20 20 A0 20 A0 20 A0 A0 A0 A0 20 20 20   .  .. . .    ...
0000000976    20 A0 20 A0 20 20 20 A0 20 A0 A0 20 20 A0 20 A0   . . ... .  .. . 
0000000992    20 A0 A0 A0 20 A0 20 20 20 A0 A0 A0 20 20 A0 A0   .   . ...   ..  
0000001008    20 A0 A0 20 20 A0 A0 20 20 A0 A0 20 20 20 20 A0   .  ..  ..  .... 
0000001024    20 A0 20 A0 20 20 A0 20 20 A0 A0 20 A0 20 20 A0   . . .. ..  . .. 
0000001040    20 A0 A0 20 A0 20 A0 20 20 A0 20 A0 20 A0 A0 20   .  . . .. . .  .
0000001056    20 A0 A0 A0 A0 20 20 20 20 A0 A0 20 20 A0 A0 A0   .    ....  ..   
0000001072    20 A0 A0 20 A0 A0 20 A0 20 A0 20 A0 A0 20 20 A0   .  .  . . .  .. 
0000001088    20 A0 20 A0 20 20 A0 20 20 A0 A0 A0 20 A0 20 A0   . . .. ..   . . 
0000001104    20 A0 A0 20 A0 A0 20 20 20 A0 A0 20 A0 A0 A0 A0   .  .  ...  .    
```

Before Data Injection the real data is **visible** in the Hexadecimal Utility Software.

After Data Injection the real data is **invisible** in the Hexadecimal Utility Software.

This type of data security is used to send top confidential information and classified documents unnoticed and unsuspicious due to its data hiddenness. Perfect file exchange communication.

Steganography

Keep your Stories and Memories inside your picture

Upload your Picture on Dropbox with Classified Documents

The decent way in sending confidential documents thru the email.

Encryption is suspicious

Upload images to any Cloud Storage with your confidential documents

The File injected in the image is secure and safe from Virus infection

Impossible to retrieve the injected information from the image Pixel's Noise

SECURE DATA ACCESS CONCEPT

Please always remember that any Cipher that is visible to the eyes of the attacker is vulnerable to attack and the vulnerabilities can be exploited at any point in time, just time matters.

SDC invincible cipher

Steganography

If encryption is so unbreakable,
Why do businesses and governments
keep getting hacked?

thank you