Package: hightR (via r-universe)

September 7, 2024

Title HIGHT Algorithm

Version 0.3.0

Author Yongwoo Kim [aut, cre]

Maintainer Yongwoo Kim <yw_kim@outlook.com>

Description HIGHT(HIGh security and light weigHT) algorithm is a block cipher encryption algorithm developed to provide confidentiality in computing environments that demand low power consumption and lightweight, such as RFID(Radio-Frequency Identification) and USN(Ubiquitous Sensor Network), or in mobile environments that require low power consumption and lightweight, such as smartphones and smart cards. Additionally, it is designed with a simple structure that enables it to be used with basic arithmetic operations, XOR, and circular shifts in 8-bit units. This algorithm was designed to consider both safety and efficiency in a very simple structure suitable for limited environments, compared to the former 128-bit encryption algorithm SEED. In December 2010, it became an ISO(International Organization for Standardization) standard. The detailed procedure is described in Hong et al. (2006) <doi:10.1007/11894063_4>.

License GPL-3

Encoding UTF-8

Imports stats

URL https://github.com/Yongwoo-Eg-Kim/hightR

BugReports https://github.com/Yongwoo-Eg-Kim/hightR/issues

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Repository https://yongwoo-eg-kim.r-universe.dev

RemoteUrl https://github.com/yongwoo-eg-kim/hightr

RemoteRef HEAD

RemoteSha 84eb2291d5e0398bd0f963fdd54c3e5ef2d61383

Contents

`hight_dec` *Perform decryption using HIGHT.*

Description

HIGHT (HIGh security and light weigHT) is a symmetric key block cipher algorithm designed for use in resource-constrained environments such as embedded systems and wireless sensor networks. Outputs 64-bit ciphertext from 128-bit master key and 64-bit plaintext. This process can be repeated multiple times.

Usage

```
hight_dec(C, MK, mode, IV = NULL, output = "int")
```

Arguments

`C`	Encrypted plaintext by HIGHT.
`MK`	Master Key. This is used to encrypt other keys that are used to encrypt and decrypt data. This should be typically kept secret and is only accessible to authorized users who need to use it for encryption and decryption operations. It should have a length of 16 and must have a value from 0 to 255.
`mode`	Please select one from 'ecb'(Electric CodeBook mode),'cfb'(Cipher FeedBack mode),'cbc'(Cipher Block Chaining mode),'ofb'(Output FeedBack mode).
`IV`	Initialization Vector. The IV is usually generated randomly and is different for each encryption operation. It is combined with the encryption key to produce a unique key for each encryption operation. Its length must be equal to 8, which is a unit of cryptographic block, and the value range must also have a value from 0 to 255. This parameter will be ignored in ECB mode.
`output`	Support 'hex'(e.g. '0x66') string or 'int'(e.g. 102) for output format.

Value

Returns a numeric vector decrypted by the HIGHT algorithm.

References

Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B. S., ... & Chee, S. (2006). HIGHT: A new block cipher suitable for low-resource device. In Cryptographic Hardware and Embedded Systems-CHES 2006: 8th International Workshop, Yokohama, Japan, October 10-13, 2006. Proceedings 8 (pp. 46-59). Springer Berlin Heidelberg.

Examples

```
# Encryption and Decryption (CBC mode)
MK = c(0x88, 0xE3, 0x4F, 0x8F, 0x08, 0x17, 0x79, 0xF1,
       0xE9, 0xF3, 0x94, 0x37, 0x0A, 0xD4, 0x05, 0x89)
IV = c(0x26, 0x8D, 0x66, 0xA7, 0x35, 0xA8, 0x1A, 0x81)
P = c(0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07)
C = hight_enc(P,MK,mode = 'CBC', IV, output='int')
hight_dec (C,MK , mode = 'CBC', IV, output = 'int')

# Check decrypted text is same with the Plaintext
hight_dec (C, MK , mode = 'CBC', IV, output = 'int')==P
```

---

hight_enc *Perform encryption using HIGHT.*

Description

HIGHT (HIGh security and light weigHT) is a symmetric key block cipher algorithm designed for use in resource-constrained environments such as embedded systems and wireless sensor networks. Outputs 64-bit ciphertext from 128-bit master key and 64-bit plaintext. This process can be repeated multiple times.

Usage

```
hight_enc(P, MK, mode, IV = NULL, output = "int")
```

Arguments

P	Plaintext. Its length must be a multiple of 8 and must have a value from 0 to 255.
MK	Master Key. This is used to encrypt other keys that are used to encrypt and decrypt data. This should be typically kept secret and is only accessible to authorized users who need to use it for encryption and decryption operations. It should have a length of 16 and must have a value from 0 to 255.
mode	Please select one from 'ECB'(Electric CodeBook mode),'CFB'(Cipher Feed-Back mode),'CBC'(Cipher Block Chaining mode),'OFB'(Output FeedBack mode).
IV	Initialization Vector. The IV is usually generated randomly and is different for each encryption operation. It is combined with the encryption key to produce a unique key for each encryption operation. Its length must be equal to 8, which is a unit of cryptographic block, and the value range must also have a value from 0 to 255. This parameter will be ignored in ECB mode.
output	Support 'hex'(e.g. '0x66') string or 'int'(e.g. 102) for output format.

Value

Returns a numeric vector encrypted by the HIGHT algorithm.

References

Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B. S., ... & Chee, S. (2006). HIGHT: A new block cipher suitable for low-resource device. In Cryptographic Hardware and Embedded Systems-CHES 2006: 8th International Workshop, Yokohama, Japan, October 10-13, 2006. Proceedings 8 (pp. 46-59). Springer Berlin Heidelberg.

Examples

```
# Encryption (CBC mode)
MK = c(0x88, 0xE3, 0x4F, 0x8F, 0x08, 0x17, 0x79, 0xF1,
       0xE9, 0xF3, 0x94, 0x37, 0x0A, 0xD4, 0x05, 0x89)
IV = c(0x26, 0x8D, 0x66, 0xA7, 0x35, 0xA8, 0x1A, 0x81)
P = c(0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07)
C = hight_enc(P, MK, mode = 'CBC', IV, output='int')
print(C)

# Encryption (ECB mode)
MK = c(0x88, 0xE3, 0x4F, 0x8F, 0x08, 0x17, 0x79, 0xF1,
       0xE9, 0xF3, 0x94, 0x37, 0x0A, 0xD4, 0x05, 0x89)
# IV is not used in ECB mode.
P = c(0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07)
C = hight_enc(P, MK, mode = 'ECB',  output='int')
print(C)
```

Index