

Network Working Group  
Request for Comments: 4196  
Category: Standards Track

H.J. Lee  
J.H. Yoon  
S.L. Lee  
J.I. Lee  
KISA  
October 2005

## The SEED Cipher Algorithm and Its Use with IPsec

### Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

### Copyright Notice

Copyright (C) The Internet Society (2005).

### Abstract

This document describes the use of the SEED block cipher algorithm in the Cipher Block Chaining Mode, with an explicit IV, as a confidentiality mechanism within the context of the IPsec Encapsulating Security Payload (ESP).

## 1. Introduction

### 1.1. SEED

SEED is a national industrial association standard [TTASSEED] and is widely used in South Korea for electronic commerce and financial services that are operated on wired and wireless communications.

SEED is a 128-bit symmetric key block cipher that has been developed by KISA (Korea Information Security Agency) and a group of experts since 1998. The input/output block size of SEED is 128-bit and the key length is also 128-bit. SEED has the 16-round Feistel structure. A 128-bit input is divided into two 64-bit blocks, and the right 64-bit block is an input to the round function with a 64-bit subkey that is generated from the key scheduling.

SEED is easily implemented in various software and hardware, and it can be effectively adopted to a computing environment with restricted resources, such as mobile devices and smart cards.

SEED is robust against known attacks including DC (Differential cryptanalysis), LC (Linear cryptanalysis), and related key attacks. SEED has gone through wide public scrutinizing procedures. It has been evaluated and is considered cryptographically secure by credible organizations such as ISO/IEC JTC 1/SC 27 and Japan CRYPTREC (Cryptography Research and Evaluation Committees)[ISOSEED][CRYPTREC].

The remainder of this document specifies the use of SEED within the context of IPsec ESP. For further information on how the various pieces of ESP fit together to provide security services, please refer to [ARCH], [ESP], and [ROAD].

## 1.2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document (in uppercase, as shown) are to be interpreted as described in RFC 2119 [KEYWORDS].

## 2. The SEED Cipher Algorithm

All symmetric block cipher algorithms share common characteristics and variables, including mode, key size, weak keys, block size, and rounds. The following sections contain descriptions of the relevant characteristics of SEED.

The algorithm specification and object identifiers are described in [ISOSEED] [SEED]. The SEED homepage, [http://www.kisa.or.kr/seed/seed\\_eng.html](http://www.kisa.or.kr/seed/seed_eng.html), contains a wealth of information about SEED, including a detailed specification, evaluation report, test vectors, and so on.

### 2.1. Mode

NIST has defined 5 modes of operation for the Advanced Encryption Standard (AES) [AES] and other FIPS-approved ciphers [MODES]: CBC (Cipher Block Chaining), ECB (Electronic Codebook), CFB (Cipher FeedBack), OFB (Output FeedBack), and CTR (Counter). The CBC mode is well-defined and well-understood for symmetric ciphers, and is currently required for all other ESP ciphers. This document specifies the use of the SEED cipher in the CBC mode within ESP. This mode requires an Initialization Vector (IV) that is the same size as the block size. Use of a randomly generated IV prevents generation of identical ciphertext from packets that have identical data that spans the first block of the cipher algorithm's block size

The IV is XOR'd with the first plaintext block before it is encrypted. Then for successive blocks, the previous ciphertext block is XOR'd with the current plaintext before it is encrypted.

More information on the CBC mode can be obtained in [MODES] [CRYPTO-S]. For use of the CBC mode in ESP with 64-bit ciphers, please see [CBC].

## 2.2. Key Size and Numbers of Rounds

SEED supports 128-bit key and has the 16-round Feistel structure.

## 2.3. Weak Keys

At the time this document was written, there were no known weak keys for SEED.

## 2.4. Block Size and Padding

SEED uses a block size of 16 octets (128 bits).

Padding is required by SEED to maintain a 16-octet (128-bit) blocksize. Padding MUST be added, as specified in [ESP], such that the data to be encrypted (which includes the ESP Pad Length and Next Header fields) has a length that is a multiple of 16 octets.

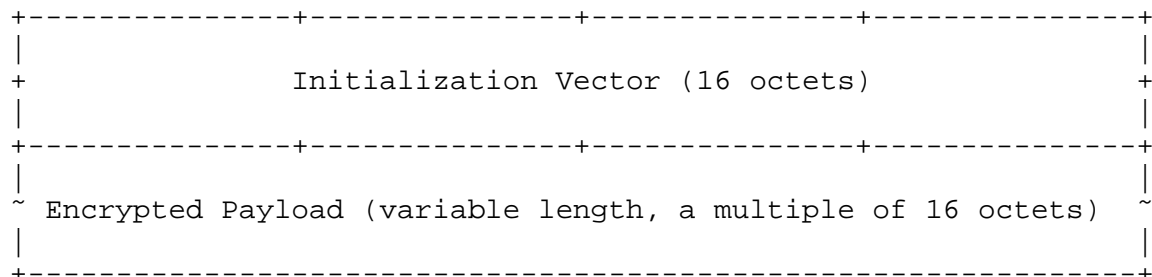
Because of the algorithm specific padding requirement, no additional padding is required to ensure that the ciphertext terminates on a 4-octet boundary (i.e., maintaining a 16-octet blocksize guarantees that the ESP Pad Length and Next Header fields will be right aligned within a 4-octet word). Additional padding MAY be included, as specified in [ESP], as long as the 16-octet blocksize is maintained.

## 2.5. Performance

Performance figures of SEED are available at [http://www.kisa.or.kr/seed/seed\\_eng.html](http://www.kisa.or.kr/seed/seed_eng.html)

### 3. ESP Payload

The ESP Payload is made up of the Initialization Vector(IV) of 16 octets followed by the encrypted payload. Thus, the payload field, as defined in [ESP], is broken down according to the following diagram:



The IV field MUST be the same size as the block size of the cipher algorithm being used. The IV MUST be chosen at random and MUST be unpredictable.

Including the IV in each datagram ensures that decryption of each received datagram can be performed, even when some datagrams are dropped or re-ordered in transit.

To avoid CBC encryption of very similar plaintext blocks in different packets, implementations MUST NOT use a counter or other low-hamming distance source for IVs.

### 4. Test Vectors

The first 2 test cases test SEED-CBC encryption. Each test case includes key, the plaintext, and the resulting ciphertext. All data are hexadecimal numbers (not prefixed by "0x").

The last 4 test cases illustrate sample ESP packets using SEED-CBC for encryption. All data are hexadecimal numbers (not prefixed by "0x").

Case #1	: Encrypting 32 bytes (2 blocks) using SEED-CBC with 128-bit key			
Key	: ed2401ad	22fa2559	91bafdb0	1fef697
IV	: 93eb149f	92c9905b	ae5cd34d	a06c3c8e
PlainText	: b40d7003	d9b6904b	35622750	c91a2457
	5bb9a632	364aa26e	3ac0cf3a	9c9d0dcb
CipherText	: f072c5b1	a0588c10	5af8301a	dcd91dd0
	67f68221	55304bf3	aad75ceb	44341c25

Case #2 : Encrypting 64 bytes (4 blocks) using SEED-CBC with  
128-bit key

Key	: 88e34f8f	081779f1	e9f39437	0ad40589
IV	: 268d66a7	35a81a81	6fbad9fa	36162501
PlainText	: d76d0d18	327ec562	b15e6bc3	65ac0c0f
	8d41e0bb	938568ae	ebfd92ed	1affa096
	394d20fc	5277ddfc	4de8b0fc	e1eb2b93
	d4ae40ef	4768c613	b50b8942	f7d4b9b3
CipherText	: a293eae9	d9aebfac	37ba714b	d774e427
	e8b706d7	e7d9a097	228639e0	b62b3b34
	ced11609	cef2abaa	ec2edf97	9308f379
	c31527a8	267783e5	cba35389	82b48d06

Case #3 : Sample transport-mode ESP packet (ping 192.168.123.100)

Key	: 90d382b4	10eeba7a	d938c46c	ec1a82bf
SPI	: 4321			
Source address	: 192.168.123.3			
Destination address	: 192.168.123.100			
Sequence number	: 1			
IV	: e96e8c08	ab465763	fd098d45	dd3ff893

Original packet :

IP header (20 bytes) : 45000054 08f20000 4001f9fe c0a87b03 c0a87b64

Data (64 bytes) :

08000ebd	a70a0000	8e9c083d	b95b0700
08090a0b	0c0d0e0f	10111213	14151617
18191a1b	1c1d1e1f	20212223	24252627
28292a2b	2c2d2e2f	30313233	34353637

Augment data with :

Padding : 01020304 05060708 090a0b0c 0d0e

Pad length : 0e

Next header : 01 (ICMP)

Pre-encryption Data with padding, pad length and next header(80 bytes):

08000ebd	a70a0000	8e9c083d	b95b0700
08090a0b	0c0d0e0f	10111213	14151617
18191a1b	1c1d1e1f	20212223	24252627
28292a2b	2c2d2e2f	30313233	34353637
01020304	05060708	090a0b0c	0d0e0e01

Post-encryption packet with SPI, Sequence number, IV :  
 IP Header : 45000054 08f20000 4001f9fe c0a87b03 c0a87b64  
 SPI/Seq # : 00004321 00000001  
 IV : e96e8c08 ab465763 fd098d45 dd3ff893  
 Encrypted Data (80 bytes) :  
 e7ebaa03 cf45ef09 021b3011 b40d3769  
 be96ebae cd4222f6 b6f84ce5 b2d5cdd1  
 60eb6b0e 5a47d16a 501a4d10 7b2d7cc8  
 ab86ba03 9a000972 66374fa8 f87ee0fb  
 ef3805db faa144a2 334a34db 0b0f81ca

Case #4 : Sample transport-mode ESP packet  
 (ping -p 77 -s 20 192.168.123.100)  
 Key : 90d382b4 10eeba7a d938c46c ec1a82bf  
 SPI : 4321  
 Source address : 192.168.123.3  
 Destination address : 192.168.123.100  
 Sequence number : 8  
 IV : 69d08df7 d203329d b093fc49 24e5bd80

Original packet:  
 IP header (20 bytes) : 45000030 08fe0000 4001fa16 c0a87b03 c0a87b64  
 Data (28 bytes) :  
 0800b5e8 a80a0500 a69c083d 0b660e00 77777777 77777777 77777777

Augment data with :  
 Padding : 0102  
 Pad length : 02  
 Next header : 01 (ICMP)

Pre-encryption Data with padding, pad length and  
 next header(32 bytes):  
 0800b5e8 a80a0500 a69c083d 0b660e00  
 77777777 77777777 77777777 01020201

Post-encryption packet with SPI, Sequence number, IV :  
 IP header : 4500004c 08fe0000 4032f9c9 c0a87b03 c0a87b64  
 SPI/Seq # : 00004321 00000008  
 IV : 69d08df7 d203329d b093fc49 24e5bd80  
 Encrypted Data (32 bytes) :  
 b9ad6e19 e9a6a2fa 02569160 2c0af541  
 db0b0807 e1f660c7 3ae2700b 5bb5efd1

Case #5 : Sample tunnel-mode ESP packet (ping 192.168.123.200)  
 Key : 01234567 89abcdef 01234567 89abcdef  
 SPI : 8765  
 Source address : 192.168.123.3  
 Destination address : 192.168.123.200  
 Sequence number : 2  
 IV : f4e76524 4f6407ad f13dc138 0f673f37

Original packet :  
 IP header (20 bytes) : 45000054 09040000 4001f988 c0a87b03 c0a87bc8  
 Data (64 bytes) :  
 08009f76 a90a0100 b49c083d 02a20400  
 08090a0b 0c0d0e0f 10111213 14151617  
 18191a1b 1c1d1e1f 20212223 24252627  
 28292a2b 2c2d2e2f 30313233 34353637

Augment data with :  
 Padding : 01020304 05060708 090a  
 Pad length : 0a  
 Next header : 04 (IP-in-IP)

Pre-encryption Data with original IP header, padding, pad length and next header (96 bytes) :  
 45000054 09040000 4001f988 c0a87b03  
 c0a87bc8 08009f76 a90a0100 b49c083d  
 02a20400 08090a0b 0c0d0e0f 10111213  
 14151617 18191a1b 1c1d1e1f 20212223  
 24252627 28292a2b 2c2d2e2f 30313233  
 34353637 01020304 05060708 090a0a04

Post-encryption packet with SPI, Sequence number, IV :  
 IP header : 4500008c 09050000 4032f91e c0a87b03 c0a87bc8  
 SPI/Seq # : 00008765 00000002  
 IV : f4e76524 4f6407ad f13dc138 0f673f37  
 Encrypted Data (96 bytes):  
 2638aa7b 05e71b54 9348082b 67b47b26  
 c565aed4 737f0bcb 439c0f00 73e7913c  
 3c8a3e4f 5f7a5062 003b78ed 7ca54a08  
 c7ce047d 5bec14e4 8cba1005 32a12097  
 8d7f5503 204ef661 729b4ea1 ae6a9178  
 59a5caac 46e810bd 7875bd13 d6f57b3d

Case #6 : Sample tunnel-mode ESP packet  
 (ping -p ff -s 40 192.168.123.200)  
 Key : 01234567 89abcdef 01234567 89abcdef  
 SPI : 8765  
 Source address : 192.168.123.3  
 Destination address : 192.168.123.200  
 Sequence number : 5  
 IV : 85d47224 b5f3dd5d 2101d4ea 8dffab22

Original packet :  
 IP header (20 bytes) :  
 45000044 090c0000 4001f990 c0a87b03 c0a87bc8  
 Data (48 bytes) :  
 0800d63c aa0a0200 c69c083d a3de0300  
 ffffffff ffffffff ffffffff ffffffff  
 ffffffff ffffffff ffffffff ffffffff

Augment data with :  
 Padding : 01020304 05060708 090a  
 Pad length : 0a  
 Next header : 04 (IP-in-IP)

Pre-encryption Data with original IP header, padding, pad length and next header (80 bytes):  
 45000044 090c0000 4001f990 c0a87b03  
 c0a87bc8 0800d63c aa0a0200 c69c083d  
 a3de0300 ffffffff ffffffff ffffffff  
 ffffffff ffffffff ffffffff ffffffff  
 ffffffff 01020304 05060708 090a0a04

Post-encryption packet with SPI, Sequence number, IV :  
 IP header : 4500007c 090d0000 4032f926 c0a87b03 c0a87bc8  
 SPI/Seq # : 00008765 00000005  
 IV : 85d47224 b5f3dd5d 2101d4ea 8dffab22  
 Encrypted Data (80 bytes) :  
 311168e0 bc36ac4e 59802bd5 192c5734  
 8f3d29c8 90bab276 e9db4702 91f79ac7  
 79571929 c170f902 ffb2f08b d448f782  
 31671414 ff29b7e0 168e1c87 09ba2b67  
 a56e0fbc 4ff6a936 d859ed57 6c16ef1b



## 5. Interaction with IKE

This section describes the use of IKE [IKE] to establish IPsec ESP security associations (SAs) that employ SEED in CBC mode.

### 5.1. Phase 1 Identifier

For Phase 1 negotiations, the object identifier of SEED-CBC is defined in [SEED].

```
algorithm OBJECT IDENTIFIER ::= { iso(1) member-body(2) korea(410)
kisa(200004) algorithm(1) }
```

```
id-seedCBC OBJECT IDENTIFIER ::= { algorithm seedCBC(4) }
```

### 5.2. Phase 2 Identifier

For Phase 2 negotiations, IANA has assigned an ESP Transform Identifier of (21) for ESP\_SEED\_CBC.

### 5.3. Key Length Attribute

Since the SEED supports 128-bit key lengths, the Key Length attribute is set with 128 bits.

### 5.4. Hash Algorithm Considerations

HMAC-SHA-1 [HMAC-SHA] and HMAC-MD5 [HMAC-MD5] are currently considered of sufficient strength to serve both as IKE generators of 128-bit SEED keys and as ESP authenticators for SEED encryption using 128-bit keys.

## 6. Security Considerations

No security problem has been found on SEED. SEED is secure against all known attacks including Differential cryptanalysis, Linear cryptanalysis, and related key attacks. The best known attack is only an exhaustive search for the key (by [CRYPTREC]). For further security considerations, the reader is encouraged to read [CRYPTREC], [ISOSEED], and [SEED-EVAL].

## 7. IANA Considerations

IANA has assigned ESP Transform Identifier (21) to ESP\_SEED\_CBC.

## 8. Acknowledgments

The authors want to thank Ph.D Haesuk Kim of Future Systems Inc. and Brian Kim of OULLIM Information Technology Inc. for providing expert advice on Test Vector examples.

## 9. References

### 9.1. Normative References

- [CBC] Pereira, R. and R. Adams, "The ESP CBC-Mode Cipher Algorithms", RFC 2451, November 1998.
- [ESP] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.
- [IKE] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.
- [KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [SEED] Park, J., Lee, S., Kim, J., and J. Lee, "The SEED Encryption Algorithm", RFC 4009, February 2005.
- [TTASSEED] Telecommunications Technology Association (TTA), South Korea, "128-bit Symmetric Block Cipher (SEED)", TTAS.KO-12.0004, September, 1998 (In Korean)  
<http://www.tta.or.kr/English/new/main/index.htm>

### 9.2. Informative Reference

- [AES] NIST, FIPS PUB 197, "Advanced Encryption Standard(AES)", November 2001.  
<http://csrc.nist.gov/publications/fips/fips197/fips-197.ps>,pdf}
- [ARCH] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.
- [CRYPTO-S] Schneier, B., "Applied Cryptography Second Edition", John Wiley & Sons, New York, NY, 1995, ISBN 0-471-12845-7.
- [CRYPTREC] Information-technology Promotion Agency (IPA), Japan, CRYPTREC. "SEED Evaluation Report", February, 2002  
[http://www.kisa.or.kr/seed/seed\\_eng.html](http://www.kisa.or.kr/seed/seed_eng.html)

- [HMAC-MD5] Madson, C. and R. Glenn, "The Use of HMAC-MD5-96 within ESP and AH", RFC 2403, November 1998.
- [HMAC-SHA] Madson, C. and R. Glenn, "The Use of HMAC-SHA-1-96 within ESP and AH", RFC 2404, November 1998.
- [ISOSEED] ISO/IEC JTC 1/SC 27 N3979, "IT Security techniques - Encryption Algorithms - Part3 : Block ciphers", June 2004.
- [MODES] Symmetric Key Block Cipher Modes of Operation, <http://www.nist.gov/modes/>.
- [ROAD] Thayer, R., N. Doraswamy and R. Glenn, "IP Security Document Roadmap", RFC 2411, November 1998.
- [SEED-EVAL] KISA, "Self Evaluation Report", [http://www.kisa.or.kr/seed/data/Document\\_pdf/SEED\\_Self\\_Evaluation.pdf](http://www.kisa.or.kr/seed/data/Document_pdf/SEED_Self_Evaluation.pdf)

#### Authors' Address

Hyangjin Lee  
Korea Information Security Agency  
Phone: +82-2-405-5446  
Fax : +82-2-405-5319  
EMail : jiinii@kisa.or.kr

Jaeho Yoon  
Korea Information Security Agency  
Phone: +82-2-405-5434  
Fax : +82-2-405-5219  
EMail : jhyoon@kisa.or.kr

Seoklae Lee  
Korea Information Security Agency  
Phone: +82-2-405-5230  
Fax : +82-2-405-5219  
EMail : sllee@kisa.or.kr

Jaeil Lee  
Korea Information Security Agency  
Phone: +82-2-405-5200  
Fax : +82-2-405-5219  
EMail: jilee@kisa.or.kr

## Full Copyright Statement

Copyright (C) The Internet Society (2005).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

