Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.25, NO.2, Apr. 2015 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2015.25.2.271

패딩 오라클 공격에 따른 다양한 패딩방법의 안전성 분석

김 기 문,[1†] 박 명 서,[2] 김 종 성,[2] 이 창 훈,[3] 문 덕 재,[4] 홍 석 희[4‡]
[1]한국인터넷진흥원, [2]국민대학교, [3]서울과학기술대학교, [4]고려대학교

Safety Analysis of Various Padding Techniques on Padding Oracle Attack

Kim-Kimoon,[1†] Park-Myungseo,[2] Kim-Jongsung,[2] Lee-Changhoon,[3]
Moon-Dukjae,[4] Hong-Seokhee[4‡]
[1]Korea Internet & Security Agency, [2]Kookmin University,
[3]Seoul national university of science and technology, [4]Korea University

요 약

인터넷 뱅킹이나 전자상거래 같은 응용 환경에서 개인정보 및 민감한 정보를 보호하기 위해서 다양한 암호 알고리즘들을 사용한다. 하지만 안전성이 검증된 암호 알고리즘을 사용하여 중요 정보를 암호화 하더라도 운영모드, 패딩방법 등 암호화를 적용하는 방법이 올바르지 못하면 암호화된 중요 정보들이 노출 된다는 연구결과와 방법들이 소개되고 있다. 이러한 공격방법 중 대표적인 사례가 패딩 오라클 공격(Padding Oracle Attack)이다. 본 논문에서는 블록암호의 CBC(Cipher Block Chainning) 운영모드에 적용 가능한 12가지 패딩방법에 대하여 패딩오라클 공격의 가능성을 분석하였다. 그 결과, 3가지의 안전한 패딩방법과 9가지의 안전하지 않은 패딩방법으로 분류할 수 있다. 3가지의 안전한 패딩방법 분석을 통해 패딩 오라클 공격에 내성을 가질 수 있도록 안전한 패딩방법 설계 시 고려해야 할 5가지 사항에 대하여 제안하고자 한다.

ABSTRACT

We use various types of cryptographic algorithms for the protection of personal and sensitive informations in the application environments, such as an internet banking and an electronic commerce. However, recent researches were introduced that if we implement modes of operation, padding method and other cryptographic implementations in a wrong way, then the critical information can be leaked even though the underlying cryptographic algorithms are secure. Among these attacking techniques, the padding oracle attack is representative. In this paper, we analyze the possibility of padding oracle attacks of 12 kinds of padding techniques that can be applied to the CBC operation mode of a block cipher. As a result, we discovered that 3 kinds were safe padding techniques and 9 kinds were unsafe padding techniques. We propose 5 considerations when designing a safe padding techniques to have a resistance to the padding oracle attack through the analysis of three kinds of safe padding techniques.

Keywords: Padding Oracle Attack, CBC Mode, Block Cipher

I. 서 론

최근 12년 6월 블록암호의 패딩 규칙을 이용하여 RSA社 인증제품의 암호키를 13분 만에 찾는 패딩 오라클 공격이 소개되었다.[1] 패딩 하는 바이트의 길이 정보가 포함되는 패딩 방법이라면 패딩 오라클 공격에 취약 할 수 있다.

패딩 오라클 공격은 블록암호의 CBC(Cipher Bl

접수일(2015년 1월 23일), 수정일(2015년 3월 3일),
게재확정일(2015년 3월 3일)

† 주저자. harrykim@kisa.or.kr
‡ 교신저자. shhong@korea.ac.kr(Corresponding author)

ock Chainning) 운영모드 뿐만 아니라, 스트림 모드인 CFB 운영모드에서 가능하지만, 본 논문에서는 블록암호의 CBC 운영모드의 패딩기법에 대한 안전성 분석만 다룬다. 패딩 오라클 공격은 암호 알고리즘을 이용하여 메시지를 암호화할 경우, 입력 크기에 맞추기 위해 메시지의 마지막에 적당한 값을 덧붙인다. 만약 공격자에게 메시지의 패딩이 옳은지 아닌지의 여부를 판단하는 오라클이 있다면, 이 오라클을 이용하여 임의의 암호문에 대응하는 메시지를 알아낼 수 있다. [Fig. 1]에서처럼 오라클은 공격자가 질문한 암호문을 복호화하여 얻은 평문의 패딩이 옳은지 아닌지를 판단하여 VALID 또는 INVALID 값을 공격자에게 대답한다. 공격자는 암호화된 암호문을 중간에서 가로챌 수 있고, 이 과정에서 획득된 정보들을 통하여 올바른 평문을 알아낼 수 있다.

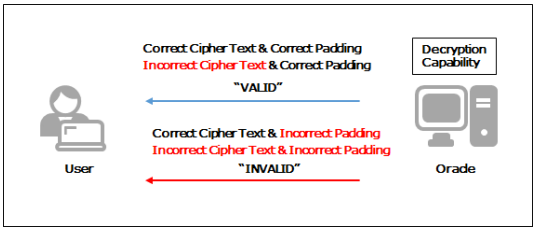

Fig. 1. Padding Oracle Attack

II. 패딩 오라클 공격 및 관련 연구

본 절에서는 패딩 오라클 공격에 대한 자세한 공격 방법 및 지금까지 발표된 다양한 패딩 오라클 공격 사례들에 대해 소개한다. 운영모드에 대한 패딩 오라클 공격은 EUROCRYPT 2002에 Vaudenay에 의해 처음으로 소개되었다. Vaudenay는 다양한 응용 환경(SSL/TLS, IPSec, WTLS 등)에 사용되는 CBC 운영 모드에 대한 패딩 오라클 공격을 발표하였다.[2] CBC 운영모드는 동일한 평문 블록과 암호문 블록 쌍이 발생하지 않도록 전 단계의 암·복호화 결과가 현 단계에 영향을 주는 모드이다. CBC 운영 모드에 대한 패딩 오라클 공격은 CBC 복호화 모드[Fig. 2]에서 적용 되며, 이때의 패딩 방법은 CBC-PAD를 가정한다.

CBC-PAD 방법은 PKCS #7에서 명시하고 있는 패딩 방법이다.[4] 공격자는 초기벡터(IV) 또는 암호문(Ciphertext)을 변경하여 서버에 보낸다. [Fig. 3]에서처럼 서버는 공격자가 보낸 암호문을 복호화하

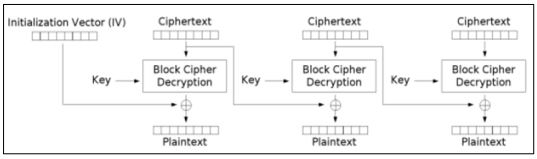

Fig. 2. Decryption Processing on CBC Operation Mode

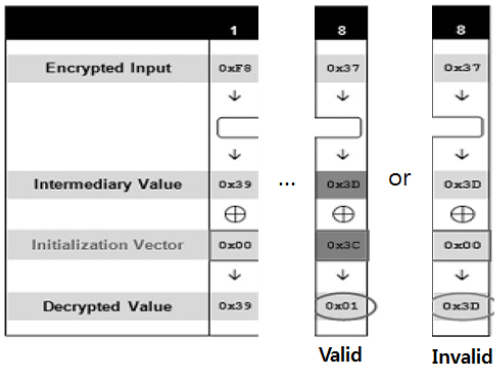

Fig. 3. Server sends to Hacker about VALID or INVALID message.

여 패딩을 체크해 보고, 패딩에 대한 응답(VALID 또는 INVALID)을 공격자에게 보낸다.

이를 이용하여, 공격자는 서버가 VALID 응답을 보낼 때까지 초기벡터 또는 암호문을 변경[Fig. 4]하여 보낸다. 공격자가 변경한 값 중, 서버가 VALID 응답을 보냈을 때의 값(Initialization Vector)을 이용하여 중간 값(Intermediary Value)을 구할 수 있다. 이렇게 구한 중간 값과 공격자가 변경하기 전의 초기벡터 또는 암호문을 XOR하면 평문을 복구할 수 있다.

이 공격은 Black, Urtubia에 의해 다양한 패딩 방법을 사용하는 운영 모드에 대해서 일반화 되었다.

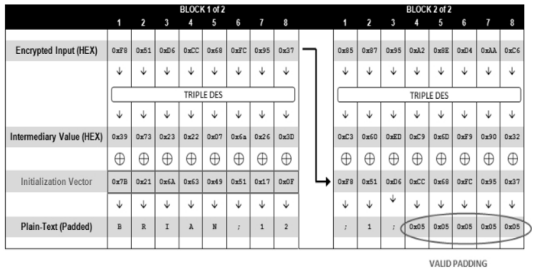

Fig 4. For example, Finding the last byte of intermediate value by changing the last byte of initialization vector.

[3] 또한, Paterson, Yau와 Klima, Rosa 등은 패딩 오라클 공격이 가능한 새로운 환경을 제시하였다. [4][5]

2010년 Juliano Rizzo와 Thai Duong은 Practical Padding Oracle Attacks이라는 제목으로 논문을 발표하였다[5]. 이 논문에서는 실제 환경에서 잠재적인 패딩 오라클 O를 찾는 방법 및 존재에 대해 확인하는 방법을 소개하였다. 또한 CAPTCHA, JSF view states 복호화 등 실제 서비스 되고 있는 환경에서 패딩 오라클 공격에 취약한 시스템에 공격을 수행하여 평문을 찾는 시나리오에 대해 소개하였다. [7] 마지막으로 지금까지 암호문을 복호화하기 위해 사용되었던 패딩 오라클 O를 공격자가 원하는 평문을 암호화하기 위해 사용하는 CBC-R 암호화라는 방식의 평문 암호화를 소개하였다. CBC-R 암호화는 Key를 모르는 상태에서 공격자 원하는 평문을 암호화하는 방법으로 기존 패딩 오라클 공격에 대해 진보된 방법이다.

2013년 5월 IEEE S&P에서 영국의 RHUL(Royal Holloway University of London)의 N. Alfardan과 K. Paterson은 블록 암호 알고리즘을 이용한 CBC 운영 모드에 대한 패딩 오라클 공격법을 이용하여 TLS(Transport Layer Security) 프로토콜을 통해 암호화 통신되는 데이터에서 평문정보를 획득할 수 있다는 결과를 소개하였다.[8] 이 공격법은 기존의 패딩 오라클 공격의 방어법이 적용된 암호화 방식에 대한 공격법으로 Lucky Thirteen attack이라고 불리고 있다. TLS 프로토콜은 웹, 전자메일, SMS 및 VoIP 등 다양한 환경에서 적용되고 있으므로 이 공격의 파급효과가 매우 커지고 있다.

III. 다양한 패딩 방법의 오라클 가능성 분석

본 절에서는 본 논문의 분석 대상인 12가지 패딩 방법에 대한 설명 및 오라클 공격 가능성을 분석하여 소개 한다.

Table 1. Byte-based and Bit-based padding

Byte-based(8)		Bit-based(4)
CBC-PAD, ESP-PAD, XY-PAD, BOZ-PAD,	PAIR-PAD, ABYT-PAD, NIST-PAD, SSH2-PAD	OZ-PAD, ABIT-PAD, ISO(9797-1)방법3, ISO(10118-1)방법3

IV. 다양한 패딩 방법

패딩 방법은 처리단위에 따라 바이트와 비트로 나뉜다. 아래〈Table 1〉와 같이 바이트 및 비트 기반의 패딩 방법에 대하여 분류하고 각각 패딩 오라클 공격에 대한 내성을 분석하도록 한다.

4.1 CBC-PAD

이 패딩 방법은 바이트 단위로 적용이 된다. 평문 데이터의 크기가 n바이트 이며, n을 알고리즘의 블록 사이즈로 나눈 나머지를 m이라 하자. m이 0이 아닐 경우, n이 해당 블록사이즈의 정수배가 되도록 평문 데이터의 끝에 필요한 바이트 수를 넣어 패딩 한다. m이 0인 경우에는 패딩 방법이 사용됨을 표기하기 위해 블록사이즈의 바이트수 '$0xXX\cdots XX$'을 추가한다.

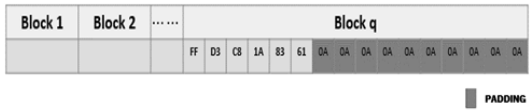

Fig. 5. Padding techniques of CBC-PAD

4.2 ESP-PAD

ESP-PAD는 바이트 기반의 패딩 방법이다. 패딩 바이트의 수에 따라 패딩 배턴이 결정된다. 패딩 바이트가 n바이트라 가정한다면 패딩 패턴은 $0x0102\cdots n$이다. 각 패딩 바이트 수에 따른 패딩 패턴은 다음과 같다.

Fig. 6. Padding techniques of ESP-PAD

4.3 XY-PAD

XY-PAD는 바이트 기반의 패딩 방법이다. 이 패딩 방법은 우선 2 바이트의 서로 다른 X, Y를 결정한다. 패딩 바이트에 따라 맨 처음 패딩 바이트는 X, 그 이후의 패딩 바이트에는 Y를 붙여 패딩 패턴을 만들

어 주는 방식이다. 패딩 바이트가 n 바이트라 가정한 다면 패딩 패턴은 $0xXYY\cdots\ YY$이다. 각 패딩 바이트 수에 따른 패딩 패턴은 다음과 같다.

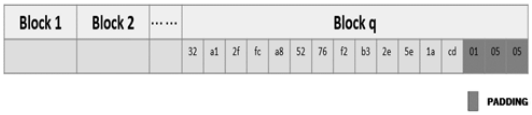

Fig. 7. Padding techniques of XY-PAD

4.4 OZ-PAD

OZ-PAD는 비트 기반의 패딩 방법이다. 이 패딩 방법은 맨 마지막 블록의 유효 데이터가 m 비트 일 때, $m+1$번째 비트를 '1'로 패딩 후 그 이후 비트는 모두 '0'으로 패딩해 주는 방식이다. 이 패딩 방법에 대해 패딩 제거할 때는 맨 마지막 블록의 뒤에서부터 '0'을 제거하며 마지막으로 '1'을 제거하는 방식이다.

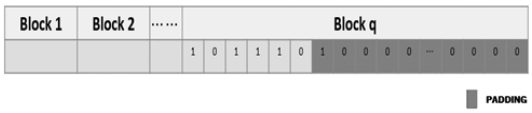

Fig. 8. Padding techniques of OZ-PAD

4.5 BOZ-PAD

BOZ-PAD는 바이트 기반의 패딩 방법으로 마지막 평문 데이터 72 비트에 56 비트 패딩을 한 그림이다. 평문 데이터를 128 비트로 만들기 위해서 평문 데이터 끝에 '1'을 패딩한 후 나머지 비트를 '0'으로 패딩한다.

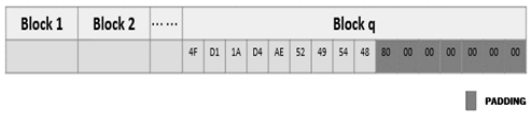

Fig. 9. Padding techniques of BOZ-PAD

4.6 PAIR-PAD

PAIR-PAD는 바이트 기반의 패딩 방법이다. 이 패딩방법은 XY-PAD의 약점을 피하기 위해 제안되었다. 송신자가 임의의 X, Y값을 결정하여 패딩하기 때문에 수신자는 X, Y에 대한 값을 알 수 없다. 패딩 방법은 다음과 같다.

Fig. 10. Padding techniques of PAIR-PAD

4.7 ABYT-PAD

ABYT-PAD는 바이트 기반의 패딩 방법이다. 이 패딩 방법은 PAIR-PAD와 유사한 방식을 사용한다. 패딩 방법은 다음과 같다. 송신자는 평문 데이터의 마지막 블록 M의 마지막 바이트 X를 확인하고, 송신자는 X와 다른 임의의 Y를 선택하여 패딩 한다. 이때 수신자는 X와 Y를 모른다.

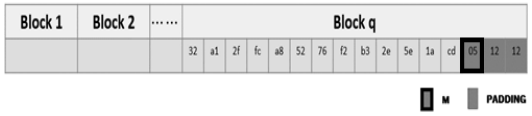

Fig. 11. Padding techniques of ABYT-PAD

4.8 ABIT-PAD

ABIT-PAD는 비트 기반의 패딩 방법이다. 이 패딩 방법은 ABYT-PAD와 동일하나 처리 단위가 비트 단위이다. 패딩 방법은 다음과 같다. 송신자는 평문 데이터의 마지막 블록 M의 마지막 비트 X를 확인하고, 송신자는 X와 반대인 값으로 패딩 한다.

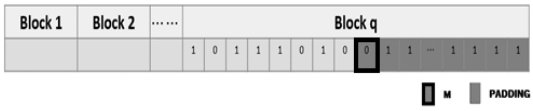

Fig. 12. Padding techniques of ABIT-PAD

4.9 NIST-PAD

NIST-PAD는 바이트 기반의 패딩 방법이다. 이 NIST-PAD는 BOZ-PAD와 유사한 패딩 방법이다. NIST-PAD는 첫 번째 패딩 바이트에 랜덤한 바이트

'R'을 선택해 채우고 나머지 패딩 바이트에 '$0x00$'을 채우는 방식이다. 이 패딩 방법의 제거 방법은 먼저 맨 마지막 블록의 최하위 바이트부터 '$0x00$'을 제거하고 마지막에 'R'을 제거하는 방식이다.

Fig. 13. Padding techniques of NIST-PAD

4.10 SSH2-PAD

SSH2-PAD는 바이트 기반의 패딩 방법이다. SSH2-PAD는 패딩할 맨 마지막 바이트에 패딩 길이 정보를 표시하고 나머지 패딩은 랜덤한 바이트를 선택하여 패딩 한다. SSH2-PAD는 유효 패딩 자리 수를 구분하기 위하여 유효 평문 데이터가 단위 블록의 배수일 때 한 블록을 추가하여 $0xr_1r_2 \cdots r_{15}10$와 같이 패딩 처리를 한다.

Fig. 14. Padding techniques of SSH2-PAD

4.11 ISO(9797-1) 방법 3

ISO(9797-1) 방법3은 비트 기반의 패딩 방법이다. 이 패딩 방법은 첫 번째 메시지 블록(L_M)에 비트 단위의 메시지 길이를 넣고, 마지막 블록에 '0'으로 패딩 하는 방법이다.

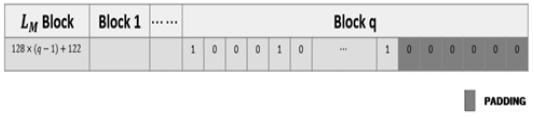

Fig. 15. Padding techniques of ISO(9797-1) Method 3

4.12 ISO(10118-1) 방법 3

ISO(10118-1) 방법3은 비트 기반의 패딩 방법이다. 이 패딩 방법은 유효 평문 데이터 P와 평문 데이터의 길이 정보를 담는 8 바이트의 L_M 사이에 비트 단위로 '$1000 \cdots 000$'을 패딩한다. 즉, 평문의 마지막 블록은 $M \| 1000 \cdots 000 \| L_M$와 같이 나타낼 수 있다.

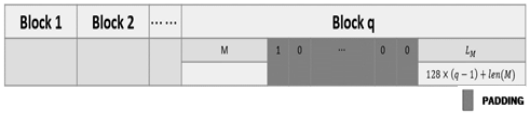

Fig. 16. Padding techniques of ISO(10118-1) Method 3

V. 오라클 공격 가능성

12가지의 패딩 방법에 대한 패딩 오라클 공격 가능성 검토 결과는 〈Table 2〉과 같다. 이때, 암호문은 q 블록으로 이루어져 있으며, 각 블록은 n 바이트 단위로 구성된 것으로 가정한다.

12가지 패딩 방법 중에서 PAIR-PAD, ABYT-PAD, ABIT-PAD 방법을 제외한 9가지 방법에 패딩 오라클 공격을 적용할 수 있음을 알 수 있다. 이중 NIST/SSH/OZ-PAD 방법은 적용은 가능하나 그 공격량이 전수조사량에 근접한 것이 이론적 분석 결과이다. 패딩 오라클 공격에 내성을 제공하는 PAIR-PAD, ABYT-PAD 및 ABIT-PAD 의 특성을 정리하면 〈Table 3〉와 같다.

Table 2. Analysis of possibility attack on 12 of different padding techniques.

Padding Techniques	Type	Attack	Complexity
CBC-PAD [2]	Byte	Yes	$128n + Log_2(n)$
ESP-PAD [9]	Byte	Yes	$128n + Log_2(n)$
XY-PAD [9]	Byte	Yes	$128n + Log_2(n)$
NIST-PAD	Byte	Yes	$2^{8n} - 1$
SSH-PAD	Byte	Yes	$2^{8n} - 2^8 + 2^4 + 1$
OZ-PAD [9]	Bit	Yes	$2^{8n} - 1$
BOZ-PAD [9]	Byte	Yes	$128n + Log_2(n)$
PAIR-PAD	Byte	No	-
ABYT-PAD	Byte	No	-
ABIT-PAD	Bit	No	-
ISO 9797-1 [5]	Bit	Yes	$8n(q-1) + 1 + Log_2(8n)$
ISO 10118-1 [5]	Bit	Yes	SB : $2^{r-1} + 2^{2r-n+1} + t$
			CB : $2^n + 2^{r-u}$

Table 3. Compare of safety padding techniques.

	PAIR-PAD	ABYT-PAD	ABIT-PAD
Type	Byte	Byte	Bit
Sender determines number of bits	2 bytes	1 byte	1 bit
Leakage the length of the messages	X	X	X
Using the Random Number Generator	O	O	X
Leakage the padding information	X	X	O
Detection of modified padding	X	X	X
Exists of Invalid padding	X	X	X

패딩 오라클 공격에 안전하다고 분석된 패딩 방법들은 주어진 메시지의 길이정보를 포함하고 있지 않으므로 길이정보 누출이 없다. 이 성질은 패딩 오라클 공격에 내성을 갖는 가장 중요한 성질이다.

하지만, PAIR-PAD와 ABYT-PAD의 경우 난수생성기를 필요로 한다. 패딩 추가 시 난수생성기를 사용하게 되면 효율성 측면에서 많은 부담이 있으므로 구현자 입장에서 고려해봐야 할 부분이다. 따라서 ABIT-PAD 의 경우와 같이 메시지의 마지막 비트 값이 결정되면 자동으로 그 보수 값을 패딩하는 방식이 선호된다. 하지만 이와 같은 경우 패딩 값을 통해 메시지의 일부 정보가 누출되므로 "정보 누출" 관점에서는 단점이 될 수 있다.

패딩 값 변조 검출여부는 공격자나 시스템 오류 등으로 인한 암호문 변조가 이루어진 경우, 주어진 패딩 방법의 체크를 통해 그 변조 여부를 확인할 수 있는지를 나타내는 것으로 위 세 가지 패딩 방식들은 암호문 변조에 대한 확인이 불가한 것으로 분석되었다. 이 성질은 POA 기법에 내성을 제공하는 주요 성질이다.

Invalid한 패딩 존재여부는 주어진 패딩 방식을 임의의 메시지에 적용할 경우, 절대 나타날 수 없는 패딩 값들의 존재 여부를 확인한 것으로 세 가지 패딩 방식에 이와 같은 패딩 값들이 없는 것으로 분석되었다.

VI. 결 론

패딩 오라클 공격은 반드시 블록기반 운영 모드에서만 적용되는 것은 아니다. 스트림 모드를 사용하는 서비스에서도 공격자가 패딩 길이나 메시지 길이 정보를 획득할 수 있는 경우 패딩 오라클 공격을 적용할 수 있다. 패딩 오라클 공격을 근본적으로 방어하는 방법은 패딩 오라클 공격이 적용 불가능한 패딩 방법을 설계하여 사용하는 방법과 GCM 및 CCM 등 암호화 인증 방식을 적용함으로써 패딩 오라클 공격을 미연에 방지하는 방법 등이 있다. 응용프로토콜에서는 패딩 오라클 공격으로부터 보호하기 위해 패딩 에러 처리 방법이 변경된 TLS v1.1, v1.2 를 인증 가능 운영모드와 함께 이용하는 것이 바람직하다.

본 논문에서는 12가지 패딩 방법 중 패딩 오라클 공격에 내성을 제공하는 PAIR-PAD, ABYT-PAD 및 ABIT-PAD를 선별을 통해, 이들의 특성을 분석하였다. 또한 이 특성 분석을 바탕으로 〈Table 4〉와 같은 5가지 설계 기준을 제시 하도록 한다.

Table 4. Five criteria for a safe padding design techniques.

Design Criteria	Correlative Padding Techniques	Vulnerability	Countermeasure
Leakage the length of the messages	CBC, ESP, XY, BOZ, ISO	Can recover the messages information through padding length leakage	Should not leak the padding or message length information
Leakage the padding information	ABYT, ABIT	Possible to recover specific message information from padding information	Should not guess the message information from padding information.
Detection of modified padding	PAIR, ABYT, ABIT	Can reduce efficiency through maliciously modify encryption	Require techniques about detecting maliciously modified ciphertext
Using the Random Number Generator	NIST, SSH, PAIR, ABYT	Using random number generator increases the complexity of implementation	Should not use the random number generator.
Exists of Invalid padding	NIST, SSH, OZ	Can be available in the theoretically attack from unused invalid padding	Should not be found the invalid padding by attacker.

〈Table 4〉의 설계기준에 맞춰 패딩 오라클 공격에 내성을 갖으며, 더불어 난수발생기 등을 사용하지 않는 안전성과 효율성을 겸비한 신규 패딩 방법에 대한 연구가 필요할 것으로 판단된다.

References

[1] Romain Bardou, "Ecient Padding Oracle Attacks on Cryptographic Hardware," CRYPTO 2012

[2] S. Vaudenay, "Security Flaws Induced by CBC Padding - Applications to SSL, IPSEC, WTLS...," Eurocrypt 2002, LNCS, vol. 2332, pp. 534-545, Springer-Verlag, 2002

[3] J. Black, H. Urtubia, "Side-Channel Attacks on Symmetric Encryption Schemes: The Case for Authenticated Encryption," USENIX, 2002.

[4] V. Klíma, T. Rosa, "Side Channel Attacks on CBC Encrypted Messages in the PKCS#7 Format," eprint2003, 2003

[5] K. G. Paterson, A. Yau, "Padding Oracle Attacks on the ISO CBC Mode Encryption Standard," CT-RSA 2004, LNCS, vol. 2964, pp. 305-323, Springer-Verlag, 2004

[6] J. Rizzo, T. Duong, "Practical Padding Oracle Attacks," USENIX WOOT 2010, 2010

[7] T. Duong, J. Rizzo, "Cryptography in the Web: The Case of Cryptographic Design Flaws in ASP.NET," IEEE Symposium on Security and Privacy 2011, 2011

[8] N. J. Alfardan, K. G. Paterson, "Lucky Thirteen: Breaking the TLS and DTLS Record Protocols," IEEE Symposium on Security and Privacy 2013, pp. 526-540, 2013

[9] John Black, Hector Urtubia(2002). "Side-Channel Attacks on Symmetric Encryption Schemes: The Case for Authenticated Encryption," USENIX 2002

〈저자소개〉

김 기 문 (Ki-moon Kim) 정회원
2011년 9월~현재: 한국인터넷진흥원(KISA) 선임연구원
2013년 9월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 암호시스템 안전성 분석, 암호 응용기술 개발

박 명 서 (Park-Myungseo) 정회원
2013년 2월: 국민대학교 수학과 졸업
2015년 2월: 국민대학교 금융정보보안학과 석사
2014년 12월~현재: 국가보안기술연구소
〈관심분야〉 정보보호, 암호 알고리즘, 이동통신보안

김 종 성 (Jong-sung Kim) 종신회원
2000년 8월/2002년 8월: 고려대학교 수학 학사/이학석사
2006년 11월: K.U.Leuven, ESAT/SCD-COSIC 정보보호 공학박사
2007년 2월: 고려대학교 정보보호대학원 공학박사
2007년 3월~2009년 8월: 고려대학교 정보보호기술연구센터 연구교수
2009년 9월~2013년 2월: 경남대학교 e-비즈니스학과 조교수
2013년 3월~현재: 국민대학교 수학과 조교수
2014년 3월~현재: 국민대학교 일반대학원 금융정보보안학과 조교수
〈관심분야〉 정보보호, 암호 알고리즘, 디지털 포렌식

이 창 훈 (Chang-Hoon Lee) 종신회원
2001년 2월: 한양대학교 수학과(이학사)
2003년 2월: 고려대학교 정보보호대학원(공학석사)
2008년 2월: 고려대학교 정보보호대학원(공학박사)
2009년 3월~2011년 2월 : 한신대학교 컴퓨터공학부 전임강사
2011년 3월~2012년2월: 한신대학교 컴퓨터공학부 조교수
2012년 3월~현재: 서울과학기술대학교 컴퓨터공학과 조교수
〈관심분야〉 정보보호, 암호학, 디지털포렌식, 컴퓨터이론

문 덕 재 (Duk-Jae Moon) 정회원
2000년 2월: 서울시립대학교 수학과 학사
2003년 2월: 고려대학교 정보보호대학원 석사
2015년 2월: 고려대학교 정보보호대학원 박사
2015년 2월~현재: 삼성 SDS
〈관심분야〉 블록 암호 및 스트림 암호 분석

홍 석 희 (Seok-hie Hong) 종신회원
1995년 2월: 고려대학교 수학과 학사
1997년 2월: 고려대학교 수학과 석사
2001년 8월: 고려대학교 수학과 박사
1999년 8월~2004년 2월: ㈜ 시큐리티 테크놀로지스 선임연구원
2003년 8월~2004년 2월: 고려대학교 정보보호기술연구센터 선임연구원
2004년 4월~2005년 2월: K.U.Leuven, ESAT/SCD-COSIC 박사후연구원
2005년 3월~2013년 8월: 고려대학교 정보보호대학원 부교수
2013년 9월~현재: 고려대학교 정보보호대학원 정교수
〈관심분야〉 대칭키·공개키 암호 분석 및 설계, 컴퓨터 포렌식