************************* COVER PAGE *************************

Randomized Key Chaining Modes with Unforgeability

Hsi-Chung Lin Sung-Ming Yen

Laboratory of Cryptography and Information Security
Dept of Computer Science and Information Engineering
National Central University
Chung-Li, Taiwan 320, R.O.C.
E-mail: {cs022064;yensm}@csie.ncu.edu.tw

Abstract

Block cipher based on private key cryptography has widely been employed for bulk encryption and a variety of operation modes have been developed so far. Recently, mode of operation for block cipher has been revisited in order to develop more robust cryptosystems under rigorous security models. One of this line of research is the related plaintext chaining (RPC) mode which is secure and can resist against forgeability attack. However, the RPC mode suffers from serious data expansion problem. In this paper, a randomized key chaining (RKC) mode by employing implicit encoding is proposed which provides the same security property as the RPC mode and is free from the data expansion disadvantage.

Keywords: Block cipher, Chosen ciphertext attack, Cryptography, Data expansion, Mode of operation, Private key cryptography, Unforgeability

This paper is submitted to the “Workshop on Cryptology and Information Security”

Contact author: Sung-Ming Yen (yensm@csie.ncu.edu.tw)

************************* COVER PAGE *************************

Randomized Key Chaining Modes with Unforgeability

Abstract. Block cipher based on private key cryptography has widely been employed for bulk encryption and a variety of operation modes have been developed so far. Recently, mode of operation for block cipher has been revisited in order to develop more robust cryptosystems under rigorous security models. One of this line of research is the related plaintext chaining (RPC) mode which is secure and can resist against forgeability attack. However, the RPC mode suffers from serious data expansion problem. In this paper, a randomized key chaining (RKC) mode by employing implicit encoding is proposed which provides the same security property as the RPC mode and is free from the data expansion disadvantage.

Keywords: Block cipher, Chosen ciphertext attack, Cryptography, Data expansion, Mode of operation, Private key cryptography, Unforgeability

1 Introduction

Conventionally, most of the security notions were shared when considering both public key cryptography and private key cryptography. However, distinct security requirements of these two kinds of cryptosystems exist. In the context of security analysis based on oracle model, both the encryption oracle and the decryption oracle should be protected in a private key cryptography. On the other hand, in a public key cryptography the encryption oracle is available to an attacker.

For bulk encryption of large amount of information, private key cryptography will still be the best choice. For a variety of different applications, distinct operation modes for private key cryptography have been developed. However, most of the operation modes proposed so far are vulnerable under some rigorous security models, especially the chosen ciphertext attacks and the adaptive chosen ciphertext attacks. Katz and Schneier gave an example of how to access the decryption oracle after eavesdropping a ciphertext [1] which demonstrates the applicability of adaptive chosen ciphertext attacks. Moreover, other examples of application level attacks against some real-life applications, e.g., SSL and IPSEC, were reported [2].

In [3], Katz and Yung introduced a new security notion of unforgeability for private key cryptography as well as a new operation mode, the *related plaintext chaining* (RPC) mode, which can provide unforgeability in private key cryptography. In this paper, the RPC mode will be analyzed and a new operation mode will be proposed against forgeability which is more efficient than the RPC mode.

In the Section 2, some security notions and existing operation modes will be briefly reviewed. RPC mode will be analyzed in the Section 3, then a new RKC mode will be proposed. Some possible further improvement of the proposed RKC mode will be developed in the Section 4. The Section 5 concludes this paper.

2 Preliminary Background

2.1 Related security notions

In [4], the notion of semantic security was first introduced by Goldwasser and Micali. The concept of semantic security is that anything about the plaintext which can be efficiently computed from the ciphertext can also be computed without the given ciphertext.

The notion of indistinguishability [4] emphasizes the adversary's inability to obtain any information of the corresponding plaintext from the ciphertext which has been proven to be polynomially equivalent to sematic security by Bellare et al. [5]. Non-malleability was introduced and proven to be polynomially equivalent to semantic security by Dolev et al. [6].

All the above three notions imply resistance against the chosen ciphertext security. For detailed definition and complete analysis of the relations between these notions and the adaptive/non-adaptive chosen ciphertext attacks can be found in [5, 7, 8].

The security notion of unforgeability [3] introduced by Katz and Yung emphasizes the inability of the adversary to create valid ciphertext. Three levels of unforgeability have been defined ranging from the weakest one to the strongest one:

(a) Random plaintext unforgeability: This is the weakest definition of unforgeability; in other words, it is the hardest attack for the adversary. The adversary succeeds if he can create the corresponding ciphertext of a plaintext which is randomly selected from the message domain as a challenge.

(b) Chosen plaintext unforgeability: In this attack, the adversary succeeds if he can create any valid pair of plaintext and ciphertext. Unlike the previous one, in this definition, the plaintext is chosen by the adversary himself but not a challenge message.

(c) Existential unforgeability: This is the strongest definition of unforgeability. The adversary succeeds if he can create any valid ciphertext. Note that the adversary need not have any idea about the corresponding plaintext. Obviously, this is the easiest attack for the adversary.

2.2 Modes of operation

In order to facilitate a variety of practical applications, many operation modes for block ciphers were developed, e.g., ECB, CBC, CFB, and OFB modes are defined for DES [9–12]. Bellare et al. also proposed the XOR mode [5] and the HCBC mode [13]. Security analysis of operation modes had also been extensively studied. The result is that all the previous operation modes are insecure under the adaptive/non-adaptive chosen ciphertext attacks. Readers are suggested to refer to [1, 2, 5, 14–17] for further detailed cryptanalysis.

3 Modes with Unforgeability

In [3], the related plaintext chaining (RPC) mode was proposed to provide unforgeability. One major drawback of the RPC mode is the large ciphertext expansion which makes the RPC mode inefficient for most applications. In this section, the RPC mode will be briefly reviewed and analyzed, then a new secure operation mode under the consideration of existential unforgeability will be proposed.

3.1 The RPC mode

The development of the RPC mode is a straightforward employment of serial numbers stamping within every data blocks such that any insertion or deletion of ciphertext will make the ciphertext be invalid. It is expected that the related serial numbers will not be continuous if modification, insertion, or deletion occurs. Moreover, two special blocks START and END mark the beginning and the end of the message in order to avoid the possible truncation attack. Basically, the RPC mode is an application of the encode-then-encipher encryption reported in [18]. Firstly, encode the plaintext into n blocks and append a START symbol and an END symbol as follow:

$$\underbrace{\mathsf{START}, \overbrace{P_1, P_2, \cdots, P_n}^{n \text{ blocks}}, \mathsf{END}}_{n+2 \text{ blocks}}$$

where the bit length of the two special symbols is identical to that of P_i. Then, encipher the encoded plaintext message as:

$$\mathcal{E}(\mathsf{START}, s), \mathcal{E}(P_1, s+1), \mathcal{E}(P_2, s+2), \cdots, \mathcal{E}(P_n, s+n), \mathcal{E}(\mathsf{END}, s+n+1)$$

where the serial number s acts as a counter in the stateful version of RPC mode. Note that the bit length of P_i plus the bit length of s equals to the data block size of the private key cryptosystem being considered.

The RPC mode is provably secure under the chosen ciphertext attack, but there are still some problems for the RPC mode. One major drawback of the RPC mode is the large ciphertext expansion. In [3], a typical example of using AES with a 32-bit serial number leads to a 33% of data expansion. This data expansion ratio is of course non-negligible. In block ciphers with smaller block size, this problem becomes even more serious. Another problem for the RPC mode is that no message block P_i can be identical to the special symbols START or END, otherwise ambiguity will occur. This limits the valid message space.

3.2 The proposed RKC mode

The advantage of encode-then-encipher technique has been discussed in [18, 19] and the RPC mode is an application of this technique by employing *explicit encoding* of using the serial number, the starting block, and the ending block. In this paper, we propose the concept of *randomized key chaining* (RKC) mode by employing *implicit encoding* such that instead of adding continuous serial numbers, the proposed RKC mode uses randomized key chaining (to be described later) to provide binding among the cipher blocks. There is no explicit encoding field in each message block, so no data expansion will be caused.

In the encoding step, the plaintext is divided into b-bit blocks and a b-bit random number block is appended at the beginning and at the end of the message blocks. Note that parameter b is the block size (in bit) of the underlying block cipher. Then, the encoded plaintext is encrypted by using randomized encipher keys where the encipher key of block P_i is the bitwise XOR of the previous plaintext block P_{i-1} and the master secret key sk. For simplification, we assume that the key size (in bit) is equivalent to the block size. The detailed structure of the proposed RKC mode is depicted in Fig. 1 and an algorithmic description can be found in Fig. 2.

When the message receiver decrypts the ciphertext, he can firstly decipher each cipher block C_i $(i > 0)$ by using the decipher key which is the bitwise XOR of the previous plaintext block P_{i-1} and the master key sk. For C_0, the decipher key (identical to the corresponding encipher key) is (IV $\oplus$ sk). After completely deciphering, the receiver can verify the validity of the ciphertext easily by comparing the first and the last block of the corresponding plaintext. The message is valid only if those two plaintext blocks are identical. Otherwise, the received ciphertext is invalid. Note that the initialization vector (IV) needs not to be secret and can even be a fixed value for public use. Moreover, in most applications, the master key sk is in fact a session key. So a public IV with fixed value is sufficient for most security applications.

In some applications, the length of the transferred ciphertext is not explicitly defined. In those cases, ambiguity will occur when any plaintext block is the same as the random number. To remedy this drawback, we introduce the idea of random block. For some block cipher systems, a full

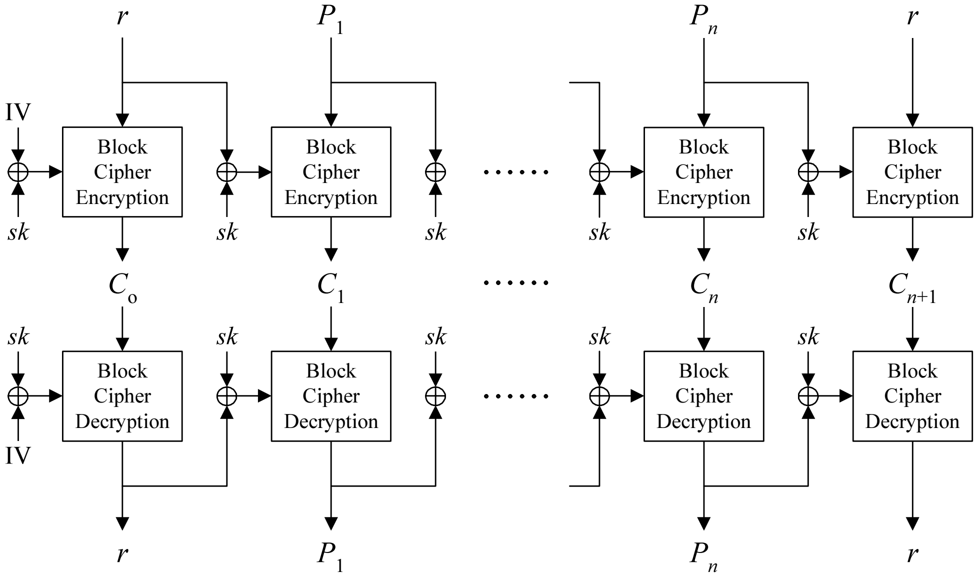

Figure 1: Encryption and decryption structures of the RKC mode

```
Algorithm $\mathcal{E}$-RKC($P$)
    parse $P$ as $P_1,\ldots,P_n$,
        where $|P_i|=b$
        $r \leftarrow \{0,1\}^b$
    $C_0 = \mathcal{E}_{(\mathrm{IV} \oplus sk)}(r)$
    $C_1 = \mathcal{E}_{(r \oplus sk)}(P_1)$
    for $I = 2,\ldots n$ do
        $C_i = \mathcal{E}_{(P_{i\text{-}1} \oplus sk)}(P_\mathrm{i})$
    $C_{n+1} = \mathcal{E}_{(P_n \oplus sk)}(r)$
    return $C = C_0,\ldots,C_{n+1}$
```

```
Algorithm $\mathcal{D}$-RKC($C$)
    parse $C$ as $C_0,\ldots,C_{n+1}$,
        where $|C_i| = b$
    $r = \mathcal{D}_{(\mathrm{IV} \oplus sk)}(C_0)$
    $P_1 = \mathcal{D}_{(r \oplus sk)}(C_1)$
    for $I = 2,\ldots n$ do
        $P_i = \mathcal{D}_{(P_{i\text{-}1} \oplus sk)}(C_i)$
    $P_{n+1} = \mathcal{D}_{(P_n \oplus sk)}(C_{n+1})$
    if $P_{n+1} \neq r$ return $\perp$
    return $P = P_1,\ldots,P_n$
```

Figure 2: Encryption and decryption algorithms of the RKC mode

block is too big a place for a random number, e.g., in AES the block size is 128 bits. In fact, a 64-bit random number is secure enough. So, instead of considering the whole block as a random number, a random block can be divided into two parts, a random number r and a length field for indicating the message length in block. With this length field, no ambiguity will occur.

In this RKC structure, drawbacks of the RPC mode are solved, and ciphertext expansion is only two blocks which is negligible.

3.3 Security analysis of the RKC mode

With this RKC mode, it is infeasible for an adversary to forge a ciphertext, to insert, or to delete ciphertext blocks without being detected. Any insertion, deletion, or modification on a ciphertext block will violate the randomized key chaining property since each randomized encipher key is derived from both the master secret key and the previous message block.

First, any modification on a cipher block will make it mapping into a totally different plaintext block. Consequently, the next cipher block will be decrypted to a wrong plaintext block, too. Finally, the last block will not be decrypted to the correct value r. Similarly, any insertion or deletion will make subsequent blocks decrypted to wrong plaintext. Eventually, the last decrypted block will not be identical to the first block.

Some adversaries may try to duplicate one block and to insert the duplicated one right after the original one; or others may try to delete blocks reside between two identical ciphertext blocks. Note especially that both kinds of attack cannot succeed since two *identical* cipher blocks in one ciphertext message map to *different* plaintext. This is the most important property of the proposed RKC mode due to the proposed concept of *randomized key chaining.*

The RKC mode also satisfies the unforgeability of the strongest level and this can be proven by following the Definition 5 of [3]. In order to show the relation between the RKC mode and adaptive/non-adaptive chosen ciphertext security, we provide two intuitive evidences here, and the complete proof will appear in the full version of this paper.

(a) Similar to the RPC mode, the proposed RKC mode is also an application of the encode-then-encipher technique. The encoding algorithm of the RKC mode is a *rare-collision encoding* where the probability of collision is $\frac{1}{2^{|r|}}$ where $|r|$ is the bit length of r. According to Theorem 1 of [18], this encoding is a good PRP (pseudo random permutation) and it ensures the resulting encryption reaches semantic security.

(b) Assume that a scheme is secure when the decryption oracle is not accessible. There exists the same level of security with the assumption when an adversary can only input invalid ciphertext into the accessible decryption oracle. As already described, in the proposed RKC mode, it is infeasible for an adversary to generate a valid ciphertext. This intuition therefore implies to adaptive chosen ciphertext secure due to the Theorem 1 of [3].

4 Segmentation for RKC Mode

One drawback of RKC mode is that the parallelizable problem. In RPC mode, both encryption and decryption process can be done in parallel. However, in the proposed RKC mode, only encryption process can be done in completely parallel. In the decryption procedure of RKC mode, P_i is a necessary information for the key-retrieval process in the decryption procedure of C_{i+1}. So in the original RKC mode, decryption process should be done block by block. This key chaining design ensure the integrity of ciphertext in an efficient way, but cause another efficiency issue. But this problem can be reduced by segmentation. Segmentation can be done in various ways according to various applications.

The idea of segmentation is simple, firstly we deal with the unforgeability of each section which is constructed of blocks. Secondly, we ensure the unforgeability of the whole ciphertext which is the combination of all unforgeable sections. In the following, three kind of segmentation methods are presented.

4.1 Fixed-length segmentation

The most intuitional segmentation strategy is the fixed-length segmentation. For a plaintext with $m \times n$ blocks, it can be view as a m-segment plaintext, in which each segment is constructed by n blocks. The structure of this idea is shown in Fig. 3.

Clearly, the encipher key of the random number r_1 is $(\mathrm{IV} \oplus sk)$, while the encipher key of r_i is $(r_{i-1} \oplus sk)$ when $i \geq 2$. Note that the random number r_m should be identical to r_1. Then any modification on a block will make that segment invalid (i.e., the first block and the last block of that segment are not identical). Similarly, modification which focus on segments will make the whole ciphertext invalid since r_1 and r_m are not equivalent. An important property of this segmentation is that although the structure of segmentation looks a little tree like, the encipher key (of each block) is still a chain for the whole ciphertext but not only a chain for each segment.

With segmentation, encryption procedure is still completely parallelizable, while the ability of parallel decryption depends on the parameter m. The encryption procedure is still encode-then-encipher, but encode in another way.

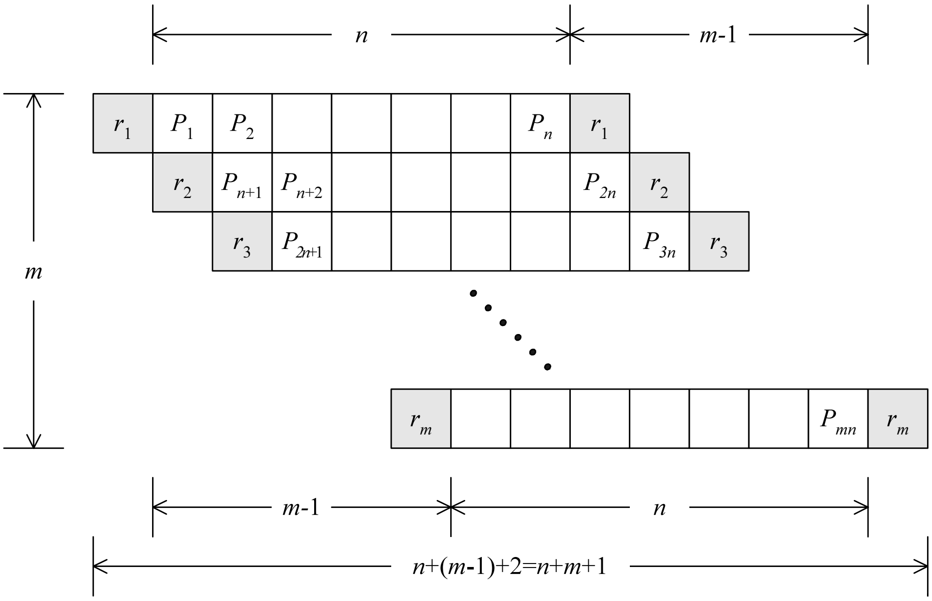

Figure 3: Fixed-length segmentation

4.2 Decreased-length segmentation

The method described in the previous subsection solve the problem of parallel, and one can customized his own parameter m and n according to his application. However, the performance of segmentation can be raise again.

In the fixed-length method, decryption processes of different segments are not stop at the same time. When segment one is already done its decryption, it has to wait other segments. If every segment can stop at the same time, the decryption time can be reduced. For the same message of $m \times n$ blocks where m is the number of segment, we can divide those $m \times n$ blocks into several decreased-length segments as in Fig. 4. For simplification, we assume that $n > \frac{m-1}{2}$ here.

The length of the first segment is $n + (\frac{m-1}{2})$, and the length (in block) of the i-th segment is

$$n + (\frac{m-1}{2}) - (i-1) = n + (\frac{m+1}{2}) - i$$

and the sum (in block) of those segments is still $m \times n$.

$$\sum_{i=1}^{m} [n + (\frac{m+1}{2}) - i] = m \times n$$

Clearly, for the same message with $m \times n$ blocks , the decrypting time down to $n + (\frac{m+1}{2}) + 1$ from $n + m + 1$ while the expansion is the same $2m$ blocks. For big m, the improvement is extensively large.

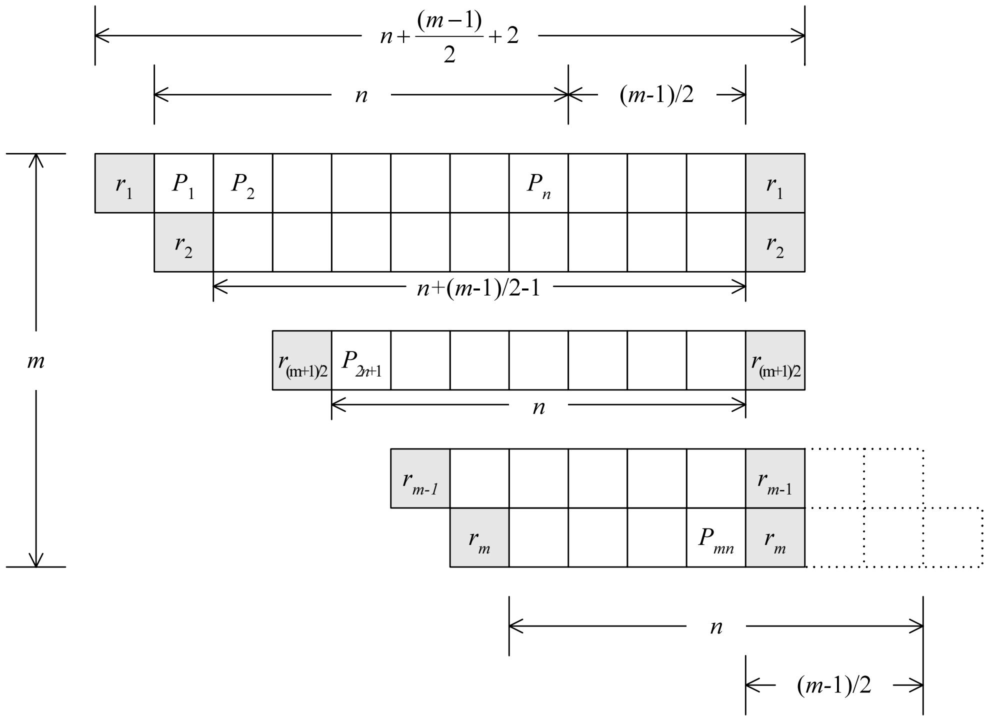

Figure 4: Decreased-length segmentation

4.3 Random-length segmentation

The segmentation methods describe in previous sections are deterministic, although the encrypting result is non-deterministic. Here we present a probabilistic segmentation method. Either in fixed-length or decreased-length segmentation method, the bit length of the random number is the same as the block size. In the random-length segmentation, the idea of random block described in Section 3.2 is adopted again. Here, the random block is separated into two parts, a random number r and a *next* address. The random number follows the same definition as before, while the *next* address identifies the address of next random block with different random number(the first block of the next segment). Note that the value $next_m$ should set to zero (the first block of the first section) to identify this is the last section. And the key chaining method is the same as above two. The structure is shown in Fig. 5.

Obviously, the random-length segmentation is a general form of this technique. Another advantage of the random-length segmentation is the update ability of ciphertext. Rather than re-encrypt the whole plaintext again when users do some modification or edition on the message, update those correspondent sections will make the whole ciphertext valid.

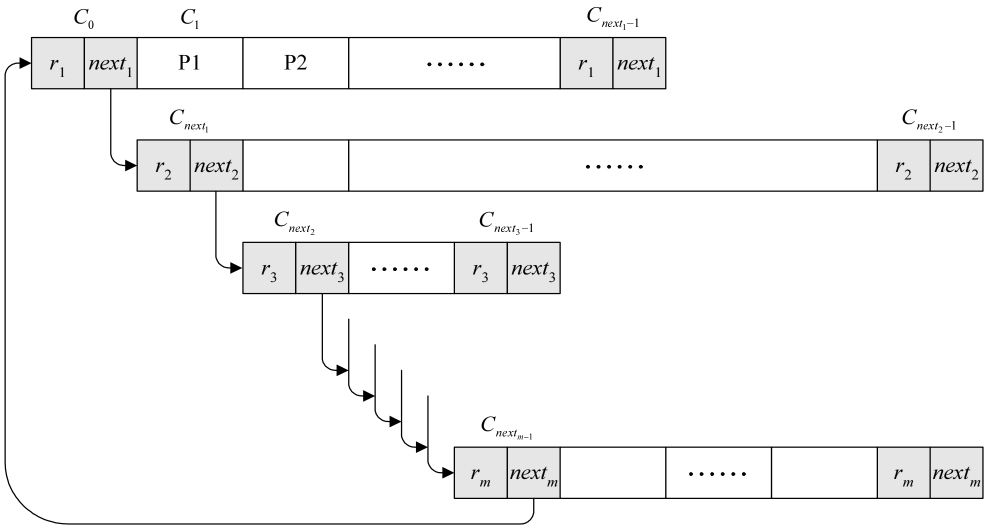

Figure 5: Random-length segmentation

5 Conclusions

Key idea of the proposed RKC mode is to remove the serial number embedded in each data block, instead the concept of randomized key chaining is employed. With this design, the requirement of unforgeability can be achieved efficiently.

The simple RKC mode proposed in Section 3.2 suffers only negligibly small amount of ciphertext expansion and the encryption process can be completely parallelizable. If for some reasons, parallel decryption is desired, RKC mode with segmentation can achieve this goal. Under this design, an example of 65536 cipher blocks with 256 processors in parallel, the ciphertext expansion is less than one percent.

When compared with the RPC mode, efficiency improvement of the proposed RKC mode is remarkable, and the problem of ambiguity of special symbols no longer exists. When the segmentation approach is employed, encryption updating can be easily achieved in the proposed RKC mode.

References

[1] J. Katz and B. Schneier, "A chosen ciphertext attack against several e-mail encryption protocols," *Proc. of the 9th USENIX Security Symposium*, 2000.

[2] S. Vaudenay, "Security flaws induced by CBC padding – applications to SSL, IPSEC, WTLS ...," *Advances in Cryptology – EUROCRYPT 2002*, Lecture Notes in Computer Science, Vol.2332, Springer Verlag, pp.534–545, 2002.

[3] J. Katz and M. Yung, "Unforgeable encryption and chosen ciphertext secure modes of operation," *Fast Software Encryption, FSE 2000*, Lecture Notes in Computer Science, Vol.1978, Springer Verlag, pp.284–299, 2001.

[4] O. Goldreich , S. Micali, “Probabilistic encryption,” *J. of computer and System Sciences*, Vol.28, pp.270–299, April 1984.

[5] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway, “A concrete security treatment of symmetric encryption,” *Proc. of the 38th Symposium on Foundations of Computer Science*, IEEE, 1997.

[6] D. Dolev, C. Dwork, and M. Naor, “Non-malleable cryptography” *Proc. of the 23rd Annual Symposium on the Theory of Computing, STOC 1991*, ACM, 1991.

[7] J. Katz and M. Yung, “Complete characterization of security notions for probabilistic private-key encryption,” *Proc. of the Thirty-Second Annual ACM Symposium on Theory of Computing, STOC 2000*, ACM, pp.245–254, 2000.

[8] M. Bellare, A.Desai, D. Pointcheval, P. Rogaway, “Relations among notions of security for public-key encryption schemes,” *Advances in Cryptology – CRYPTO ’98*, Lecture Notes in Computer Science, Vol.1462, Springer Verlag, pp.26–45, 1998.

[9] ANSI X3.106, “American National Standard for Information Systems – Data Encryption Algorithm – modes of operation,” American National Standards Institute, 1983.

[10] ISO 8372, “Information processing – modes of operation for a 64-bit block cipher algorithm,” International Organization for Standardization, Geneva, Switzerland, 1987.

[11] National Bureau of Standards, NBS FIPS PUB 74, “Guidelines for implementing and using the NBS data encryption standard,” U.S. Department of Commerce, April 1981.

[12] National Bureau of Standards, NBS FIPS PUB 81, “DES modes of operation,” U.S. Department of Commerce, December 1980.

[13] M. Bellare, A. Boldyreva, L. Knudsen, C. Namprempre, “Online ciphers and the hash-CBC construction,” *Advances in Cryptology – CRYPTO 2001*, Lecture Notes in Computer Science, Vol.2139, Springer Verlag, pp.292–309, 2001.

[14] E. Biham, “Cryptanalysis of multiple modes of operation,” *Journal of cryptology*, Vol.11, pp.45–58, 1998.

[15] E. Biham and L.K. Knudsen, “Cryptanalysis of the ANSI X9.52 CBCM mode,” *Advances in Cryptology – EUROCRYPT ’98*, Lecture Notes in Computer Science, Vol.1403, Springer Verlag, pp.100–111, 1998.

[16] C.J.A. Jansen, and D.E. Boekee, “Modes of blockcipher algorithms and their protection against active eavesdropping,” *Advances in Cryptology – EUROCRYPT ’87*, Lecture Notes in Computer Science, Vol.304, Springer Verlag, pp.281–286, 1988.

[17] B. Preneel, M. Nuttin, V. Rijmen, and J. Buelens, “Cryptanalysis of the CFB mode of the DES with a reduced number of rounds,” *Adavnces in Cryptology – CRYPTO ’93*, Lecture Notes in Computer Science, Vol.773, Springer Verlag, pp.212–223, 1988.

[18] M. Bellare and P. Rogaway, “Encode-then-encipher encryption: how to exploit nonces or redundancy in plaintexts for efficient cryptography,” *Advances in Cryptology – ASIACRYPT 2000*, Lecture Notes in Computer Science, Vol.1976, Springer Verlag, pp.317–330, 2000.

[19] A. Desai, “New paradigms for constructing symmetric encryption schemes secure against chosen-ciphertext attack,” *Advances in Cryptology – CRYPTO 2000*, Lecture Notes in Computer Science, Vol.1880, Springer Verlag, pp.394–412, 2000.