EURASIP Journal on Applied Signal Processing 2005:12, 1923–1927

Cryptanalysis of the Two-Dimensional Circulation Encryption Algorithm

Christophe De Cannière
Computer Security and Industrial Cryptography (COSIC), Department of Electrical Engineering (ESAT), Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, 3001 Leuven-Heverlee, Belgium
Email: christophe.decanniere@esat.kuleuven.be

Joseph Lano
Computer Security and Industrial Cryptography (COSIC), Department of Electrical Engineering (ESAT), Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, 3001 Leuven-Heverlee, Belgium
Email: joseph.lano@esat.kuleuven.be

Bart Preneel
Computer Security and Industrial Cryptography (COSIC), Department of Electrical Engineering (ESAT), Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, 3001 Leuven-Heverlee, Belgium
Email: bart.preneel@esat.kuleuven.be

Received 13 February 2004; Revised 25 November 2004; Recommended for Publication by Andy Wu

We analyze the security of the two-dimensional circulation encryption algorithm (TDCEA), recently published by Chen et al. in this journal. We show that there are several flaws in the algorithm and describe some attacks. We also address performance issues in current cryptographic designs.

Keywords and phrases: chaos-based cryptography, cryptanalysis, security evaluation, TDCEA, known-plaintext attack.

1. INTRODUCTION

In symmetric-key cryptography, two parties share a secret key K to encrypt messages using a cipher. Symmetric encryption techniques are used to efficiently encrypt data. Two common types of ciphers are commonly used nowadays: block ciphers and stream ciphers.

Block ciphers encrypt blocks of data (typically 64 or 128 bits) in a fixed key-dependent way. The design of block ciphers is a well-studied area of research. The best known block ciphers are the Data Encryption Standard (DES) [2] and the Advanced Encryption Standard (AES) [3]. In the past decade, many new attacks on block ciphers have emerged, the most important ones being differential [4] and linear [5] cryptanalysis. Differential cryptanalysis is an example of a chosen-plaintext attack, while linear cryptanalysis is a known-plaintext attack. A good design should at least be resistant to these attacks.

Stream ciphers, on the other hand, generate a pseudorandom key stream independent of the plaintext. This key stream is then used to encrypt the plaintext character per character in a time-varying way.

In this paper, we will study the security of the recently published two-dimensional circulation encryption algorithm (TDCEA) [1]. This design can be seen as a block cipher, but it also has some properties of a stream cipher. It encrypts blocks of 64 bits at a time by combining each block with the secret key.

The outline of this paper is as follows. In Section 2, we briefly describe TDCEA. In Section 3, we study the security of TDCEA. We show several flaws in the algorithm and describe a known-plaintext attack that breaks the cipher in less than 10 seconds on a 1.5 GHz PC. We also explain why we believe that it will not be possible to improve the design to be cryptographically sound, as TDCEA lacks many of the desirable properties of a state-of-the-art symmetric encryption algorithm. In Section 4, we address the tradeoffs that need to be made between performance and security of a design. We then discuss the use of concepts from chaos theory in cryptography in Section 5, and conclude in Section 6.

2. BRIEF DESCRIPTION OF TDCEA

In this section, we give a brief description of TDCEA. For a full description, we refer to [1]. The secret key of TDCEA

consists of a 17-bit value μ ($3 < \mu < 4$), a 17-bit initial state $x(0)$ ($0 < x(0) < 1$), and two 3-bit values α and β.[1] The plaintext is encrypted in blocks of 64 bits, which corresponds to eight pixels. For every block $p(i)$ ($i = 0, 1, \ldots$), we calculate a new value for the internal state $x(0)$ with the following logistic map:

$$x(i) = \mu \cdot x(i-1) \cdot (1 - x(i-1)). \quad (1)$$

The ciphertext $c(i)$ is then obtained by arranging $p(i)$ in a matrix, and performing rotations on all rows and columns of this matrix. By how many positions each row and column is rotated is dependent on α, β, and $x(i)$.

3. SECURITY OF TDCEA

3.1. Flaws of TDCEA

In this section, we list several flaws of TDCEA.

The key of TDCEA is too short

The effective length of the secret key $(\mu, x(0), \alpha, \beta)$ is only 40 bits. Our unoptimized implementation of TDCEA runs at about 1 million encryptions per second on a 1.5 GHz PC. This means that exhaustive search (trying all possible keys until the right key is found) takes only about 12 days on a single PC. On a large cluster of computers, the key can be found in few minutes. A secret key of at least 80 bits is nowadays the minimum requirement for security against exhaustive search. To make matters worse, TDCEA only uses 23 bits (α, β, and $x(i)$) to encrypt a plaintext block p_i, which makes divide-and-conquer attacks on the key space possible.

TDCEA only permutes the plaintext

According to the principles of confusion and diffusion introduced by Shannon [6], a strong cipher should use a combination of substitutions and permutations. This can be found in two popular schemes for block ciphers, namely Feistel networks (such as DES) and substitution-permutation networks (such as AES). However, TDCEA only permutes the values of 8 consecutive pixels. It is easy to see that only permuting an image will not hide all of its properties. For instance, an entirely white image will remain entirely white after encryption. Especially pictures with low entropy will still be recognizable after encryption. We have tried to encrypt such pictures and in many cases they are still very recognizable after encryption.

TDCEA is noniterative

When we consider TDCEA as a block cipher, we see that it consists of operations such as multiplications and rotations, which are commonly used in block ciphers. In order to resist cryptanalytic attacks, a strong block cipher is built out of many iterations of the same function. For instance, DES consists of 16 rounds, and AES consists of 10, 12, or 14 rounds of the same function, every time with a different round key. TDCEA only consists of one round, which means there is little hope it will resist linear and differential cryptanalysis.

The key distribution of TDCEA is weak

In a good key distribution system, compromise of one session key should not compromise the master key. In TDCEA, a session key is encrypted by an exclusive or with the master key (see the full description of TDCEA in [1]). This means that compromise of one single session key will also compromise the master key, and thus all previous and future session keys.

The logistic map is not a good pseudorandom number generator

The sequence $x(i)$ becomes quickly periodic. Especially for small values of μ, this is a problem as the period will be very small and an attacker will observe repetition in the permutation used in different blocks. For instance, for μ between 3 and 3.45, the logistic map will oscillate between just two values.

3.2. Known-plaintext attack on TDCEA

The authors claim that TDCEA resists a known-plaintext attack. We will now show that this is not correct by describing an algorithm that breaks TDCEA with only 24 known-plaintext bytes, which is equivalent to three known-plaintext blocks.

The attacker has at his disposal three plaintext blocks p_0, p_1, and p_2, and the corresponding ciphertexts c_0, c_1, and c_2. We use the fact that TDCEA only uses 23 bits (α, β, and $x(i)$) to encrypt a plaintext block p_i (cf. Section 3). The attacker now proceeds as follows to recover the plaintext.

Step 1. The attacker encrypts the plaintext p_0 by trying all values for α, β, and $x(0)$. This means trying $2^{23} \approx 8\,400\,000$ possibilities, which requires a few seconds on a PC. He then checks whether the obtained ciphertext is equal to the actual ciphertext c_0. If this is the case, he has found a valid guess α_g, β_g, and $x(0)_g$. One typically obtains very few valid guesses: 4 is a typical value.

Step 2. For the valid guesses α_g, β_g obtained in Step 1, the attacker now tries all values of $x(1)$ to find those for which p_1 encrypts to c_1. Again a very small list of possibilities α_g, β_g, and $x(1)_g$ is obtained.

Step 3. For the guesses α_g, β_g valid in both Steps 1 and 2, the attacker now tries all values of $x(2)$ to find those for which p_2 encrypts to c_2. Again a very small list of possibilities α_g, β_g, and $x(2)_g$ is obtained.

Step 4. The attacker now exhaustively searches the small list obtained in Steps 1 through 3 for values $(x(0)_g, x(1)_g, x(2)_g)$

[1] α and β are used in the algorithm to determine the number of positions by which an 8-bit word is rotated. Thus the only thing we need to know about α and β is their value modulo 8.

for which α_g and β_g are the same and for which the following equation holds:

$$\frac{x(1)_g}{x(0_g)\cdot(1-x(0)_g)} = \frac{x(2)_g}{x(1)_g\cdot(1-x(1)_g)}. \quad (2)$$

Considering (1), one can easily see that the guesses $(x(0)_g, x(1)_g, x(2)_g)$ for which the above equality holds are with high probability the correct values used in TDCEA, that the corresponding α_g and β_g are also the correct values, and that the fraction in (2) is equal to the secret μ. We have thus obtained the whole secret key of TDCEA.

This algorithm has been implemented in C on a Xeon 1.5 GHz. It breaks TDCEA in less than 10 seconds.

3.3. Further comments on the security of TDCEA

We have investigated whether the basic structure of TDCEA can be improved so that the algorithm becomes secure.

A first thing that should be done is to increase the key size of the algorithm to prevent the simple known-plaintext attack described above. This implies a substantial increase in the size of the multiplier, which will also affect the speed and the area required by the encryption algorithm.

However, even a huge secret key will not make the cipher secure. An essential flaw of TDCEA is the fact that it only permutes the plaintext as noted above. This problem will remain the same irrespective of the key size. Besides the visible problems of the algorithm, it is also easy to recover the secret key. For instance, one can mount a chosen-plaintext attack as follows: encrypt 8 pixels such that only 1 bit in the circulation matrix is one and all other bits are zero. In the ciphertext, we can see where this bit ended up and thus we know how it has been rotated. As this rotation is directly dependent on the key, this gives us information on the secret key. Working in this way, it will be easy to collect enough information to recover the entire key.

As explained in Section 3.1, many other problems will need to be overcome in order to make the security of TDCEA acceptable. We do not believe that it will be possible to make a secure and efficient algorithm out of the basic building blocks of TDCEA. We will discuss this further in the next section.

4. ON THE TRADEOFF BETWEEN PERFORMANCE AND SECURITY

It is clear that a minimal requirement for a good symmetric-key algorithm is that it should be secure, as there is no point in using an insecure encryption algorithm. In practice, it is required that the algorithm have a sufficiently large secret key and that there is no attack on the algorithm faster than exhaustive search. For instance, the five block ciphers selected for the final of the Advanced Encryption Standard (AES) development effort [7] fulfill this requirement for 128-bit, 192-bit, and 256-bit keys.

To be used in practice, an algorithm should also have a good performance in various applications. In software, this is expressed in the number of cycles the processor needs to encrypt a byte of plaintext (cycles/byte). In hardware, good performance is a combination of high throughput and low gate count.

Rijndael, the algorithm that has been selected as the AES, achieves very good performance in both software and hardware. Moreover, the design can be implemented in hardware either with a very low gate count and with a more than reasonable throughput (e.g., [8] describes an ASIC implementation using 5400 gates and encrypting 300 Mbps) or optimized for speed and thus heavily pipelined (e.g., [9] describes an ASIC implementation using 173 000 gates and encrypting 2290 Mbps). The AES is a cost-effective practical solution that can be used in most applications, certainly including multimedia data transmission.

In some rare cases where a very low gate count combined with a high throughput is required, it may not be possible to use AES. In these cases, as noted in [10], a stream cipher may achieve a better tradeoff between throughput, gate count, and security. An interesting question is whether the building blocks of TDCEA could provide a solution in such cases. TDCEA has two main building blocks, a multiplication and key-dependent rotations. We will now explain why these building blocks will not achieve a better tradeoff between throughput, gate count, and security than Rijndael.

Multiplications are not the best choice in hardware. For the AES finalists, it has been shown that the block ciphers using multiplication had a significantly longer critical path and also needed more area than those not using multiplication operations; see [11].

The key-dependent rotations of the matrix also will not offer a good tradeoff. The horizontal and vertical rotations have to be performed sequentially, which will make the execution of the algorithm slow compared with other diffusion methods. Moreover, one can note that depending on the secret key, a different number of cycles are performed. This fact will make the system vulnerable to side-channel attacks such as timing and power analysis; see [12, 13].

5. ON CHAOS THEORY AND CRYPTOGRAPHY

In the past years, many new cryptographic algorithms based on chaotic concepts have been published. According to Kocarev [14], *"despite a huge number of papers published in the field of chaos-based cryptography, the impact that this research has made on conventional cryptography is rather marginal."*

Indeed, most of these new designs are too slow or insecure. Often they are both insecure and slow. We believe that this is due to an insufficient knowledge of the state of the art in cryptanalysis by the designers of such systems, as their designs do not resist the most basic of cryptanalytic attacks.

However, it is possible to use chaotic concepts to build ciphers that seem to do well on both security and performance. For example, the stream cipher Rabbit [15] has a good performance on a Pentium and the first analysis indicates that it has sufficient resistance against cryptanalytic attacks [16].

6. CONCLUSION

We have shown several flaws of the TDCEA algorithm and implemented an attack that breaks the cipher with 24 bytes of known plaintext. The attack runs in less than 10 seconds on a PC. TDCEA is thus highly insecure and should not be used. We have also explained why we believe that the building blocks of TDCEA are not suitable to achieve a good tradeoff between security and performance in a state-of-the-art symmetric encryption algorithm.

We recommend the use of standard encryption algorithms, such as the Advanced Encryption Standard or the block ciphers in the NESSIE [17] portfolio, in practical applications. These standards have undergone an extensive security analysis, achieve very good tradeoffs between performance and security, and can be used in almost all applications.

ACKNOWLEDGMENTS

This work has been supported by the Concerted Research Action (GOA) Mefisto. The first author is an FWO Research Assistant, sponsored by the Fund for Scientific Research–Flanders. The second author's research is financed by a Ph.D. grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

REFERENCES

[1] H. Chen, J. Guo, L. Huang, and J. Yen, "Design and realization of a new signal security system for multimedia transmission," *EURASIP Journal on Applied Signal Processing*, vol. 2003, no. 13, pp. 1291–1305, 2003.

[2] National Institute of Standards and Technology, *FIPS-46: Data Encryption Standard*, January 1977.

[3] National Institute of Standards and Technology, *FIPS-197: Advanced Encryption Standard*, November 2001.

[4] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Journal of Cryptology*, vol. 4, no. 1, pp. 3–72, 1991.

[5] M. Matsui, "Linear cryptanalysis method for DES cipher," in *Proc. EUROCRYPT '93*, T. Helleseth, Ed., vol. 765 of *Lecture Notes in Computer Science*, pp. 386–397, Springer-Verlag, Lofthus, Norway, May 1993.

[6] C. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28-4, pp. 656–715, 1949.

[7] National Institute of Standards and Technology, *Advanced Encryption Standard*, http://csrc.nist.gov/CryptoToolkit/aes/.

[8] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A compact Rijndael hardware architecture with S-box optimization," in *Proc. ASIACRYPT '01*, C. Boyd, Ed., vol. 2248 of *Lecture Notes in Computer Science*, pp. 239–254, Springer-Verlag, Gold Coast, Australia, December 2001.

[9] I. Verbauwhede, P. Schaumont, and H. Kuo, "Design and performance testing of a 2.29 Gb/s Rijndael processor," *IEEE J. Solid-State Circuits*, vol. 38, no. 3, pp. 569–572, 2003.

[10] ECRYPT Network of Excellence, *The State of the Art of Stream Ciphers*, Brugge, Belgium, October 2004, http://www.isg.rhul.ac.uk/research/projects/ecrypt/stvl/sasc.html.

[11] T. Ichikawa, T. Kasuya, and M. Matsui, "Hardware evaluation of the AES finalists," in *Proc. 3rd AES Conference*, pp. 279–285, New York, NY, USA, April 2000.

[12] P. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," in *Proc. CRYPTO '96*, N. Koblitz, Ed., vol. 1109 of *Lecture Notes in Computer Science*, pp. 104–113, Springer-Verlag, Santa Barbara, Calif, USA, August 1996.

[13] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. CRYPTO '99*, M. Wiener, Ed., vol. 1666 of *Lecture Notes in Computer Science*, pp. 388–397, Springer-Verlag, Santa Barbara, Calif, USA, August 1999.

[14] L. Kocarev, "Chaos-based cryptography: a brief overview," *IEEE Circuits Syst. Mag.*, vol. 1, no. 3, pp. 6–21, 2001.

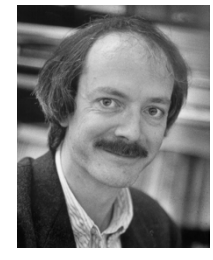
[15] M. Boesgaard, M. Vesterager, T. Pedersen, J. Christiansen, and O. Scavenius, "Rabbit: a new high-performance stream cipher," in *Proc. FSE '03*, T. Johansson, Ed., vol. 2887 of *Lecture Notes in Computer Science*, pp. 307–329, Springer-Verlag, Lund, Sweden, February 2003.

[16] V. Rijmen, *Analysis of Rabbit*, www.cryptico.com, September 2003.

[17] NESSIE Consortium, *Portfolio of Recommended Cryptographic Primitives*, 2003, http://www.cryptonessie.org.

Christophe De Cannière received his M.S. degree in electrical engineering from the Katholieke Universiteit Leuven, Belgium, in 2001. Currently, he is pursuing a Ph.D. degree at the COSIC Research Group, the Electrical Engineering Department, KU Leuven. He is supported by the Fonds voor Wetenschappelijk Onderzoek Vlaanderen (FWO) and is working in the field of symmetric encryption under the supervision of Professor Bart Preneel.

Joseph Lano received his M.S. degree in electromechanical engineering, option electronics, from the Katholieke Universiteit Leuven, Belgium, in 2002. Currently, he is pursuing a Ph.D. degree at the COSIC Research Group, the Electrical Engineering Department, KU Leuven. He is supported by a Ph.D. grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen), and is working in the field of symmetric encryption under the supervision of Professor Bart Preneel.



Bart Preneel received the M.S. degree in electrical engineering and the Doctorate in applied sciences in 1987 and 1993, respectively, both from the Katholieke Universiteit Leuven, Belgium. He is a Professor in the Electrical Engineering Department, Katholieke Universiteit Leuven, and a Visiting Professor at the TU Graz, Austria. Together with Professor J. Vandewalle, he is heading the Research Group COSIC at the KU Leuven, which currently has 35 members. He has held visiting professor positions at the Ruhr-University Bochum, Germany, at the University of Bergen, Norway, and the University of Ghent, Belgium. He was also a Research Fellow at the Department of Electrical Engineering and Computer Sciences, the University of California at Berkeley. His main research interests are cryptology and information security. He has authored and coauthored more than 180 articles in international journals and conference proceedings.

He is a Vice-President of the International Association for Cryptologic Research (http://www.iacr.org) and Chairman of the Leuven Security Excellence Consortium (http://www.l-sec.be). Currently he is a Project Manager of ECRYPT (http://www.ecrypt.eu.org), the EU-funded European Network of Excellence on Cryptology and Watermarking. In 2003, he received the European Information Security Award in the area of academic research. He is a Member of the Editorial Board of the Journal of Cryptology and of the ACM Transactions on Information and System Security.