

# Cryptanalysis Technique to Evaluate the Strength of Ciphers

by Toshio Tokita, Yasuyuki Sakai and Katsuyuki Takashima\*

The design of secure ciphers (cryptosystems) generally requires cryptanalytical techniques with which to evaluate their strength. This paper provides a summary of methods of evaluating the strengths of common-key block ciphers and public-key cryptosystems. It also introduces software developed by Mitsubishi Electric Corporation to evaluate the security and the performance of cryptosystems.

## Common-Key Block-Cipher Strength Evaluation

Here we explain differential and linear cryptanalysis, methods commonly used for common-key block ciphers.

DIFFERENTIAL CRYPTANALYSIS AND DIFFERENTIAL CHARACTERISTIC PROBABILITIES. Differential cryptanalysis, proposed by Biham, et al, in 1990 is based on the principle that it is possible to characterize statistically variations between plaintexts and ciphertexts, where the characteristics can be used to guess key information. Generally, the cipher strength against this cryptanalysis method is expressed in terms of maximum average differential probability, as in Eq. 1, with lower maximum average differential probability indicating more secure ciphers. Here,  $\Delta P (\neq 0)$  and  $\Delta C$  are the amounts of change in plaintext  $P$  and ciphertext  $C$ , respectively, where (+) indicates a bitwise exclusive OR.

$$DP_{\max} = \max_{\Delta P \neq 0, \Delta C} \text{Prob} \{F(P + \Delta P) + F(P) = \Delta C\}$$

..... Eq. 1

However, for any given encryption algorithm the accurate calculation of the  $DP_{\max}$  value is extremely difficult because of the computational complexity involved. Given this, instead of performing the calculation, the algorithm  $C = F(P)$  is broken down into small component functions  $F_1, F_2, F_3, \dots$ , in the form of  $C = F_n(\dots(F_2(F_1(P))))$ , where generally the maximum differential characteristic probability as defined in Eq. 2 is used as the indicator for the strength against the differential cryptanalysis method.

$$DP'_{\max} = \max \Pi \text{Prob}\{F_i(P_i + \Delta P_i) + F_i(P_i) = \Delta P_{i+1}\}$$

..... Eq. 2

On the other hand, the number of plaintext and ciphertext pairs required for success with differential cryptanalysis is inversely proportional to the maximum differential characteristic probability, where the smaller this probability value, the more secure the encryption.

LINEAR CRYPTANALYSIS AND LINEAR CHARACTERISTIC PROBABILITIES. Linear cryptanalysis was proposed by the corporation in 1993, based on the principle that it is possible to characterize statistically the relationship between plaintexts, ciphertexts, and the bits in the key, where the characteristics can be used to guess key information. The strength of a cipher against linear cryptanalysis is expressed in terms of the maximum average linear probability as defined in Eq. 3, where the smaller the maximum average linear probability, the more secure the encryption.

Here  $\Gamma P$  and  $\Gamma C (\neq 0)$  indicate the mask values of the plaintext  $P$  and the ciphertext  $C$ , respectively, and  $(\cdot)$  indicates the parity of the value that is calculated as the logical AND for each bit.

$$LP_{\max} = \max_{\Gamma C \neq 0, \Gamma P} |2 \cdot \text{Prob}\{P \cdot \Gamma P = C \cdot \Gamma C\} - 1|^2$$

..... Eq. 3

However, for any given encryption algorithm the accurate calculation of the  $LP_{\max}$  value is extremely difficult because of the computational complexity involved. Given this, instead of performing the calculation, the algorithm  $C = F(P)$  is broken down into small component functions  $F_1, F_2, F_3, \dots$ , in the form of  $C = F_n(\dots(F_2(F_1(P))))$ , where generally the maximum differential characteristic probability as defined in Eq. 4 is used as the indicator for the strength of the linear cryptanalysis.

$$LP'_{\max} = \max \Pi |2 \cdot \text{Prob}\{P_i \cdot \Gamma P_i = P_{i+1} \cdot \Gamma P_{i+1}\} - 1|^2$$

..... Eq. 4

On the other hand, the number of plaintext and ciphertext pairs required for success in linear cryptanalysis is inversely proportional to the maximum differential characteristic probability,

\*Toshio Tokita, Yasuyuki Sakai and Katsuyuki Takashima are with the Information Technology R&D Center.

where the smaller this probability value, the more secure the encryption.

**OTHER STRENGTH EVALUATIONS.** See reference [2] for other cryptanalytical methods not described here (for example, truncated differential cryptanalysis, higher-order cryptanalysis, etc).

There is also the case where a common-key block cipher is used in a mode such as OFB for random-number generation. Generally, an evaluation of randomness in such cases requires statistical methods, considering long-period characteristics, linear complexity, equal 0/1 frequency, etc.

**Public-Key Cryptosystem Strength Evaluations**

Generally, public-key cryptosystems are designed basing their security on the intractability of the following problems in number theory:

1. The integer-factorization problem
2. The finite-field discrete-logarithm problem
3. The elliptic-curve discrete-logarithm problem

Here, we call these “integer-factorization based public-key cryptosystems,” “discrete-logarithm based public-key cryptosystems” and “elliptic curve discrete-logarithm based public-key cryptosystems” and explain each of them below.

**INTEGER-FACTORIZATION BASED PUBLIC-KEY CRYPTOSYSTEM STRENGTH EVALUATIONS.** The RSA cryptosystem, which is the most common public-key scheme in use today, is founded on the security provided by the intractability of the integer-factorization problem. The most obvious way to attack the RSA cryptosystem is by factorizing the publicly known composite modulus  $n$  (the product of two distinct primes that are themselves private information). There is a considerable literature on factoring algorithms.

The running time of some factoring algorithms depends solely on the size of  $n$ . They include the quadratic-sieve method and the number-field sieve method.

In contrast, some algorithms are tailored to perform better when the composite modulus  $n$  is of a special type. The running times of such algorithms therefore typically depend on certain properties of the factors of  $n$ . They include Pollard’s rho method, the elliptic curve method, the  $p-1$  method, and the  $p+1$  method.

Although the difficulty of solving the integer-factorization problem typically increases with the size of the composite number (with the computational complexity increasing in an order termed sub-exponential time), when the com-

posite is the product of prime factors that have certain properties, then the factorization can be done quickly even if the modulus itself is large.

**DISCRETE-LOGARITHM BASED PUBLIC-KEY CRYPTOSYSTEM STRENGTH EVALUATIONS.** The discrete-logarithm problem is one that is used broadly in parallel with the factorization problem. For a given prime number  $p$ , a generator  $g$  of the multiplicative group of  $Z_p$ , and an element  $y$  of the multiplicative group of  $Z_p$ , the problem is to find an integer  $x$  such that  $y \equiv g^x \pmod{p}$ .

Here,  $x$  is known as the base- $p$  discrete logarithm. For example,  $x$  is 4 when  $4 \equiv 3^x \pmod{7}$ . Although logarithm calculations for real numbers are easy, the logarithm calculations for base  $p$ , or in other words, logarithmic calculations in a discrete domain are difficult when  $p$  is large.

The ElGamal cryptosystem is typical of those that base their security on the intractability of this problem. Additionally, even in the most common key-agreement protocol, the Diffie-Hellman (DH) protocol, advantage is taken of the fact that the discrete-logarithm problem is difficult to solve. (Note that in the DH key-agreement protocol, security is based on the DH assumption, which is a stronger assumption than the discrete-logarithm assumption.)

One possible attack on the ElGamal cryptosystem and the DH key-agreement protocol is to calculate the private information  $x$  from the public information  $y$ ,  $g$ , and  $p$ . Such methods include Pollar’s rho method, the Pohlig-Hellman method, the index-calculus method, and the number-field sieve method. The discrete-logarithm problem, like the factorization problem, is generally more difficult to solve when the various parameters are larger (that is, the number of calculations required increases in what is termed sub-exponential time); however, when parameters having specific characteristics are selected, the discrete-logarithm calculations can be done easily regardless of the size of the parameters.

**ELLIPTIC-CURVE DISCRETE-LOGARITHM PUBLIC-KEY CRYPTOSYSTEM STRENGTH EVALUATIONS.** Algorithms for RSA and ElGamal cryptosystems use integer (or more precisely, “finite field”) additions and multiplications. Similarly, the elliptic-curve cryptosystem is of the same type in that it uses addition on an elliptic curve. (In Fig. 1, point S is the result of adding point Q and point R.) The computational complexity problem of the elliptic curve discrete logarithm is formulated in terms of addition on an elliptic curve, and the elliptic-curve cryptosystem, as typified by the

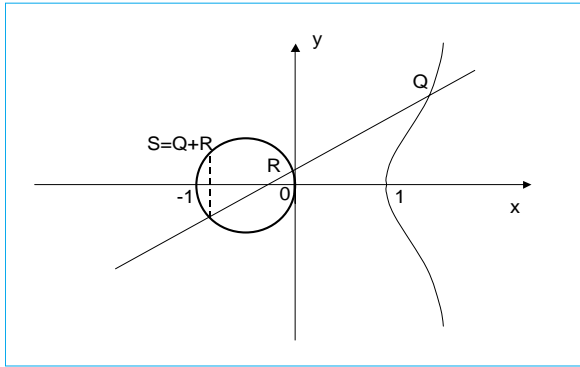


Fig. 1 Example of elliptic curve :  $y^2 = x^3 - x$

elliptic-curve ElGamal cryptosystem, bases its security on that difficulty. Generally no algorithms have been established by which to solve the elliptic-curve discrete-logarithm problem efficiently, but there are efficient cryptanalytic methods for specific elliptic curves.

Methods for attacking specific elliptic-curve cryptosystem parameters include the Pohlig-Hellman method, the MOV method, the FR method, and the Satoh-Araki-Semaev-Smart (SASS) method. In order to preserve sufficient security against the MOV method or the FR method, the MOV (FR) reduction degree of the elliptic curve must be large. Additionally, in order to be secure against the SASS method, the trace of the Frobenius endomorphism on the elliptic curve must not be one. Furthermore, in order to be secure from the Pohlig-Hellman method, the number of rational points on the elliptic curve must be a (pseudo-) prime. Establishing the parameters of the elliptic-curve cryptosystem to fulfill these criteria requires the

number of rational points on the elliptic curve to be calculated with some ingenuity. The so-called SEA method, which places no restrictions on the characteristics of finite fields, has been well established for such calculations for over eight years. On the other hand, the recently-established Satoh method is particularly useful when the characteristic is small, and the Skjernaa, FGH and AGM methods are improvements of it.

These fast algorithms are based on number theory, and the search for improved algorithms continues. Recently the Weil-descent method (and in particular, the GHS method) has been discovered as one method of cryptographic attack, and as a result, the current recommendation for protection is to use a group of rational points on an elliptic curve, whose values of  $(x, y)$  coordinates are in a prime degree extension field over prime field.

**Cipher-Performance Analysis Software**

This section will discuss the cipher-performance analysis software developed by Mitsubishi Electric. This is divided into software for common-key and public-key cryptosystems, and evaluates the strength of a cipher against any of the methods discussed above.

COMMON-KEY CRYPTOSYSTEM PERFORMANCE EVALUATION SOFTWARE. Fig. 2 shows the structure of an approach to common-key cryptosystem performance analysis software. Summaries will be given for each of the software components comprising the common-key cryptosystem performance evaluation software.

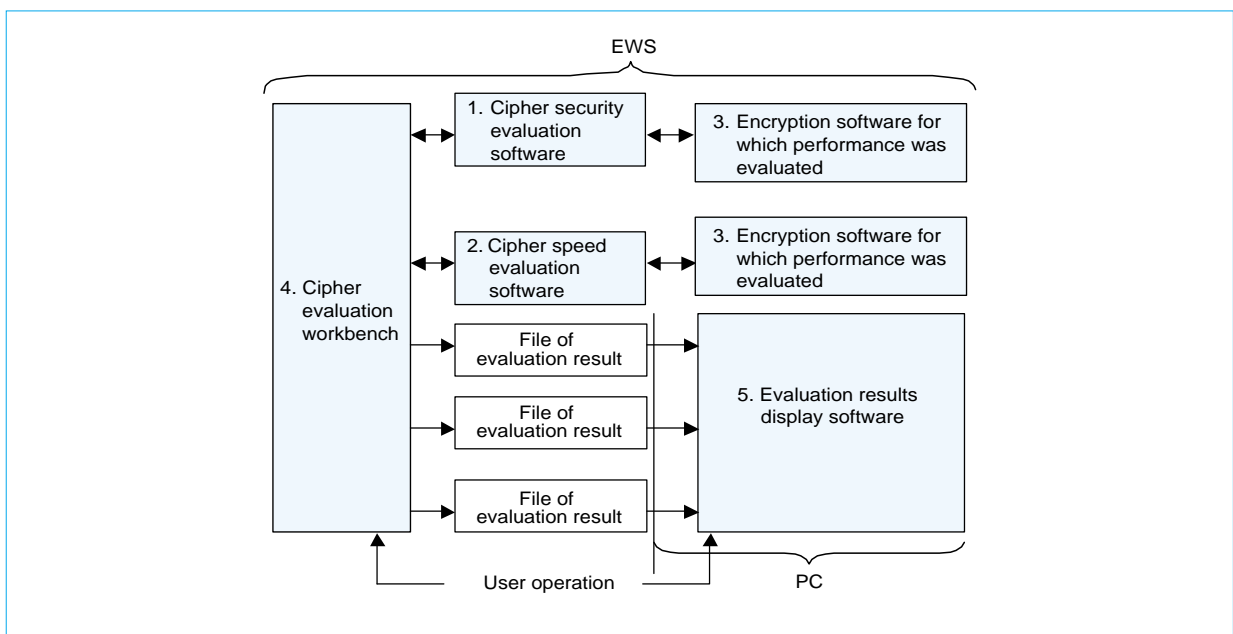


Fig. 2 Evaluation software for common-key cryptosystem (outline)

1. Cipher security-evaluation software  
This software evaluates the differential characteristic probabilities and the linear characteristic probabilities of the cipher to be evaluated. Additionally, when a pseudo-random number generator is used, it investigates frequency tests as a measure of randomness, and investigates collision tests and linear complexity.
2. Cipher speed-evaluation software  
This evaluates the processing speed (as an absolute value) of the encryption/decryption processes on a specific platform for the algorithm to be evaluated. Virtual platforms can also be designated and the processing speeds will be evaluated as a relative speed, relative to the virtual platform
3. Encryption software for which performance was evaluated  
The encryption algorithms for the ciphers to be evaluated had functions for absolute speed evaluations and security evaluations for AES (Rijndael), Serpent, CAST-256, and Twofish, and functions that were subjected to relative speed evaluations.
4. Cipher-evaluation workbench  
GUI functions are used when setting the evaluation parameters.
5. Evaluation-results display software  
GUI functions are used when displaying the evaluation results.

**PUBLIC-KEY CRYPTOSYSTEM PERFORMANCE EVALUATION SOFTWARE.** Fig. 3 shows the structure of an approach to public-key cryptosystem performance analysis software. A brief summary of the various software components comprising

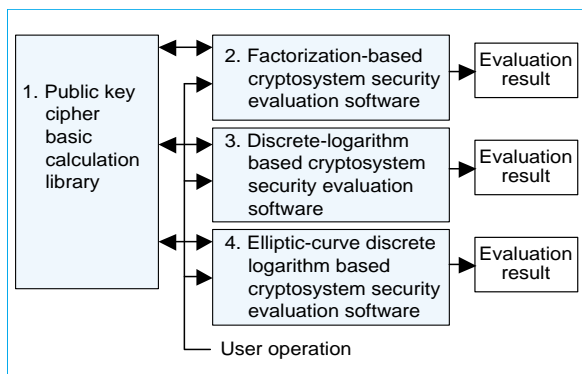


Fig. 3 Evaluation software for public-key cryptosystem (outline)

the public-key cryptosystem security evaluation software is presented below.

1. Public-key cryptosystem basic calculation library  
This is a library of basic calculations for public-key cryptosystems.
2. Security evaluation software for integer-factorization based cryptosystems  
This performs security analysis for integer-factorization based public-key cryptosystems using the quadratic-sieve and elliptic-curve methods of factoring.
3. Security-evaluation software for discrete-logarithm based cryptosystems  
This performs security analysis for discrete-logarithm based public-key cryptosystems using the Pohlig-Hellman and index-calculus methods.
4. Security-evaluation software for elliptic-curve discrete-logarithm based cryptosystems  
This performs security analysis for elliptic-curve discrete-logarithm based public-key cryptosystems based on counting the number of rational points on an elliptic curve (SEA method), on the trace of the Frobenius endomorphism on the elliptic curve, and on the MOV conditions.

Cipher-evaluation technologies are progressing daily based on innovations in cryptanalysis methods. As a result, the cipher-strength evaluation software described in this paper will need to undergo enhancements to handle each new innovation. □

**Note:** The cipher performance evaluation software described in this paper includes results from the Cipher Strength Evaluation Technology Development Project by the Department for the Development of Fundamental Technologies for the Next-Generation Digital Industry, of the Information-Technology Promotion Agency, Japan.

**References:**

- [1] Information-Technology Promotion Agency, Japan Security Center: Cipher Technology Evaluation Report: Cryptrec Report 2000 (2001-3)
- [2] Communications Broadcast Mechanisms: Common key Block Cipher Selection/Design/Evaluation Documents (2000-6)
- [3] Cohen, H. : A Course in Computational Algebraic Number Theory, Springer-Verlag (1993)
- [4] Shin'ichi Amada, et al.: Cipher Performance Evaluation Software Development, SCIS2000-A51 (2000-1)