Elastic Block Ciphers: The Basic Design

Debra Cook[*]
Bell Labs
Murray Hill, NJ, USA
dcook@bell-labs.com

Angelos Keromytis[†]
Columbia University
Dept. of Computer Science
New York, NY, USA
angelos@cs.columbia.edu

Moti Yung
RSA Labs, EMC Corp, and
Columbia University
Dept. of Computer Science
moti@cs.columbia.edu

ABSTRACT

We introduce the concept of an *elastic block cipher,* which refers to stretching the supported block size of a block cipher to any length up to twice the original block size while incurring a computational workload that is proportional to the block size. We define a method for converting any existing block cipher into an elastic block cipher and mention our analysis of the construction.

Categories and Subject Descriptors

E.0 [**General**]: Data Encryption

General Terms

block ciphers, algorithms, encryption

Keywords

variable-length block ciphers, elastic block ciphers

1. INTRODUCTION

Standard block ciphers are designed around one or a small number of block sizes, with most supporting 128-bit blocks. In applications, the length of the data to be encrypted is often not a multiple of the supported block size. This results in the use of plaintext-padding schemes that impose computational and space overheads by appending bits to the data. When the data being encrypted is relatively small, the padding can become a significant portion of the encrypted data. For example, encrypting a database at the field or row level to allow for efficient querying can easily result in a substantial amount of padding. When the plaintext is between one and two blocks, an elastic block cipher allows all of the bits to be encrypted as a single block, avoiding the need to use a mode of encryption and creating a stronger binding across the ciphertext bits compared to the ciphertext produced by a mode of encryption, such as CBC.

We introduce the concept of an *elastic block cipher,* which allows us to ”stretch” the supported block size of a block cipher up to a length double the original block size, while increasing the computational workload proportionally to the block size. This, together with modes of operation, permits block sizes to be set based on an application's requirements, allowing, for example, a non-traditional block size to be used for all blocks, or a traditional block size to be used for all but the last block in a given mode of operation. We propose a general method for creating an elastic block cipher from an existing block cipher. Our intent is not to design a new *ad-hoc* cipher, but to systematically build upon existing block ciphers. Our method consists of a network structure that uses the round function from an existing block cipher, allowing us to treat the round function of the original cipher as a black box and reuse its properties. This results in the security of the elastic version of a cipher being directly related to that of the original cipher.

Previous proposals for converting existing block ciphers into variable-length ones focused on treating a block cipher as a black box and combining it with other operations [2, 8]. While such an approach allows the security of the variable-length block cipher to be defined in terms of original block cipher, the resulting constructions require multiple applications of the original block cipher, making them computationally inefficient compared to padding. These methods may be valuable in providing modes of encryption that preserve the length of the data but they do not address how to design block ciphers to support variable-length blocks. There have also been ad-hoc attempts to design a variable-length block cipher from scratch [9, 11]. Ciphertext stealing is a way of preserving the length of the data when using a mode of encryption (as opposed to having the block cipher support a range of block sizes). It involves padding the partial plaintext block using ciphertext from the previous block and treats the partial block as a full block instead of adjusting the computational work to the actual number of bits. Furthermore, the partial block must be decrypted before its preceding block.

2. METHOD

2.1 Notation and Definitions

We use the following notation and definitions when describing the construction of elastic block ciphers.

- G denotes any existing block cipher with a fixed-length block size that is structured as a sequence of rounds. By default, any block cipher that is not structured as a

[*] This work was done at Columbia University.

[†] This work was partially supported by NSF Grants ITR CNS-04-26623 and CPA CCF-05-41093.

ASIACCS'07, March 20-22, 2007, Singapore.

sequence of rounds is viewed as having a single round.

- A cycle in G is the sequence of steps in which all b bits have been processed by the round function. For example, in AES [7], the round function is a cycle. In a balanced Feistel network, a sequence of two applications of the round function, which processes $\frac{b}{2}$ bits in each application, is a cycle. In RC6 [10], the sequence of four applications of the round function is a cycle.
- r denotes the number of cycles in G.
- b denotes the block length of the input to G in bits.
- y is an integer in the range $[0, b]$.
- G' denotes the modified G with a $(b + y)$-bit input for any valid value of y. G' will be referred to as the elastic version of G.
- r' denotes the number of rounds in G'.
- A bit (position) input to a block cipher is called *active* in a round if the bit is input to the round function.
- The round function of G' is one cycle of G.

2.2 Construction

Our algorithm converts the encryption and decryption functions of existing block ciphers to accept blocks of size b to $2b$ bits, where b is the block size of the original block cipher. Our method uses a network structure, the elastic network shown in Figure 1, into which the cycle of the original block cipher is inserted. This allows the properties of the original block cipher's round function to be reused. The elastic network creates a permutation on $b + y$ bits from a round function that processes b bits, where $0 \leq y \leq b$.

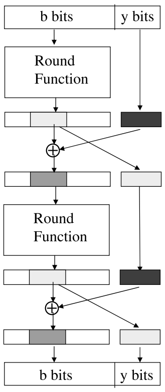

Figure 1: Elastic Network

The properties we require of the network structure are:

- It provides a permutation on $b + y$ bits for any $0 \leq y \leq b$ where b is the block size of the fixed-length block cipher.
- It is a single, generic, construction that can be used with any block cipher.
- The cycle of any existing b-bit block cipher becomes a component of the structure without any modification to it.
- The number of rounds is not set by the structure, but rather the round function can be applied as many times as needed by a specific cipher.
- The rate of diffusion for $b+y$ bits is defined in terms of the rate of diffusion for b bits in the fixed-length block cipher.
- The operations involved in the structure allow for efficient implementations in terms of time and memory requirements.

Our process of converting a fixed-length block cipher into an elastic block cipher involves inserting the cycle of the fixed-length block cipher into the elastic network, adding initial and final key-dependent permutations, adding or expanding initial and end-of-round whitening, and determining the number of rounds required. The general structure of the method is shown in Figure 2.

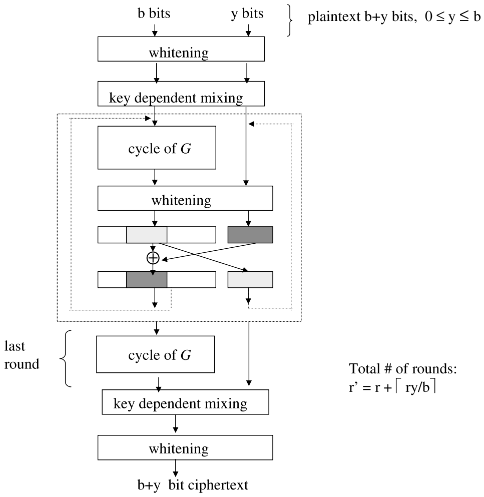

Figure 2: Elastic Block Cipher Structure

The following steps convert G with a fixed, b-bit, block size into its elastic version, G', that can process $b + y$ bits, for $0 \leq y \leq b$.

1. Set the number of rounds, r', such that each of the $b + y$ bits is input to and active in the same number of cycles in G' as each of the b bits is in G. $r' = r + \lceil \frac{ry}{b} \rceil$.
2. Apply initial and end-of-round whitening (XORing with expanded-key bits) to all $b+y$ bits. If G includes these whitening steps, the steps are modified to include all $b + y$ bits. If G does not have these whitening step, the steps are added when creating G'.
3. Prior to the first round and after the last round, apply a key-dependent mixing step that permutes or mixes the bits in a manner that any individual bit is not guaranteed to be in the rightmost y bits with a probability of 1. The leftmost b bits that are output from the initial mixing step are the input to the first round function. The initial mixing step is between the initial whitening and first round function. The final mixing step is after the last round function and prior to the final whitening.
4. Alternate which y bits are left out of the round function by XORing the y bits left out of the previous round function with y bits from the round function's output, then swap the result with the y bits left out of the previous round. This step is performed after the end of round whitening. Specifically:
   (a) Let Y denote the y bits that were left out of the round function.

(b) Let X denote some subset of y bits from the round function's output of b bits. A different set of X bits (in terms of position) is selected in each round.

(c) Set $Y \leftarrow X \oplus Y$.

(d) Swap X and Y to form the input to the next round.

This "swap step" may be added to the last round if it is required that all rounds be identical. However, having it after the last round does not provide additional security.

The result, G', is a permutation on $b + y$ bits. Its inverse, the decryption function, consists of the network applied in reverse and the round function replaced by its inverse.

The method is designed for G' to be equivalent to G, with the possible addition of whitening and the key-dependent mixing steps, when the data is an integral number of b-bit blocks, while accommodating a range of b to $2b$-bit blocks. We note that if complete diffusion (every bit impacting all other bits) occurs after q cycles in G then it occurs after at most $q + 1$ rounds in G'.

The elastic version of a block cipher requires a greater number of expanded-key bits than the fixed-length version. In our implementations, we used a stream cipher as the key schedule in order to significantly increase the randomness of the expanded key bits over those produced by existing key schedules, to allow as many expanded-key bits as needed to be produced without having to alter the key schedule for each block cipher, and to illustrate the concept of a standard key schedule that is independent of the specific block cipher. Using a stream cipher does incur a performance penalty over existing key schedules, but increasing the randomness of expanded-key bits aids in the prevention of attacks by reducing attacks due to the key schedule and decreasing the possibility that an attacker can obtain additional expanded-key bits after recovering only a few expanded-key bits.

3. SUMMARY OF RESULTS

We briefly mention here the main results in our analysis of elastic block ciphers. Detailed results are available in [3] and will be the subjects of future publications. The analysis justifies our choice of structure and steps for creating elastic block ciphers. In order to verify the security of our design we analyzed both the general approach and instantiations of elastic block ciphers.

First we employed a "reduction method" that exploits the elastic network structure and the fact that we used the round function of the original cipher as a black box. We are able to relate the security of elastic block ciphers in general against practical attacks to the security of the original ciphers against such attacks. We proved that the elastic version of a block cipher is secure against any attack that attempts to recover the key or expanded-key bits if the original cipher is secure against the attack. This result eliminates the need to analyze each elastic block cipher individually against practical attacks (such as linear and differential cryptanalysis) if the fixed-length versions are secure against such attacks.

Second, we considered specific attacks to provide a more concrete analysis. We proved that any algebraic equations relating the expanded-key, plaintext and ciphertext bits of the elastic version can be converted to equations for the fixed-length version in polynomial time and memory We show how the probability that a differential characteristic holds in the elastic version of a block cipher can be calculated using the probability a differential holds through one cycle of the original cipher, and applied the technique to elastic versions of AES and MISTY1 [5].

Third, by viewing the network in an ideal form (similar to the analysis performed by Luby and Rackoff on Feistel networks [4]) where the round functions are independently chosen pseudorandom permutations (PRP) on b bits, we prove that a three round elastic network is a PRP and a five round elastic network is a strong PRP on $b + y$ bits, $0 \leq y \leq b$.

Fourth, in order to demonstrate our method, we created four examples of elastic block ciphers from AES, Camellia [1], MISTY1 and RC6. For each example, we compared the performance of the elastic version to that of the original cipher with padding. We tested the randomness of the ciphertext in the four elastic examples using the statistical tests used by NIST in the AES competition [6]. The results indicate sufficient randomness in the ciphertext and no obvious design flaw in the elastic block ciphers. The support for variable-sized blocks also allows for new modes of encryption to be defined.

4. REFERENCES

[1] K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima, and T. Tokita. Camellia: A 128-Bit Block Cipher Suitable for Multiple Platforms - Design and Analysis. In *Proceedings of Selected Areas in Cryptography, LNCS 2012, Springer-Verlag*, pages 39–56, 2000.

[2] M. Bellare and P. Rogaway. On the Construction of Variable Length-Input Ciphers. In *Proceedings of Fast Software Encryption, LNCS 1636, Springer-Verlag*, 1999.

[3] D. Cook. Elastic Block Ciphers. Ph.D. Thesis, Columbia University, July 2006.

[4] M. Luby and C. Rackoff. How to Construct Pseudorandom Permutations from Pseudorandom Functions. *Siam Journal of Computing*, 17(2), April 1988.

[5] M. Matsui. Specification of MISTY1 - a 64-bit Block Cipher. Manuscript, Mitsubishi Electric Corporation, September 2000.

[6] NIST. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, NIST Special Publication 800-22. `csrc.nist.gov/publications/nistir`, 2001.

[7] NIST. FIPS 197 Advanced Encryption Standard (AES), 2001.

[8] S. Patel, Z. Ramzan, and G. Sundaram. Efficient Constructions of Variable-Input-Length Block Ciphers. In *Proceedings of Selected Areas in Cryptography 2004, LNCS 3357, Springer-Verlag*, 2004.

[9] J. Reeds. III,. Cryptosystem for Cellular Telephony. US Patent 5,159,634, 1992.

[10] Rivest, Robshaw, Sidney, and Yin. RC6 Block Cipher. `http://www.rsa.security.com/rsalabs/rc6`, 1998.

[11] R. Schroeppel. Hasty Pudding Cipher. `http://www.cs.arizona.edu/rcs/hpc`, 1998.