Single-Path Authenticated-Encryption Scheme Based on Universal Hashing

Soichi Furuya[1] and Kouichi Sakurai[2]

[1] Systems Development Laboratory, Hitachi, Ltd.,
292 Yoshida-cho, Totsuka-ku, Yokohama, 244-0817 Japan
soichi@sdl.hitachi.co.jp

[2] Dept. of Computer Science and Communications Engineering, Kyushu University,
6-10-1 Hakozaki, Higashi-ku, Fukuoka 812-8581 Japan
sakurai@csce.kyushu-u.ac.jp

Abstract. An authenticated-encryption scheme is frequently used to provide a communication both with confidentiality and integrity. For stream ciphers, i.e., an encryption scheme using a cryptographic pseudo-random-number generator, this objective can be achieved by the simple combination of encryption and MAC generation. This naive approach, however, introduces the following drawbacks; the implementation is likely to require two scans of the data, and independent keys for the encryption and MAC generations must be exchanged. The single-path construction of an authenticated-encryption scheme for a stream cipher is advantageous in these two aspects but non-trivial design.

In this paper we propose a single-path authenticated-encryption scheme with provable security. This scheme is based on one of the well-known ϵ-almost-universal hash functions, the evaluation hash. The encryption and decryption of the scheme can be calculated by single-path operation on a plaintext and a ciphertext. We analyze the security of the proposed scheme and give a security proof, which claims that the security of the proposed scheme can be reduced to that of an underlying PRNG in the *indistinguishability from random bits*. The security model we use, *real-or-random*, is one of the strongest notions amongst the four well-known notions for confidentiality, and an encryption scheme with *real-or-random* sense security can be efficiently reduced to the other three security notions. We also note that the security of the proposed scheme is tight.

Keywords: Stream cipher, mode of operation, provable security, message authentication, real-or-random security.

1 Introduction

A symmetric-key encryption is a cryptographic primitive that is mainly used to provide confidentiality to communicated (or stored) data against adversaries who do not possess the secret key. However, many cryptographic protocols implicitly use symmetric-key encryption to provide not only data confidentiality but also data integrity [R97].

K. Nyberg and H. Heys (Eds.): SAC 2002, LNCS 2595, pp. 94–109, 2003.

Although a secure communication in terms of data confidentiality can be achieved by using an encryption scheme, use of an encryption scheme does not always provide data integrity at the same time. Typically, data integrity is achieved by making use of an independent mechanism to generate a message authentication code (MAC).

A naive solution to achieve the two securities is a simple combination of two mechanisms, namely, an encryption and a MAC generation. However, we note that this simple approach does have some drawbacks. One is that the encoding and decoding mechanisms must manage two keys independently, e.g., random generation, exchange, storage, and discarding of the two keys. More importantly, there is another drawback in that the typical software implementation encoding and decoding processes are not single-path operation. This potential drawback is critical even for modern computers when they are dealing with, for example, streaming multimedia data. A construction to void those two drawbacks is not a trivial problem.

For the block cipher, there are reports of recent studies where an efficient mode of operation providing data authenticity as well as data confidentiality was provided [GD01, J01, RBBK01]. The modes in all cases demonstrated that they can provide the two securities independently if the underlying block cipher is treated as a pseudorandom permutation. On the other hand, there have been fewer reports on stream ciphers, i.e., an encryption scheme based on a secure key stream.

In this paper, we present an approach to construct an encryption scheme based on a key stream and analyze its security. Our main objectives of the construction are: *1.* An encryption scheme that operates with single-path calculation on a plaintext or ciphertext; and *2.* An encryption scheme using only one initialized key stream without compromising any security.

Our start point is a typical stream cipher; that is a bitwise xor operation of a key stream and a plaintext stream. The security of this scheme can be proven in terms of the confidentiality in the strongest sense. The information theoretic approach of Shannon's theorem proves this fact [S49]. For the computational approach, a similar technique that proves the security of CTR mode can be used to provide four major notions of confidentiality [BDJR97].

On the other hand, there is a well-known construction for MAC schemes that fits a stream cipher. We chose the Wegman-Carter construction [WC81] to embed the integrity mechanism into the Vernam cipher. The Wegman-Carter construction is a provably secure approach to the generation of a MAC, using universal hashing [CW79] and one-time paddness. Because there are a number of (almost) universal hash functions using a pseudorandom sequence, the adoption of an additional Wegman-Carter's MAC mechanism to a stream cipher looks less expensive[1].

The design of universal hashing is important not only for MAC generation but also for operations on databases. There are many reports of extensive re-

[1] Golić proposed primitive-converting constructions based on a PRNG [G00]. These are rather theoretical works and less efficient than dedicated universal hash functions.

search on the construction of a universal hashing. The evaluation hash analyzed by Shoup [S96] has piqued our interest for three reasons. *R1:* The required length of random bits is constant and short. *R2:* All the operations used in the evaluation hash are invertible, the hash achieves sufficient performance both in software and hardware implementations. *R3:* Security reduction is very small with comparison to the hash length. These characteristics play important roles in making the proposed scheme practical. Because of *R1*, the output length of the PRNG required for encryption (or decryption) can be reduced by a factor of two in comparison with the most inefficient construction, such as the combination of a stream cipher and MMH [HK97]. *R2* enables the proposed scheme to become a single-path operation both for encryption and decryption. The security bound of *R3* is critically important for the security bound of the proposed scheme.

In the latter part of this paper, we analyze the security of a scheme based on the studied construction. We mainly use the security notion in terms of *real-or-random* sense [BDJR97]. The reason we use this notion is that the notion can be efficiently reduced to three other well-known notions. This means that a scheme with a provable security of the *real-or-random* sense can be said to be as secure as the currently known strongest schemes.

The proof consists of two independent parts, i.e., security proofs for data confidentiality and data authenticity. In both proofs, we use *indistinguishability of PRN from random bits.* This notion of security can be also efficiently reduced to other notions of security, *left-or-right* sense, *semantic* security, and *find-then-guess* security [BDJR97].

There are known results on single-path authenticated-encryption schemes. Especially for the modes of block ciphers, there are designs and security analyses on XCBC [GD01], XIAPM [J01] and OCB [RBBK01]. As for stream ciphers, there are also results of authenticated-encryption schemes. Taylor's work [T93] is one of the practical message authentication mechanisms that is based on a PRNG. Although Taylor also describes the enhancement of his MAC scheme to achieve both confidentiality and integrity in [T93], the required additional length pseudorandom sequence over Vernam cipher is not a constant. Therefore, for a longer message the additional cost to a Vernam cipher cannot be negligible.

The chain and sum primitive [JV98] is another efficient scheme to achieve authenticity with an encryption based only on a PRNG. Moreover, the additional pseudorandom number consumption is constant. However, the scheme mandatorily requires a sequential two-path process. The scheme initially calculates the chain and sum of the plaintext, then it encrypts the Vernam cipher with a PRNG keyed by the resultant value of the chain and sum. This means that the intermediate value, which must not be disclosed for security, is as big as the size of message. Therefore, if this scheme deals a very long message, the same size of secure storage is required.

Many reports on the design of universal hashing are relevant to our past work and future problems. The MMH [HK97] and the Square hash [PR99] are efficient universal hashing schemes based on a PRNG. However, either of the two modes consumes a pseudorandom sequence increasingly proportional to a message's

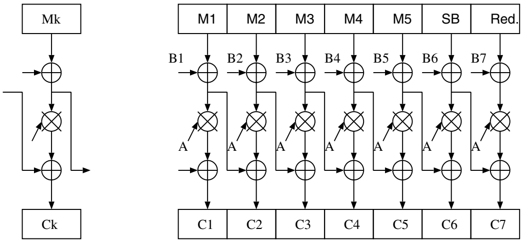

Fig. 1. Encryption block diagram (S01).

length. NMH universal hashing is used in UMAC [BHKKR99] and NMH is an extremely efficient universal hashing scheme. Also we note that whether or not the UMAC construction (instead of Wegman-Carter's) can be used to embed the message authenticity into a stream cipher is an open problem to us.

This paper is organized as follows: We introduce the studied encryption scheme S01 in Section 2 followed by two security discussions. The confidentiality is discussed in Section 3 and the integrity is in Section 4. In Section 5, summarize the feasibility of implementing the proposed scheme from the practical point of view. Finally, we offer our concluding remarks in Section 6.

2 Studied Model

Let n be a parameter that specifies the block length. The encryption takes two input data: an mn-bit message M and an n-bit redundancy R; using n-bit non-zero key stream and $(m+3)n$ key stream, the encryption generates $(m+2)n$-bit ciphertext. The decryption also takes two input data: an $c'n$-bit ciphertext C' and an n-bit redundancy R; using n-bit non-zero key stream and $(c'+1)n$-bit key stream, the decryption outputs either a forgery detection signal ϕ or $(c'-2)n$-bit message M'. If the length of a ciphertext does not change, $m+2=c'$.

For both encryption and decryption, three key streams are used, namely S_A, S_B, and S_P. The key streams are generated as follows:

Key stream generation:

S_A n-bit Non-zero key stream: Generate n-bit pseudorandom number. If the generated number is zero, discard it and generate n-bit random number again until n-bit non-zero key stream is obtained.

S_B $(m+2)n$-bit key stream: Generate $(m+2)n$-bit pseudorandom number following the S_A generation.

S_P n-bit key stream: Generate n-bit pseudorandom number following the S_B generation.

The encryption consists of three processes: data padding, data masking and data randomization. This is depicted in Fig. 1.

Encryption process

Padding Append S_P and R at the end of the message M. Therefore, the padded message M' is $(m+2)n$-bit length.

Masking Generate $F = M' \oplus S_B$.

Data randomization Divide F into n-bit blocks. Each block is multiplied by S_A over a finite field $\mathbf{F}2^n$. Each block of the resultant data is XORed by the previous block of F. Then concatenate all blocks to generate the ciphertext C.

We describe the mathematical description of S01 encryption scheme [FS01].

$$\begin{aligned}
M' &= M|S_P|R, \\
M' &= (M'_1, \ldots, M'_{m+2}), \\
F_0 &= 0, \\
F_i &= M_i \oplus S_{Bi}, && (1) \\
C_i &= (S_A \otimes F_i) \oplus F_{i-1}, && (2) \\
C &= C_1|C_2|\cdots|C_{m+2}.
\end{aligned}$$

For Equations (1) and (2), $1 \leq i \leq m+2$.

The decryption is a combination of the inverse of encryption and message authentication. We leave only its mathematical description.

$$\begin{aligned}
F_0 &= 0, \\
C' &= (C'_1, C'_2, \ldots, C'_{c'}), \\
F_i &= (C_i \oplus F_{i-1}) \otimes S_A^{-1}, && (3) \\
M'_i &= F_i \oplus S_{Bi}, && (4) \\
M' &= M'_1|M'_2|\cdots|M'_{c'-2}, \\
S'_P &= M'_{c'-1}, \\
R' &= M'_{c'}. && (5)
\end{aligned}$$

The last two blocks of M', i.e., S'_P and R', are used to check the integrity of the ciphertext. If and only if $S'_P = S''_P$ and $R' = R$, the decryption outputs M'. Otherwise output the forgery detection signal ϕ.

3 Confidentiality

In this section, we prove the confidentiality of the S01 scheme. To discuss the security, we have to determine the notion of confidentiality. There are four major notions of confidentiality in the symmetric-key setting. We study the confidentiality in the *real-or-random* sense, since the security in the *real-or-random* sense can be efficiently reduced to that of the other three notions. That means that the *real-or-random* sense is one of the strongest security notions.

Real-or-random setting The encryption oracle tosses a coin ($= \{0, 1\}$)to decide which game to play. There are two games in this setting. In Game 1, in response to an input message M, the oracle (ignores the message and) generates a secret random string with the same length to M and encrypts it under a randomly-chosen key $\mathcal{K}$. In Game 2, in response to an input message M, the oracle encrypts M under a randomly-chosen key $\mathcal{K}$. The adversary is a deterministic algorithm. He is allowed to generate a message and make oracle calls so that the adversary obtains the oracle outputs. Based on the knowledge of these oracle queries and oracle outputs, the adversary outputs one bit value $J^{rr} = \{0, 1\}$. The advantage of the adversary is defined as

$$\mathrm{Adv}^{rr} \overset{\mathrm{def}}{=} \Big|\Pr(J^{rr} = 1|\text{Game 1}) - \Pr(J^{rr} = 1|\text{Game 2})\Big|.$$

We call that encryption scheme $(t, \mu; \epsilon)$-secure if $\mathrm{Adv}^{rr} \leq \epsilon$ holds for any adversary with computational time $t_A \leq t$ and total oracle-query length $\mu_A \leq \mu$. Following these definitions we introduce the main theorem concerning the security of the S01 scheme.

Theorem 1. *A pseudorandom number generator (PRNG)π is $(t_\pi, \mu_\pi; \epsilon_\pi)$-secure in the indistinguishability from random bits, i.e.,*

$$\mathrm{Adv}^{rr}_{\pi} = \Pr(J^{rr}_{\pi} = 1|\mathrm{Game}(\pi)) - \Pr(J^{rr}_{\pi} = 1|\mathrm{Game}(\$)) \leq \epsilon_\pi,$$

with computational time t_π and amount of query μ_π, where Game(π) *and* Game($) *are games with the oracle outputting the π sequence and random sequence, respectively.*

The encryption scheme S01 with π, i.e., S01$_\pi$ is $(t_{S01\cdot\pi}, \mu_{S01\cdot\pi}; \epsilon_{S01\cdot\pi})$ secure in the real-or-random sense, where $(t_{S01\cdot\pi}, \mu_{t_{S01\cdot\pi}}; \epsilon_{S01\cdot\pi}) = (t_\pi - c_1\mu_\pi + c_2, \mu_\pi - c_1c_2, \epsilon_\pi)$.

Proof: We prove this through contradiction. Assume that an adversary $A^{rr}_{S01\cdot\pi}$ can $(t_{S01\cdot\pi}, \mu_{S01\cdot\pi}; \epsilon_{S01\cdot\pi})$-break $S01_\pi$ in the real-or-random sense. We construct a new adversary A^{rr}_{π} that can $(t_\pi, \mu_\pi; \epsilon_\pi)$-break π in the indistinguishability from random bits.

Let $\mathcal{O}_\pi(\cdot)$ be A^{rr}_{π}'s oracle. A^{rr}_{π} will run $A^{rr}_{S01\cdot\pi}$, using $\mathcal{O}_\pi(\cdot)$ to provide an appropriate simulation of $A^{rr}_{S01\cdot\pi}$'s oracle $\mathcal{O}_{S01\cdot\pi}(\cdot)$ as indicated below.

Algorithm A^{rr}_{π}

1. Invoke $A^{rr}_{S01\cdot\pi}$, and obtain $A^{rr}_{S01\cdot\pi}$'s outputs a message, M and a redundancy R,
2. (Gen. S_A) Obtain n-bit oracle output to generate S_A. If $S_A = 0$, repeat the generation until $S_A \neq 0$.
3. (Gen. S_B) Calculate the length of M (the number of n-bit blocks in M, m) and obtain $(m + 2)$-block (or equivalently $(m + 2)n$-bit) oracle output.
4. (Gen. S_P) Obtain n-bit oracle output to generate S_P.
5. Using S_A, S_B, S_P, and R, encrypt M to generate the ciphertext C.

6. Send C to $A^{rr}_{S01\cdot\pi}$.
7. Obtain $J^{rr}_{S01\cdot\pi} = \{0, 1\}$, the output of $A^{rr}_{S01\cdot\pi}$.
8. Output $J^{rr}_{\pi} = J^{rr}_{S01\cdot\pi}$, and terminate this run.

Let $\Pr[E]$ stand for the probability of event E occuring. Then, $\mathrm{Adv}^{rr}_{A_{S01\cdot\pi}}$ is defined as follows:

$$\begin{aligned}\mathrm{Adv}^{rr}_{A_{S01\cdot\pi}} &= \Pr[J^{rr}_{S01\cdot\pi} = 1|\mathrm{Game}(S01\cdot\pi)]\\ &- \Pr[J^{rr}_{S01\cdot\pi} = 1|\mathrm{Game}(\$)]\\ &\geq \epsilon_{S01\cdot\pi}.\end{aligned}$$

$\mathcal{O}_\pi(\cdot)$ outputs either (Game π) the output sequence of π, or (Game \$) the random sequence.

We now compute A^{rr}_π's advantage, $\mathrm{Adv}^{rr}_{A_\pi}$.

$$\begin{aligned}\mathrm{Adv}^{rr}_{A_\pi} &= \Pr[J^{rr}_\pi = 1|\mathrm{Game}(\pi)]\\ &- \Pr[J^{rr}_\pi = 1|\mathrm{Game}(\$)]\\ &= \Pr[J^{rr}_{S01\cdot\pi} = 1|\mathrm{Game}(S01\cdot\pi)]\\ &- \Pr[J^{rr}_\pi = 1|\mathrm{Game}(\$)] \qquad (6)\\ &= \Pr[J^{rr}_{S01\cdot\pi} = 1|\mathrm{Game}(S01\cdot\pi)]\\ &- \Pr[J^{rr}_{S01\cdot\pi} = 1|\mathrm{Game}(\$)] \qquad (7)\\ &\geq \epsilon_{S01\cdot\pi}.\end{aligned}$$

Equations (6) and (7) are obtained because of Lemma 1 and 2. Therefore, we have

$$\begin{aligned}t_\pi &= t_{S01\cdot\pi} + c_1 \cdot \mu_{S01\cdot\pi},\\ \mu_\pi &= \mu_{S01\cdot\pi} + 4n + \frac{n}{2^n} + \frac{n}{2^{2n}} + \cdots\\ &= \mu_{S01\cdot\pi} + c_2,\\ \epsilon_\pi &= \epsilon_{S01\cdot\pi}.\end{aligned}$$

where $c_2 = 3n + \frac{n}{1-2^{-n}}$. Therefore, we solve $\mu_{S01\cdot\pi}$, $t_{S01\cdot\pi}$, and $\epsilon_{S01\cdot\pi}$.

$$\begin{aligned}\mu_{S01\cdot\pi} &= \mu_\pi - c_2,\\ t_{S01\cdot\pi} &= t_\pi - c_1 \cdot \mu_{S01\cdot\pi}\\ &= t_\pi - c_1 \cdot (\mu_\pi - c_2)\\ &= t_\pi - c_1 \cdot \mu_\pi + c_1 c_2,\\ \epsilon_{S01\cdot\pi} &= \epsilon_\pi.\end{aligned}$$

Therefore, A^{rr}_π can $(t_{S01\cdot\pi}, \mu_{S01\cdot\pi}; \epsilon_{S01\cdot\pi})$-break π in the indistinguishability from random bits. □

Lemma 1. *In the proof of Theorem 1, the following equation holds.*

$$\Pr[J^{rr}_\pi = 1|\mathrm{Game}(\pi)] = \Pr[J^{rr}_{S01\cdot\pi} = 1|\mathrm{Game}(S01\cdot\pi)].$$

In other words, in Game(π)*, i.e., a game with* $\mathcal{O}_\pi$ *outputting* π *sequence, the probability that* A^{rr}_π *outputs* $J^{rr}_\pi = 1$ *is equal to the probability that* $A^{rr}_{S01\cdot\pi}$ *outputs* $J^{rr}_{S01\cdot\pi} = 1$ *in* Game(π)*.*

Proof: In the game with $\mathcal{O}_\pi$ outputting π sequence, $A^{rr}_{S01\cdot\pi}$ (invoked by A^{rr}_π) receives a ciphertext C encrypted by $S01 \cdot \pi$. Therefore the operations of $A^{rr}_{S01\cdot\pi}$ are exactly the same as $A^{rr}_{S01\cdot\pi}$ playing in Game ($S01 \cdot \pi$). Because A^{rr}_π always outputs $J^{rr}_{S01\cdot\pi}$ such that $J^{rr}_{S01\cdot\pi} = J^{rr}_\pi$,

$$\Pr[J^{rr}_\pi = 1|\text{Game}(\pi)] = \Pr[J^{rr}_{S01\cdot\pi} = 1|\text{Game}(S01 \cdot \pi)].$$

□

Lemma 2. *In the proof of Theorem 1, the following equation holds.*

$$\Pr[J^{rr}_\pi = 1|\text{Game}(\$)] = \Pr[J^{rr}_{S01\cdot\pi} = 1|\text{Game}(\$)]. \tag{8}$$

In other words, in Game($\$$)*, i.e., a game with* $\mathcal{O}_\pi$ *outputting a random sequence, the probability that* A^{rr}_π *outputs* $J^{rr}_\pi = 1$ *is equal to the probability that* $A^{rr}_{S01\cdot\pi}$ *outputs* $J^{rr}_{S01\cdot\pi} = 1$ *in* Game(π)*.*

Proof: At first we evaluate the left-hand-side term of Equation (8), i.e., $\Pr[J^{rr}_\pi = 1|\text{Game}(\$)]$. As we mentioned in the proof of Lemma 1, the relation $J^{rr}_\pi = J^{rr}_{S01\cdot\pi}$ always holds. Therefore,

$$\Pr[J^{rr}_\pi = 1|\text{Game}(\$)] = \Pr[J^{rr}_{S01\cdot\pi} = 1|\text{Game}(S01 \cdot \$)]. \tag{9}$$

Note that $A^{rr}_{S01\cdot\pi}$ is a deterministic algorithm that generates a message M and output $J_{S01\cdot\pi}$ =0(random number)/1(ciphertext) in response to a ciphertext C. There exist a function $f_{A^{rr}_{S01\cdot\pi}}$ such that for $\forall(P, C)$, $f_{A^{rr}_{S01\cdot\pi}}(P, C) = J^{rr}_{S01\cdot\pi}$. Let $\mathbb{C}_{A_{S01\cdot\pi},0} = \{C : f_{A^{rr}_{S01\cdot\pi}}(C) = 0\}$ and we have following equations.

$$\begin{aligned}\Pr(J^{rr}_\pi = 0|\text{Game}(\$)) &= \sum_{C\in\mathbb{C}} \Pr(C)\\ &= \sum_{C\in\mathbb{C}} \frac{1}{2^{|C|}}\\ &= \frac{|\mathbb{C}|}{2^{|C|}}.\end{aligned} \tag{10}$$

Because of the uniformity of a random number as the input C, Equation (10) can be evaluated as $\Pr(C) = \frac{1}{2^{|C|}}$.

We now evaluate the right-hand-side term of Equation (8), i.e., $\Pr[J^{rr}_{S01\cdot\pi} = 1|\text{Game}(\$)]$.

$$\begin{aligned}\Pr(J^{rr}_{S01\cdot\pi} = 0|\text{Game}(\$)) &= \sum_{C\in\mathbb{C}} \Pr(C)\\ &= \sum_{C\in\mathbb{C}} \frac{1}{2^{|C|}}\\ &= \frac{|\mathbb{C}|}{2^{|C|}}.\end{aligned} \tag{11}$$

Because of the uniformity of the ciphertext that $A^{rr}_{S01\cdot\pi}$ receives (stated and proven in Proposition 1), Equation (11) is obtained.

Hence, Equation (9) holds and the following relation also holds.

$$\Pr[J^{rr}_{\pi}=1|\mathrm{Game}(\$)]=\Pr[J^{rr}_{S01\cdot\pi}=1|\mathrm{Game}(\$)].$$

□

We now show Proposition 1 and its proof.

Proposition 1. *Uniformity of ciphertext of* $S01\cdot\$$*: If the scheme* $S01\cdot\$$ *uses random number, ciphertexts distributes uniformly. The distribution is independent of a message.*

Proof: Let M' be the padded message. More specifically a secret random padding S_P and redundancy data R are appended to M in order to generate M'. Because of Proposition 2 shown in the latter part of the paper, for an arbitrary M', the number of key streams (S_A, S_B) that maps a message M' on to a ciphertext C, is 2^n-1, which is independent of M'. Therefore, for an arbitrary M' the probability that the corresponding ciphertext coincides with C is $(2^n-1)/2^{mn}$ (over possible key streams).

Remember that the way to generate M' out of M is independent of (S_A, S_B). Hence, for an arbitrary message M, M's ciphertext distributes uniformly, as well.

□

Proposition 2. *For a arbitrary* (M', C) *pair there are exactly* 2^n-1 *key streams of* (S_A, S_B) *that maps a message* M' *to a ciphertext* C*. Moreover each key-stream candidate for a fixed* (M', C) *has distinct* S_A *value. Hence,* S_A *value cannot be determined only from a* (M', C)*.*

Proof: From Equations (1) and (2), we solve S_{Bi}'s recursively.

$$S_{B1}=(C_1\oplus F_0)\otimes S_A^{-1}\oplus M'_1, \tag{12}$$
$$S_{Bi}=(C_i\oplus M'_{i-1}\oplus S_{Bi-1})\otimes S_A^{-1}\oplus M'_i. \tag{13}$$

For an arbitrary S_A, S_B sequence is uniquely determined. Therefore, (S_A, S_B) pairs that map M' to C exist at least as many as S_A's, i.e., 2^n-1.

We then fix S_A value and prove that only one S_B sequence maps M' to C. In total we conclude there exist (2^n-1) key streams (S_A, S_B) that map M' to C.

We assume two different key streams, (S_A, S'_B) and (S_A, S''_B). Note that these two share common S_A value. When these two key streams encrypt M',

$$\begin{aligned}F'_{i+1}&=P_i\oplus S'_{Bi},\\ C'_i&=(F'_{i+1}\otimes S_A)\oplus F'_i,\\ F''_{i+1}&=P_i\oplus S''_{Bi},\\ C''_i&=(F''_{i+1}\otimes S_A)\oplus F''_i.\end{aligned}$$

Let j be the least index where S'_B and S''_B differs. From Equations (1) and (2), we have $F'_j = F''_j$, $S'_B \neq S''_B$ and $F'_{j+1} \neq F''_{j+1}$. Therefore, $C'_j \neq C''_j$.

Hence, there is no message M' that is mapped to the same C under two different keys streams that shares the same S_A.

We therefore conclude that the number of key streams (S_A, S_B) that encrypts M' and generates ciphertext C is $2^n - 1$. □

4 Integrity

In this section we study security in terms of message integrity. We first define an model of the adversary. We assume the adversary that can choose a message and the redundancy and obtain the corresponding ciphertext. Because the studied model is a stream cipher, the key stream is used in the one-time-pad manner, or equivalently any part of the key stream is never re-used. An adversary forges a ciphertext out of the knowledge of a message, the redundancy and the corresponding ciphertext. The adversary is capable of altering even the length of the ciphertext. The aim of this study is to give an upperbound probability of a successful forgery. We give Theorem 2 about the security of the studied encryption scheme.

Theorem 2. *A pseudorandom number generator (PRNG)π is $(t_\pi, \mu_\pi; \epsilon_\pi)$-secure in the indistinguishability from random bits, i.e.,*

$$\mathrm{Adv}_\pi^{rr} = \Pr(J_\pi^{rr} = 1|\mathrm{Game}(\pi)) - \Pr(J_\pi^{rr} = 1|\mathrm{Game}(\$)) \leq \epsilon_\pi,$$

with computational time t_π and amount of query μ_π, where Game(π) *and* Game($) *are games with the oracle outputting the π sequence and random sequence, respectively.*

The encryption scheme S01 with π, S01$_\pi$ is $(t_{S01\cdot\pi}^{alter}, \mu_{S01\cdot\pi}^{alter}; p_{S01\cdot\pi}^{alter})$ secure against the adversary, i.e., an adversary cannot generate a successful forgery with computational time $(t_{S01\cdot\pi}^{alter}$ data complexity $\mu_{S01\cdot\pi}^{alter}$ and the upperbound of the successful probability $p_{S01\cdot\pi}^{alter}$, where $(t_{S01\cdot\pi}^{alter}, \mu_{S01\cdot\pi}^{alter}, p_{S01\cdot\pi}^{alter}) = (t_\pi - c_3 \cdot \mu_\pi + c_3 c_4, \mu_\pi - c_4, \epsilon_\pi + (m+1)/(2^n - 1))$.

Proof: We prove this through contradiction. Assume that an adversary $A_{S01\cdot\pi}^{alter}$ can $(t_{S01\cdot\pi}^{alter}, \mu_{S01\cdot\pi}^{alter}; p_{S01\cdot\pi}^{alter})$-forge $S01_\pi$. We construct a new adversary A_π^{rr} that can $(t_\pi^{rr}, \mu_\pi^{rr}; \epsilon_\pi^{rr})$-break π in the indistinguishability from random bits.

Let $\mathcal{O}_\pi(\cdot)$ be A_π^{rr}'s oracle. A_π^{rr} will run $A_{S01\cdot\pi}^{alter}$, using $\mathcal{O}_\pi(\cdot)$ to provide an appropriate simulation of $A_{S01\cdot\pi}^{alter}$'s oracle $\mathcal{O}_{S01\cdot\pi}(\cdot)$. More specifically, $A_{S01\cdot\pi}^{alter}$ is a deterministic algorithm that generates a message and a redundancy. In response to the ciphertext C, or equivalently the output of the oracle $\mathcal{O}_{S01\cdot\pi}(\cdot)$, $A_{S01\cdot\pi}^{alter}$ outputs the forged ciphertext C', which is different from C. The probability that C' passes the integrity check of the decryption is $p_{S01\cdot\pi}^{alter}$.

We indicate the algorithm of the constructed adversary below.

Algorithm A_π^{rr}

1. Invoke $A_{S01\cdot\pi}$, and obtain $A_{S01\cdot\pi}$'s outputs. The output consists of a message M and a redundancy R,
2. (Gen. S_A) Obtain n-bit oracle output to generate S_A. If $S_A = 0$, repeat the generation until $S_A \neq 0$.
3. (Gen. S_B) Calculate the length of M (the number of n-bit blocks in M, m) and obtain $(m+2)$-block (or equivalently $(m+2)n$-bit) oracle output.
4. (Gen. S_P) Obtain n-bit oracle output to generate S_P.
5. Using S_A, S_B, S_P, and R, encrypt M to generate the ciphertext C.
6. Send C to $A_{S01\cdot\pi}^{alter}$.
7. Obtain C', the output of $A_{S01\cdot\pi}^{alter}$.
8. Check the validity of C'. If the decryption result is the forgery detection signal ϕ, output $J_\pi^{rr} = 0$. Otherwise, output $J_\pi^{rr} = 1$.
9. Terminate this run.

Let $\mathcal{O}_\pi(\cdot)$ be the oracle of A_π^{rr}. $\mathcal{O}_\pi(\cdot)$ outputs either (Game π) the output sequence of π, or (Game \$) the random sequence.

We now compute A_π^{rr}'s advantage, $\mathrm{Adv}_{A_\pi}^{rr}$.

$$\begin{aligned}\mathrm{Adv}_{A_\pi}^{rr} &= \Pr(J_\pi^{rr} = 1|\mathrm{Game}(\pi)) \\ &\quad - \Pr(J_\pi^{rr} = 1|\mathrm{Game}(\$)).\end{aligned}$$

Because of the following Lemma 3, the following equation holds.

$$\mathrm{Adv}_{A_\pi}^{rr} = p_{S01\cdot\pi}^{alter} - \Pr(J_\pi^{rr} = 1|\mathrm{Game}(\$)).$$

From Lemma 4, we have

$$\mathrm{Adv}_{A_\pi}^{rr} \geq p_{S01\cdot\pi}^{alter} - (m+1)/(2^n - 1).$$

Therefore, we can evaluate the cost and advantage of A_π^{rr} as follows:

$$\begin{aligned}t_\pi^{rr} &= t_{S01\cdot\pi}^{alter} + c_3 \cdot \mu_{S01\cdot\pi}^{alter} \\ \mu_\pi^{rr} &= \mu_{S01\cdot\pi}^{alter} + 4n + \frac{n}{2^n} + \frac{n}{2^{2n}} + \cdots \\ &= \mu_{S01\cdot\pi}^{alter} + c_4, \\ \epsilon_\pi &= p_{S01\cdot\pi}^{alter} - (m+1)/2^n,\end{aligned}$$

where $c_4 = 3n + \frac{n}{1-2^{-n}}$. We finally solve $t_{S01\cdot\pi}^{alter}, \mu_{S01\cdot\pi}^{alter}, p_{S01\cdot\pi}^{alter}$ and obtain the following relations:

$$\begin{aligned}\mu_{S01\cdot\pi}^{alter} &= \mu_\pi - c_4, \\ t_{S01\cdot\pi}^{alter} &= t_\pi - c_3 \cdot \mu_{S01\cdot\pi} \\ &= t_\pi - c_3 \cdot (\mu_\pi - c_4) \\ &= t_\pi - c_3 \cdot \mu_\pi + c_3 c_4, \\ p_{S01\cdot\pi}^{alter} &= \epsilon_\pi + (m+1)/2^n.\end{aligned}$$

□

Lemma 3. *In the proof of Theorem 2, the following equation holds.*

$$\Pr[J_\pi^{rr} = 1|\text{Game}(\pi)] = p_{S01\cdot\pi}^{alter}.$$

In other words, in Game(π)*, i.e., a game with* $\mathcal{O}_\pi$ *outputting* π *sequence, the probability that* A_π^{rr} *outputs* $J_\pi^{rr} = 1$ *is equal to the probability that* $A_{S01\cdot\pi}^{alter}$ *outputs a valid forged ciphertext in* Game(π)*.*

Proof: In the game with $\mathcal{O}_\pi$ outputting π sequence, $A_{S01\cdot\pi}^{alter}$ (invoked by A_π^{rr}) receives a ciphertext C encrypted by $S01 \cdot \pi$. Therefore, the operations of $A_{S01\cdot\pi}^{alter}$ are exactly the same as $A_{S01\cdot\pi}^{alter}$ playing in Game($S01 \cdot \pi$). Note that A_π^{rr} outputs $J_{S01\cdot\pi} = 1$ if and only if $A_{S01\cdot\pi}^{alter}$ outputs the successful forgery C'. Therefore, we have the following equation.

$$\Pr[J_\pi^{rr} = 1|\text{Game}(\pi)] = p_{S01\cdot\pi}^{alter}.$$

□

Lemma 4. *In the proof of Theorem 2, the following equation holds.*

$$\Pr[J_\pi^{rr} = 1|\text{Game}(\$)] \leq (m+1)/(2^n - 1).$$

In other words, in Game($)*, i.e., a game with* $\mathcal{O}_\pi$ *outputting a random sequence, the probability that* A_π^{rr} *outputs* $J_\pi^{rr} = 1$ *is at most* $(m+1)/(2^n - 1)$*.*

Proof: We first consider attacks without changing the length of the ciphertext (namely, the attack is limited only to changing the ciphertext value). The proofs for the cases of (1) eliminating the length of the ciphertext and (2) appending new ciphertext blocks will be discussed in the latter part of the proof.

Assuming that $A_{S01\cdot\pi}^{alter}$ with known-plaintext tries to alter ciphertext and the decryptor receives the maliciously altered ciphertext, C', instead of C. The one of the necessary objective of the attacker is to alter a message such that the recovered (modified) message has the same redundancy. Due to the algebraic structure of the scheme, a successful forgery must have special relation that includes an unknown S_A value. The proof of the S_A's uncertainty out of a known plaintext is given in Proposition 2.

We construct the necessary condition to match R. From Equations (3) and (4), the recovered message M' of the forged ciphertext is:

$$\begin{aligned} F'_1 &= F_1, \\ F'_i &= (C'_i \oplus F'_{i-1}) \otimes S_A^{-1}, \\ M'_i &= F'_i \oplus S_{Bi}. \end{aligned}$$

$A_{S01\cdot\pi}^{alter}$ also knows the original message, so that he has following equations:

$$\begin{aligned} F_1 &= F_1, \\ F_i &= (C_i \oplus F_{i-1}) \otimes S_A^{-1}, \\ M_i &= F_i \oplus S_{Bi}. \end{aligned}$$

From these equations, we solve R and R'. Consequently we have

$$R = M_{m+2} = S_{Bm+2} \oplus \bigoplus_{i=1}^{m+2} C_{m+3-i} S_A^{-i} \oplus F_0 S_A^{-(m+2)},$$

$$R' = M'_{m+2} = S_{Bm+2} \oplus \bigoplus_{i=1}^{m+2} C'_{m+3-i} S_A^{-i} \oplus F_0 S_A^{-(m+2)}.$$

The necessary condition is to hold $R = R'$. Then we have a condition.

$$0 = \delta_{m+2} S_A^{-1} \oplus \delta_{m+1} S_A^{-2} \oplus \delta_m S_A^{-3} \oplus \ldots \oplus \delta_2 S_A^{-(m+1)} \oplus \delta_1 S_A^{-(m+2)}. \qquad (14)$$

Remember that $A^{alter}_{S01\cdot\pi}$ is a deterministic algorithm and $A^{alter}_{S01\cdot\pi}$ determines δ without the knowledge of S_A (Proposition 2). Then for a fixed δ value, there exist at most $m+1$ roots for non-zero S_A of Equation (14). Because S_A is random and independent of the $A^{alter}_{S01\cdot\pi}$'s actions, the probability of a successful forgery is upperbounded by $(m+1)/(2^n - 1)$.

Eliminating ciphertext length: In this part of the proof, we concentrate on the case in which the ciphertext is shortened. Here we use $S_P = S'_P$ as a necessary condition for the successful forgery.

Let c' be the length of the forged ciphertext ($c' < m+2$). Then the decryptor will identify $M'_{c'-1}$ as S_P. More specifically,

$$S'_P = M'_{c'-1} = S_{Bc'-1} \oplus \bigoplus_{i=1}^{c'-1} C'_{c'-i} S_A^{-i} \oplus F_0 S_A^{-(c'-1)}. \qquad (15)$$

On the other hand, the decryption scheme generates S_P as the next pseudorandom block after $S_{Bc'}$. Therefore,

$$S_P = S_{Bc'+1} = M_{c'+1} \oplus \bigoplus_{i=1}^{c'+1} C'_{c'+2-i} S_A^{-i} \oplus F_0 S_A^{-(c'+1)}. \qquad (16)$$

From Equations (12), and (13), $S_{Bc'-1}$ can be expressed only by the public value and S_A as follows:

$$\begin{aligned} S_{Bc'-1} &= M_{c'-1} \oplus S_A^{-1} C_{c'-1} \oplus S_A^{-2} C_{c'-2} \oplus \cdots \oplus S_A^{-(c'-1)} C_1 \oplus F_0 S_A^{-(c'-1)} \\ &= M_{c'-1} \oplus F_0 S_A^{-(c'-1)} \oplus \bigoplus_{i=1}^{c'-1} C_{c'-i} S_A^{-i}. \end{aligned} \qquad (17)$$

From Equations (15), (16) and (17), the necessary condition for matching S_P and S'_P is expressible only with known values (M, C, C'), and an unknown independent value S_A as follows:

$$0 = M_{c'+1} \oplus M_{c'-1} \oplus F_0 S_A^{-(c'+1)} \oplus C'_2 S_A^{-c'} \oplus C'_1 S_A^{-(c'+1)} \oplus \bigoplus_{i=1}^{c'-1} (\delta_{c'-i} \oplus C_{c'+2-i}) S_A^{-i}. \qquad (18)$$

Being similar to the previous case of attack, $A^{alter}_{S01\cdot\pi}$ determines δ, c', and C' without the knowledge of S_A (Proposition 2). Then for a fixed set of (δ, c', C'), there exist at most $c' + 1$ roots for non-zero S_A of Equation (18). Because S_A is random and independent of the $A^{alter}_{S01\cdot\pi}$'s actions, the probability of a successful forgery is upperbounded by $(c' + 1)/(2^n - 1) \leq m/(2^n - 1)$.

Appending new ciphertext blocks: We briefly describe the proof of the case in which the adversary appends a new ciphertext block (in addition to changing the existing ciphertext blocks). Let c' be the length of the appended ciphertext $(c' > c)$.

In this case we consider the difficulty of controlling R'. From Equation (5), R' can be expressed by

$$R' = S_{Bc'} \oplus M_{c'} \oplus F_{c'}. \tag{19}$$

Note that what $A_{S01\cdot\pi}$ can control is limited to something about $F_{c'}$. $S_{Bc'}$ is a new value that has never appeared in the encryption process, i.e., $S_{Bc'}$ is independently random from any public value. Therefore, although he can control the actual value of $F_{c'}$, the probability that $R' = R$ is $1/2^n$. □

5 Implementation and Efficiency

In this section, we discuss the implementation and efficiency of the proposed scheme. For the practical parameters, we set the block length to be 64 bits for all evaluations.

In terms of software implementation, we implemented the proposed scheme together with the PANAMA stream cipher [DC98]. As a result, we have performances of 202 Mbps (encryption) and 207 Mbps (decryption) on a 600-MHz Alpha processor. The code is written in C-language and compiled by a DEC cc compiler.

The hardware suitability of our scheme is very high because of operations in $\mathbf{F}2^{64}$. We estimated additional hardware cost to the PRNG. Generally speaking, there are considerable trade-offs between performance and hardware size. Our evaluation demonstrates two instances: the maximum throughput model and the smallest gate size model. All estimations were evaluated with a 0.35-μm CMOS process.

For the maximum throughput, the multiplication is implemented with full logical expression, and is estimated by the size (the number of gates) and the propagation delay. As a result of an estimation with Verilog-HDL, the multiplication in $\mathbf{F}2^{64}$ is implemented in a 36-K gate and its propagation delay is 5.4 ns. This is an optimized circuit in the sense of the propagation delay. The circuit performs up to 150 MHz and encrypts a block (64-bit) with a clock. Thus, the maximum throughput is estimated to be 9.6 Gbps (when the PRNG performs fast enough).

Similarly we estimated the smallest gate count implementation, in which a multiplication in a finite field is realized very cheaply with a linear feedback shift register. we found that the additional circuit to the PRNG can be implemented with no more than 3 K gate. This circuit works with clocks up to 800 MHz. Since a block en(de)cryption takes 65 clocks, this smallest gate count instance practically performs about 200 Mbps at 200 MHz.

6 Concluding Remarks

We proposed an encryption scheme both for data confidentiality and data integrity that uses a $n(m+4)$-bit random number stream, where n is the length of a block and m is the number of blocks in a message. We proved the probability of a forgery in known-plaintext environments.

The proposed scheme is practical as we demonstrated implementations both on software and hardware. In particular, hardware suitability is very high since operations in the scheme are suitable for a hardware platform. Therefore, either the additional gate count is very small, or the maximum throughput can be very high with acceptable gate counts. Because of the possibility of parallel PRNG, the maximum throughput of 9.6 Gbps is a reasonably realistic estimation.

As for efficiency, the scheme has two more advantages. First of all, the scheme achieves single-path encryption scheme so that streaming data can be also dealt with using a limited hardware resource. Secondly, the PRNG is independent of either the intermediate value or the message. This means that parallel computation and precomputation are very easy and effective for increasing the maximum throughput.

An additional issue regarding security is that any part of an actual key stream value cannot be determined by an adversary so that the most likely target to crack, the PRNG, is unreachable directly from the attacker. This cannot be construed to mean any concrete additional security. However, in a cryptographic attack on a PRNG, the attacker must make use of the actual value, then the scheme itself may bring a certain hedge against attacks.

References

[BDJR97] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway, “A Concrete Security Treatment of Symmetric Encryption: Analysis of the DES Modes of Operation,” *Proceedings of the 38th Symposium on Foundations of Computer Science, IEEE*, 1997, full paper is available at `http://www-cse.ucsd.edu/users/mihir/`.

[BGV96] A. Bosselaers, R. Govaerts, and J. Vandewalle, “Fast Hashing on the Pentium,” *Advances in Cryptology, —CRYPTO’96, LNCS Vol. 1109, Springer-Verlag*, 1996.

[BKR94] M. Bellare, J. Kilian, and P. Rogaway, “The Security of Cipher Block Chaining,” *Advances in Cryptology, —CRYPTO’94, LNCS Vol. 839, Springer-Verlag*, 1994.

[BHKKR99] J. Black and S. Halevi, H. Krawczyk, T. Krovets, P. Rogaway, “UMAC: Fast and Secure Message Authentication,” *Advances in Cryptology, —CRYPTO’99, LNCS Vol. 1666, Springer-Verlag*, 1999.

[CW79] L. Carter and M. Wegman, “Universal Hash Functions,” *Journal of Computer and System Sciences, Vol. 18*, 1979.

[DC98] J. Daemen and C. Clapp, "Fast Hashing and Stream Encryption with PANAMA," *Fast Software Encryption, 5th International Workshop, FSE'98, Proceedings, LNCS Vol. 1372, Springer-Verlag*, 1998.

[FS01] S. Furuya, D. Watanabe, Y. Seto, and K. Takaragi, "Integrity-Aware Mode of Stream Cipher," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E85-A No.1, pp.58–65, 2002.

[HK97] S. Halevi and H. Krawczyk, "MMH: Software Message Authentication in the Gbit/second Rates," *Fast Software Encryption, 4th International Workshop, FSE'97, LNCS Vol. 1267, Springer-Verlag*, 1997.

[GD01] V. D. Gligor and P. Donescu, "Fast Encryption and Authentication: XCBC Encryption and XECB Authentication Modes," In *Preproceedings of FSE 2001, 8th Fast Software Encryption Workshop*, Yokohama Japan, 2001.

[G00] J. D. Golić, "Modes of Operation of Stream Ciphers," *Selected Areas in Cryptography, 7th Annual International Workshop, SAC 2000 Proceedings, LNCS Vol. 2012, Springer-Verlag*, 2001.

[JV98] M. H. Jakubowski and R. Venkatesan, "The Chain & Sum Primitive and Its Applications to MACs and Stream Ciphers," *Advances in Cryptology, —EUROCRYPT'98, LNCS Vol. 1403, Springer-Verlag*, 1998.

[J97] T. Johansson, "Bucket Hashing with Small Key Size," *Advances in Cryptology, —EUROCRYPT'97, LNCS Vol. 1233, Springer-Verlag*, 1997.

[J01] C. S. Jutla, "Encryption Modes with Almost Free Message Integrity," *Advances in Cryptology, —EUROCRYPT2001, LNCS Vol. 2045, Springer-Verlag*, 2001.

[KY00] J. Katz and M. Yung, "Unforgeable Encryption and Chosen Cipher Secure Modes of Operation," *Fast Software Encryption, 7th International Workshop, FSE2000, LNCS Vol. 1978, Springer-Verlag*, 2001.

[NP99] W. Nevelsteen and B. Preneel, "Software Performance of Universal Hash Functions," *Advances in Cryptology, —EUROCRYPT'99, LNCS Vol. 1592, Springer-Verlag*, 1999.

[PR99] S. Patel and Z. Ramzan, "Square Hash: Fast Message Authentication via Optimized Universal Hash Functions," *Advances in Cryptology, —CRYPTO'99, LNCS Vol. 1666, Springer-Verlag*, 1999.

[PvO96] B. Preneel and P. van Oorschot, "On The Security of Two MAC Algorithms," *Advances in Cryptology, —EUROCRYPT'96, LNCS Vol. 1070, Springer-Verlag*, 1996.

[R97] M. Roe, "Cryptography and Evidence," *Doctoral Dissertation with the University of Cambridge*, 1997. available at `http://www.ccsr.cam.ac.uk/techreports/index.html`.

[RBBK01] P. Rogaway, M. Bellare, J. Black, and T. Krovetz, "OCB: A Block-Cipher Mode of Operation for Efficient Authenticated Encryption," *Eights ACM conference on computer and communications security CCS-8, ACM Press*, 2001.

[S49] C. E. Shannon, "A Mathematical Theory of Communication," *Bell Systems Technical Journal*, Vol.28, No.4, 1949.

[S96] V. Shoup, "On Fast And Provably Secure Message Authentication Based on Universal Hashing," *Advances in Cryptology, —CRYPTO'96, LNCS Vol. 1109, Springer-Verlag*, 1996.

[T93] R. Taylor, "An Integrity Check Value Algorithm for Stream Ciphers," *Advances in Cryptology, —CRYPTO'93, LNCS Vol. 773, Springer-Verlag*, 1993.

[WC81] M. Wegman and L. Carter, "New Hash Functions And Their Use in Authentication And Set Equality," *Journal of Computer and System Sciences, Vol. 22*, 1981.