Security Evaluation Report of MULTI-S01

2001年12月14日

東京工業大学

岩田 哲

Security Evaluation Report of MULTI-S01

December 14, 2001

Abstract

In this report, we give the security evaluation of MULTI-S01. Our main results are summarized as follows:

- We confirmed that the security claims in the self-evaluation report of MULTI-S01 are correct.
- However, the security definitions in the self-evaluation report are extremely weak (compared to the standard security definitions). We explain why the definitions are weak.
- We then give standard security definitions which the authors must consider.
- We also give outlines of how to prove the security of MULTI-S01 in these standard settings.

As a consequence, we suggest that MULTI-S01 should be considered as a standard *after* completing its security proof in the standard definitions, and we do *not* recommend the adoption of MULTI-S01 as a standard at present.

1 Introduction

The purpose of this report is to evaluate the security of MULTI-S01. In particular, we are interested in the security of MULTI-S01 in the sense of reduction based cryptography. That is, one gives reductions of the form

"If a certain primitive is secure then the scheme based on it is secure."

For our purpose, this becomes

"If PANAMA is secure then MULTI-S01 is secure."

This gives us the best way to gain assurance that a cryptographic scheme does what was intended, provided that:

- the assumption "PANAMA is secure" is reasonable, and
- the statement "MULTI-S01 is secure" means really secure.

The first claim of this report is that the security assumption on PANAMA in the self-evaluation report is too strong, and far from reasonable. And the next claim is that the security definitions on MULTI-S01 in the self-evaluation report are far from "secure" (in both privacy and integrity). In particular, the authors only consider the security against "known-ciphertext attack with single ciphertext" for privacy, and "known-plaintext attack with single plaintext-ciphertext pair" for integrity (we do not claim that their security claims are wrong, we confirmed that they are true).

The lack of security proof (in the standard model) is not because of the structure of MULTI-S01. We next give the standard security assumption (for PANAMA) and definitions (for MULTI-S01) precisely, and indeed, we claim that it is possible to prove its security against adaptive chosen-plaintext attack for both privacy and integrity (which, in turn, gives the security against the strongest form of adaptive chosen ciphertext attack) with reasonable assumption on PANAMA.

We also give the outlines of how to prove the security of MULTI-S01 in these standard settings (but this report will *not* give the complete proof. The authors should complete the proof).

This paper is organized as follows: In Section 2, we give the formalization of MULTI-S01. In Section 3, we give the standard assumption (for PANAMA) and definitions (for general encryption schemes). In Section 4, we give the standard security definitions (for MULTI-S01) and how the theorems should look like. In Section 5, we give outlines of proof. We give some remarks in Section 6 and conclude in Section 7.

2 Preliminaries

We are interested in the security of MULTI-S01 in the sense of reduction based cryptography.

For this purpose, we need to formalize MULTI-S01. In particular, we have to clarify the input and the output of the algorithm, and what the adversary knows or doesn't. The latter one should be done carefully since

MULTI-S01 uses nonce Q (or "deviation parameter" according to [10], but it is usually called "nonce"), and redundancy data R.

We understand that MULTI-S01 consists of three algorithms, key generation algorithm MULTI-S01-$\mathcal{K}$, encryption algorithm MULTI-S01-$\mathcal{E}$, and decryption MULTI-S01-$\mathcal{D}$. Also it uses PANAMA as a building block.

FORMAL DESCRIPTION OF MULTI-S01-$\mathcal{K}$: The input to the algorithm is `Coins` (or a random tape) and it outputs a key $K \in \{0,1\}^{256}$. Therefore

$$\text{MULTI-S01-}\mathcal{K} : \texttt{Coins} \rightarrow \{0,1\}^{256}.$$

Usually we omit mention of the argument of MULTI-S01-$\mathcal{K}$, thinking of MULTI-S01-$\mathcal{K}$ as a probabilistic algorithm. In particular, when we write

$$K \overset{R}{\leftarrow} \text{MULTI-S01-}\mathcal{K},$$

we understand that K is chosen uniformly at random from $\{0,1\}^{256}$.

FORMAL DESCRIPTION OF PANAMA: The input to PANAMA is a key $K \in \{0,1\}^{256}$, and a nonce $Q \in \{0,1\}^{256}$ (or "deviation parameter" according to [10], but it is usually called "nonce"). In this formalization, we imagine that $n \in \{0, \ldots, 2^{32} - 2\}$, which specifies the number of output blocks, is also the input to the algorithm (otherwise the algorithm doesn't know how many blocks it should output).

The output of PANAMA is $A \in \{0,1\}^{64} \setminus \{0^{64}\}$, $B \in \{0,1\}^{64n}$, and $S \in \{0,1\}^{64}$.

Therefore

$$\begin{aligned}\text{PANAMA} : \{0,1\}^{256} \times \{0,1\}^{256} \times \{0, \ldots, 2^{32} - 2\} \\ \rightarrow \{0,1\}^{64} \setminus \{0^{64}\} \times \{0,1\}^{64n} \times \{0,1\}^{64},\end{aligned}$$

and we write

$$(A, B, S) \leftarrow \text{PANAMA}_K(Q, n).$$

The adversary does not know K, while the adversary may know (or, strong adversary may even choose) Q and n (this is why we write $\text{PANAMA}_K(Q, n)$ rather than $\text{PANAMA}(K, Q, n)$).

Further details of PANAMA is not necessary in this report. See [6, 10] for more details.

FORMAL DESCRIPTION OF MULTI-S01-$\mathcal{E}$: It takes three inputs: message $M \in \{0,1\}^{\leq 2^{38}-128}$ (any bit strings less than $2^{38} - 128$ bits), redundancy data $R \in \{0,1\}^{64}$, and a key $K \in \{0,1\}^{256}$ (according to [10]).

In this report, we do *not* consider the case for M longer than $2^{38} - 128$ bits (Definitions and theorems in this report can be easily extended to handle this case).

The authors must be more careful about the treatment of the redundancy data R. It is unclear from [10] how we should formalize it. In particular, it is unclear if it is secret, or public (for adversary) and how it is transmitted from the sender to the receiver. The careful treatment of (both public and secret) redundancy data is in [1].

In this formalization we treat the redundancy data R as a publicly known data chosen by the sender each time of the encryption. We do not care how it is chosen. Also, we do not specify how it is transmitted from the sender to the receiver. In particular we do not include the redundancy data in ciphertext. Since we assume R is public, the adversary (and the receiver) knows R (We are not quite sure if this is the intention of the authors, but this makes the adversary stronger, and the security results more meaningful).

We also treat the nonce Q as the input to the algorithm (otherwise the sender cannot create (A, B, S) from PANAMA. We do not understand why the authors did not treat the nonce Q as the input, if there are any reason for this, the authors must explain them). As is the case of the redundancy data R, we treat the nonce Q as a publicly known data chosen by the sender each time of the encryption. We do not care how it is chosen, but we forbid from choosing the same nonce twice. Also, we do not specify how it is transmitted.

The output of MULTI-S01-$\mathcal{E}$ is the ciphertext $C \in \{0,1\}^{64n}$ where $n = \lceil m/64 \rceil + 2$ and m is the length of the message M in bits.

Therefore

$$\text{MULTI-S01-}\mathcal{E} : \{0,1\}^{256} \times \{0,1\}^{256} \times \{0,1\}^{\leq 2^{38}-128} \times \{0,1\}^{64} \to \{0,1\}^{64n}$$

and we write

$$C \leftarrow \text{MULTI-S01-}\mathcal{E}_K(Q, M, R).$$

The adversary does not know K, while the adversary may know (or may even choose) Q, M and R (this is why we write MULTI-S01-$\mathcal{E}_K(Q, M, R)$).

The detailed algorithm of MULTI-S01-$\mathcal{E}_K(Q, M, R)$ is described below.

```
Algorithm MULTI-S01-$\mathcal{E}_K(Q, M, R)$
    $n \leftarrow \lceil |M|/64 \rceil + 2$;
    $(A, B, S) \leftarrow$ PANAMA$(Q, n)$;
    $P \leftarrow$ Pad$(M, S, R)$;
    compute $C$ from $(A, B, P)$;
    output $C$;
```

Pad(M, S, R) works as follows.

$$\mathtt{Pad}(M, S, R) = (M, 0^*, S, R)$$

where $|(M, 0^*)|$ is the least multiple of 64, longer than or equals $|M|$.

Details of computation of C from (A, B, P) are not necessary in this report. See [10] for more details.

Formal description of MULTI-S01-$\mathcal{D}$: It takes three inputs: ciphertext $C \in \{0,1\}^{\leq 2^{38}}$ (any bit strings less than 2^{38} bits), redundancy data $R \in \{0,1\}^{64}$, and a key $K \in \{0,1\}^{256}$ (according to [10]).

As in MULTI-S01-$\mathcal{E}$, we also treat the nonce Q as the input to the algorithm.

The output of MULTI-S01-$\mathcal{D}$ is either:

- The plaintext $M' \in \{0,1\}^{64(n-2)}$ where $n = \lceil c/64 \rceil$ and c is the length of the ciphertext C in bits, or
- the distinguished "reject" signal.

Therefore

$$\begin{aligned}\text{MULTI-S01-}\mathcal{D} : \{0,1\}^{256} \times \{0,1\}^{256} \times \{0,1\}^{\leq 2^{38}} \times \{0,1\}^{64} \\ \rightarrow \{0,1\}^{64(n-2)} \cup \{\text{“reject”}\}\end{aligned}$$

and we write

$$M' \leftarrow \text{MULTI-S01-}\mathcal{D}_K(Q, C, R) \text{ or “reject”} \leftarrow \text{MULTI-S01-}\mathcal{D}_K(Q, C, R).$$

Further details are not necessary in this report. See [10] for more details.

3 Security assumption and definitions

We start by looking at Theorem 1 of [11].

Theorem 1 [11] *Assuming that the PRNG* PANAMA *is secure, MULTI-S01 provides perfect confidentiality and integrity. The probability of successful forgery is no more than* $(n+1)/2^{64}$.

We confirmed that this theorem is true. In what follows, we carefully examine what this theorem tells us. To be precise, we look at:

- In Section 3.1, what the assumption "PANAMA is secure" means.
- In Section 3.2, what "perfect confidentiality" means.
- In Section 3.3, what "forgery" means.

3.1 Security assumption on PANAMA

It is unclear what "PANAMA is secure" means from the context, and thus, this assumption is vague. It seems that this means:

"the output of PANAMA *is uniformly distributed random bits."*

This is because each of A and B_i is treated as a uniformly distributed random bits over $\{0,1\}^{64} \setminus \{0^{64}\}$ and $\{0,1\}^{64}$ respectively in the proof of Theorem 1 of [11]. It is clear that this assumption is too strong, unacceptable and also wrong (the entropy of the output of PANAMA is at most the entropy of key K, and it is usual that A, B is longer than 256 bits).

The assumption should look like:

"it is difficult for any adversary to distinguish the output of PANAMA *from uniformly distributed random bits."*

Then this assumption is true unless otherwise someone breaks PANAMA. In what follows, we formalize this assumption (especially, "difficult" is unclear in the above notion).

SECURITY OF PANAMA: We cannot follow the complete standard definition of PRNG, because MULTI-S01 uses Q, which works as a nonce (and works as seed in PANAMA together with K, and the adversary may see the nonce Q, i.e., the adversary may see a part of the seed).

In this formalization, we give the adversary every possible advantage (some of them may be more than is available in real life). These advantages include:

- We allow the adversary to choose the nonce Q and n, (but not K) and see the corresponding output (A, B, S).
- We allow the adversary to know q input-output pairs (though we forbid the adversary from choosing the same nonce twice).

The adversary's job is to distinguish the outputs of PANAMA from random bits of equal length. Informally, we say that "PANAMA is secure" if any "reasonable" adversary cannot do a "good job."

Definition 3.1 (PANAMA **is** $(t_P, q_P, \mu_P, \epsilon_P)$**-secure)** *We say that* PANAMA is $(t_P, q_P, \mu_P, \epsilon_P)$-secure, *if for any adversary A which runs in time at most t_P, makes at most q_P oracle queries, these totaling at most μ_P bits,*

$$\begin{aligned}\mathtt{Adv}^{\mathtt{pr}}_{\mathcal{A}} \stackrel{\mathrm{def}}{=} \Pr\Big(K \stackrel{R}{\leftarrow} \mathit{MULTI\text{-}S01\text{-}}\mathcal{K} : \mathcal{A}^{Panama_K(\cdot,\cdot)} = 1\Big)\\ - \Pr\Big(\mathcal{A}^{\$^{|\cdot,\cdot|}} = 1\Big) \leq \epsilon_P.\end{aligned}$$

The notation $\mathcal{A}^{Panama_K(\cdot,\cdot)}$ indicates $\mathcal{A}$ with an oracle which, in response to a query (Q, n), returns $(A, B, S) \leftarrow$ PANAMA$_K(Q, n)$*. The notation $\mathcal{A}^{\$^{|\cdot,\cdot|}}$ indicates $\mathcal{A}$ with an oracle which, in response to a query (Q, n), returns $(A, B, S) \stackrel{R}{\leftarrow} \{0,1\}^{64} \setminus \{0^{64}\} \times \{0,1\}^{64n} \times \{0,1\}^{64}$.*

3.2 Security definition for privacy

Definition 3.2 (Perfect confidentiality [11]) *Let P be the concatenated data of M, S, and R. The necessary and sufficient condition for* perfect confidentiality *is* $\Pr(P \mid C) = \Pr(P)$*, where C represents the ciphertext.*

We do not understand why the authors chosen this security definition. If there are some reasons, the authors must explain them.

The intuition of this definition can be stated as follows: the encryption algorithm has perfect confidentiality if and only if it is secure against *ciphertext-only attack,* where the adversary is restricted to have *only one* ciphertext.

It is obvious that the perfect confidentiality does not meet the security requirement of modern cryptography. The standard security definition considers the security against *adaptive chosen plaintext attack,* which is much stronger than ciphertext-only attack with single ciphertext.

Here are the standard security definitions. The authors should evaluate their algorithm in one of the following four notions (or stronger one stated later).

In the following four notions, the encryption scheme Π is a triple $(\mathcal{K}, \mathcal{E}, \mathcal{D})$, where $\mathcal{K}$ is a finite set, $\mathcal{E}$ is a probabilistic algorithm and $\mathcal{D}$ is a deterministic algorithm. Encryption algorithm $\mathcal{E}$ takes strings $K \in \mathcal{K}$, and $M \in \{0,1\}^*$, and returns a string $C \leftarrow \mathcal{E}_K(M)$. Decryption algorithm $\mathcal{D}$ takes strings $K \in \mathcal{K}$, and $C \in \{0,1\}^*$, and returns $\mathcal{D}_K(C)$ which is either a string $M \in \{0,1\}^*$ or a distinguished symbol "reject."

Real-or-random: The idea is that an adversary cannot distinguish the encryption of plaintext from the encryption of an equal-length of random bits.

Definition 3.3 (Real-or-random [3]) *Encryption scheme* $\Pi = (\mathcal{E}, \mathcal{D}, \mathcal{K})$ *is said to be* (t, q, μ, ϵ)*-secure in the* real-or-random *sense, if for any adversary* $\mathcal{A}$ *which runs in time at most* t*, makes at most* q *oracle queries, these totaling at most* μ *bits,*

$$\mathtt{Adv}^{\mathtt{rr}}_{\mathcal{A}} \overset{\mathrm{def}}{=} \Pr\left(K \overset{R}{\leftarrow} \mathcal{K} : \mathcal{A}^{\mathcal{E}_K(\cdot)} = 1\right) - \Pr\left(K \overset{R}{\leftarrow} \mathcal{K} : \mathcal{A}^{\mathcal{E}_K(\$^{|\cdot|})} = 1\right) \leq \epsilon.$$

The notation $\mathcal{A}^{\mathcal{E}_K(\cdot)}$ *indicates* $\mathcal{A}$ *with an oracle which, in response to a query* M*, returns* $y \leftarrow \mathcal{E}_K(M)$*. The notation* $\mathcal{A}^{\mathcal{E}_K(\$^{|\cdot|})}$ *indicates* $\mathcal{A}$ *with an oracle which, in response to a query* M*, choose* $M' \overset{R}{\leftarrow} \{0,1\}^{|M|}$ *and then returns* $C \leftarrow \mathcal{E}_K(M')$*.*

Left-or-right: The idea is that an adversary cannot distinguish from each other the encryption of an equal-length strings.

Definition 3.4 (Left-or-right [3]) *Encryption scheme* $\Pi = (\mathcal{E}, \mathcal{D}, \mathcal{K})$ *is said to be* (t, q, μ, ϵ)*-secure in the* left-or-right *sense, if for any adversary* $\mathcal{A}$ *which runs in time at most* t*, makes at most* q *oracle queries, these totaling at most* μ *bits,*

$$\begin{aligned}\mathtt{Adv}^{\mathtt{lr}}_{\mathcal{A}} \overset{\mathrm{def}}{=} & \Pr\left(K \overset{R}{\leftarrow} \mathcal{K} : \mathcal{A}^{\mathcal{E}_K(\mathtt{left}(\cdot,\cdot))} = 1\right) \\ & - \Pr\left(K \overset{R}{\leftarrow} \mathcal{K} : \mathcal{A}^{\mathcal{E}_K(\mathtt{right}(\cdot,\cdot))} = 1\right) \leq \epsilon.\end{aligned}$$

The notation $\mathcal{A}^{\mathcal{E}_K(\mathtt{left}(\cdot,\cdot))}$ *indicates* $\mathcal{A}$ *with an oracle which, in response to query* (M_1, M_2)*, returns* $C \leftarrow \mathcal{E}_K(M_1)$*. The notation* $\mathcal{A}^{\mathcal{E}_K(\mathtt{right}(\cdot,\cdot))}$ *indicates*

$\mathcal{A}$ with an oracle which, in response to a query (M_1, M_2), returns $C \leftarrow \mathcal{E}_K(M_2)$.

FIND-THEN-GUESS: This is an adaptation of the notion of polynomial security given in [9, 15]. We only present the definition without details. See [3] for more details.

Definition 3.5 (Find-then-guess [3]) *Encryption scheme $\Pi = (\mathcal{E}, \mathcal{D}, \mathcal{K})$ is said to be (t, q, μ, ϵ)-secure in the* find-then-guess *sense, if for any adversary $\mathcal{A}$ which runs in time at most t, makes at most q oracle queries, these totaling at most μ bits,*

$$\begin{aligned}\mathtt{Adv}^{\mathtt{fg}}_{\mathcal{A}} \stackrel{\text{def}}{=} 2 \cdot \Pr\Big(&K \stackrel{R}{\leftarrow} \mathcal{K}; (x_0, x_1, s) \leftarrow \mathcal{A}^{\mathcal{E}_K(\cdot)}(\mathtt{find});\\ &b \stackrel{R}{\leftarrow} \{0,1\}; y \leftarrow \mathcal{E}_K(x_b) : \mathcal{A}^{\mathcal{E}_K(\cdot)}(\mathtt{guess}, y, s) = b\Big) - 1 \leq \epsilon.\end{aligned}$$

SEMANTIC: Goldwasser and Micali [9] explain semantic security by saying that whatever can be efficiently computed about the plaintext given the ciphertext can also be computed in the absence of ciphertext. We only present the definition without details. See [3] for more details.

Definition 3.6 (Semantic [3]) *Let f : $\mathtt{Message\text{-}Space} \rightarrow \{0,1\}^*$ be a function and let $\mathcal{M} = \{\mathcal{M}_\gamma\}_{\gamma \in \{0,1\}^{\leq m}}$ be an m-distribution on* Message-Space*. Encryption scheme $\Pi = (\mathcal{E}, \mathcal{D}, \mathcal{K})$ is said to be (t, q, μ, ϵ)-secure in the* semantic *sense, for f over $\mathcal{M}$, if*

$$\mathtt{Adv}^{\mathtt{sm}}_{\mathcal{A}}(f, \mathcal{M}) \stackrel{\text{def}}{=} E\left(K \stackrel{R}{\leftarrow} \mathcal{K}; (\gamma, s) \leftarrow \mathcal{A}^{\mathcal{E}_K(\cdot)}(\mathtt{select}) : \alpha(a, \gamma, s)\right) \leq \epsilon,$$

where

$$\alpha(a, \gamma, s) = \Pr\left(x \stackrel{R}{\leftarrow} \mathcal{M}_\gamma; y \leftarrow \mathcal{E}_K(x) : \mathcal{A}^{\mathcal{E}_K(\cdot)}(\mathtt{predict}, y, s) = f(x)\right) - p^*_{f, \mathcal{M}_\gamma}$$

for any adversary $\mathcal{A}$ which runs in time at most t, makes at most q oracle queries, these totaling at most μ bits.

The reasons why we claim these definitions are standard are:

- There are reductions between the above four notions (real-or-random is the strongest one).

- Many encryption modes are evaluated under one of these four notions (CBC mode, XOR mode [3], CFB mode, OCFB mode [2], IAPM, IACBC [12], XCBC mode [8], and many others).
- One of the above four notions together with the integrity of ciphertexts (see the next subsection and [4, 5, 13]) implies the indistinguishability of the strongest form of adaptive chosen-ciphertext attack (which, in turn, is equivalent to non-malleability [7] under adaptive chosen ciphertext attack [14]).

Yet, there is a stronger notion of privacy, asserting indistinguishability from random bits [16]. This notion is easily seen to imply all of the above four notions, and by tight reductions. This notion was first adopted in the security proof of OCB mode [16], which uses nonce. MULTI-S01 also uses nonce, Q, and we recommend that the authors should adopt this notion. Also, as in Section 5, we strongly believe that MULTI-S01 can be proven to be secure in this strongest notion.

In the following notion, we consider a nonce-using encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ and an associated number s (the nonce length). Here $\mathcal{K}$ is a finite set, $\mathcal{E}$ and $\mathcal{D}$ are deterministic algorithms. Encryption algorithm $\mathcal{E}$ takes strings $K \in \mathcal{K}$, $Q \in \{0,1\}^s$ and $M \in \{0,1\}^*$, and returns a string $C \leftarrow \mathcal{E}_K(Q, M)$. Decryption algorithm $\mathcal{D}$ takes strings $K \in \mathcal{K}$, $Q \in \{0,1\}^s$ and $C \in \{0,1\}^*$, and returns $\mathcal{D}_K(Q, C)$ which is either a string $M \in \{0,1\}^*$ or a distinguished symbol "reject."

INDISTINGUISHABILITY FROM RANDOM BITS: The idea is that an adversary cannot distinguish the encryption of plaintext from random bits of equal length.

Definition 3.7 (Indistinguishability from random bits [16]) *A nonce-using encryption scheme* $\Pi = (\mathcal{E}, \mathcal{D}, \mathcal{K})$ *is said to be* (t, q, μ, ϵ)*-secure in the* indistinguishability from random bits *sense, if for any adversary* $\mathcal{A}$ *which runs in time at most* t*, makes at most* q *oracle queries, these totaling at most* μ *bits,*

$$\mathtt{Adv}_{\mathcal{A}}^{\mathtt{ir}} \stackrel{\mathrm{def}}{=} \Pr\left(K \xleftarrow{R} \mathcal{K} : \mathcal{A}^{\mathcal{E}_K(\cdot,\cdot)} = 1\right) - \Pr\left(K \xleftarrow{R} \mathcal{K} : \mathcal{A}^{\$^{|\cdot,\cdot|}} = 1\right) \leq \epsilon.$$

The notation $\mathcal{A}^{\mathcal{E}_K(\cdot,\cdot)}$ *indicates* $\mathcal{A}$ *with an oracle which, in response to a query* (Q, M)*, returns* $C \leftarrow \mathcal{E}_K(Q, M)$*. It is assumed that* $|C| = l(|M|)$ *depends only on* $|M|$*, where* $C \leftarrow \mathcal{E}_K(Q, M)$*. The notation* $\mathcal{A}^{\$^{|\cdot,\cdot|}}$ *indicates* $\mathcal{A}$ *with an oracle which, in response to a query* (Q, M)*, returns* $C \xleftarrow{R} \{0,1\}^{l(|M|)}$*.*

3.3 Security definition for integrity

Definition 3.8 (Definition of attacker [11]) *An adversary knows a known-plaintext consisting of message M, redundant data R, and corresponding ciphertext C. His goal is to generate a different ciphertext whose last two blocks pass the receiver's redundancy data test.*

Again, we do not understand why the authors chosen this security definition. If there are some reasons, the authors must explain them.

The intuition of this definition can be stated as follows: the encryption algorithm is secure if it is secure against *known-plaintext attack,* where the adversary is restricted to have *only one* plaintext-ciphertext pair.

Modern cryptography gives the adversary every possible advantage (sometimes more than is available in real life). These advantages include:

- First, we consider the security against *adaptive chosen plaintext attack,* which is stronger than known-plaintext attack with single plaintext-ciphertext pair.

- Second, we allow the adversary to know q plaintext-ciphertext pairs (as opposed to know only one pair).

- Third, we allow the adversary to choose nonce, Q (though we forbid the adversary from choosing the same nonce twice).

- Finally, the nonce used in the forgery attempt may coincide with a nonce used in one of the adversary's queries (this is also allowed in MULTI-S01).

Here is the standard security definition of integrity. We recommend the authors to evaluate their algorithm in this setting.

INTEGRITY OF CIPHERTEXTS: We consider a nonce-using encryption scheme. An adversary $\mathcal{A}$ is *nonce-respecting* if it never repeats a nonce: if $\mathcal{A}$ asks its oracle a query (Q, M), it will never subsequently ask its oracle a query (Q, M'), regardless of its coins (if any) and regardless of oracle responses. The adversary $\mathcal{A}$ *forges* if $\mathcal{A}$ is nonce-respecting, $\mathcal{A}$ output (Q, C) where $\mathcal{D}_K(Q, C) \neq$ "reject", and $\mathcal{A}$ made no earlier query (Q, M) which resulted in a response C.

Definition 3.9 (Integrity of ciphertexts [4, 5, 13, 16]) *A nonce-using encryption scheme* $\Pi = (\mathcal{E}, \mathcal{D}, \mathcal{K})$ *is said to be* (t, q, μ, ϵ)*-secure in the* integrity of ciphertexts *sense, if for any adversary* $\mathcal{A}$ *which runs in time at most* t*, makes at most* q *oracle queries, these totaling at most* μ *bits,*

$$\mathtt{Adv}^{\mathtt{ic}}_{\mathcal{A}} \stackrel{\mathrm{def}}{=} \Pr\left(K \stackrel{R}{\leftarrow} \mathcal{K} : \mathcal{A}^{\mathcal{E}_K(\cdot,\cdot)}\ \textit{forges}\right) \leq \epsilon.$$

The notation $\mathcal{A}^{\mathcal{E}_K(\cdot,\cdot)}$ *indicates* $\mathcal{A}$ *with an oracle which, in response to a query* (Q, M)*, returns* $C \leftarrow \mathcal{E}_K(Q, M)$*. We stress that the nonce used in the forgery attempt may coincide with a nonce used in one of the adversary's queries.*

The reasons why we claim this definition is standard are:

- Many encryption modes are evaluated under this notion (IAPM, IACBC [12], XCBC mode [8], OCB mode [16], and many others).
- The above notion together with the indistinguishability form random bits implies the indistinguishability of the strongest form of adaptive chosen-ciphertext attack (which, in turn, is equivalent to non-malleability [7] under adaptive chosen ciphertext attack [14]).

4 Definitions, and Theorems for MULTI-S01

We recommend the authors to adopt Definition 3.7 for privacy and Definition 3.9 for integrity (see Section 3 for reasons of these choice).

Unfortunately, we cannot follow these definitions directly, since MULTI-S01 uses redundancy data R (which is not considered in Definition 3.7 and Definition 3.9), and we have to modify these definitions. In Section 4.1, we show how to modify these definitions (these are just examples. The authors may use other definitions). And in Section 4.2 we show how the corresponding theorems should look like.

We say that MULTI-S01 is a nonce, redundancy data-using encryption scheme (MULTI-S01-$\mathcal{K}$, MULTI-S01-$\mathcal{E}$, MULTI-S01-$\mathcal{D}$), where

$$\begin{cases} \text{MULTI-S01-}\mathcal{E} : \{0,1\}^{256} \times \{0,1\}^{256} \times \{0,1\}^{\leq 2^{38}-128} \times \{0,1\}^{64} \\ \qquad \rightarrow \{0,1\}^{64n} \\ \text{MULTI-S01-}\mathcal{D} : \{0,1\}^{256} \times \{0,1\}^{256} \times \{0,1\}^{\leq 2^{38}} \times \{0,1\}^{64} \\ \qquad \rightarrow \{0,1\}^{64(n-2)} \cup \{\text{“reject”}\} \end{cases}$$

and we write

$$\begin{cases} C \leftarrow \text{MULTI-S01-}\mathcal{E}_K(Q, M, R), \text{ and} \\ M' \leftarrow \text{MULTI-S01-}\mathcal{D}_K(Q, C, R) \text{ or “reject”} \leftarrow \text{MULTI-S01-}\mathcal{D}_K(Q, C, R). \end{cases}$$

4.1 Definitions for MULTI-S01

We consider a nonce, redundancy data-using encryption scheme. An adversary $\mathcal{A}$ is *nonce-respecting* if it never repeats a nonce: if $\mathcal{A}$ asks its oracle a query (Q, M, R), it will never subsequently ask its oracle a query (Q, M', R'), regardless of its coins (if any) and regardless of oracle responses. All adversaries are assumed to be nonce-respecting.

Definition 4.1 (Indistinguishability from random bits) *We say that MULTI-S01 is (t, q, μ, ϵ)-secure in the indistinguishability from random bits sense, if for any nonce-respecting adversary $\mathcal{A}$ which runs in time at most t, makes at most q oracle queries, these totaling at most μ bits,*

$$\begin{aligned} \mathtt{Adv}_{\mathcal{A}}^{\mathtt{ir}} \stackrel{\text{def}}{=} & \Pr\left(K \stackrel{R}{\leftarrow} \textit{MULTI-S01-}\mathcal{K} : \mathcal{A}^{\text{MULTI-S01-}\mathcal{E}_K(\cdot,\cdot,\cdot)} = 1\right) \\ & - \Pr\left(\mathcal{A}^{\$|\cdot,\cdot,\cdot|} = 1\right) \leq \epsilon. \end{aligned}$$

The notation $\mathcal{A}^{\text{MULTI-S01-}\mathcal{E}_K(\cdot,\cdot,\cdot)}$ indicates $\mathcal{A}$ with an oracle which, in response to a query (Q, M, R), returns $C \leftarrow$ MULTI-S01-$\mathcal{E}_K(Q, M, R)$. The notation $\mathcal{A}^{\$|\cdot,\cdot,\cdot|}$ indicates $\mathcal{A}$ with an oracle which, in response to a query (Q, M, R), returns $C \stackrel{R}{\leftarrow} \{0, 1\}^{64n}$, where $n = \lceil m/64 \rceil + 2$ and m is the length of the message M in bits.

The adversary $\mathcal{A}$ *forges* if $\mathcal{A}$ is nonce-respecting, $\mathcal{A}$ output (Q, C, R) where $\mathcal{D}_K(Q, C, R) \neq$ “reject”, and $\mathcal{A}$ made no earlier query (Q, M, R) which resulted in a response C.

Definition 4.2 (Integrity of ciphertexts) *We say that MULTI-S01 is (t, q, μ, ϵ)-secure in the integrity of ciphertexts sense, if for any adversary $\mathcal{A}$ which runs in time at most t, makes at most q oracle queries, these totaling at most μ bits,*

$$\mathtt{Adv}_{\mathcal{A}}^{\mathtt{ic}} \stackrel{\text{def}}{=} \Pr\left(K \stackrel{R}{\leftarrow} \textit{MULTI-S01-}\mathcal{K} : \mathcal{A}^{\text{MULTI-S01-}\mathcal{E}_K(\cdot,\cdot,\cdot)} \textit{ forges}\right) \leq \epsilon.$$

The notation $\mathcal{A}^{\text{MULTI-S01-}\mathcal{E}_K(\cdot,\cdot,\cdot)}$ indicates $\mathcal{A}$ with an oracle which, in response to a query (Q, M, R), returns $C \leftarrow$ MULTI-S01$_K(Q, M, R)$.

4.2 Theorems for MULTI-S01

In this section, we show how the theorems should look like, rather than Theorem 1 in [11]. Note that these theorems are drafts, and we did *not* complete the proof.

Theorem 4.1 (Privacy) *There is a constant* $\alpha > 0$ *such that the following is true. Suppose that* PANAMA *is* $(t_P, q_P, \mu_P, \epsilon_P)$*-secure then MULTI-S01 is* (t, q, μ, ϵ)*-secure in the indistinguishability from random bits sense, where*

$$t = t_P - \alpha(\mu + q),\ q = q_P,\ \mu = \mu_P - 128q,\ \text{and } \epsilon = \epsilon' + \epsilon_P,$$

and $\epsilon' = 0$*.*

Theorem 4.2 (Integrity) *There is a constant* $\alpha > 0$ *such that the following is true. Suppose that* PANAMA *is* $(t_P, q_P, \mu_P, \epsilon_P)$*-secure then MULTI-S01 is* (t, q, μ, ϵ)*-secure in the integrity of ciphertexts sense, where*

(relations between $t, t_P, q, q_P, \mu, \mu_P, \epsilon$ *and* ϵ_P *should be specified here.)*

We leave the details of Theorem 4.2.

5 Outlines of proof

To prove these theorems, it is enough to give information-theoretic bounds. Here we assume that the adversary has unbounded computational power, and thus we do not restrict the adversary's running time (we omit the corresponding argument).

We say that PANAMA *is ideal* if, in response to a query (Q, n), it returns $(A, B, S) \xleftarrow{R} \{0,1\}^{64} \setminus \{0^{64}\} \times \{0,1\}^{64n} \times \{0,1\}^{64}$ (that is, the same as $\$^{|\cdot,\cdot|}$ oracle in Definition 3.1). We denote the ideal PANAMA by PANAMA*. MULTI-S01 uses PANAMA as a building block. We denote MULTI-S01 that uses PANAMA* (instead of PANAMA) by MULTI-S01*. That is,

```
Algorithm MULTI-S01*-$\mathcal{E}_K(Q, M, R)$
    $n \leftarrow \lceil |M|/64 \rceil + 2$;
    $(A, B, S) \leftarrow$ PANAMA*$(Q, n)$;
    $P \leftarrow$ Pad$(M, S, R)$;
    compute $C$ from $(A, B, P)$;
    output $C$;
```

The information-theoretic bounds are presented below:

Lemma 5.1 (Main lemma for privacy) *MULTI-S01* is* (q, μ, ϵ')*-secure in the indistinguishability from random bits sense, where* $\epsilon' = 0$.

Lemma 5.2 (Main lemma for integrity) *MULTI-S01* is* (q, μ, ϵ')*-secure in the integrity of ciphertexts sense, where (*ϵ' *should be specified here).*

We leave ϵ' in Lemma 5.2.

5.1 Proof of Theorem 4.1

Given Lemma 5.1, proof of Theorem 4.1 is fairly standard.

Informally, we proceed as follows:

1. We first show that MULTI-S01 and MULTI-S01* are indistinguishable.
2. We know from Lemma 5.1 that MULTI-S01* is secure.
3. Therefore MULTI-S01 is secure.

Thus, at the moment, we concentrate on proving that MULTI-S01 and MULTI-S01* are indistinguishable. We prove this by a contradiction argument. Suppose that some adversary $\mathcal{C}$ can distinguish MULTI-S01 from MULTI-S01*, we show that this would mean PANAMA is not secure (which is a contradiction).

Formally, assume that an adversary $\mathcal{C}$ can (t, q, μ, ϵ_P)-break MULTI-S01 from MULTI-S01*. That is, some nonce-respecting adversary $\mathcal{C}$ which runs in time at most t, makes at most q oracle queries, these totaling at most μ bits,

$$\begin{aligned} &\Pr\left(K \xleftarrow{R} \text{MULTI-S01-}\mathcal{K} : \mathcal{C}^{\text{MULTI-S01-}\mathcal{E}_K(\cdot,\cdot,\cdot)} = 1\right) \\ &- \Pr\left(K \xleftarrow{R} \text{MULTI-S01-}\mathcal{K} : \mathcal{C}^{\text{MULTI-S01}^*\text{-}\mathcal{E}_K(\cdot,\cdot,\cdot)} = 1\right) > \epsilon_P. \end{aligned} \tag{1}$$

The notation $\mathcal{C}^{\text{MULTI-S01-}\mathcal{E}_K(\cdot,\cdot,\cdot)}$ indicates $\mathcal{C}$ with an oracle which, in response to a query (Q, M, R), returns $C \leftarrow$ MULTI-S01-$\mathcal{E}_K(Q, M, R)$ (by using PANAMA to compute (A, B, S)). The notation $\mathcal{C}^{\text{MULTI-S01}^*\text{-}\mathcal{E}_K(\cdot,\cdot,\cdot)}$ indicates $\mathcal{C}$ with an oracle which, in response to a query (Q, M, R), returns $C \leftarrow$ MULTI-S01*-$\mathcal{E}_K(Q, M, R)$ (by using PANAMA* to compute (A, B, S)).

We build another adversary $\mathcal{B}$ that $(t_P, q_P, \mu_P, \epsilon_P)$-breaks PANAMA.

Note that $\mathcal{B}$ has an oracle $\mathcal{O}(\cdot,\cdot)$ which is either Panama or Panama*. Our adversary $\mathcal{B}$ simply runs $\mathcal{C}$ and outputs whatever the output of $\mathcal{C}$. In order to run $\mathcal{C}$, $\mathcal{B}$ simulates its oracle by using its own oracle $\mathcal{O}$. In more details:

```
Algorithm $\mathcal{B}^{\mathcal{O}(\cdot,\cdot)}$
  run $\mathcal{C}$;
  when $\mathcal{C}$ makes its oracle query $(Q, M, R)$:
    $n \leftarrow \lceil |M|/64 \rceil + 2$;
    $(A, B, S) \leftarrow \mathcal{O}(Q, n)$;
    $P \leftarrow \mathsf{Pad}(M, S, R)$;
    compute $C$ from $(A, B, P)$;
    return $C$ as the answer to the oracle query;
  when $A$ outputs a bit $b$:
    output $b$;
```

It is clear that the total number of oracle queries made by $\mathcal{B}$ is at most q (which is q_P) since the total number of oracle queries made by $\mathcal{C}$ is at most q.

Suppose that $\mathcal{C}$ makes a query (Q, M, R) whose length is $256 + |M| + 64$. The length of the corresponding (A, B, S) is $64 + 64n + 64$, where $n = \lceil |M|/64 \rceil + 2$. Then $\mathcal{B}$ returns C to $\mathcal{C}$ which is $64n$ bits. This means that $|C| = |(A, B, S)| - 128$ each time $\mathcal{C}$ makes a query. Since $\mathcal{C}$ makes at most q oracle queries, these totaling at most μ bits, $\mathcal{B}$ receives at most $\mu + 128q$ (which is μ_P) bits from its oracle.

It is clear that $\mathcal{B}$'s running time is t plus the time to compute Pad at q points plus time to compute C from (A, B, P) for q times plus additional time which is $\alpha'(\mu + 128q + q(256 + 32))$, where the constant α' depends only on details of the model of computation. Thus, $\mathcal{B}$'s running times is $t + \alpha(\mu + q)$ for some constant α (which is t_P).

Further, an inspection of this algorithm makes clear that it supplies to $\mathcal{C}$ a perfect simulation of MULTI-S01 (if $\mathcal{O}$ is Panama) and a perfect simulation of MULTI-S01* (if $\mathcal{O}$ is Panama*). Therefore

$$\begin{aligned}&\Pr\left(K \overset{R}{\leftarrow} \text{MULTI-S01-}\mathcal{K} : \mathcal{C}^{\text{MULTI-S01-}\mathcal{E}_K(\cdot,\cdot,\cdot)} = 1\right)\\ &= \Pr\left(K \overset{R}{\leftarrow} \text{MULTI-S01-}\mathcal{K} : \mathcal{B}^{\text{Panama}_K(\cdot,\cdot)} = 1\right)\end{aligned}$$

and

$$\begin{aligned}&\Pr\left(K \overset{R}{\leftarrow} \text{MULTI-S01-}\mathcal{K} : \mathcal{C}^{\text{MULTI-S01*-}\mathcal{E}_K(\cdot,\cdot,\cdot)} = 1\right)\\ &= \Pr\left(K \overset{R}{\leftarrow} \text{MULTI-S01-}\mathcal{K} : \mathcal{B}^{\text{Panama*}_K(\cdot,\cdot)} = 1\right).\end{aligned}$$

This means that

$$\begin{aligned}&\Pr\left(K \overset{R}{\leftarrow} \text{MULTI-S01-}\mathcal{K} : \mathcal{B}^{\mathrm{Panama}_K(\cdot,\cdot)} = 1\right)\\ &- \Pr\left(K \overset{R}{\leftarrow} \text{MULTI-S01-}\mathcal{K} : \mathcal{B}^{\mathrm{Panama}^*{}_K(\cdot,\cdot)} = 1\right) > \epsilon_P\end{aligned}$$

from (1). Therefore the adversary $\mathcal{B}$ $(t_P, q_P, \mu_P, \epsilon_P)$-breaks PANAMA.

This contradicts the assumption that PANAMA is $(t_P, q_P, \mu_P, \epsilon_P)$-secure. Therefore for any adversary $\mathcal{C}$ which runs in time at most t, makes at most q oracle queries, these totaling at most μ bits,

$$\begin{aligned}&\Pr\left(K \overset{R}{\leftarrow} \text{MULTI-S01-}\mathcal{K} : \mathcal{C}^{\text{MULTI-S01-}\mathcal{E}_K(\cdot,\cdot,\cdot)} = 1\right)\\ &- \Pr\left(K \overset{R}{\leftarrow} \text{MULTI-S01-}\mathcal{K} : \mathcal{C}^{\text{MULTI-S01}^*\text{-}\mathcal{E}_K(\cdot,\cdot,\cdot)} = 1\right) \le \epsilon_P.\end{aligned} \tag{2}$$

Now consider an adversary $\mathcal{A}$ which runs in time at most t, makes at most q oracle queries, these totaling at most μ bits, tries to break MULTI-S01 in the indistinguishability from random bits sense. That is, we are interested in $\mathtt{Adv}^{\mathtt{ir}}_{\mathcal{A}}$, which is

$$\Pr\left(K \overset{R}{\leftarrow} \text{MULTI-S01-}\mathcal{K} : \mathcal{A}^{\text{MULTI-S01-}\mathcal{E}_K(\cdot,\cdot,\cdot)} = 1\right) - \Pr\left(\mathcal{A}^{\$|\cdot,\cdot,\cdot|} = 1\right).$$

This is equivalent to

$$\begin{aligned}&\Pr\left(K \overset{R}{\leftarrow} \text{MULTI-S01-}\mathcal{K} : \mathcal{A}^{\text{MULTI-S01-}\mathcal{E}_K(\cdot,\cdot,\cdot)} = 1\right)\\ &- \Pr\left(K \overset{R}{\leftarrow} \text{MULTI-S01-}\mathcal{K} : \mathcal{A}^{\text{MULTI-S01}^*\text{-}\mathcal{E}_K(\cdot,\cdot,\cdot)} = 1\right)\\ &+ \Pr\left(K \overset{R}{\leftarrow} \text{MULTI-S01-}\mathcal{K} : \mathcal{A}^{\text{MULTI-S01}^*\text{-}\mathcal{E}_K(\cdot,\cdot,\cdot)} = 1\right)\\ &- \Pr\left(\mathcal{A}^{\$|\cdot,\cdot,\cdot|} = 1\right).\end{aligned}$$

This value is at most $\epsilon_P + \epsilon'$ (from (2) and Lemma 4.1) which is ϵ.

5.2 Proof of Theorem 4.2, and main lemmas

Proof of Theorem 4.2 can be done with small modification to the proof of Theorem 4.1, while we leave the proofs of main lemmas. We are quite sure with the result of Lemma 5.1 (we think $\epsilon' = 0$ in Theorem 4.1 and Lemma 5.1 are correct). The intuition is as follows: consider

$$\Pr\left(K \overset{R}{\leftarrow} \text{MULTI-S01-}\mathcal{K} : \mathcal{A}^{\text{MULTI-S01}^*\text{-}\mathcal{E}_K(\cdot,\cdot,\cdot)} = 1\right).$$

Each time the adversary makes a query, each block of the corresponding ciphertext C is a uniform random value (B_i's) xor'd with some other independent value (M, A, S, and R), we have that C is uniformly distributed and independent of the query apart from its length.

Therefore the distribution of C when $\mathcal{A}$'s oracle is MULTI-S01*-$\mathcal{E}_K(\cdot,\cdot,\cdot)$ is the same as the distribution of C when $\mathcal{A}$'s oracle is $\$^{|\cdot,\cdot,\cdot|}$.

We leave the details of the above discussion, Theorem 4.2, and the value of ϵ' in Lemma 5.2.

6 Some remarks

6.1 Comments on the padding algorithm

Padding algorithm $\mathtt{Pad}(M, S, R)$ works as follows.

$$\mathtt{Pad}(M, S, R) = (M, 0^*, S, R)$$

where $|(M, 0^*)|$ is the least multiple of 64, longer than or equals $|M|$.

This is not good.

Reason: Let $M_i = 0^i$ for $1 \leq i \leq 64$ (that is, $M_1 = 0, M_2 = 00, M_3 = 000$, etc). Then we have $\mathtt{Pad}(M_i, S, R) = \mathtt{Pad}(M_j, S, R)$ for any i, j, S and R such that $1 \leq i < j \leq 64$. This means that the corresponding ciphertexts are all identical, that is, if the adversary knows the ciphertext of message M_1, he also knows the ciphertexts of $M_2, \ldots, M_{64}$.

Further, suppose that C is the ciphertext of M_1. Then the decryption algorithm outputs M_{64} as the corresponding plaintext (which is not M_1).

These facts do not affect the theoretical security proof (Theorem 4.1 and Theorem 4.2), but may be a problem in some practical applications, and we recommend the authors to fix this problem. Below, we present two simple solutions.

Solution 1: Padding algorithm $\mathtt{Pad}_1(M, S, R)$ works as follows.

$$\mathtt{Pad}_1(M, S, R) = (M, 10^*, S, R)$$

where $|(M, 10^*)|$ is the least multiple of 64, *longer* than $|M|$. This means that we always pad 10^* (even if $|M|$ is a multiple of 64). We easily see that

$$\mathtt{Pad}_1(M, S, R) \neq \mathtt{Pad}_1(M', S, R)$$

for any M, M', S and R such that $M \neq M'$. This means that we alway can remove 10^* from $P = \mathsf{Pad}_1(M, S, R)$.

SOLUTION 2: The second padding algorithm requires two S's. Thus we have to change PANAMA to output (A, B, S_1, S_2) instead of (A, B, S). Padding algorithm $\mathsf{Pad}_2(M, S_1, S_2, R)$ works as follows.

$$\mathsf{Pad}_2(M, S, R) = \begin{cases} (M, 10^*, S_1, R) & \text{if } |M| \text{ is a multiple of 64,} \\ (M, 10^*, S_2, R) & \text{if } |M| \text{ is not a multiple of 64,} \end{cases}$$

where $|(M, 10^*)|$ is the least multiple of 64, longer than or equals $|M|$. This algorithm saves the ciphertext expansion if $|M|$ is a multiple of 64.

6.2 Should compare MULTI-S01 with IAPM, OCB, XCBC

There are three major authenticated encryption algorithms with provable security. They are IAPM [12], OCB mode [16], and XCBC [8]. The authors should compare MULTI-S01 with IAPM, OCB, and XCBC (for both security and efficiency).

6.3 Comment on redundancy data

We do not see why the authors decided to use the redundancy data. We think the security analysis in [11] works even if we omit the redundancy data. We also think that the security proof in this report works even if we omit the redundancy data.

7 Conclusion

In this report, we evaluated the security of MULTI-S01. Our main results are summarized as follows:

- We confirmed that the security claims in the self-evaluation report of MULTI-S01 are correct.

- However, the security definitions in the self-evaluation report are extremely weak (compared to the standard security definitions). We pointed out that the assumption on PANAMA is too strong and wrong, adaptive chosen plaintext attack should be considered for privacy instead of ciphertext-only attack with single ciphertext, and adaptive chosen plaintext attack should be considered for integrity instead of known-plaintext attack with single plaintext-ciphertext pair.

- We then gave the standard security definitions, indistinguishability from random bits, and integrity of ciphertexts, which the authors must consider.
- We also gave the outlines of how to prove the security of MULTI-S01 in these standard settings.

As a consequence, we suggest that MULTI-S01 should be considered as a standard *after* completing its security proof in the standard definitions, and we do *not* recommend the adoption of MULTI-S01 as a standard at present.

References

[1] J. An, and M. Bellare. Does encryption with redundancy provide authenticity? *Advances in Cryptology — EUROCRYPT 2001, LNCS 2045,* Springer-Verlag, 2001.

[2] A. Alkassar, A. Geraldy, B. Pfitzmann, and A. Sadeghi. Optimized self-synchronizing mode of operation. *Fast Software Encryption, FSE 2001, LNCS,* Springer-Verlag.

[3] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption: Analysis of the DES modes of operation. *38th Annual Symposium on Foundations of Computer Science, FOCS '97,* IEEE, 1997.

[4] M. Bellare and C. Namprempre. Authenticated encryption: relations among notions and analysis of the generic composition paradigm. *Advances in Cryptology — ASIACRYPT 2000, LNCS 1976,* Springer-Verlag, 2000.

[5] M. Bellare and P. Rogaway. Encode-then-encipher encryption: how to exploit nonces or redundancy in plaintexts for efficient encryption. *Advances in Cryptology — ASIACRYPT 2000, LNCS 1976,* Springer-Verlag, 2000.

[6] J. Daemen, and C. Clapp. Fast hashing and stream encryption with PANAMA. *Fast Software Encryption, FSE 98, LNCS 1372,* Springer-Verlag, 1998.

[7] D. Delov, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM J. Comput.,* vol. 30, no. 2, pp. 391–437, 2000.

[8] V. Gligor, and P. Donescu. Fast encryption and authentication: XCBC encryption and XECB authentication modes. *Fast Software Encryption, FSE 2001, LNCS,* Springer-Verlag.

[9] O. Goldreich, and S. Micali. Probabilistic encryption. *J. of Computer and System Sciences,* vol. 28, 1984.

[10] Hitachi, Ltd. A symmetric key encryption algorithm: MULTI-S01. `http://www.sdl.hitachi.co.jp/crypto/s01/call3e.pdf`.

[11] Hitachi, Ltd. Self-evaluation report MULTI-S01. `http://www.sdl.hitachi.co.jp/crypto/s01/call4e.pdf`.

[12] C. Jutla. Encryption modes with almost free message integrity. *Advances in Cryptology — EUROCRYPT 2001, LNCS 2045,* Springer-Verlag, 2001.

[13] J. Katz and M. Yung. Unforgeable encryption and adaptively secure modes of operations. *Fast Software Encryption, FSE 2000, LNCS 1978,* Springer-Verlag, 2000.

[14] J. Katz and M. Yung. Complete characterization of security notions for probabilistic encryption. *32nd annual ACM Symposium on the Theory of Computing, STOC 2000,* ACM, 2000.

[15] S. Micali, C. Rackoff, and R. Sloan. The notion of security for probabilistic cryptosystems. *SIAM J. Comput.,* vol. 17, no. 2, 1988.

[16] P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: a block-cipher mode of operation for efficient authenticated encryption. *Proceedings of ACM Conference on Computer and Communications Security, ACM CCS 2001*, ACM, 2001.