# Encryption Modes with Almost Free Message Integrity

Charanjit S. Jutla

IBM T. J. Watson Research Center,
Yorktown Heights, NY 10598-704

**Abstract.** We define a new mode of operation for block encryption which in addition to assuring confidentiality also assures message integrity. In contrast, previously for message integrity a separate pass was required to compute a cryptographic message authentication code (MAC). The new mode of operation, called Integrity Aware CBC (IACBC) requires a total of $m + 2$ block encryptions on a plain-text of length $m$ blocks. The well known CBC (cipher block chaining) mode requires $m$ block encryptions. The second pass of computing the CBC-MAC essentially requires additional $m$ block encryptions. A new highly parallelizable mode (IAPM) is also shown to be secure for both encryption and message integrity.

## 1 Introduction

Symmetric key encryption is an integral part of world of communication today. It refers to the schemes and algorithms used to communicate data secretly over an insecure channel between parties sharing a secret key. It is also used in other scenarios like data storage.

There are two primary aspects of any security system: *confidentiality* and *authentication*. In its most prevalent form, confidentiality is attained by encryption of bulk digital data using *block ciphers*. The block ciphers (e.g. DES [15]), which are used to encrypt fixed length data, are used in various chaining modes to encrypt bulk data. One such mode of operation is cipher block chaining (CBC) ([1,9,14]). The security of CBC has been well studied [2].

Cipher block chaining of block ciphers is also used for authentication between parties sharing a secret key. The CBC-MAC (CBC Message Authentication Code) is an international standard [10]. The security of CBC-MAC was demonstrated in [4]. Authentication in this setting is also called *Message Integrity*.

Despite similar names, the two CBC modes, one for encryption and the other for MAC are different, as in the latter the intermediate results of the computation of the MAC are kept secret. In fact in most standards (TLS, IPsec [19,17]) and proprietary security systems, two different passes with two different keys, one each of the two modes is used to achieve both confidentiality and authentication.

Nevertheless, it is enticing to combine the two passes into one so that in a single cipher block chaining pass, both confidentiality and authentication are as-

B. Pfitzmann (Ed.): EUROCRYPT 2001, LNCS 2045, pp. 529–544, 2001.

sured. Many such attempts have been made, which essentially use a simple checksum or manipulation detection code (MDC) in the chaining mode ([16,13,6]). Unfortunately, all such previous schemes are susceptible to attacks (see e.g. [18]).

We mention here that there are two alternative approaches to authenticated encryption. The first is to generate a MAC using universal hash functions as in UMAC ([3]). UMACs on certain architectures can be generated rather fast. However, UMAC suffers from requiring too much key material or a Pseudorandom number generator (PRNG) to expand the key. In another scheme, block numbers are embedded into individual blocks to thwart attacks against message integrity ([11]). However, this makes the cipher-text longer.

In this paper, we present a new variant of CBC mode, which in a single pass achieves both confidentiality and authentication. To encrypt a message of length $m$ blocks, it requires a total of $(m + \log m)$ block encryptions. All other operations are simple operations, like exclusive-or. To contrast this with the usual CBC mode, the encryption pass requires $m$ block encryptions, and the CBC-MAC computation requires another $m$ block encryptions.

Our new mode of operation is also simple. A simpler (though not as efficient) version of the mode just requires a usual CBC encryption of the plain-text appended with the checksum (MDC), with a random initial vector $r$. As already mentioned, such a scheme is susceptible to message integrity attacks. However, if one "whitens" the complete output with a random sequence, the scheme becomes secure against message integrity attacks. Whitening just refers to xor-ing the output with a random sequence. The random sequence could be generated by running the block cipher on $r + 1$, $r + 2$, ... $r + m$ (but with a different shared key). This requires $m$ additional cryptographic operations, and hence is no more efficient than generating a MAC.

The efficiency of the new mode comes from proving that the output whitening random sequence need only be pair-wise independent. In other words, if the output whitening sequence is $s_1$, $s_2$,...$s_m$, then each $s_i$ is required to be random, but only pairwise-independent of the other entries. Such a sequence is easily generated by performing only $\log m$ cryptographic operations like block encryption. A simple algebraic scheme can also generate such a sequence by performing only two cryptographic operations.

In fact, an even weaker condition than pair-wise independence suffices. A sequence of uniformly distributed $n$-bit random numbers $s_1$, $s_2$,...$s_m$, is called *pair-wise differentially-uniform* if for every $n$-bit constant c, and every pair $i$, $j$, $i \neq j$, probability that $s_i \oplus s_j$ is $c$ is $2^{-n}$. We show that the output whitening sequence need only be pair-wise differentially-uniform. A simple algebraic scheme can generate such a sequence by performing only one cryptographic operation.

The pair-wise independent sequence generated to assure message integrity can also be used to remove chaining from the encryption mode while still assuring confidentiality. This results in a mode of operation for authenticated encryption which is highly parallelizable. Once again, we show that a pair-wise differentially-uniform sequence suffices to guarantee security of both confidentiality and authentication in this parallelizable version.

Recently and independently, Gligor and Donescu ([7]) also described a mode of operation similar to CBC (but not the parallelizable mode) which has built-in message integrity, although with a slightly weaker security bound than our construction.

The rest of the paper is organized as follows. Section 2 describes the new mode of operation. Section 3 gives definitions of random permutations, and formalizes the notions of security, for both confidentiality and message integrity. In section 4 we prove that the new (parallelizable) scheme is secure for message integrity. In section 5 we state the secrecy theorem of the new mode of operation.

## 2 The New Modes of Operation

We begin by defining two properties of sequence of random numbers which are slightly weaker than the well known pair-wise independence property. The first property also appeared in [8].

### 2.1 Pairwise Differentially-Uniform Random Numbers

**Definition 2.1** (pair-wise differentially-uniform): A sequence of uniformly distributed $n$-bit random numbers $s_1, s_2, ..., s_z$, is called *pair-wise differentially-uniform* if for every $n$-bit constant c, and every pair $i$, $j$, $i \neq j$, probability that $s_i \oplus s_j$ is $c$ is $2^{-n}$.

**Definition 2.2** A sequence of random numbers $s_1$,$s_2$,...,$s_z$ uniformly distributed in GFp, is called *pair-wise differentially-uniform in* GFp if for every constant $c$ in GFp, and every pair $i$, $j$, $i \neq j$, probability that $(s_i - s_j) \bmod p$ is $c$ is $1/p$.

### 2.2 The New Modes – IACBC and IAP

Now we describe the new modes of operation for encryption, which also guarantee message integrity. We will describe the parallelizable mode in more detail, as it is for this mode that we provide detailed proofs in this paper.

The mode similar to CBC is called **IACBC** for *integrity aware cipher block chaining*. It is described in Fig 1. The parallelizable mode is called **IAPM** for *integrity aware parallelizable mode*. It is described in Fig 2. We now give more details for IAPM. After reading the details for IAPM, the definition of IACBC will be clear from Fig 1.

Let $n$ be the block size of the underlying block cipher (or pseudo-random permutation). For now we assume that if the block cipher requires keys of length $k$, then this mode of operation requires two keys of length $k$ . Let these keys be called $K1$ and $K2$. From now on, we will use $f_x$ to denote the encryption function under key $x$. The same notation also holds for pseudo-random permutations.

The message to be encrypted $P$, is divided into blocks of length $n$ each. Let these blocks be $P_1, P_2, ...P_{z-1}$. As in CBC, a random initial vector of length $n$ (bits) is chosen. This random vector $r$ is expanded into $t = O(\log z)$ new random vectors $W_1, ...W_t$ using the block cipher and key $K2$ as follows:

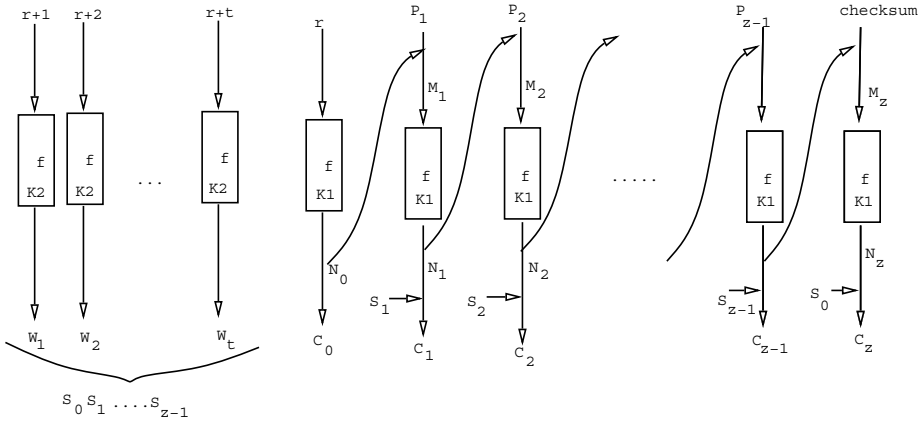


**Fig. 1.** Encryption with Message Integrity (IACBC)

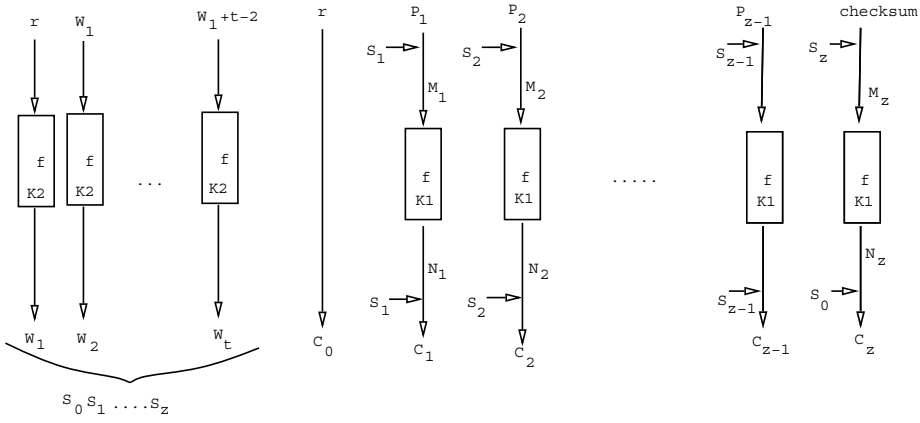


**Fig. 2.** Parallelizable Encryption with Message Integrity (IAPM)

$W_1 = f_{K2}(r)$
for $i = 2$ to $t$ do
$W_i = f_{K2}(W_1 + i - 2)$
end for

As we will show in section 4, with high probability, the $t$ vectors are independent. The $t$ random and independent vectors are used to prepare $z + 1$ new pair-wise differentially-uniform random vectors $S_0, S_1, ..., S_z$. There are several ways to generate such a sequence, some requiring $t$ to be only one. Such a scheme will be described towards the end of this section. For now, consider the following method using subsets ($t = \lceil \log(z + 2) \rceil$):

for $i = 1$ to $2^t - 1$ do
Let $< a_1, a_2, ...a_t >$ be the binary representation of $i$

$S_{i-1} = \sum_{j=1}^{t} (a_j \cdot W_j)$
end for

The summation in the for loop above is an xor-sum.
The cipher-text message $C = < C_0, C_1, ..., C_z >$ is generated as follows (see Figure 2). The encryption pseudo-code follows:

$C_0 = r$
for $i = 1$ to $z - 1$ do
$M_i = P_i \oplus S_i$
$N_i = f_{K1}(M_i)$
$C_i = N_i \oplus S_i$
end for
checksum $= \sum_{i=1}^{z-1} P_i$
$M_z =$ checksum $\oplus S_z$
$N_z = f_{K1}(M_z)$
$C_z = N_z \oplus S_0$

Again, the summation above is an xor-sum. Note that $S_0$ is used in the last step.

It is easy to see that the above scheme is invertible. The inversion process yields blocks $P_1, P_2, ..., P_z$. The decrypted plain-text is $< P_1, P_2, ..., P_{z-1} >$. Message integrity is verified by checking $P_z = P_1 \oplus P_2 \oplus ... \oplus P_{z-1}$.

The random vectors $W_1, ...W_t$ can also be generated as in Fig 1, in which case $C_0$ is set to $f_{K1}(r)$ (instead of $r$).

There are many other ways of generating the pair-wise differentially-uniform vectors $S_0, S_1, ..., S_z$ ($z < 2^n$). One could generate a sequence of pairwise differentially uniform vectors by an algebraic construction in GFp as follows: generate two random vectors $W_1$, and $W_2$, and then let $S_i = (W_1 + W_2 * i) \bmod p$, where $p$ is a prime of appropriate size. For example, if the block cipher has block size 64 bits, $p$ could be chosen to be $2^{64} - 257$. This leads to a fast implementation.

A sequence of $2^n - 1$ $n$-bit uniform random numbers, which are pair-wise differentially uniform, can also be generated by viewing the $n$-bit numbers as elements of GF($2^n$). Consider, $S_i = e(i) \cdot W$, where $W$ is a random number in GF($2^n$), $e(i)$ is a one to one function from $\mathcal{Z}_{2^n-1}$ to non-zero elements of GF($2^n$), and the multiplication is in GF($2^n$). Then $S_i$ is a pair-wise differentially uniform sequence of uniformly distributed random numbers. Note that this requires generation of only one $W$ (i.e. $t = 1$).

The GFp construction with only one $W$, instead of two, is not pair-wise differentially uniform (as opposed to the previous construction in GF($2^n$)). However, it is pair-wise differentially uniform in GFp (see definition 2.2). More precisely, the sequence $S_i = (W_1 * i)$ *mod* $p$, is pair-wise differentially uniform in GFp (assuming $W_1$ is uniformly distributed in GFp). Such a sequence can be used securely in a slight variant of the mode described above where "whitening" now refers to addition modulo $2^n$ (see section 4.2).

## 3 Encryption Schemes: Message Security with Integrity Awareness

We give definitions of schemes which explicitly define the notion of secrecy of the input message. Of course, pseudo-random permutations can be used to build encryption schemes which guarantee such message secrecy ([2], [12]).

In addition, we also define the notion of message integrity. Moreover, we allow arbitrary length input messages (upto a certain bound).

Let Coins be the set of infinite binary strings. Let $l(n) = 2^{O(n)}$, and $w(n) = O(n)$. Let $\mathcal{N}$ be the natural numbers.

**Definition** A (probabilistic, symmetric, stateless) encryption scheme with message integrity consists of the following:

- **initialization:** All parties exchange information over private lines to establish a private key $x \in \{0,1\}^n$. All parties store $x$ in their respective private memories, and $|x| = n$ is the security parameter.
- **message sending with integrity:**

$$\text{Let } E : \{0,1\}^n \times \text{Coins} \times \mathcal{N} \times \{0,1\}^{l(n)} \to \{0,1\}^{l(n)} \times \mathcal{N}$$

$$D : \{0,1\}^n \times \mathcal{N} \times \{0,1\}^{l(n)} \to \{0,1\}^{l(n)} \times \mathcal{N}$$

$$\text{MDC} : \mathcal{N} \times \{0,1\}^{l(n)} \to \{0,1\}^{w(n)}$$

be polynomial-times function ensembles. In $E$, the third argument is supposed to be the length of the plain-text, and $E$ produces a pair consisting of cipher-text and its length. Similarly, in $D$ the second argument is the length of the cipher-text. We will drop the length arguments when it is clear from context. The functions $E$ and $D$ have the property that for all $x \in \{0,1\}^n$, for all $P \in \{0,1\}^{l(n)}$, $c \in$ Coins

$$D_x(E_x(c, P)) = P \| \text{MDC}(P)$$

We will usually drop the random argument to $E$ as well, and just think of $E$ as a probabilistic function ensemble.It is also conceivable that MDC may depend on Coins, cipher-text.

**Definition** (*Security under Find-then-Guess* [2]) Consider an adversary $A$ that runs in two stages. During the adversary's find stage he endeavors to come up with a pair of equal length messages, $P^0$, $P^1$, whose encryptions he wants to tell apart. He also retains some state information $s$. In the adversary's guess stage he is given a random cipher-text $y$ for one of the plain-texts $P^0$, $P^1$, together with $s$. The adversary is said to "win" if he correctly identifies the plain-text.

An Encryption Scheme is said to be $(t, q, \mu, \epsilon)$-secure in the find-then-guess sense, if for any adversary $A$ which runs in time at most $t$ and asks at most $q$ queries , these totaling at most $\mu$ bits,

$$Adv_A \stackrel{\text{def}}{=} 2 \cdot Pr[(P^0, P^1, s) \leftarrow A^{E_x(\cdot)}(\text{find});\ b \leftarrow \{0,1\};\ y \leftarrow E_x(P^b) : A^{E_x(\cdot)}(guess, y, s) = b] - 1 \ \leq \ \epsilon$$

The following notion of security is also called *integrity of ciphertext* ([5]). **Definition** (*Message Integrity):* Consider an adversary $A$ running in two stages. In the first stage (*find*) $A$ asks $r$ queries of the oracle $E_x$. Let the oracle replies be $C^1, ...C^r$. Subsequently, $A$ produces a cipher-text $C'$, different from each $C^i$, $i \in [1..r]$. Since $D$ has length of the cipher-text as a parameter, the breakup of $D_x(C')$ as $P'\|P''$, where $|P''| = w(n)$, is well defined. The adversary's success probability is given by

$$\text{Succ} \stackrel{\text{def}}{=} Pr[\text{MDC}(P') = P'']$$

An encryption scheme is secure for message integrity if for any adversary A, A's success probability is negligible.

## 4 Message Integrity

In this section we show that the mode of operation IAPM in Fig 2 guarantees message integrity with high probability.

In the following theorem, we will assume that the block cipher (under a key $K1$) is a a random permutation $F$. We also assume that the $t$ $W$'s are generated using an independent random permutation $G$ (for instance, using a different key $K2$ in a block cipher).

Let the adversary's queries in the first stage be $p^1, P^2, ...P^m$. We write $p^1$ in lower case, as for each adversary $p^1$ is fixed. All random variables will be denoted by upper case letters. Let the corresponding ciphertexts be $C^1, ..., C^m$. We will use $C$ to denote the sequence of ciphertext messages $C^1, ..., C^m$. For all random variables corresponding to a block, we will use superscripts to denote the message number, and subscripts to denote blocks in a particular message. Thus $C^i_j$ will be the random variable representing the $j$th block in ciphertext message $i$. More precisely, this variable should be written $C^i_j(F, G)$, as it is a function of the two permutations. However, we will drop the arguments when it is clear from context.

Let the adversary's query in the second stage be cipher-text $C'$, different from all ciphertexts in the first stage. We will use primed variables to denote the variables in the second stage.

We will use $W$ to denote the set of variables $\{W^i_j : i \in [1..m], j \in [1..t]\} \cup \{W'_j, j \in [1..t]\}$. We will use $S^i$ ($S'$) to denote masks or "whitening" blocks generated using $W^i$ ($W'$ resp). Any method can be used to generate $S^i$ from $W^i$, as long as $S^i_j$ are pairwise differentially uniform. For a particular adversary, $S^i_j$ is a function of permutation $G$ and the initial vector, and hence should (more precisely) be written as $S^i_j(G, C^i_0(F, G))$ ($C^i_0(F, G)$ being the IV used to generate $W^i_1$). But, we will drop the arguments as it will be clear from context. For any constant $r$, we will denote by $S^i_j(r)$ the random variable $S^i_j(G, r)$.

The variables $M$ and $N$ are as in Fig 2. For example, $M^i_j = P^i_j \oplus S^i_j$.

We start with some informal observations to aid the reader in the eventual formal proof. Since the new ciphertext $C'$ is different from all old ciphertexts,

it must differ from each old ciphertext $C^i$ in a least block number, say $d(i)$. For each $C^i$ (except at most one $C^k$), the block number $d(i) = 0$, with high probability. In Lemma 3 we show that with high probability $N'_{d(k)}$ is different from all old $N^i_j$, and all other new $N'$ blocks (except for a special case). Thus, $M'_{d(k)}$ is random. Then it follows (Theorem 1) that in either case the checksum is unlikely to validate.

We first prove the theorem for schemes in which the pairwise differentially uniform sequence is generated using only one $W$, i.e. $t = 1$. The general case is addressed in a later subsection.

**Theorem 1.** *Let $A$ be an adversary attacking the message integrity of IAPM ($t = 1$) with random permutations $F$ and $G$. Let $A$ make at most $m$ queries in the first stage, totaling at most $\mu$ blocks. Let $u = \mu + m$. Let $v$ be the maximum number of blocks in the second stage. Then for adversary $A$,*

$$Succ < (2 * u^2 + m^2 + (m+1)^2 + u + v + 2 + o(1)) * 2^{-n}$$

**Proof:**

In the first stage the adversary makes queries with a total of at most $m$ plaintext messages (chosen adaptively). W.l.o.g. assume that the adversary actually makes exactly $m$ total message queries in the first stage. Let $L^i$ be the random variable representing the length of ciphertext $C^i$ (i.e. the checksum block has index $L^i - 1$). Similarly, $L'$ will denote the length of $C'$.

We prove that either the adversary forces the following event E0, or the event E1 happens with high probability. In either case the checksum validates with low probability.

The first event E0 is called deletion attempt, as the adversary in this case just truncates an original ciphertext, but retains the last block.

**Event** E0 (*deletion attempt*): There is an $i \in [1..m]$, such that $2 \leq L' < L^i$, and

$$(i)\ \ \forall j \in [0..L'-2] : C'_j = C^i_j$$

$$\text{and } (ii)\ \ C'_{L'-1} = C^i_{L^i-1}$$

Event E1 says that there is a block in the new ciphertext $C'$, such that its $N$ variable is different from all previous $N$s (i.e. from original ciphertexts from the first stage), and also different from all other new $N$s.

**Event** E1: there is an $x \in [1..L'-1]$ such that

$$(i)\ \ \forall s \in [1..m] \forall j \in [1..L^s - 1]\ : N'_x \neq N^s_j$$

$$\text{and } (ii)\ \ \forall j \in [1..L'-1], j \neq x\ : N'_x \neq N'_j$$

We next show that in both cases (i.e E0 or E1) the checksum validates with low probability.

For the case that E0 happens, we have (since $S' = S^i$ and $N'_{L'-1} = N^i_{L^i-1}$),

$$(\sum_{j=1}^{L'-1} P'_j = 0) \wedge \mathrm{E0} \Rightarrow \sum_{j=1}^{L'-2} (P^i_j) + M^i_{L^i-1} + S^i_{L'-1} = 0$$
$$\equiv \sum_{j=1}^{L'-2} (P^i_j) + \sum_{j=1}^{L^i-2} (P^i_j) + S^i_{L^i-1} + S^i_{L'-1} = 0$$

Note that $r^i$ can be chosen after $P^i$ has been determined (as $P^i$ is a deterministic function of $C^1, \ldots, C^{i-1}$), and hence the $S^i$s are independent of $P^i$. Since the $S^i$s are pairwise differentially uniform and $L' < L^i$, the above event happens with probability at most $2^{-n}$.

For the case E1, by Lemma 2, the checksum validates with probability at most $1/(2^n - u - v)$

Thus the adversary's success probability is upper bounded by

$$\Pr[\neg(E0 \vee E1)] + \frac{1}{2^n - (u+v)} + \frac{1}{2^n}$$

which by Lemma 3 is at most

$$(u^2 + m^2 + u + v + 2) * 2^{-n} + (u^2 + (m+1)^2) * 2^{-n} + O(u+v) * 2^{-2n}$$

□

**Lemma 2:** $\Pr[\sum_{j=1}^{L'-1} P'_j = 0 \mid E1] \leq \frac{1}{2^n - (u+v)}$

*Proof:* $F$ being a random permutation, under E1, $F^{-1}(N'_x)$ can not take values already assigned to $F^{-1}(N^s_j)$, $s \in [1..m]$, $j \in [1..L^s - 1]$. Also, $F^{-1}(N'_x)$ can be chosen after $F^{-1}(N'_j)$ have been assigned values ($j \neq x$). Thus, under the condition that event E1 has happened we have that $M'_x = F^{-1}(N'_x)$ can take any of the other values, i.e. excluding the following (at most) $(\mu + m) + L' - 2$ values, with equal probability (independently of $C$, $C'$, $r^i$, $i \in [1..m]$, $G$, and hence independently of $W$, and independent of E1 itself):

- values already taken by $M^s_1, ..., M^s_{L^s-1}$, for each $s$, and
- the values to be taken (or already fixed) by $M'_j$, $j \in [1..L'-1]$, $j \neq x$.

Now, $\sum_{j=1}^{L'-1} P'_j = 0$ iff

$$F^{-1}(N'_x) \;=\; M'_x = \sum_{j=1, j\neq x}^{L'-1} (M'_j \oplus S'_j) \;\oplus S'_x$$

Given any value of the RHS, since the LHS can take (at least) $2^n - (u + v - 2)$ values, the probability of LHS being equal to RHS is at most $1/(2^n - (u+v))$. □

**Lemma 3:** *Let events E0,E1 be as in Theorem 1. Then,*

$$\mathrm{Prob}[\neg(\mathrm{E0} \vee \mathrm{E1})] < (u^2 + m^2 + u + v) * 2^{-n} + (u^2 + (m+1)^2) * 2^{-n}$$

*Proof:* We first calculate the probability of event $(E0 \vee E1)$ happening under the assumption that $F$ and $G$ are random functions (instead of random permutations). Since $F$ (and $G$) is invoked only $u$ times ($(m+1)$ times resp.), a standard

argument shows that the error introduced in calculating the probability of event (E0 $\vee$ E1) is at most $(u^2 + (m+1)^2) * 2^{-n}$.

We now consider an event, which says that all the $M$ variables are different. The goal is to claim independence of the corresponding $N$ variables, and hence the $C$ variables. However, the situation is complicated by the fact that the condition that all the $M^i_j$ variables for some $i$ are different, may cause the variables $C^{i'}_j$, for $i' < i$, to be no more independent. However, a weaker statement can be proved by induction. To this end, consider the **event** E2($y$), for $y \leq m$:

$$\forall i, i' \in [1..y], \forall j, j', j \in [1..L^i - 1], j' \in [1..L^{i'} - 1], (i,j) \neq (i',j') : \ (M^i_j \neq M^{i'}_{j'})$$

Event E2($m$) will also be denoted by E2.

We also predicate on the event that all the initial variables $C^i_0$ are different. Let $E3$ be the **event** that

$$\forall i, j \in [1..m], i \neq j \ : \ C^i_0 \neq C^j_0$$

For $\overrightarrow{r} = r^1, ..., r^m$, all $r^i$ different, let $E3(\overrightarrow{r})$ be the event that for all $i \in [1..m]$, $C^i_0 = r^i$.

Let $l()$ be the length of the first ciphertext (determined by the adversary). We will use constant $c^i$ to denote strings of arbitrary block length. We will use $c$ to denote the sequence $c^1, ..., c^m$. The function $|\cdot|$ is used below to represent length of a message in blocks. Given a sequence of ciphertext messages $c^1, ..., c^i$, $i \leq m$, let $l(c^1, ..., c^i)$ be the length of the $(i+1)$th ciphertext (which is determined by the adversary, and therefore is a deterministic function of $c^1, ...c^i$). Recall that each ciphertext includes the block $C^i_0$, which is just $r^i$ under $E3(\overrightarrow{r})$. Also, since $C'$ is a deterministic function of $C$, given $c^1, ..., c^m$ let the ciphertext in the second stage be $c'$ with length $l'$. We have

$$\begin{aligned} \Pr[\neg(E0 \vee E1) \ \wedge \ E2 \mid E3(\overrightarrow{r}) \,] \ = \ & \sum_{c^1 : \, |c^1| = l()} \ ... \ \sum_{c^i : \, |c^i| = l(c^{i-1}, ..., c^1)} \ ... \\ ... \ & \sum_{c^m : \, |c^m| = l(c^{m-1}, ..., c^1)} \Pr[\neg(E0 \vee E1) \ \wedge \bigwedge_i C^i = c^i \ \wedge \ E2 \mid E3(\overrightarrow{r}) \,] \end{aligned} \tag{1}$$

In this sum, if for some $i$, $c^i_0 \neq r^i$, then the inside expression is zero. Also, if event E0 holds for $c$ (which determines $c'$), then the inside expression above for that $c$ is zero. So, from now on, we will assume that E0 does not hold for $C = c$. Then, the inside expression above becomes:

$$\begin{aligned} & \Pr[\neg(E0 \vee E1) \wedge \bigwedge_i C^i = c^i \wedge E2 \mid E3(\overrightarrow{r})] \\ & \leq \min_{x \in [1..l'-1]} \Big\{ \sum_{s \in [1..m], j \in [1..|c^s|-1]} \Pr[(N'_x = N^s_j) \wedge \bigwedge_i C^i = c^i \wedge E2 \mid E3(\overrightarrow{r})] \\ & \qquad + \sum_{j \in [1..l'-1], j \neq x} \Pr[(N'_x = N'_j) \wedge \bigwedge_i C^i = c^i \wedge E2 \mid E3(\overrightarrow{r})] \Big\} \end{aligned}$$

For each $s, j$, we have $(N'_x = N^s_j)$ iff $(S'_{x^*} \oplus S^s_{j^*}) = (C'_x \oplus C^s_j)$, where $S'_{x^*}, S^s_{j^*}$ are the masks that are used for these ciphertext blocks. That is, $j^* = j$ if $j < |c^s| - 1$ and $j^* = 0$ otherwise, and similarly $x^* = x$ if $x < l' - 1$ and $x^* = 0$ otherwise (Similarly for $j \neq x$ we have $(N'_x = N'_j)$ iff $(S'_{x^*} \oplus S'_{j^*}) = (C'_x \oplus C'_j)$).

Since each of the summands in the expression above has a conjunct $C = c$ for some constant string $c$ (and since the forged ciphertext $C'$ is a function of $C$), it follows that each of the summands in the first sum can be written as $\Pr[(S'_{x^*}(c'_0) \oplus S^s_{j^*}(c^s_0) = c'_x \oplus c^s_j) \wedge C = c \wedge E2 \mid E3(\overrightarrow{r})]$. Note that $S'_{x^*}(c'_0) \oplus S^s_{j^*}(c^s_0)$ can in some cases be identically zero. As $c$ is some constant string, then $c'_x \oplus c^s_j$ is also constant, and recall that the variables $S(c_0)$ depend only on the choice of $G$. Thus, each of these summands (if $S'_{x^*}(c'_0) \oplus S^s_{j^*}(c^s_0)$ is not identically zero) can be bounded by

$$\begin{aligned}
&\Pr[S'_{x^*}(c'_0) \oplus S^s_{j^*}(c^s_0) = c'_x \oplus c^s_j \ \wedge C = c \wedge E2 \mid E3(\overrightarrow{r})] \\
&= \ \Pr[C = c \wedge E2 \mid S'_{x^*}(c'_0) \oplus S^s_{j^*}(c^s_0) = c'_x \oplus c^s_j \ \wedge E3(\overrightarrow{r})] \\
&\quad * \ \Pr[S'_{x^*}(c'_0) \oplus S^s_{j^*}(c^s_0) = c'_x \oplus c^s_j \mid E3(\overrightarrow{r})] \\
&\leq \ (2^{-n})^{\mu} \ * \ \Pr[S'_{x^*}(c'_0) \oplus S^s_{j^*}(c^s_0) = c'_x \oplus c^s_j \mid E3(\overrightarrow{r})]
\end{aligned}$$

where the last inequality follows by Claim 5 with $\mu = \sum_{i \in [1..m]} (l(c^{i-1}, \ldots, c^1) - 1)$. A similar inequality holds for the summands in the second sum (i.e. $N'_x = N'_j$ case). Thus, by Claim 4, the inside expression in equation (1) is at most $2^{-n\mu} * (u + v) * 2^{-n}$. Since we have $2^{n\mu}$ summands, it follows that

$$\Pr[\neg(E0 \vee E1) \wedge E2 \mid E3(\overrightarrow{r})] \leq (u + v) * 2^{-n}$$

Finally, we calculate $\Pr[\neg(E0 \vee E1)]$

$$\begin{aligned}
&Pr[\neg(E0 \vee E1)] \\
&\leq \Pr[\neg(E0 \vee E1) \wedge E2 \mid E3] + \Pr[\neg E2 \mid E3] + \Pr[\neg E3] \\
&\leq \Pr[\neg E3] + \\
&\quad \sum_{r^1, \ldots, r^m} ((\Pr[\neg(E0 \vee E1) \wedge E2 \mid E3(\overrightarrow{r})] + \Pr[\neg E2 \mid E3(\overrightarrow{r})]) * \Pr[E3(\overrightarrow{r}) | E3]) \\
&\leq m^2 * 2^{-n} + (u + v) * 2^{-n} + (u)^2 * 2^{-n}
\end{aligned}$$

where the last inequality follows by Claim 6. □

**Claim 4:** For each constant $c$ (and its corresponding $c'$) for which event E0 does not hold, and constant $\overrightarrow{r}$ with distinct values, there is an $x \in [1..l' - 1]$ such that

(i) $\forall s \in [1..m] \forall j \in [1..|c^s| - 1]$:
if $S'_{x^*}(c'_0) \oplus S^s_{j^*}(c^s_0)$ is identically zero then $c'_x \oplus c^s_j \neq 0$, otherwise

$$\Pr[S'_{x^*}(c'_0) \oplus S^s_{j^*}(c^s_0) = c'_x \oplus c^s_j \mid E3(\overrightarrow{r})] \leq 2^{-n},$$

(ii) $\forall j \in [1..|l'-1], j \neq x,$:

$$\Pr[S'_{x^*}(c'_0) \oplus S'_{j^*}(c^s_0) = c'_x \oplus c'_j \mid E3(\overrightarrow{r})] \leq 2^{-n}$$

*Proof:* These are the different cases (we will drop the argument from $S^s$ and $S'$ as it will be clear from context):
(**a**) (*New IV*) If for all $i \in [1..m]$, $c'_0 \neq r^i$, then we choose $x = 1$. In that case $N'_1 = N'_j$ is same as $C'_1 \oplus C'_j = S'_1 \oplus S'_{j^*}$, where $j^* = j$ if $j \neq (l'-1)$, and $j^* = 0$ otherwise. Thus, for $j \in [1..l'-1], j \neq x$, since $S'$ is pairwise differentially uniform, probability of $(S'_1 \oplus S'_{j^*} = c'_1 \oplus c'_j)$ is $2^{-n}$ (even under $E3(\overrightarrow{r})$).

Similarly, $N'_1 = N^s_j$ is same as $C'_1 \oplus C^s_j = S'_1 \oplus S^s_{j^*}$, where $j^* = j$ if $j \neq |c^s|-1$, and $j^* = 0$ otherwise. Under event $E3(\overrightarrow{r})$, and the fact that $c'_0$ is different from all $r^i$, we have that $S'_1 \oplus S^s_{j^*}$ is uniformly distributed.
(**b**) There exists a $k$, $k \in [1..m]$ such that $c'_0 = r^k$. For all other $k' \in [1..m]$, $c'_0 \neq r^k$. Thus $S' = S^k$. We have several cases:
(b1) (*truncation attempt*) If $c'$ is a truncation of $c^k$, then we let $x = l'-1$ which is the index of the last block of $c'$.
(b2) (*extension attempt*) If $c'$ is an extension of $c^k$, then we let $x = |c^k|-1$ which is the index of the last block of $c^k$.
(b3) Otherwise, let $x$ be the least index in which $c'$ and $c^k$ are different.

In all the cases (b1), (b2) and (b3), conjunct (ii) is handled as in (a).

In case (b1), $N'_x = N^s_j$ is same as $C'_{l'-1} \oplus S^k_0 = C^s_j \oplus S^s_{j^*}$, where $j^* = j$ if $j \neq |c^s|-1$, and $j^* = 0$ otherwise. Now, for $s = k$, $j^* = 0$ (in which case $S'_0 \oplus S^s_j$ is identically zero), we have $c'_x \oplus c^s_j = c'_{l'-1} \oplus c^k_{|c^k|-1}$. This quantity is not zero, since E0 (the deletion attempt) doesn't hold for $c$. Otherwise, $S'_0 \oplus S^s_{j^*} = S^k_0 \oplus S^s_j$ is uniformly distributed.

In case (b2), $N'_x = N^s_j$ is same as $C'_{|c^k|-1} \oplus S^k_{|c^k|-1} = C^s_j \oplus S^s_{j^*}$, where $j^* = j$ if $j \neq |c^s|-1$, and $j^* = 0$ otherwise. When $s = k$, $j^*$ is never $|c^k|-1$, and hence $S^k_{|c^k|-1} \oplus S^s_{j^*}$ is uniformly distributed.

In case (b3), $N'_x = N^s_j$ is same as $C'_x \oplus S^k_{x^*} = C^s_j \oplus S^s_{j^*}$, where $j^* = j$ if $j \neq |c^s|-1$, and $j^* = 0$ otherwise, and $x^* = x$ if $x \neq (l'-1)$, and $x^* = 0$ otherwise. If $s = k$, and $j^* = x^*$, then either $j^* = x^* = 0$, or $j = x$. In the latter case, $c'_x \oplus c^s_j = c'_x \oplus c^k_x$, which is non-zero as $x$ is the index in which $c'$ and $c^k$ differ. In the former case, $j = |c^k|-1$, and $x = (l'-1)$. In this case, $c'_x \oplus c^s_j = c'_{l'-1} \oplus c^k_{|c^k|-1}$. If this quantity is zero, then since $x\,(= (l'-1))$ was the least index in which $c^k$ and $c'$ differed, event E0 would hold for $c$, leading to a contradiction. In other cases, $S^k_{x^*} \oplus S^s_{j^*}$ is uniformly distributed. □

Recall that $E3(\overrightarrow{r})$ is the event that all $C^i_0$ are distinct (and set to $\overrightarrow{r}$).

**Claim 5**: Let $l_1$ be the length of the first ciphertext. Let $y \leq m$. For any constant lengths $l_i$ $(i \in [2..y])$ and constant strings $c^i$, $(i \in [1..y], |c^i| = l_i)$, and any function $G$ independent of $F$,

$$\Pr[\bigwedge_{i \in [1..y]} C^i = c^i \wedge E2(y) \mid G \wedge E3(\overrightarrow{r})] \leq (2^{-n})^{\mu}$$

where $\mu = \Sigma_{i \in [1..y]}(l^i - 1)$.

*Proof:* The above probability is zero unless for all $i \in [2..y]$, $l^i = l(c^1, ..., c^{i-1})$. From now on, we will assume that the $l^i$ are indeed such.

We do induction over $y$, with base case $y = 0$.
The base case is vacuously true, as $\mu = 0$ and conditional probability of TRUE is 1.
Now assume that the lemma is true for $y$. We prove the lemma for $y + 1$. The explanation for the inequalities is given below the sequence of inequalities.

$$\begin{aligned}
&\Pr[\bigwedge_{i\in[1..y+1]} C^i = c^i \wedge E2(y+1) \mid G \wedge E3(\overrightarrow{r})] \\
&\leq \Pr[C^{y+1} = c^{y+1} \mid \bigwedge_{i\in[1..y]} C^i = c^i \wedge E2(y+1) \wedge G \wedge E3(\overrightarrow{r})] \\
&\quad * \Pr[\bigwedge_{i\in[1..y]} C^i = c^i \wedge E2(y+1) \mid G \wedge E3(\overrightarrow{r})] \\
&\leq (2^{-n})^{l^{y+1}-1} * \Pr[\bigwedge_{i\in[1..y]} C^i = c^i \wedge E2(y) \mid G \wedge E3(\overrightarrow{r})] \\
&\leq (2^{-n})^{\Sigma_{i\in[1..y]}(l^i-1)}
\end{aligned}$$

The second inequality follows because under the condition $E2(y+1)$, all the $M_j^{y+1}$ are different from the previous $M$, and hence the sequence of variables, for all $j \in [1..L^{y+1} - 1]$, $F(M_j^{y+1})$ can take all possible $(2^n)^{(L^{y+1}-1)}$ values, independently of $G$, and $F(M_j^{\leq y})$, and hence also all ciphertext messages till index $t$. Hence, the sequence $C_j^{y+1} = F(M_j^{y+1}) \oplus S_j^{y+1}$ can take all possible values. Moreover, $L^{y+1} = l(c^1, ..., c^y) = l^{y+1}$.

The last inequality follows by induction. □

**Claim 6:** For every fixed $\overrightarrow{r}$ with distinct values,

$$Pr[\neg E2 \mid E3(\overrightarrow{r})] < u^2 * 2^{-n}$$

*Proof:* Recall that Event E2 is

$$\forall i, i' \in [1..m], \forall j, j', j \in [1..L^i], j' \in [1..L^{i'}], (i,j) \neq (i',j') : \ (M_j^i \neq M_{j'}^{i'})$$

Under $E3(\overrightarrow{r})$, we have
(a) The set of variables $\{W_1^i\}$, $i \in [1..m]$, are uniformly random and independent variables.
(b) For each $i$, the variable $W_1^i$ is independent of all ciphertext messages $C^{i'}$, $i' < i$, and hence all plaintext messages $P^{i'}$, $i' \leq i$. This follows because $W_1^i$ can be chosen after $C^{i'}$, $i' < i$ have been chosen.

Given $E3(\overrightarrow{r})$, the probability that event E2 does not happen is at most $(\Sigma_{i\in[1..m]} L^i)^2 * 2^{-n}$, which is at most $u^2 * 2^{-n}$. This is seen as follows:

$$\Pr[M_j^i = M_{j'}^{i'}] \ = \ \Pr[P_j^i \oplus S_j^i = P_{j'}^{i'} \oplus S_{j'}^{i'}] \ = \ \Pr[S_j^i = S_{j'}^{i'} \oplus P_j^i \oplus P_{j'}^{i'}]$$

Without loss of generality, let $i \geq i'$. Then from (b) above it follows that this probability is at most $2^{-n}$ (if $i = i'$, then we also use the fact that the sequence $S$ is pairwise differentially uniform). □

### 4.1 General Case

We now prove the scheme IAPM ($t \geq 1$) secure for message integrity. Here $F$ and $G$ are independent random permutations.

**Theorem 4:** *Let $A$ be an adversary attacking the message integrity of IAPM ($t \geq 1$) with random permutations $F$ and $G$. Let $A$ make at most $m$ queries in the first stage, totaling at most $\mu$ blocks. Let $u = \mu + m$. Let $v$ be the maximum number of blocks in the second stage. Then for adversary $A$,*

$$Succ < (2 * u^2 + 2tm^2 + tm + t^2(m+1)^2 + 3t(2m+1)(u+v) + 2 + o(1)) * 2^{-n}$$

*Proof Sketch:* We first calculate the adversary's success probability assuming that $G$ is a random function. Then, the error introduced in the probability because of this approximation is at most $((t(m+1))^2 * 2^{-n})$.

The differences in the proof from that of Theorem 1 are (i) we can not assume a priori, that the sequence $S^i$ is pairwise differentially uniform, (ii) E3($\overrightarrow{r}$) as defined in Lemma 3 does not imply that $S^i$ is independent of $S^j$, for $i \neq j$, (iii) in proof of Theorem 1, the case of event E0 requires $S^i$ to be pairwise differentially uniform, and (iv) in claim 4 case (a), $S'(c'_0)$ is not necessarily independent of all $S^i(r^i)$.

To this end, Event E3 is now defined to be the event that all entries in the following (multi-) set are different:

$$\{C^i_0, i \in [1..m]\} \cup \{G(C^i_0) + j - 1, i \in [1..m], j \in [1..t-1]\}$$

For $\overrightarrow{r} = r^1, ..., r^m$, all $r^i$ different, let E3($\overrightarrow{r}$) be the event E3 and that for all $i \in [1..m]$, $C^i_0 = r^i$.

For $\overrightarrow{r}$=$r^1, ..., r^m$, all $r^i$ different, $\Pr[\neg\ \text{E3}(\overrightarrow{r})] \leq (2tm^2 + tm) * 2^{-n}$

Under event E3, for all $i \in [1..m]$, the sequence $S^i$ is pairwise differentially uniform, and is independent of $S^j$ ($j \in [1..m]$, $j \neq i$). Now (in Theorem 1) the case of event E0 is also handled under the condition $E3(\overrightarrow{r})$.

In Claim 4, case (a) (i.e. New IV) now requires showing that $S'(c'_0)$ (with $c'_0$ different from all $r^i$) is independent of all $S^i(r^i)$ ($i \in [1..m]$).

Consider the following events (note that $W^i_1 = G(r^i)$):

**Event** E4: $\forall i \in [1..m], \forall j \in [1..t-1]\ : c'_0 \neq W^i_1 + j - 1$.

**Event** E5: $\forall i \in [1..m] : |G(c'_0) - W^i_1| > t\ \wedge\ |G(c'_0) - r^i| > t\ \wedge\ |G(c'_0) - c'_0| > t$

Now given that, for all $k \in [1..m], c'_0 \neq r^k$, and under event E4, it is the case that $c'_0$ has never been an oracle query to $G$, and thus $\Pr[\neg\text{E5} \mid \text{E4} \wedge \text{E3}(\overrightarrow{r})]$ $< 2t(2m+1) * 2^{-n}$. Also, $\Pr[\neg\ \text{E4} \mid \text{E3}(\overrightarrow{r})] \leq mt * 2^{-n}$.

Under events E4, E5 and E3($\overrightarrow{r}$), and $c'_0$ different from all $r^i$, $S'(c'_0)$ is indeed independent of previous $S^i(r^i)$, and is also pairwise differentially uniform. □

### 4.2 Modes Using GFp

In another variant of IACBC and IAPM, a pair-wise differentially uniform sequence in GFp is employed for "whitening" the output (and the input for parallel modes). However, now "whitening" refers to adding modulo $2^n$, instead of performing an exclusive-or operation. Theorems 1 and 5 also hold for encryption schemes which employ sequences which are pair-wise differentially-uniform in GFp; the success probabilities, however are now in terms of $2/p$ instead of $1/2^n$. The condition $N'_i = N_j$ would now translate to $C'_i - S_i = C_j - S_j$, which is the same as $S_i - S_j = C'_i - C_j$ (here the subtraction is $n$-bit integer subtraction). It can be shown that if $S_i, S_j$ are independent of $C', C$, then the probability of this event is at most $2/p$.

## 5 Message Secrecy

We state the theorem for security under the Find-then-Guess notion of security. The proof follows standard techniques ([2]).

**Theorem 5:** *Let $A$ be an adversary attacking the encryption scheme IAPM in Figure 2 (with $f$ being a random permutation $F$) in the find-then-guess sense, making at most $q$ queries, totaling at most $\mu$ blocks. Then,*

$$Adv_A \leq (2\mu^2) \cdot \frac{1}{2^n}$$

## 6 Security of IACBC

**Theorem 6:** *Let $A$ be an adversary attacking the message integrity of IACBC with random permutations $F$ and $G$. Let $A$ make at most $m$ queries in the first stage, totaling at most $\mu$ blocks. Let $u = \mu + m$. Let $v$ be the maximum number of blocks in the second stage. Then for adversary $A$,*

$$Succ < (2 * (u+1)^2 + 2tm^2 + t^2(m+1)^2 + 3tmu + 2(u+v+1) + 2 + o(1)) * 2^{-n}$$

Theorem 5 continues to hold for IACBC. Proofs of theorem 5, 6 and IACBC variant of theorem 5 will be given in the full version of the paper.

**Acknowledgment**

I am extremely grateful to Shai Halevi and Pankaj Rohatgi for help with the proof of message integrity. I would also like to thank Pankaj for helping me simplify the overall scheme, and Shai for going through the paper in excruciating detail and making numerous helpful suggestions.

I would also like to thank Don Coppersmith, Johan Hastad, Nick Howgrave-Graham, J.R. Rao, Ron Rivest, Phil Rogaway, and referees for helpful suggestions.

## References


1. ANSI X3.106, "American National Standard for Information Systems - Data Encryption Algorithm - Modes of Operation", American National Standards Institute, 1983.
2. M. Bellare, A. Desai, E. Jokiph, P. Rogaway, "A Concrete Security Treatment of Symmetric Encryption: Analysis of the DES Modes of OPeration", 38th IEEE FOCS, 1997
3. J. Black, S. Halevi, H. Krawczyk, T. Krovetz and P.Rogaway, "UMAC: Fast and secure message authentication", *Advances in Cryptology-Crypto 99*, LNCS 1666, 1999
4. M. Bellare, J. Kilian, P. Rogaway, "The Security of Cipher Block Chaining", CRYPTO 94, LNCS 839, 1994
5. M. Bellare, C. Namprempre, "Authenticated Encryption: Relations among notions and analysis of the generic composition paradigm", Proc. Asiacrypt 2000, T. Okamoto ed., Springer Verlag 2000
6. V.D. Gligor, P.Donescu, "Integrity Aware PCBC Encryption Schemes", 7th Intl. Workshop on Security Protocols, Cambridge, LNCS, 1999
7. V.D. Gligor, P. Donescu, "Fast Encryption Authentication: XCBC Encryption and XECB Authentication Modes", `http://csrc.nist.gov/encryption/modes/workshop1`
8. Hugo Krawczyk, "LFSR-based Hashing and Authentication", Proc. Crypto 94. LNCS 839, 1994
9. ISO 8372, " Information processing - Modes of operation for a 64-bit block cipher algorithm", International Organization for Standardization, Geneva, Switzerland, 1987
10. ISO/IEC 9797, "Data cryptographic techniques - Data integrity mechanism using a cryptographic check function employing a block cipher algorithm", 1989
11. J. Katz and M. Yung, "Unforgeable Encryption and Adaptively Secure Modes of Operation", Fast Software Encryption 2000.
12. M. Luby, "Pseudorandomness and Cryptographic Applications", Princeton Computer Science Notes, Princeton Univ. Press, 1996
13. C.H. Meyer, S. M. Matyas, "Cryptography: A New Dimension in Computer Data Security", John Wiley and Sons, New York, 1982
14. National Bureau of Standards, NBS FIPS PUB 81, "DES modes of operation", U.S. Department of Commerce, 1980.
15. National Bureau of Standards, Data Encryption Standard, U.S. Department of Commerce, FIPS 46 (1977)
16. RFC 1510,"The Kerberos network authentication service (V5)", J. Kohl and B.C. Neuman, Sept 1993
17. Security Architecture for the Internet Protocol, RFC 2401, `http://www.ietf.org/rfc/rfc2401.txt`
18. S.G. Stubblebine and V.D. Gligor, "On message integrity in cryptographic protocols", Proceedings of the 1992 IEEE Computer Society Symposium on Research in Security and Privacy, 1992.
19. The TLS Protocol, RFC2246, `http://www.ietf.org/rfc/rfc2246.txt`