Integrity-Aware Parallelizable Cipher Feedback Mode for Real-time Cryptography

Prosanta Gope
Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan, R.O.C
Email: prosanta.nitdgp@gmail.com

Conventional Cipher Feedback Mode (CFB) can allow the transmission unit to be shorter than the block-cipher length. Eventually, it causes no delay and even any message expansion unlike the ECB and CBC mode of operation where encryption cannot begin unless and until a complete block of full-length (say 64 bits) plain-text data is available. However, because of stalling during the block encryption, CFB cannot provide low latency, low jitter; these are two imperative properties in the sense of real-time cryptography. For that, it is important that the input stream should not wait for the key-stream to be generated; that means, key-streams are required to be arranged in advance, which cannot be expected in case of the conventional CFB mode. Besides, the conventional Cipher Feedback Mode is also incompetent for such real-time crypto systems, where the integrity of the message is also greatly desirable along with privacy. In this article, we propose a variant of Cipher Feedback Mode, called, Integrity-Aware, Parallelizable Cipher Feedback Mode (IAP-CFB), which can guarantee all the aforesaid requirements, such as, low latency, low jitter, privacy, and integrity assurance, etc.

Keywords: Real-time cryptography, Integrity-Aware, Parallelizable, Cipher feedback mode.

1. MOTIVATION AND REQUIREMENTS

Examples of real-time applications requiring security include wireless communications (like mobile communication), distributed managements of distributed networks, access and control of remote sites (physical security management, medical equipment's), etc. In general, typical real-time cryptography requirements differ significantly from the conventional cryptography in a number of ways, where a real-time cryptography often demands the following properties:

Integrity awareness: Detection of message corruption is essential, particularly for actions with serious consequences.

Low latency: Input to output delay which is more imperative than throughput.

Low jitter: Low jitter denotes the processing time for each message packet should be same. There is little or no more time for per message key scheduling.

Parallelism: Encryption and decryption process in the real-time crypto-system should also guarantee parallelism.

Unfortunately, basic stream cipher modes (like CFB, CTR, and OFB) used in real-time applications can only provide privacy without integrity protection. Besides, because of stalling during the block encryption the conventional Cipher Feedback Mode [1-2] cannot even ensure the properties like low latency and low jitter, which are indeed essential in the sense of real-time cryptography. Furthermore, in CFB, the current cipher-text unit is fed back to the shift register for generating the key-stream output for the very next input of the plain-text unit. Accordingly, we cannot expect another imperative characteristic called parallelism, in the conventional Cipher Feedback Mode.

1.1 Necessity of Integrity Awareness in Real-time System

Real-time system usually needs to prevent message forgeries and unauthorized message modification. Corrupt control messages can cause disasters directly. Integrity can be supported by including the predictable values in the (extended) plain-text message. The classical way of achieving this is by appending a cryptographic hash of the plain-text of the message. On the other hand, a less computationally costly alternative is also possible when the cipher provides suitable feedback of the plain-text into the subsequent cipher-text, eventually affecting an expected value at the end of the message. In many real-time systems, specifically, those involving at retrofit or roll over, existing frame check data can be included in the encryption as predictable postfix integrity value.

1.2 Our Idea

Here, we introduce the concept of the single pass Authenticated Encryption (AE) [3-13], which is basically a cryptographic approach, where privacy and integrity can be assured together in a

single pass, with almost-free additional computational burden. Certainly, it is dissimilar to all of the three (E&M, MTE, and ETM) traditional AE approaches of generic compositions mentioned in [14], where, encryption and authentication are performed separately. While, in case of our single pass Authenticated Encryption, encryption and authentication can proceed in parallel. In that case, we need not require any MAC or CRC to be produced for integrity checking. Now, to construct an integrity-aware, parallelizable CFB mode, here at first we introduce a tactic to convert the conventional CFB into a single pass Authenticated Encryption mode. In order to do that, and to make it suitable for real-time environment, we introduce the concept of intentional delay, i.e. *t* of M_i blocks in the resultant system, where *t* denotes the time required for each block of encryption. In other words, we can say that because of the intentional delay *t* of M_i blocks, the plain-text inputs appearing at $M_{t+1}, M_{t+2}, \ldots, M_n$ need not to wait for the key-stream to be generated. In fact, that helps the proposed mode of operation to ensure low latency, low jitter, parallelism, and even helps to provide the integrity awareness as well.

Therefore, the remainder of this article is organized as follows. In Section 2, we present our proposed our real-time based our integrity-aware, parallelizable CFB mode, called IAP-CFB. A relevant discussion based on the security evidence and the performance of the proposed IAP-CFB mode of operation is presented in Section 3 and Section 4, respectively. Finally, the concluding remarks are given in Section 5.

2. INTEGRITY-AWARE PARALLELIZABLE CIPHER FEEDBACK MODE

In this section, we propose a new single pass authenticated encryption mode, called integrity-aware parallelizable Cipher Feedback mode (IAP-CFB), which can fulfill the aforesaid requirements of the real-time environment. Now, assuming that both the encryption and decryption are being done on a regular basis and the encryption and decryption algorithm for the message $\mathrm{M} < M_1, M_2, \ldots, M_n >$, consisting of *n* number of *r*-bit blocks, where the parameters *n, r,* along with the key (*K*) size can vary depending upon the block cipher that is used. Now, we assume that the communication system used here is the *r*-bit transmission units, more precisely, IAP-CFB (shown in Fig.1) uses *p*–bit of shift registers consisting of $x = p/r$ positions $X_1, X_2, \ldots, X_x$ of *r*-bit transmission units, where $x = n + t$. Here, we utilize the concept of the

intentional delay (mentioned earlier), to support the integrity of the message by feeding the previous plain-text M_{i-t} of $r < p$ bits in the subsequent one. Now, if we consider that the delay is t of M_i that means, before the appearance of plain-text inputs $M_{t+1}, M_{t+2},..., M_n$ for generating the cipher-texts $C_{t+1}, C_{t+2},..., C_n$ the desired key-stream outputs $O_{t+1}, O_{t+2},..., O_n$ will be ready. Besides, the similar scenario can also be seen during decryption, where before appearance of the cipher-texts input $C_{t+1}, C_{t+2},..., C_n$, the desired key-stream outputs $O_{t+1}, O_{t+2},..., O_n$ will be arranged. In other words, it can be argued that our key-stream is real-time. Now, at the beginning the initial shift register X_1 starts with an IV + 1 value of p-bit, where IV denotes the initial vector. Then during the period of intentional delay the shift register $X_2,..., X_t$ will contain the incremented value of the IV, where the IV is updated through the counter interface Δ. After the delay, in order to complete the rest of the operations, IAP-CFB updates the shift register X_{t+1}, by extracting the right most $(p - r)$ bits of X_t and appending C_1 to the right most side of the r-bit and the similar operation will continue for the rest of the shift register $X_{t+2},..., X_x$, where $C_2,..., C_n$ will be appending as the right most r-bit. Besides, after the intentional delay, each register contents from X_{t+1} to X_x will be XORed with the plain-texts appeared in M_{i-t} and during the XOR operation, the rightmost $(p - r)$ bits of every plain-text M_{i-t} is required to be padded with 0's. Thereafter, the resultant XOR outputs are encrypted using a block algorithm (say AES), and then the MSB_r of the outputs O_{t+1} to O_n are XORed with the real-time input plain-texts M_{t+1} to M_n on the basis of their arrival. Eventually, that will constitute the cipher-text outputs C_{t+1} to C_n. Here, the final t cipher-texts are being used as indicators, which specify if there is any change in cipher-stream in transit that must be reflected on several subsequent plain-texts and simultaneously at least on one of the indicators $C_{n+1}, C_{n+2},..., C_x$ at the decryption end, where $x = n + t$. In other words, based on the parameter t in intentional delay, exactly equal numbers of indicators will be produced. As a result of that for any message $M < M_1, M_2,..., M_n >$, the cipher-stream $C < C_1, C_2,..., C_x >$ is generated. The encryption and decryption algorithm of the proposed IAP-CFB mode of operation can be represented as follows

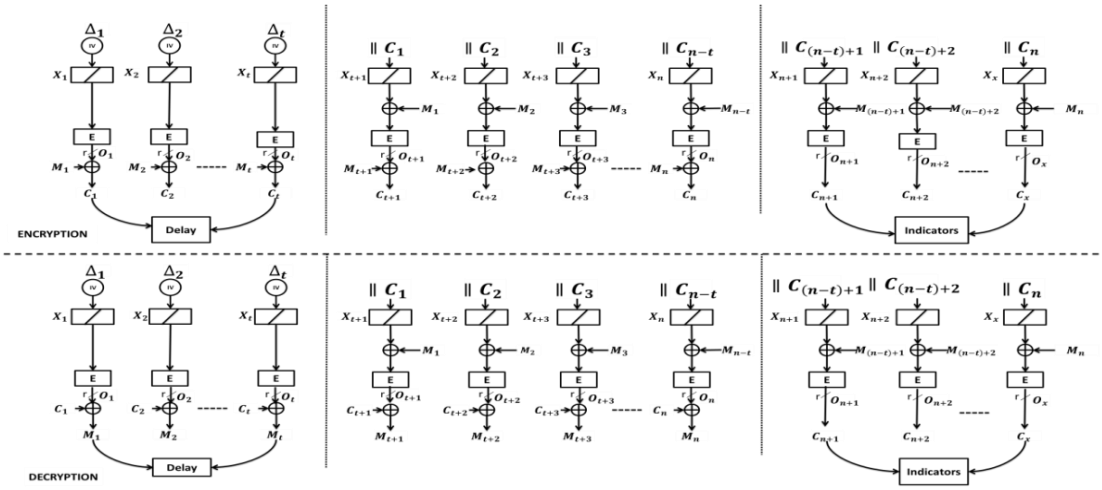

Fig. 1. Integrity-Aware Parallelizable Cipher Feedback Mode

Algorithm. Encryption and Decryption of the IAP-CFB

Algorithm $Enc_K^{IAP-CFB}(M_1, M_2, ..., M_n)$
Begin
$\Delta_0 \leftarrow Init(\text{IV})$// *initial value*
//*considering delay is t of* M_i
for $i = 1$ *to* x ***do*** // $x = n + t$
  while $i \le t$ ***then do***
    $Inc_i \leftarrow \Delta_{i-1} + 1$
    $X_i \leftarrow Inc_i(\Delta_i)$
    $O_i \leftarrow Enc_K(\Delta_i)$
    $C_i \leftarrow M_i \oplus MSB_r(O_i)$
    $i \leftarrow i + 1$
  end of while
  while $i > t$ && $i \le n$ ***then do***
    $ShiftReg_r(X_{i-1})$
    $X_i \leftarrow (X_{i-1} \parallel C_{i-t})$
    $O_i \leftarrow Enc_K(M_{i-t} \| 0...0 \oplus X_i)$
    $C_i \leftarrow M_i \oplus MSB_r(O_i)$
    $i \leftarrow i + 1$
  end of while
// *for indicators*
  while $i > n$ && $i \le x$ ***then do***
    $ShiftReg_r(X_{i-1})$
    $X_i \leftarrow (X_{i-1} \parallel C_{i-t})$
    $O_i \leftarrow Enc_K(M_{i-t} \| 0...0 \oplus X_i)$
    $C_i \leftarrow MSB_l(O_i)$// *where* $l > r \lesssim p$
    $i \leftarrow i + 1$
  end of while
end for
return $C_1, C_2, ..., C_n, C_{n+1}, ..., C_x$
End

Algorithm $Dec_K^{IAP-CFB}(C_1, C_2, ..., C_n)$
Begin
$\Delta_0 \leftarrow Init(\text{IV})$// *initial value*
//*considering delay is t of* M_i
for $i = 1$ *to* x ***do*** // $x = n + t$
  while $i \le t$ ***then do***
    $Inc_i \leftarrow \Delta_{i-1} + 1$
    $X_i \leftarrow Inc_i(\Delta_i)$
    $O_i \leftarrow Enc_K(\Delta_i)$
    $M_i \leftarrow C_i \oplus MSB_r(O_i)$
    $i \leftarrow i + 1$
  end of while
  while $i > t$ && $i \le n$ ***then do***
    $ShiftReg_r(X_{i-1})$
    $X_i \leftarrow (X_{i-1} \parallel C_{i-t})$
    $O_i \leftarrow Enc_K(M_{i-t} \| 0...0 \oplus X_i)$
    $M_i \leftarrow C_i \oplus MSB_r(O_i)$
    $i \leftarrow i + 1$
  end of while
// *for indicators*
  while $i > n$ && $i \le x$ ***then do***
    $ShiftReg_r(X_{i-1})$
    $X_i \leftarrow (X_{i-1} \parallel C_{i-t})$
    $O_i \leftarrow Enc_K(M_{i-t} \| 0...0 \oplus X_i)$
    $C_i \leftarrow MSB_l(O_i)$// *where* $l > r \lesssim p$
    $i \leftarrow i + 1$
  end of while
end for
return $M_1, M_2, ..., M_n, C_{n+1}, ..., C_x$
if Check$(C_{n+1}, C_{n+2}, ..., C_x)$ is **true** ***then return M***
else return **INVALID**
End

In the above algorithm of IAP-CFB, the size of each indicator is *l*-bit, where $l > r \lesssim p$. Here, $ShiftReg_r(X_{i-1})$ denotes the operation of the shift register where left most *r* bits contents of the

previous shift register X_{i-1} are shifted left. Whereas $(X_{i-1} \parallel C_{i-t})$ denotes the appending of the previous $(i - t)$th cipher-text C_{i-t} at the right most *r*-bit position of the shift register X_{i-1}. Now, like the conventional CFB, here also decryption does not involve for calling the decryption function, this would be advantage of running a block cipher in the stream cipher in a stream mode in a case where the decryption function for the block algorithm is slower than the encryption.

3. SECURITY CONSIDERATIONS

In this section, we provide security evidence for the proposed IAP-CFB mode of operation against the following security issues, and those are indeed essential for any secure encryption scheme.

- Left-or-Right Security chosen plain-text attack denoted by (LOR-CPA).
- Integrity of cipher-text denoted by (INT-CTXT).
- Indistinguishability of encryptions under the chosen cipher-text attack denoted by (IND-CCA).

3.1 Left-or-Right Security (LoR)

LoR security was first introduced by Bellare et al. in [15] as a strong form of CPA security. The attack can be implemented as a game between an active adversary (left-right distinguisher) $\mathcal{A}$ and an encryption oracle $\mathcal{E}_{K,b}$, which contains a key *K* and a bit $b \in \{0, 1\}$. In each iteration, $\mathcal{A}_{lor}$ chooses two plain-texts M_i^0 and M_i^1 with $|M_i^0| = |M_i^1|$ and gives them to $\mathcal{E}_{K,b}$. The encryption oracle return $C_i = Enc_K(M_i^b)$, where the cases b = 0 and b = 1 are called left and right case. At the end, $\mathcal{A}_{lor}$ outputs a bit e, meant as a guess at b. The adversary's advantage $\mathcal{A}dv_{lor}$ is defined as the probability difference of output $e = 0$ in the two cases. Now, the adversary's resources are parameterized by its maximum running time *t*, the number of queries *q* and their total length μ, where the maximum probability of success is ε.

Definition 3.1 (LoR Security). *An encryption scheme (Gen, Enc, Dec) is* (t, q, μ, ε) *secure in the left-or-right sense if for any adversary* $\mathcal{A}_{lor}$ *which runs in the time at most t and ask at most q queries, totaling at most* μ bits.

$$\mathcal{A}dv_{lor} = \Pr[\mathcal{A}_{lor}^{\mathcal{E}_{K,0}} = 0 \mid K \leftarrow Gen] - \Pr[\mathcal{A}_{lor}^{\mathcal{E}_{K,1}} = 0 \mid K \leftarrow Gen] \leq \varepsilon$$

The above definition describes the probability of that $\mathcal{A}_{lor}$ outputs $e = 0$ when interacting with the oracle containing b = 0, and b = 1. Now, the LoR security of the proposed scheme breaks down at the first repetition of the value of shift register *X* and when the adversary has the full command on every plain-texts feedback M_{i-t} along with the values in *X*. Hence, if $X_i = X_j$ (shift register positioning after the intentional delay *t* of M_i) for $i \neq j$, and the plain-text feedback inputs (M_{i-t}, M_{j-t}), XORed to X_i and X_j are also equal and which implies $O_i = O_j$. Their encryption results in an equal string value $C_i \oplus C_j = M_i^b \oplus M_j^b$. Hence, an adversary can win the LoR game $M_i^0 \oplus M_j^0 \neq M_i^1 \oplus M_j^1$ when all the aforesaid condition holds. Therefore, we stress that the security of IAP-CFB is bounded by the birthday paradox i.e. until repetition of the value of the shift register *X*, and the input plain-texts feedback i.e. $M_{i-t} = M_{j-t}$. However, it will be quite challenging for the adversary $\mathcal{A}_{lor}$ to have full command on every plain-texts feedback M_{i-t}, where *t* may always vary.

Lemma 3.1: [Security of IAP-CFB with RF]*: Let* $\Re^{p, r}$ *be a random function family such that, for any t, q and* $\mu \leq r.q$*, the input feedback plain-texts* $M_{i-t} = M_{j-t}$*, where* $i \neq j$*, then the advantages of an adversary* $\mathcal{A}$ *attacking the CPA privacy of the IAP-CFB, instantiated with* $\Re$*, denoted by IAP-CFB* [$\Re$]*, is*

$$\mathcal{A}dv^{lor}_{IAP-CFB[\Re]}(t, q, \mu) \leq \frac{q(q-1)}{2^{p+1}}$$

PROOF: Conceive, the probabilities in the LoR game with bit b as $\Pr_b$, so, for instance, the advantage of the adversary can be written as $\mathcal{A}dv_{lor} = \Pr_0[e = 0] - \Pr_1[e = 0]$. We distinguish whether a collision occurs during the attack or not. Let $\mathbb{C}$ be the collision event, it contains all executions of the game where $i \neq j$ exist with $1 \leq i,\ j \leq q$ and $X_i = X_j$, the input feedback of plain-texts $M_{i-t} = M_{j-t}$. Its complement is called $\overline{\mathbb{C}}$. If there is no collision then each $O_i = f(X_i)$ is considered as randomly and independently chosen. That will cause C_i to be random and independent of $C_1, ..., C_{i-1}$ and $M_1^b, ..., M_i^b$. Hence, we can say that collision probability in round *i* does not depend on b and overall we can write

$$\Pr[\mathbb{C}] = \Pr_0[\mathbb{C}] = \Pr_1[\mathbb{C}] \qquad \text{(i)}$$

Eventually, collisions will help adversary, as, if there is no collision occurs, then the adversary outputs $e = 0$ with the same probability for b = 0 and b = 1.

$$\Pr_0[e = 0 \mid \overline{\mathbb{C}}] = \Pr_1[e = 0 \mid \overline{\mathbb{C}}] \qquad \text{(ii)}$$

Now, from (i) and (ii) we can derive adversary's advantages as follows:

$$\begin{aligned}
\mathcal{A}dv_{lor} &= \Pr_0[e = 0] - \Pr_1[e = 0] \\
&= \Pr_0[e = 0 \mid \mathbb{C}]\ \Pr_0[\mathbb{C}] + \Pr_0[e = 0 \mid \overline{\mathbb{C}}]\Pr_0[\overline{\mathbb{C}}] \\
&\quad - \Pr_1[e = 0 \mid \mathbb{C}]\ \Pr_1[\mathbb{C}] - \Pr_1[e = 0 \mid \overline{\mathbb{C}}]\Pr_1[\overline{\mathbb{C}}] \\
&= \Pr[\mathbb{C}](\Pr_0[e = 0 \mid \mathbb{C}] - \Pr_1[e = 0 \mid \mathbb{C}])
\end{aligned}$$

So, $\mathcal{A}dv_{lor} \leq \Pr[\mathbb{C}]$

Now, for the collision probability, it is irrelevant to merely use the birthday formula because X_i and X_j are not independent if $|j - i| < n$ and that implies the overlapping of X_i and X_j. We define the stream $S = \frac{\Delta}{\text{IV}} C_1, ..., C_{q-1}$ of all the collision-relevant transmission units those are shifted through X_t until the end, *q*th encryption. $\frac{\Delta}{\text{IV}}$ denotes the incremented value of the IV (based on the delay *t*). The length of the S is $L = (n + q - 1)p$ bits, and the shift register contents

are X_i = S[i],…S[i + n –1] for i =1,...,q. Now, we derive the number $col_{i,j}$ of streams with a collision $X_i = X_j$, where the input feedback of plain-texts $M_{i-t} = M_{j-t}$ for possible pair (i, j), when $1 \le i < j \le q$.

Without overlapping, where $j \ge i + n$:

As $X_i = X_j$, so, there are 2^p values for the shift register contents of both rounds. Remaining (S–2p) bits offer 2^{S-2p} possible values. Accordingly, $col_{i,j} = 2^p.2^{S-2p} = 2^{S-p}$.

With overlapping, where $i < j < i + n$:

Considering, z = j – i. Then X_i and X_j together use p + zr bits, and those have 2^{zr} possible values. The rest S – p – zr bits provide 2^{S-p-zr} possibilities. Hence, the $col_{i,j} = 2^{zr}.2^{S-p-zr} = 2^{S-p}$.

There are q (q -1) /2 possible pairs (i, j). Accordingly, the number of col of streams S with at least one collision less than $q(q-1)2^{S-p-1}$. Thus $\Pr[\overline{\mathbb{C}}] = (2^S - col) / 2^S > 1 - q(q-1)2^{-p-1}$.

So, we can write, $\Pr[\mathbb{C}] \le \mathcal{A}dv^{lor-cpa}_{IAP-CFB[\Re]}(t, q, \mu) \le \frac{q(q-1)}{2^{p+1}}$.

Above proof is based on the assumption that, the input feedback of plain-texts $M_{i-t} = M_{j-t}$. However, if they are not equal, then that will certainly effect on the possibility of the occurrences of collision. Precisely, the results of the operations ($M_{i-t} \oplus X_i$ and $M_{j-t} \oplus X_j$) while $MSB_r|M_{i-t}| \ne MSB_r|M_{j-t}|$, $X_i = X_j$ and $i \ne j$, are expected to be diffused during the block encryption and eventually that will also constitute the resultant outputs $O_i \ne O_j$. That may eventually makes the adversary's task even more difficult and simultaneously improves the CPA security of the proposed IAP-CFB. Besides, this also implies that, the input feedback plain-text $MSB_r|M_{i-t}|$ does not deteriorate the security of the conventional Cipher Feedback mode, in fact, surely improves its integrity level.

3.2. Integrity of Cipher-text (INT-CTXT)

INT-CTXT (integrity of cipher-text) requires that it be computationally infeasible to produce a cipher-text not previously produced by the sender. In general, INT-CTXT can be achieved through an unforgeable integrity token. Now, if we consider that the set of integrity tokens $\mathbf{I} \in \{C_{n+1},...,C_x\}$ (Indicators)} used in IAP-CFB, as an authentic and unforgeable one, which can be defined as follows.

Definition 3.2 [Authenticity of the set of Integrity Tokens I] *Assuming that the set of integrity tokens* $\mathbf{I} \in \{C_{n+1}...C_x\}$ *be a triple of efficient algorithms i.e.* **I**= (*GEN, TAGS, VER*). *where* **I** *is considered to be a secure and unforgeable one if an adversary* $\mathcal{A}$ *is not able to make a successful existential forgery, meaning to produce a valid* $(M' < M_1, M_2, ..., M_n >, T' < C_{n+1},...,C_x >)$ *at the decryption end by changing in any cipher* C_j *of* $C' < C_1, C_2, ..., C_n >$ *under chosen cipher message attack in time t, with q number of queries*

$$Adv_{\mathbf{I}}^{auth}(t,\ q) = \Pr[VER_{\mathrm{K}}\ (\mathcal{A}_{TAGS_{\mathrm{K}(.,.)}}) = 1 \mid K \leftarrow GEN]\ \leq \frac{q(q-1)}{2^{p+1}}.$$

In the above definition, the parameter *GEN* denotes the key generation algorithm, whereas the parameter *TAGS* specifies the generation of the set of integrity tokens during encryption (T_{Enc}) and decryption (T_{Dec}), which can be expressed as follows:

while $i > n\ \&\&\ i \leq x$ ***then do***

$ShiftReg_r(X_{i-1})$

$X_i \leftarrow (X_{i-1} \parallel C_{i-t})$

$O_i \leftarrow Enc_K(M_{i-t} \| 0...0 \oplus X_i)$

$C_i \leftarrow MSB_l(O_i)$// *where* $l > r \lesssim p$

$i \leftarrow i + 1$

end of while

Here, the verification process (*VER*) is carried out by prudently checking the value of the indicators $C_{n+1},...,C_x$, in other words, by verifying that whether each $T_{Enc_i} = T_{Dec_i}$ or not. Now, by paraphrasing our definition we can say that an adversary $\mathcal{A}$ forges the set integrity tokens **I** if,

without prior knowledge of the key $K \leftarrow GEN$, by changing any desired cipher-text C_j of C' $< C_1, C_2, ..., C_n >$ he is able to produce a desired valid message $M' < M_1, M_2, ..., M_n >$ along with the authentic integrity token set T' at the decryption end such that $T_{Enc} = T_{Dec} = T'$, which is only possible if the adversary can distinguish whether a collision occurs or not, where the advantage of $\mathcal{A}$ is $\frac{q(q-1)}{2^{p+1}}$ (**already proved in lemma 3.1**). Hence, we claim that the proposed scheme IAP-CFB along with the integrity token **I**, is secure as under any key *K* that the adversary cannot forge a cipher-text in time *t* with probability better than $\frac{q(q-1)}{2^{p+1}}$. In this way, the IAP-CFB can assure INT-CTXT (integrity of cipher-text) and simultaneously INT-PTXT (integrity of plain-text). The Bellare et al. [14] already proved the implication INT-CTXT $\rightarrow$ INT-PTXT using the Theorem 3.1, which can be restated as Theorem 1 shown below.

Theorem 1 *Let IAP-CFB = (K, E, D) be an encryption scheme. Then for any adversary* $\mathcal{A}$, $Adv^{int\text{-}ptxt}_{IAP-CFB}(\mathcal{A}) \leq Adv^{int\text{-}ctxt}_{IAP-CFB}(\mathcal{A})$.

So far, we have shown that the proposed IAP-CFB mode of operation can assure INT-CTXT, INT-PTXT, along with the LOR-CPA security. Bellare et al. [14] already proved the implication INT-CTXT ^ LOR-CPA → IND-CCA, which specifies that encryption scheme that is both IND-CPA secure and INT-CTXT secure, is also IND-CCA secure. Accordingly, we can argue that IAP-CFB is IND-CCA secure. Moreover, as the IAP-CFB can resist IND-CCA, which also implies NM-CPA, accordingly, the proposed IAP-CFB mode of operation can provide NM-CPA security [16-17].

4. CONCLUSION

In this article, we have identified the various requirements for real-time cryptography. Based on that, a single pass, parallelizable Authenticated Encryption mode IAP-CFB has been designed, which can guarantee to fulfill those unique requirements. Subsequently, we have analyzed the

security of the IAP-CFB mode of operation, where we have seen that the proposed scheme can ensure various imperative security properties like LOR-CPA, IND-CCA, etc.

REFERENCES

[1] Schneier, B. (1996) Applied Cryptography, John Wiley & Sons, New York, 2nd edition, 197-211.

[2] ISO/IEC 9797. Data cryptographic techniques–Data integrity mechanism using a cryptographic check function employing a block cipher algorithm. 1989.

[3] Jutla, C. (2001) Encryption modes with almost free message integrity. In *Advances in Cryptology–EUROCRYPT 2001*, B. Pfitzmann, Ed., Vol. 2045 of *Lecture Notes in Computer Science*, Springer-Verlag, , 529-544.

[4] Rogaway, P (2004). Efficient Instantiations of tweakable Blockciphers and Refinements to Modes OCB and PMAC, Proceeding of the ASIACRYPT 2004, LNCS, Vol. 3329, 16-31, Springer, Heidelberg.

[5] Rogaway, P. Bellare, M. and Black, J. (2003) OCB: A block-cipher mode of operation for efficient authenticated encryption. ACM Transactions on Information and System Security (TISSEC) 6.3 365- 403.

[6] Gligor, V. Donescu, P. (2001) Fast Encryption and Authentication: XCBC Encryption and XECB Authentication Modes, 2nd NIST Workshop on AES Modes of Operation, Santa Barbara, USA.

[7] Iwata, T. Kurosawa, K. (2003) OMAC: One–key CBC MAC, Proceedings of Fast Software Encryption 2003, LNCS vol. 2887, Springer–Verlag.

[8] Bellare, M. Rogaway, P. Wagner, D (2004). The EAX Mode of Operation Proceedings of Fast Software Encryption 2004, LNCS vol 3017, Springer-Verlag.

[9] Kohno, T. Viega, J. Whiting, D. (2004) CWC: A High-Performance Conventional Authenticated Encryption Mode, Proceedings of Fast Software Encryption 2004, LNCS Vol 3017, Springer-Verlag.

[10] Ferguson, N. Whiting, D. Schneier, B. Kelsey, J. Lucks, S. and Kohno, T.(2003). Helix: Fast encryption and authentication in a single cryptographic primitive. In Fast Software Encryption, 10th International Workshop, FSE 2003, T. Johansson, Ed., Lecture Notes in Computer Science, Springer-Verlag.

[11] Muller, F. (2004) Differential Attacks against the Helix Stream-cipher, Proceedings of Fast Software Encryption 2004, LNCS Vol. 3017, Springer–Verlag.

[12] Watanabe, D. Furuya, S. (2004) A MAC forgery attack on SOBER-128, Proceedings of Fast Software Encryption 2004, LNCS Vol. 3017, Springer–Verlag.

[13] Driscoll, K. (2002) Beep-Beep: Embedded real-time encryption, Proceedings of Fast Software Encryption 2002, LNCS Vol. 2365, pp. 164–178, Springer-Verlag Berlin Heidelberg.

[14] Bellare, M. Namprempre, C. (2008) Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. Journal of Cryptology, 21(4), 469–491.

[15] Bellare, M. Desai, A. JokiPii, E. and Rogaway, P. (1997) 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. A revised version is available online at http://www-cse.ucsd.edu/users/mihir

[16] Dolev, D. Dwork, C. and Naor, M. Non-malleable cryptography, *Proc.* 23rd *Annual Symposium on the Theory of Computing*.

[17] Bellare, M. Desai, A. Pointcheval, D. and Rogaway, P. (1998) Relations among notions of security for public-key encryption schemes, *Proc. Advances in Cryptology - CRYPTO'98*,

LNCS, vol. 1462. Springer-Verlag.