mixFeed

Designers/Submitters:
Bishwajit Chakraborty - Indian Statistical Institute,Kolkata
Mridul Nandi - Indian Statistical Institute, Kolkata, India

bishu.math.ynwa@gmail.com
mridul.nandi@gmail.com

September 21, 2019

1 Introduction

In this document, we propose a new scheme for authenticated encryption with associated data (AEAD) based on AES'128/128 [7] block cipher. Here, we introduce a new mode which we call *Minimally Xored Feedback* mode (mixFeed) based on any block cipher with some involved key-scheduling algorithm. Our mode (on top of the n-bit block cipher) requires only n-bit xor to process each n-bit blocks. The name can also be justified for the fact that we use a mixture of Plaintext and Ciphertext as the feedback to the underlying blockcipher.

Another aspect of the mixFeed is that, we use nonce-dependent key. This would help to get higher security beyond conventional model (such as reasonable security against leakage of nonce-dependent key).

2 mixFeed Specification

2.1 Notations and Conventions

We fix positive even integers n, κ, and t to denote the *block size*, *key size*, and *tag size* respectively in bits. Our input nonce size is one byte less than the block size. $\mathsf{E}n/\kappa$ denotes a block cipher family E, parametrized by the block length n and key length κ. In this paper we use AES'128/128 and so $n = 128$, nonce size $= 120$ and $\kappa = 128$. Note that AES'128/128 is same as the original AES128/128 except that we use mixcolumn operation at the last round.

We fix the tag size to be 128. Note that one can always truncate the tag to a small size if required.

We use $\{0,1\}^+$ and $\{0,1\}^n$ to denote the set of all non-empty (binary) strings, and n-bit strings, respectively. λ denotes the empty string and $\{0,1\}^* = \{0,1\}^+ \cup \{\lambda\}$. For all practical purposes: we use little-endian format of indexing, and assume all binary strings are *byte-oriented*, i.e. belong in $(\{0,1\}^8)^*$. For any string $B \in \{0,1\}^+$, $|B|$ denotes the number of bits in B, and for $0 \le i \le |B|-1$, b_i denotes the i-th bit of B, i.e. $B = b_{|B|-1} \cdots b_0$. where b_0 is the least significant bit (LSB) and $b_{|B|-1}$ is the most significant bit (MSB). Given a nonempty bit string B of size $x < n$, we denote $\mathsf{pad}(B)$ as $0^{n-x-1}1B$. Thus we always pad the extra bits from MSB side. When $x = n$, we define $\mathsf{pad}(B)$ as B itself. The chop function chops either the most significant or least significant bits. For $k \le n$, and $B \in \{0,1\}^n$, $\lfloor B \rfloor_k := B_{k-1} \ldots B_0$ and $\lceil B \rceil_k := B_{n-1} \ldots B_{n-k}$.

For $B \in \{0,1\}^+$, $(B_{\ell-1}, \ldots, B_0) \overset{n}{\leftarrow} B$, denotes the n-bit *block parsing* of B into $(B_{\ell-1}, \ldots, B_0)$, where $|B_i| = n$ for $0 \le i \le \ell-2$, and $1 \le |B_{\ell-1}| \le n$. For $A, B \in \{0,1\}^+$, and $|A| = |B|$, $A \oplus B$ denotes the "bitwise XOR" operation on A and B. For $A, B \in \{0,1\}^+$, $A \| B$ denotes the "string concatenation" operation on A and B.

We will use a compact representation of if-else statement by the following expression $P\ ?\ b\ :\ c$ where P is some mathematical statement. This evaluates to b if P is true and c otherwise. $P_1\ \&\ P_2\ ?\ b_1\ :\ b_2\ :\ b_3\ :\ b_4$ evaluates to b_1 if both P_1 and P_2 are true, to b_2 if only P_1 is true, to b_3 if only P_2 is true and to b_4 if none of P_1 , P_2 are true.

Block Cipher: A block cipher with key size κ and block size n is a family of permutations over n-bits indexed by κ bit key. For a fixed key $k \in \{0,1\}^\kappa$, we write $E_k(\cdot) = E(k, \cdot)$. Many block cipher uses some non-trivial key-scheduling algorithm which produces round keys for each round to mask the block cipher state. Let ϕ corresponds to the function which updates the key. In other words, if K is the key of the block cipher for the current execution, $\phi(K)$ will denote the updated key. We will see details of this key update function for AES'128/128 in more details later.

2.2 Our Recommendation

In Algorithm 1 we describe our specification mixFeed based on any block cipher E. We propose (primary submission) mixFeed where E is instantiated by AES'128/128 where the last round also calls MixColumns operation of AES128/128. For the sake of completeness we describe it in Algorithm 2. This does not change any security level of AES'128/128, but it adds uniformity over all rounds.

2.3 Provenance of Constants used in Tweak Control

Our mode uses a 4-bit constant $t_3 \parallel t_2 \parallel t_1 \parallel t_0$ for processing the last block of associated data and the last block of message which distinguishes different cases regarding completeness of the last blocks. This constant value is decided from the inputs of the hardware API and are explained as follows.

- eoi : t_3 is called the end of input control bit. This bit is set to 1 if and only if the current data block being processed is the final block of the input. For all other data block processing t_3 is set to 0.

Algorithm 1 Encryption/Decryption algorithm in mixFeed. Here, λ denotes the empty string. $\bot, \top$ denotes the abort and accept symbols respectively. By $*$, we mean that the exact value is not bothered.

```
function $\textsc{mixFeed}_{[\mathsf{E}]}.\mathsf{enc}(K, N, A, M)$
    $((a, \delta_A), (m, \delta_M)) \leftarrow \mathsf{Fmt}(A, M)$
    if $a = 0, m = 0$ then
        $(T, *) \leftarrow \mathsf{E}_K(N\|0^6 10)$
        return $(\lambda, T)$
    else if $a = 0$ then $(K_N, *) \leftarrow \mathsf{E}_K(N\|0^7 1)$
    else $(K_N, *) \leftarrow \mathsf{E}_K(N\|0^8)$
    $(T, K) \leftarrow \mathsf{E}_{K_N}(N\|0^8)$
    $C \leftarrow \lambda$
    if $a \neq 0$ then $(*, T, K) \leftarrow \mathsf{proc\_txt}(T, K, A, \delta_A, +)$
    if $m \neq 0$ then $(C, T, *) \leftarrow \mathsf{proc\_txt}(T, K, M, \delta_M, +)$
    return $(C, T)$

function $\textsc{mixFeed}_{[\mathsf{E}]}.\mathsf{dec}(K, N, A, C, T)$
    $((a, \delta_A), (m, \delta_C)) \leftarrow \mathsf{Fmt}(A, C)$
    if $a = 0, m = 0$ then
        $(T', *) \leftarrow \mathsf{E}_K(N\|0^6 10)$
        return $(T = T')?\ \top : \bot$
    else if $a = 0$ then $(K_N, *) \leftarrow \mathsf{E}_K(N\|0^7 1)$
    else $(K_N, *) \leftarrow \mathsf{E}_K(N\|0^8)$
    $(T', K) \leftarrow \mathsf{E}_{K_N}(N\|0^8)$
    $M \leftarrow \lambda$
    if $a \neq 0$ then $(*, T', K) \leftarrow \mathsf{proc\_txt}(T', K, A, \delta_A, +)$
    if $m \neq 0$ then $(M, T', *) \leftarrow \mathsf{proc\_txt}(T', K, C, \delta_C, -)$
    if $T \neq T'$ then
        return $\bot$
    else
        return $(M, \top)$
```

```
function $\mathsf{Fmt}(A, M)$
    $(A_{a-1}, \ldots, A_0) \xleftarrow{n} A$
    $(M_{m-1}, \ldots, M_0) \xleftarrow{n} M$
    $\delta_A \leftarrow (n \mid |A_{a-1}|)\ \&\ (m = 0)?\ 12 : 4 : 14 : 6$
    $\delta_M \leftarrow (n \mid |M_{m-1}|)?\ 13 : 15$
    return $((a, \delta_A), (m, \delta_M))$

function $\mathsf{proc\_txt}(K_1, Y_0, D, \delta_D, dir)$
    $(D_{d-1}, \ldots, D_0) \xleftarrow{n} D$
    for $i = 0$ to $d - 1$ do
        $(X_{i+1}, D'_i) \leftarrow \mathsf{Feed}(Y_i, D_i, dir)$
        $(Y_{i+1}, K_{i+2}) \leftarrow \mathsf{E}_{K_{i+1}}(X_{i+1})$
    $X_{d+1} \leftarrow Y_d \oplus 0^{n-4}\|\delta_D$
    $(Y_{d+1}, K_{d+2}) \leftarrow \mathsf{E}_{K_{d+1}}(X_{d+1})$
    return $(D', Y_{d+1}, K_{d+2})$

function $\mathsf{Feed}(Y, D, \mathrm{dir})$
    $D' \leftarrow D \oplus \lfloor Y \rfloor_{|D|}$
    if $\mathrm{dir} = \text{"}+\text{"}$ then
        $B \leftarrow \lceil \mathsf{pad}(D') \rceil_{n/2} \| \lfloor \mathsf{pad}(D) \rfloor_{n/2}$
    if $\mathrm{dir} = \text{"}-\text{"}$ then
        $B \leftarrow \lceil \mathsf{pad}(D) \rceil_{n/2} \| \lfloor \mathsf{pad}(D') \rfloor_{n/2}$
    $X \leftarrow B \oplus Y$
    return $(X, D')$
```

- eot: t_2 is called the end of type control bit. This bit is set to 1 if and only if the current data block being processed is the last block of the same type i.e. it is the last block of message/ associated data. For all other data block processing t_2 is set to 0.
- partial: t_1 is called the partial control bit. this bit is set to 1 if data block currently being processed is a partial block, i.e. it's the data size is less than the required block size. For all other data block processes it is set to 0.
- Type: t_0 is called the type control bit and it identifies the data being processed. For the final message block processing, t_0 is set to 1. For all other data processing, t_0 is set to 0.

While processing a last data block of a type, the input of the block cipher is decided based on the 4 control bits. Fmt function outputs the δ_A, δ_M values by simply giving the integer representation of $t_3\|t_2\|t_1\|t_0$. For example if we are in the last message block and it is partial then $t_3 = 1, t_2 = 1, t_1 = 1, t_0 = 1$, making $\delta_M = 15$ In Algorithm 1. Similarly if we are processing the last associate data block which is complete and the message length is non-zero, then $t_3 = 0, t_2 = 1, t_1 = 0, t_0 = 0$ making $\delta_M = 4$.

i	1	2	3	4	5	6	7	8	9	10	11
RCON(i)	01	02	04	08	10	20	40	80	1b	36	6c

Table 1: The RCON Values

3 Security of mixFeed

Here we describe some possible strategies to attack the mixFeed mode, and give a rough estimate on the amount of data and time required to mount those attacks (see Table 2). In the following discussion:

Algorithm 2 AES'128/128 Block Cipher. To apply a chain of block cipher, we perform an extra round of AES'128/128 Key-Schedule and use that round key as the initial key of the next call of AES'128/128. As described in the Introduction the second output of Emodule only depends on the first input K and we define this function as $\phi(K)$.

```
function $\mathsf{E}(K; X)$
    $(W_{47}, \ldots, W_0) \leftarrow \mathsf{KeyGen}(K)$
    for $i = 1$ to $10$ do
        $X \leftarrow X \oplus (W_{4i-1}, W_{4i-2}, W_{4i-3}, W_{4i-4})$
        $X \leftarrow \mathsf{SubBytes}(X)$
        $X \leftarrow \mathsf{ShiftRows}(X)$
        $X \leftarrow \mathsf{MixColumns}(X)$
    $X \leftarrow X \oplus (W_{43}, W_{42}, W_{41}, W_{40})$
    $K \leftarrow (W_{47}, W_{46}, W_{45}, W_{44})$
    return $(X, K)$

function $\mathsf{KeyGen}(K)$
    $(K_{15}, \ldots, K_0) \overset{8}{\leftarrow} K$
    for $i = 0$ to $3$ do
        $W_i \leftarrow (K_{4i+3}, K_{4i+2}, K_{4i+1}, K_{4i})$
    for $i = 4$ to $47$ do
        $Y \leftarrow W_{i-1}$
        if $i\%4 = 0$ then
            $Y \leftarrow \mathsf{SubWords}(Y \lll 8)$
            $Y \leftarrow Y \oplus \mathsf{RCON}_{i/4}$
        $W_i \leftarrow W_{i-4} \oplus Y$
    return $(W_{47}, \ldots, W_0)$

function $\mathsf{SubBytes}(X)$
    $(X_{15}, \ldots, X_0) \overset{8}{\leftarrow} X$
    for $i = 0$ to $15$ do
        $X_i \leftarrow \mathrm{AS}(X_i)$
    return $X$

function $\mathsf{Shiftrows}(X)$
    $(X_{15}, \ldots, X_0) \overset{8}{\leftarrow} X$
    for $i = 0$ to $3$ do
        for $j = 0$ to $3$ do
            $Y_{4i+j} \leftarrow X_{4i+((j+i)\%4)}$
    return $Y$

function $\mathsf{MixColumns}(X)$
    $M \leftarrow \begin{pmatrix} 2 & 3 & 1 & 1 \\ 3 & 1 & 1 & 2 \\ 1 & 1 & 2 & 3 \\ 1 & 2 & 3 & 1 \end{pmatrix}$
    $Y \leftarrow M \cdot X$
    return $Y$
```

- D denotes the data complexity of the attack. This parameter quantifies the online resource requirements, and includes the total number of blocks (among all messages and associated data) processed through the underlying block cipher for a fixed master key. Note that for simplicity we also use D to denote the data complexity of forging attempts.

- T denotes the time complexity of the attack. This parameter quantifies the offline resource requirements, and includes the total time required to process the off line evaluations of the underlying block cipher. Since one call of the block cipher can be assumed to take a constant amount of time, we generally take T as the total number of off line calls to the block cipher.

Security Model	Data complexity ($\log_2 D$)	Time complexity ($\log_2 T$)
IND-CPA	60	112
INT-CTXT	50	112

Table 2: Security Claims. We remark that the given values indicate the amount of data or time required to make the attack advantage close to 1.

Notes on Security on the Modes After making q queries with σ many blocks, adversary observes inputs and outputs of the block cipher with a key which is dependent on the nonce and the current block number. Thus the security of this construction would depend on the nultikey set up. As the least significant 64 bits of inputs are random (during encryption), the multi-key attack (in the ideal cipher model) will have advantage roughly $\sigma T/2^{192}$ where T is the number of ideal cipher calls and σ is the number of encryption blocks. Similar argument will work for all decryption attempts.

We must admit that there is no conventional privacy security in case of nonce misuse.

3.1 Known Security Analysis of AES'128/128

The security of AES'128/128 is same as the security of AES128/128 as mixcolumn is a linear operation which can be peeled off from the output. The security of AES128/128 is well-established in the community.

To the best of our knowledge, the best single-key attack on AES128/128 is the biclique attack by Bogdanov et al. [1], that recovers the key in approx. 2^{126} computations. Although there is a related-key attack on full-round AES-128/192 and AES-128/256, the same attack does not apply to AES128/128, even in the usual XOR related-key setting, let alone the key scheduled related-keys. In fact, [5] shows that AES128/128 is almost as secure in related-key setting as it is in single-key setting. Recent distinguishers on AES128/128 [3, 4, 8, 2], are applicable to round-reduced variants of AES128/128, and hence not applicable in our case.

4 Design Rationale

4.1 Choice of the Mode

Our primary goal is to design a lightweight cipher that should be efficient, provide high performance and able to perform well in low end devices. In addition, we also demand robustness in security.

4.1.1 Nonce dependent key

At the very first step we compute the secret key based on nonce. So, for every encryption we use random keys. Even though due to some side channel analysis the secret key corresponding to a nonce N is released, the master key remains still secret and all encryption using nonce other than N remains good.

4.1.2 Minimally xored mixture feedback

As our name suggests, we use minimum number of xors to process each block. This makes the design simpler and having very low footprint in software. The rational behind having mixture of plaintext and ciphertext feedback is to achieve NIST aimed security. During encryption we ensure 192 bit entropy for each block process. We have 128 bit dynamic secret key and 64 bits LSB of the inputs have influence from 64 bits LSB of the previous block cipher call.

While decrypt, we have 64 bit MSB of the previous outputs goes to the correspond position of the next input. This would provide about 64 bit security for forgery attempts.

4.1.3 Single State

mixFeed has a state size as small as the block size of the underlying cipher, and it ensures good implementation characteristics both on lightweight and high-performance platforms. We moreover need not to hold the original key as we dynamically update the key based on the key scheduling algorithm used for the block cipher computation.

4.1.4 Inverse-Free

mixFeed is a inverse-free authenticated algorithm. Both encryption and verified decryption of the algorithm do not require any decryption call to the underlying twekable block cipher. This reduces the overall hardware footprint significantly, especially in the combined authenticated-encryption, verified-decryption implementations.

4.2 Choice of the Block cipher

4.2.1 Well analyzed and NIST standard

AES128/128 block cipher is well analyzed for long time and it remains secure. Moreover, in this proposal, a weaker security from AES128/128 would suffice. AES128/128 also performs very well in microcontroller based platform. We note that the last mix-column operation is included in our proposal to make it uniform over all rounds. This reduces additional MUX which was required to process last round for the original AES128/128.

4.2.2 Dynamic Key

We compute the key dynamically as key schedules goes on. This helps us not to hold the master key as well not to expose a secret key multiple times. As the key-scheduling of AES128/128 is involved, the related-key security analysis of AES128/128 expected to be much harder than conventional xor-related key.

5 Preliminaries for Proving Security of mixFeed

Here we define the Different security notions of mixFeed and the Tweakable Block Cipher[6].

5.1 Security Definitions of mixFeed

Let $\mathcal{E}nc_K, \mathcal{D}ec_K$ respectively denote the encryption and decryption algorithms of mixFeed with key K.

5.1.1 Privacy

Given an adversary $\mathscr{A}$ we define the privacy advantage of $\mathscr{A}$ against mixFeed as $\mathbf{Adv}^{priv}_{\mathsf{mixFeed}}(\mathscr{A}) = |\Pr\left[\mathscr{A}^{\mathcal{E}nc_K} = 1\right] - \Pr\left[\mathscr{A}^{\$} = 1\right]|$, where \$ returns a random output string of same length as $\mathcal{E}nc_K$. The *privacy advantage* of mixFeed is defined as

$$\mathbf{Adv}^{priv}_{\mathsf{mixFeed}}(q, \sigma, t) = \max_{\mathscr{A}} \mathbf{Adv}^{priv}_{\mathsf{mixFeed}}(\mathscr{A})$$

where the maximum is taken over all the nonce respecting adversaries $\mathscr{A}$ running in time t and making at most q many encryption queries with total number of blocks in all the queries being σ.

5.1.2 Forgery

We say that a nonce respecting oracle adversary $\mathscr{A}^{\mathcal{E}nc_K, \mathcal{D}ec_K}$ forges mixFeed if $\mathscr{A}$ is able to make a fresh query (N, A, C, T) to $\mathcal{D}_K$ such that $\mathcal{D}_K(N, A, C, T) \neq \perp$. By fresh query we mean that the adversary does not make any previous query (N, A, M) to $\mathcal{E}nc_K$ such that $\mathcal{E}nc_K(N, A, M) = (C, T)$. We say a decryption query valid if $\mathcal{D}_K(N, A, C, T) \neq \perp$. The *forging advantage* of an adversary $\mathscr{A}$ is written as

$$\mathbf{Adv}^{forge}_{\mathsf{mixFeed}}(\mathscr{A}) = \Pr\left[\mathscr{A}^{\mathcal{E}nc_K, \mathcal{D}ec_K} \text{ forges}\right]$$

and we write

$$\mathbf{Adv}^{forge}_{\mathsf{mixFeed}}(q, \sigma, t) = \max_{\mathscr{A}} \mathbf{Adv}^{forge}_{\mathsf{mixFeed}}(\mathscr{A})$$

where the maximum is taken over all adversary $\mathscr{A}$ running in time t, making at most q_e many nonce respecting encryption queries with maximum σ_e many blocks and making at most q_d many decryption queries with maximum σ_d many blocks. Define $q = q_e + q_d$, $\sigma = \sigma_e + \sigma_d$. Note that the decryption queries are not necessarily nonce respecting i.e. nonce can be repeated in the decryption queries and an encryption query and a decryption query can use the same nonce. However, all nonces used in encryption queries are distinct.

5.2 Security Definitions of Tweakable block cipher

5.2.1 TPRP-security

Let $\tilde{E}$ be an n-bit tweakable block cipher with tweak space $\mathcal{T}$. The *TPRP-advantage* of $\tilde{E}$ against an oracle adversary $\mathscr{A}$ is defined as $\mathbf{Adv}^{TPRP}_{\tilde{E}}(\mathscr{A}) = |\Pr\left[\mathscr{A}^{\tilde{E}_K} = 1\right] - \Pr\left[\mathscr{A}^{\tilde{\Pi}} = 1\right]|$ where $\tilde{\Pi}$ is chosen uniformly from the set of all functions $\tilde{\pi} : \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ where for every $tw \in \mathcal{T}$, $\tilde{\pi}(tw, \cdot)$ is a permutation on $\{0,1\}^n$. We call $\tilde{\Pi}$ a tweakable random permutation. We write,

$$\mathbf{Adv}^{TPRP}_{\tilde{E}}(q, t) = \max_{\mathscr{A}} \mathbf{Adv}^{TPRP}_{\tilde{E}}(\mathscr{A})$$

where maximum is taken over all adversaries $\mathscr{A}$ running in time t making q many tweak-input queries of the form (tw, X). We define μ*-TPRP* advantage of $\tilde{E}$ to be

$$\mathbf{Adv}^{\mu\text{-}TPRP}_{\tilde{E}}(q, t) = \max_{\mathscr{A}} \mathbf{Adv}^{TPRP}_{\tilde{E}}(\mathscr{A})$$

where the maximum is taken over all the adversaries $\mathscr{A}$ as defined above with the additional restriction that it is μ-respecting i.e. the number of queries by $\mathscr{A}$ with same input X is at most μ. When the tweakable block cipher is instantiated in the ideal cipher model, the time parameter t denotes the number of ideal cipher calls.

5.2.2 Multi-Commitment Prediction

Let $\tilde{E}$ be a tweakable block cipher. Let $\mathscr{A}$ be an adversary with oracle access to $\tilde{E}$, i.e. it can make queries of the form (tw, X) to $\tilde{E}$ to receive $\tilde{E}_K(tw, X)$. Given Such an adversary $\mathscr{A}$ consider the following game between $\mathscr{A}$ and $\tilde{E}$

PHASE 1 : $\mathscr{A}$ makes queries of the form (tw, X) and receives $Y = \tilde{E}_K(tw, X)$.

PHASE 2 : After all the queries of PHASE 1 is done,

(a) For some $k \leq \lambda$, adversary makes k many commitments of the form (tw_i, x_i, y_i) where $x_i, y_i \in \{0,1\}^{\frac{n}{2}}$.

(b) $\mathscr{A}$ makes at most λ many queries to produce at most λ many prediction tuples of the form $(tw_j, X_j)_{j\in[1,\lambda]}$ such that (tw_j, X_j) are fresh i.e. $\forall j, (tw_j, X_j)$ has never been queried before predicting it.

We say that any adversary $\mathscr{A}$ wins the λ-*multi-commitment-prediction* game if for some prediction tuple (tw^j, X^j) there exist a commitment tuple (tw_i, x_i, y_i) which was committed in Phase 2. step (a), such that

$$tw_i = tw_j; x_i = \lceil X_j \rceil_{\frac{n}{2}}; \lfloor \tilde{E}_K(tw_j, X_j) \rfloor_{\frac{n}{2}} = y_i.$$

The λ-*multi-commitment-predicting* advantage of an adversary $\mathscr{A}$ is defined as

$$\mathbf{Adv}_{\tilde{E}}^{\lambda\text{-mcp}}(\mathscr{A}) = \Pr\left[\mathscr{A}^{\tilde{E}}\text{wins the } \lambda\text{-}\textit{multi-commitment-prediction} \text{ game }\right]$$

and we write,

$$\mathbf{Adv}_{\tilde{E}}^{\lambda\text{-mcp}}(q,t) = \max_{\mathscr{A}} \mathbf{Adv}_{\tilde{E}}^{\lambda\text{-mcp}}(\mathscr{A})$$

where maximum is taken over all adversaries $\mathscr{A}$ running in time t making at most q many queries.

We define (μ, λ)-mcp advantage of $\mathscr{A}$ to be

$$\mathbf{Adv}_{\tilde{E}}^{(\mu,\lambda)\text{-mcp}}(q,t) = \max_{\mathscr{A}} \mathbf{Adv}_{\tilde{E}}^{\lambda\text{-mcp}}(\mathscr{A})$$

where the maximum is taken over all adversaries as defined above with the additional restriction that they make μ-respecting queries in PHASE 1 of the game.

In the ideal cipher model the (λ, μ)-multi commitment prediction security is defined in the same way as above with an additional restriction that the adversary doesn't make any primitive calls to E in PHASE 2.

5.2.3 Multi-Collision

Let $\tilde{E}$ be a tweakable block cipher. Define an oracle $\mathcal{O}_{\tilde{E}}$ which takes a query input of the form (tw, X, C) and returns $X' = C \oplus 0^{\frac{n}{2}} \| \lfloor Y \rfloor_{\frac{n}{2}}$ where $Y = \tilde{E}_K(tw, X)$.We say that an adversary $\mathscr{A}$ with oracle access to $\mathcal{O}$ produces a μ-multicollision if it can produce μ many trascripts of the form $(tw_i, X_i, C_i, X_i')_{i\in[1,\mu]}$ such that $X_i' = X_j'$ for all $i, j \in [1, \mu]$. The μ-*multicollision*-advantage of the adversary $\mathscr{A}$ is defined as

$$\mathbf{Adv}_{\tilde{E}}^{\mu\text{-mcoll}}(\mathscr{A}) = \Pr\left[\mathscr{A}^{\mathcal{O}_{\tilde{E}}} \text{ produces } \mu\text{-multicollision }\right]$$

and we write

$$\mathbf{Adv}_{\tilde{E}}^{\mu\text{-mcoll}}(q) = \max_{\mathscr{A}} \mathbf{Adv}_{\tilde{E}}^{\mu\text{-mcoll}}(\mathscr{A})$$

where maximum is taken over all adversaries $\mathscr{A}$ making at most q many queries.

6 Security Reductions of mixFeed

Here we give upper bounds on the *privacy advantage* and *forging advantage* of mixFeed against any adversary $\mathscr{B}$. For notational reference see Figure 4.

6.1 Privacy

Let $\mathscr{B}$ be any adversary which successfully breaks the privacy security of mixFeed. We construct a μ-respecting adversary $\mathscr{A}$ which uses $\mathscr{B}$ to break the μ-$TPRP$ security of $\tilde{E}$.

Let $\mathcal{CH}$ be a μ-$TPRP$ challenger. $\mathscr{A}$ acts as a privacy challenger for $\mathscr{B}$ as follows:

1. $\mathcal{CH}$ randomly chooses a bit $b \in \{0,1\}$. if $b = 0$, $\mathcal{CH}$ computes using $\tilde{E}$. Otherwise $\mathcal{CH}$ chooses a random function $P : T \times \{0,1\}^n \to \{0,1\}^n$ such that for all $tw_i \in T$, $P(tw_i, \star)$ are independent random permutations from $\{0,1\}^n \to \{0,1\}^n$ and computes using P.

2. on receiving encryption queries from $\mathscr{B}$, of the form (N, A, M),

   (a) $\mathscr{A}$ computes $\bar{N}, \delta_A, \delta_M$.

   (b) For all $0 \leq j \leq l+2$, $\mathscr{A}$ defines $tw_j = (N, j)$.

   (c) If number of previous queries to $\mathcal{CH}$ of the form $(\star, \bar{N})$ is less than μ then $\mathscr{A}$ queries $(tw_0, \bar{N})$ to $\mathcal{CH}$ to receive Y_0.Else it aborts.

   (d) For all $1 \leq j \leq a$,

      i. $\mathscr{A}$ computes $B_j = A_j \oplus \lfloor Y_j \rfloor_{|A|}$ and $X_j = \lceil \mathsf{pad}(B_j) \rceil_{\frac{n}{2}} \| \lfloor \mathsf{pad}(B_j) \oplus Y_j \rfloor_{\frac{n}{2}}$.

      ii. If number of previous queries to $\mathcal{CH}$ of the form $(\star, X_j)$ is less than μ then it queries (tw_j, X_j) to $\mathcal{CH}$ to receive Y_j. Else it aborts.

   (e) $\mathscr{A}$ defines $X_{a+1} = Y_{l+1} \oplus \delta_A$.

   (f) If number of previous queries to $\mathcal{CH}$ of the form $(\star, X_{a+1})$ is less than μ then it queries (tw_{a+1}, X_{a+1}) to $\mathcal{CH}$ to receive Y_{a+1}.

   (g) For $1 \leq j \leq m$,

      i. $\mathscr{A}$ computes $C_j = \lfloor M_j \oplus Y_{a+j} \rfloor_{|M|}$ and $X_{a+j+1} = \lceil \mathsf{pad}(C_j) \rceil_{\frac{n}{2}} \| \lfloor \mathsf{pad}(C_j) \oplus Y_{a+j} \rfloor_{\frac{n}{2}}$.

      ii. If number of previous queries to $\mathcal{CH}$ of the form $(\star, X_{a+j+1})$ is less than μ then it queries (tw_{a+j+1}, X_{a+j+1}) to $\mathcal{CH}$ to receive Y_{a+j+1}. Else it aborts.

   (h) $\mathscr{A}$ defines $X_{l+2} = Y_{l+1} \oplus \delta_M$.

   (i) If number of previous queries to $\mathcal{CH}$ of the form $(\star, X_{l+2})$ is less than μ then it queries (tw_{l+2}, X_{l+2}) to $\mathcal{CH}$ to receive Y_{l+3}.

   (j) Finally if $\mathscr{A}$ doesn't abort in any of the previous steps, then it defines $C := C_m \| \cdots \| C_1$ and $T := Y_{l+3}$ and sends (C, T) to $\mathscr{B}$.

3. If $\mathscr{B}$ produces the distinguishing bit b' then $\mathscr{A}$ also produces the same distinguishing bit b'.

Theorem 1. *For any privacy breaking adversary $\mathscr{B}$ of* mixFeed *any μ-TPRP adversary $\mathscr{A}$ of $\tilde{E}$ and any $\mu+1$-multicollision adversary $\mathscr{C}$ of $\tilde{E}$, we have*

$$\mathbf{Adv}^{priv}_{\mathsf{mixFeed}}(\mathscr{B}) \leq \mathbf{Adv}^{\mu\text{-}TPRP}_{\tilde{E}}(\mathscr{A}) + \mathbf{Adv}^{\mu+1\text{-}\mathsf{mcoll}}_{P}(\mathscr{C}).$$

Proof. Suppose $\mathcal{CH}$ output the bit $b \in \{0,1\}$. If $b = 1$ then $\mathcal{CH}^b(tw, X) = P(tw, X)$ and if $b = 0$ then $\mathcal{CH}^b(tw, X) = \tilde{E}_K(tw, X)$. We define the event $(\mathcal{CH} \to b) \cap (\mathscr{A}\text{Aborts})$ as $\mathscr{A}^b$ Aborts.

Note that at any stage of the game between $\mathscr{A}$ and $\mathscr{B}$, if while computing the encryption query responses for $\mathscr{B}$, $\mathscr{A}$ needs to make a new query of the form $(tw^{i_{\mu+1}}_{j_{\mu+1}}, X^{i_{\mu+1}}_{j_{\mu+1}})$ and it has already queried μ many queries of the form $(tw^{i_1}_{j_1}, X^{i_1}_{j_1}), \ldots, (tw^{i_\mu}_{j_\mu}, X^{i_\mu}_{j_\mu})$ to $\mathcal{CH}$ such that $X^{i_l}_{j_l} = X^{i_k}_{j_k}$ for all $l, k \in [1, \mu+1]$ then $\mathscr{A}$ aborts.

$$\text{Let } D^{i_l}_{j_l} = \begin{cases} \mathsf{pad}(B^{i_l}_{j_l}) \text{ for } 1 \leq j_l \leq a_i \\ \lceil Y^{i_l}_{j_l-1} \rceil_{\frac{n}{2}} \| \lfloor \delta_A \rfloor_{\frac{n}{2}} \text{ if } j_l = a_l + 1 \\ \mathsf{pad}(C^{i_l}_{j_l - a_i}) \text{ for } a_l + 2 \leq j_l \leq (a_l + m_l + 1) \\ \lceil Y^{i_l}_{j_l-1} \rceil_{\frac{n}{2}} \| \lfloor \delta_M \rfloor_{\frac{n}{2}} \text{ if } j_l = a_l + m_l + 2. \end{cases}$$

In this case, adversary $\mathscr{C}$ can produce $\mu+1$ many tuples of the form $(tw^{i_l}_{j_l-1}, X^{i_l}_{j_l-1}, D^{i_l}_{j_l}, X^{i_l}_{j_l})$ such that $(0^{\frac{n}{2}} \| \lfloor Y^{i_l}_{j_l-1} \rfloor_{\frac{n}{2}}) \oplus D^{i_l}_{j_l} = X^{i_l}_{j_l}$ where

$\mathcal{CH}^b(tw^{i_l}_{j_l-1}, Xi_{l_{j_l-1}}) = Y^{i_l}_{j_l-1}$ for all $l \in [1, \mu+1]$. Thus $\mathscr{C}$ wins the $\mu+1$-multicollision game.

Hence we have

$$\Pr\left[\mathscr{A}^b \text{ Aborts}\right] \leq \Pr\left[\mathscr{C}\text{produces } \mu+1\text{-multicollisions in the } \mathcal{CH}^b \text{ game}\right]$$
$$= \mathbf{Adv}^{\mu+1\text{-mcoll}}_{\mathcal{CH}^b}(\mathscr{C})$$

Now suppose the adversary $\mathscr{A}$ never aborts i.e never had to violate the $\mu+1$-multicollision restriction.

Notice that $\mathscr{A}$ playing the above game, perfectly simulates as a *privacy challenger* for $\mathscr{B}$. Suppose the *TPRP*- challenger $\mathcal{CH}$ randomly chooses a bit $b = 1$. Then all the $\mathcal{CH}$ queries are responded through P. Suppose $\mathscr{B}$ makes a query of the form (N, A, M). Then it is clear from the game that $\forall 0 \leq j \leq a+m+2$, tw_j are distinct and hence we have Y_j are independent and uniformly random outputs from $P(tw_j, \star)$. Then since M is known we have $C_j = \lfloor Y_{a+j} \oplus M_j \rfloor_{|M|}$ are uniformly random and $T = Y_{a+m+2}$ is uniformly random. Hence the (C, T) response from $\mathscr{A}$ is uniformly random. Hence $\mathscr{A}$ acts as a *privacy* challenger which responds to the encryption queries uniformly randomly and we have,

$$\Pr\left[\mathscr{B}^{\$} = 1 \cap \mathscr{A} \text{ doesn't Abort}\right] \leq \Pr\left[\mathscr{A}^P = 1\right].$$

Similarly if $b = 0$. Then $\mathcal{CH}$ responds to the queries of $\mathscr{A}$ correctly with respect to $\tilde{E}$ and thus the (C, T) response from $\mathscr{A}$ to a (N, A, M) query by $\mathscr{B}$ is correctly computed with respect to $\tilde{E}$. Hence $\mathscr{A}$ acts as a *privacy* challenger which responds to the encryption queries correctly with respect to $\tilde{E}$ an we have

$$\Pr\left[\mathscr{B}^{\mathcal{E}nc} = 1 \cap \mathscr{A} \text{ doesn't Abort}\right] \leq \Pr\left[\mathscr{A}^{\tilde{E}} = 1\right]$$

Now without loss of generality assume $\Pr\left[\mathscr{B}^{\$} = 1\right] \geq \Pr\left[\mathscr{B}^{\mathcal{E}nc} = 1\right]$ else we consider the adversary $\mathscr{B}^C$ which is compliment of $\mathscr{B}$ in the sense that it follows the same security game as $\mathscr{B}$ with the difference that whenever $\mathscr{B}$ outputs guessing bit b, $\mathscr{B}^C$ outputs guessing bit $\bar{b}$.

Then we have,

$$\begin{aligned}
\Pr\left[\mathscr{B}^{\$} = 1\right] - \Pr\left[\mathscr{B}^{\mathcal{E}nc} = 1\right] &\leq \Pr\left[\mathscr{B}^{\$} = 1 \cap \mathscr{A} \text{ doesn't Abort}\right] \\
&\quad + \Pr\left[\mathscr{B}^{\$} = 1 \cap \mathscr{A}\text{Aborts}\right] \\
&\quad - \Pr\left[\mathscr{B}^{\mathcal{E}nc} = 1 \cap \mathscr{A} \text{ doesn't Abort}\right] \\
&\quad - \Pr\left[\mathscr{B}^{\mathcal{E}nc} = 1 \cap \mathscr{A}\text{Aborts}\right] \\
&\leq \Pr\left[\mathscr{A}^P = 1\right] - \Pr\left[\mathscr{A}^{\tilde{E}} = 1\right] \\
&\quad + \Pr\left[\mathscr{A}^P = 1 \cap \mathscr{A}\text{Aborts}\right] \\
&\leq \Pr\left[\mathscr{A}^P = 1\right] - \Pr\left[\mathscr{A}^{\tilde{E}} = 1\right] \\
&\quad + \Pr\left[\mathscr{A}^1 \text{ Aborts}\right]
\end{aligned}$$

Hence, we have,

$$\begin{aligned}
\left|\Pr\left[\mathscr{B}^{\$} = 1\right] - \Pr\left[\mathscr{B}^{\mathcal{E}nc} = 1\right]\right| \leq &\left|\Pr\left[\mathscr{A}^P = 1\right] - \Pr\left[\mathscr{A}^{\tilde{E}} = 1\right]\right| \\
&+ \mathbf{Adv}^{\mu+1\text{-mcoll}}_{P}(\mathscr{C})
\end{aligned}$$

□

6.2 Forgery

Let $\mathscr{B}$ be any forging adversary of $\mathsf{mixFeed}$. Suppose $\mathscr{B}$ makes q_d many forging attempts with effectively σ_d many encryption blocks. We construct a $(\mu-1,\sigma_d)$-mcp adversary $\mathscr{A}$ which uses $\mathscr{B}$ to win the $(\mu-1,\sigma_d)$-multi-commitment-prediction game of $\tilde{E}$.

Let $\mathcal{CH}$ be a $(\mu-1,\lambda)$-mcp challenger. $\mathscr{A}$ acts as a forgery challenger for $\mathscr{B}$ as follows:

PHASE 1 :

1. Whenever $\mathscr{B}$ sends an encryption query of the form $(N^i, A^i, M^i)_{i\in\mathcal{E}}$,

   (a) $\mathscr{A}$ responds to the query by computing (C^i, T^i) by making the required $\tilde{E}_K$ queries to $\mathcal{CH}$.

   (b) In the previous step, $\mathscr{A}$ always follows the restriction that no more than $\mu-1$ queries to $\tilde{E}$ have the same input. Else it aborts.

2. For each $j \in [1, q_d]$, on receiving decryption queries of the form $(N^{*j}, A^{*j}, C^{*j}, T^{*j})$ from $\mathscr{B}$, $\mathscr{A}$ responds it with $\perp$, and does the following:

   (a) $\mathscr{A}$ checks if $\mathscr{B}$ has previously made any encryption query (N^i, A^i, M^i) and received output of the form (C^i, T^i) such that $T^i = T^{*j}$ and does the following:

      i. if there doesn't exist any encryption query (N^i, A^i, M^i) from $\mathscr{B}$ such that $T^i = T^{*j}$, then $\mathscr{A}$ sets $p_j = -1$.

      ii. Else if $\exists(N^i, A^i, M^i)$ such that $T^i = T^{*j}$ but $N^i \neq N^{*j}$ or $l_i \neq l^*_j$ then $\mathscr{A}$ sets $p_j = -1$.

      iii. Else if $p'_j \in \mathbb{Z}_{\geq 0}$ be such that $C^{*j}_{m^*_j-k} = C^i_{m_i-k}, \forall k \in [0, p'_j)$ and $C^{*j}_{m^*_j-p'_j} \neq C^i_{m_i-p'_j}$ but $\lceil C^{*j}_{m^*_j-p'_j}\rceil_{\frac{n}{2}} = \lceil C^i_{m_i-p'_j}\rceil_{\frac{n}{2}}$ then define $p_j = p'_j + 1$.

      iv. Else let $p'_j \in \mathbb{Z}_{\geq 0}$ be such that $C^{*j}_{m^*_j-k} = C^i_{m_i-k}, \forall k \in [0, p'_j)$ and $\lceil C^{*j}_{m^*_j-p'_j}\rceil_{\frac{n}{2}} \neq \lceil C^i_{m_i-p'_j}\rceil_{\frac{n}{2}}$ then define $p_j = p'_j$.

   (b) $\mathscr{A}$ computes Y^{*j}_k for all $k \in [0, a^*_j + m^*_j - p_j]$ with the help of $\mathcal{CH}$ following the restriction that no more than $\mu-1$ queries to $\tilde{E}$ have the same input. In that case $\mathscr{A}$ aborts.

   (c) Note that, if there exist a common prefix between (N^i, A^i, C^i) and (N^{*j}, A^{*j}, C^{*j}) then $\mathscr{A}$ already have computed upto the common prefix length during encryption query and thus need not send any new encryption query to $\mathcal{CH}$ for computation up to that point.

PHASE 2:

1. For each $j \in [1, q_d]$

   (a) If $p_j = -1$, then,

      i. Note that $\mathscr{A}$ knows $Y^{*j}_{l^*_j+1}$ from PHASE 1 .

      ii. $\mathscr{A}$ sets commitment of the form $((N^{*j}, l^*_j + 2), \lceil Y^{*j}_{l^*_j+1}\rceil_{\frac{n}{2}}, \lfloor T^{*j}\rfloor_{\frac{n}{2}})$.

   (b) Else,

      i. $\mathscr{A}$ defines $\Delta^j_{-1} = \delta_{M^{*j}} \oplus \delta^i_{M^i}$ and for each $k \in [0, p_j]$, $\Delta^j_k = \lfloor \mathsf{pad}(C^{*j}_{m^*_j-k})\rfloor_{\frac{n}{2}} \oplus \lfloor \mathsf{pad}(C^i_{m_i-k})\rfloor_{\frac{n}{2}}$; $l^*_j = a^*_j + m^*_j$.

ii. For each $j \in [1, q_d]$,and for each $k \in [0, p_j]$ $\mathscr{A}$ makes commitments of the form $(tw_k^{*j}, x_k^{*j}, y_k^{*j})$ where,

$$tw_k^{*j} = (N^{*j}, l_j^* - k + 1); x_k^{*j} = \lceil C^{*j}_{m_j^* - k} \rceil_{\frac{n}{2}}$$

$$y_k^{*j} = \lfloor Y^i_{l_i - k + 1} \rfloor_{\frac{n}{2}} \oplus \Delta^j_{k-1}.$$

2. For each $j \in [1, q_d]$
   (a) If $p_j = -1$,then
      i. It calculates $X^{*j}_{l_j^*+2} = Y^{*j}_{l_j^*+1} \oplus \delta_{M^{*j}}$.
      ii. It sets prediction of the form $((N^{*j}, l_j^* + 2), X^{*j}_{l_j^*+2}, \lfloor T^{*j} \rfloor_{\frac{n}{2}})$.
   (b) Else
      i. Note that, $\mathscr{A}$ knows $Y^{*j}_{l_j^* - p_j}$ from Phase 1 .
      ii. for $k = p_j$ to 0,
         A. $\mathscr{A}$ knows the value of $Y^{*j}_{l_j^* - k}$.
         B. $\mathscr{A}$ then sets $X^{*j}_{l_j^* - k + 1} = (0^{\frac{n}{2}} \| \lfloor Y^{*j}_{l_j^* - k} \rfloor_{\frac{n}{2}}) \oplus C^{*j}_{m_j^* - k}$.
         C. It sets $(tw_k^{*j}, X^{*j}_{l_j^* - k + 1}, y_k^{*j})$ as a prediction tuple, where tw_k^{*j}, y_k^{*j} are as defined above.
         D. finally it queries $(tw_k^{*j}, X^{*j}_{l_j^* - k + 1})$ to $\mathcal{CH}$ and receives $Y^{*j}_{l_j^* - k + 1}$.

Theorem 2. *For any forging adversary $\mathscr{B}$ of mixFeed making q_e many encryption queries with σ_e many encryption query blocks, q_d many decryption queries with σ_d many decryption query blocks , any $(\mu - 1, \sigma_d)$-mcp adversary $\mathscr{A}$ of $\tilde{E}$, and any $\mu + 1$-multicollision adversary $\mathscr{C}$ of $\tilde{E}$, we have*

$$\mathbf{Adv}^{forge}_{\textsf{mixFeed}}(\mathscr{B}) \leq \mathbf{Adv}^{(\mu-1,\sigma_d)\text{-}mcp}_{\tilde{E}}(\mathscr{A}) + \mathbf{Adv}^{(\mu+1)\text{-}\mathsf{mcoll}}_{\tilde{E}}(\mathscr{C}).$$

Claim 1. *Suppose $\mathscr{A}$ never Aborts. If $(N^{*i}, A^{*i}, C^{*i}, T^{*i})$ is a valid forgery,for some $i \in [1, q_d]$ then for some $k \in [-1, p_i]$ we have $(tw_k^{*i}, (0^{\frac{n}{2}} \| \lfloor Y^*_{a_i^* + m_i^* - k} \rfloor_{\frac{n}{2}}) \oplus C^*_{m_i^* - k}, \lfloor Y^{*j}_{a_i^* + m_i^* - k + 1} \rfloor_{\frac{n}{2}})$ is a successful prediction query tuple.*

Proof. (Claim) Let$(N^{*j}, A^{*j}, C^{*j}, T^{*j})$ is a valid forgery. If there doesn't exist any encryption query (N^i, A^i, M^i) from $\mathscr{B}$ such that $T^i = T^{*j}$ or $\exists(N^i, A^i, M^i)$ such that $T^i = T^{*j}$ but $N^i \neq N^{*j}$ or $m_j^* \neq m_i$, then we have $p_j = -1$.

In the commitment phase the adversary $\mathscr{A}$ commits $((N^{*j}, l_j^* + 2), \lceil Y^{*j}_{l_j^*+1} \rceil_{\frac{n}{2}}, \lfloor T^{*j} \rfloor_{\frac{n}{2}})$ as descibed above.

Now we have if any of the above condition is satisfied then $((N^{*j}, l_j^* + 2), X^{*j}_{l_j^*+2})$ is fresh i.e. $((N^{*j}, l_j^* + 2), X^{*j}_{l_j^*+2})$ has never been queried before by $\mathscr{A}$ to $\mathcal{CH}$, $\lceil X^{*j}_{l_j^*+2} \rceil_{\frac{n}{2}} = \lceil Y^{*j}_{l_j^*+1} \rceil_{\frac{n}{2}}$ and $\tilde{E}_K((N^{*j}, l_j^* + 2), X^{*j}_{l_j^*+2}) = T^{*j}$. Hence we see that $((N^{*j}, l_j^* + 2), X^{*j}_{l_j^*+2}, \lfloor T^{*j} \rfloor_{\frac{n}{2}})$ is a valid prediction with respect to the commitment $((N^{*j}, l_j^* + 2), \lceil Y^{*j}_{l_j^*+1} \rceil_{\frac{n}{2}}, \lfloor T^{*j} \rfloor_{\frac{n}{2}})$.

Now let $(N^{*j}, A^{*j}, C^{*j}, T^{*j})$ is a valid forgery. and let $p_j \neq -1$ is as defined before. Hence there exist a $i \in [1, q_e]$ such that $N^{*j} = N^i, a_j^* + m_j^* = a_i + m_i = l_j^*, T^{*j} = T^i$.

First let $p_j' \in \mathbb{Z}_{\geq 0}$ be such that $C^{*j}_{m_j^* - k} = C^i_{m_i - k}, \forall k \in [0, p_j')$ and $C^{*j}_{m_j^* - p_j'} \neq C^i_{m_i - p_j'}$ but $\lceil C^{*j}_{m_j^* - p_j'} \rceil_{\frac{n}{2}} = \lceil C^i_{m_i - p_j'} \rceil_{\frac{n}{2}}$. In this case $p_j = p_j' + 1$. We have by suffix property $Y^{*j}_{l_j - p_j + 1} = \lfloor Y^i_{l_j - p_j + 1} \rfloor_{\frac{n}{2}} \oplus \Delta^j_{p_j - 1}$ and $\Delta^j_{p_j - 1} \neq 0$. Since $tw^{*j}_{p_j} = tw^i_{p_j}$ we must have $X^{*j}_{l_j - p_j + 1} \neq X^i_{l_j - p_j + 1}$. And hence $(tw^{*j}_{p_j}, X^{*j}_{l_j - p_j + 1}, y^{*j}_{p_j})$ is fresh.

Now let $p_j' \in \mathbb{Z}_{\geq 0}$ be such that $C^{*j}_{m_j^* - k} = C^i_{m_i - k}, \forall k \in [0, p_j)$ and $\lceil C^{*j}_{m_j^* - p_j'} \rceil_{\frac{n}{2}} \neq \lceil C^i_{m_i - p_j'} \rceil_{\frac{n}{2}}$. Then we have $p_j = p_j'$ and by the suffix property we must have $\lfloor Y^{*j}_{l_j^* - p_j + 1} \rfloor_{\frac{n}{2}} = \lfloor Y^i_{l_j^* - p_j + 1} \rfloor_{\frac{n}{2}}$. Since $\lceil C^{*j}_{m_j^* - p_j} \rceil_{\frac{n}{2}} \neq \lceil C^i_{m_i - p_j} \rceil_{\frac{n}{2}}$

we have, $X^{*j}_{l^*_j-p_j+1} \neq X^i_{l**_j-p_j+1}$ and hence $(tw^{*j}_{p_j}, X^{*j}_{l^*_j-p_j+1}, y^{*j}_{p_j})$ is fresh where $tw^{*j}_{p_j} = (N^{*j}, l^*_j - p_j + 1)$.

In the commitment phase the adversary commits $(tw^{*j}_k, x^{*j}_k, y^{*j}_k)$ for all $k \in [0, p_j]$.
Hence if $(tw^{*j}_{p_j}, X^{*j}_{l^*_j-p_j+1}, y^{*j}_{p_j})$ is a valid prediction with respect to $(tw^{*j}_{p_j}, x^{*j}_{p_j}, y^{*j}_{p_j})$ we are done.

If not then we have $\lfloor Y^{*j}_{l^*_j-p_j+1}\rfloor_{\frac{n}{2}} \neq \lfloor Y^i_{l^*_j-p_j+1}\rfloor_{\frac{n}{2}} \oplus \Delta^j_{p_j-1}$. Hence $X^{*j}_{l^*_j-p_j+2} \neq X^i_{l^*_j-p_j+2}$ as $C^{*j}_{m^*_j-p_j+1} \oplus C^i_{m_i-p_j+1} = 0^{\frac{n}{2}} \| \Delta^j_{p_j-1}$. Hence we have $(tw^{*j}_{p_j-1}, X^{*j}_{l^*_j-p_j+2}, y^{*j}_{p_j-1})$ is fresh. Hence if it is a valid prediction with respect to the commitment $(tw^{*j}_{p_j-1}, x^{*j}_{p_j-1}, y^{*j}_{p_j-1})$ then we are done.

Inductively, suppose we have $(tw^{*j}_1, X^{*j}_{l^*_j}, y^{*j}_1)$ is not a valid prediction. Then we have $\lfloor Y^{*j}_{l^*_j}\rfloor_{\frac{n}{2}} \neq \lfloor Y^i_{l^*_j}\rfloor_{\frac{n}{2}} \oplus \Delta^j_0$. Hence $X^{*j}_{l^*_j+1} \neq X^i_{l^*_j+1}$ as $C^{*j}_{m^*_j} \oplus C^i_{m_i} = 0^{\frac{n}{2}} \| \Delta^j_0$. Hence we have $(tw^{*j}_0, X^{*j}_{l^*_j+1}, y^{*j}_0)$ is fresh. Now since we have $N^{*j} = N^i, a^*_j + m^*_j = a_i + m_i = l^*_j, T^{*j} = T^i$, we have $X^{*j}_{l^*_j+2} = X^i_{l^*_j+2}$ i.e. $Y^{*j}_{l^*_j} = Y^i_{l^*_j} \oplus 0^{\frac{n}{2}} \| \Delta^{*j}_{-1}$. Since $(N^{*j}, A^{*j}, C^{*j}, T^{*j})$ is a valid forgery, hence, we must have $\tilde{E}_K((tw^{*j}_0, X^{*j}_{l^*_j+1})) = Y^{*j}_{l^*_j+1}$. Hence, $(tw^{*j}_0, X^{*j}_{l^*_j+1}, y^{*j}_0)$ must be a valid prediction.

□

Proof. Theorem 2. For all encryption query of the form (N^i, A^i, M^i), $\mathscr{A}$ can correctly simulate $\mathcal{E}nc_K$ as it has access to $\tilde{E}_K$.

Note that at any stage of the game between $\mathscr{A}$ and $\mathscr{B}$, if while computing the encryption query responses for $\mathscr{B}$, $\mathscr{A}$ needs to make a new query of the form $(tw^{i_{\mu+1}}_{j_{\mu+1}}, X^{i_{\mu+1}}_{j_{\mu+1}})$ and it has already queried μ many queries of the form $(tw^{i_1}_{j_1}, X^{i_1}_{j_1}), \ldots, (tw^{i_\mu}_{j_\mu}, X^{i_\mu}_{j_\mu})$ to $\mathcal{CH}$ such that $X^{i_l}_{j_l} = X^{i_k}_{j_k}$ for all $l, k \in [1, \mu+1]$ then $\mathscr{A}$ aborts.

$$\text{Let } D^{i_l}_{j_l} = \begin{cases} \mathsf{pad}(B^{i_l}_{j_l}) \text{ for } 1 \leq j_l \leq a_i \\ \lceil Y^{i_l}_{j_l-1}\rceil_{\frac{n}{2}} \| \lfloor \delta_A \rfloor_{\frac{n}{2}} \text{ if } j_l = a_l + 1 \\ \mathsf{pad}(C^{i_l}_{j_l-a_i}) \text{ for } a_l + 2 \leq j_l \leq (a_l + m_l + 1) \\ \lceil Y^{i_l}_{j_l-1}\rceil_{\frac{n}{2}} \| \lfloor \delta_M \rfloor_{\frac{n}{2}} \text{ if } j_l = a_l + m_l + 2. \end{cases}$$

In this case, adversary $\mathscr{C}$ can produce $\mu+1$ many tuples of the form $(tw^{i_l}_{j_l-1}, X^{i_l}_{j_l-1}, D^{i_l}_{j_l}, X^{i_l}_{j_l})$ such that $(0^{\frac{n}{2}} \| \lfloor Y^{i_l}_{j_l-1}\rfloor_{\frac{n}{2}}) \oplus D^{i_l}_{j_l} = X^{i_l}_{j_l}$ where $\mathcal{CH}^b(tw^{i_l}_{j_l-1}, X{i_l}_{j_l-1}) = Y^{i_l}_{j_l-1}$ for all $l \in [1, \mu+1]$. Thus $\mathscr{C}$ wins the $\mu+1$-multicollision game.

Hence $\Pr[\mathscr{A} \text{ Aborts}] \leq \Pr[\mathscr{C} \text{ produces } \mu+1\text{-}\textit{multicollision}]$.

Note that, $p_i < m^*_i$ for all i-th decryption query. Hence $\mathscr{A}$ makes at most $\sum_i p_i \leq \sum_i m_i \leq \sigma_d$ many commitments and makes at most σ_d many queries in PHASE 2 to produce at most σ_d many prediction tuples.

Hence by the claim 1 we have,

$\Pr[\mathscr{A} \text{ wins } (\mu-1, \sigma_d)\text{-mcp game}]$

$$\geq \Pr[\mathscr{B} \text{ Forges } i\text{-th query for some } i \in [1, q_d] | \mathscr{A} \text{ doesn't Abort}]$$

$$\begin{aligned} \Pr[\mathscr{B} \text{ Forges}] &\leq \Pr[\mathscr{B} \text{ Forges } i\text{-th query for some } i \in [1, q_d] | \mathscr{A} \text{ doesn't Abort}] \\ &\quad + \Pr[\mathscr{A} \text{ Aborts}] \\ &\leq \Pr[\mathscr{A} \text{ wins } (\mu, \sigma_d)\text{-mcp game}] + \mathbf{Adv}^{\mu+1\text{-}\mathsf{mcoll}}_{\tilde{E}}(\mathscr{C}) \\ &= \mathbf{Adv}^{(\mu-1,\sigma_d)\text{-}mcp}_{\tilde{E}}(\mathscr{A}) + +\mathbf{Adv}^{\mu+1\text{-}\mathsf{mcoll}}_{\tilde{E}}(\mathscr{C}). \end{aligned}$$

□

7 Bounding Security Definitions of $\tilde{E}$

Here we bound the advantages of an adversary playing in different security games as defined in Section 5.1. The tweakable block cipher $\tilde{E}$ can be best understood from the following diagram.

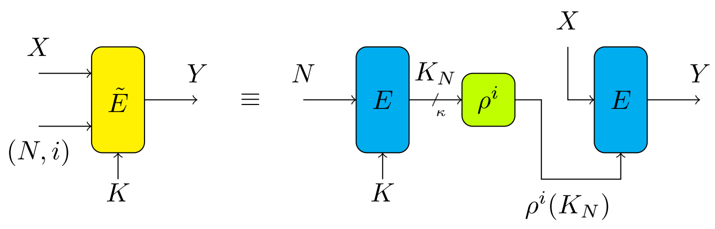

Figure 1: The tweable block cipher in mixFeed. Here ρ is the 11-th round key function in AES[7] key scheduling algorithm. ρ^i denotes i-many consecutive applications of ρ.

To bound the security definitions of $\tilde{E}$ we make the following assumption.

Assumption 1. *For any $K \in \{0,1\}^n$ chosen uniformly at random, probability that K has a period at most l is at most $\frac{l}{2^{\frac{n}{2}}}$.*

Note that our assumption is weak in the sense that for an ideal permutation the above probability is at most $\frac{l}{2^n}$. Recently Mustafa Khairallah has observed that there are at least $2^{33.77}$ keys with a period of $2^{30.08}$ in the AES Key scheduling algorithm. Note that the probability that one of these keys are used is $\frac{2^{33.77}}{2^{128}} = 2^{-94.23}$. Whereas by our assumption the probability is at most $\frac{2^{30.08}}{2^{64}} = 2^{-33.92}$. So, we conclude that his observation does not violate our assumption.

7.1 Bounding μ-TPRP Security

Here we try to bound the μ-*TPRP*-security of the tweakable block cipher $\tilde{E}$. Let $\mathscr{A}$ be any μ-respecting adversary playing the μ-*TPRP* game and makes at most t many primitive queries and d many online queries. We assume that the adversary doesn't make repetitive or redundant queries.

7.1.1 The Ideal world and Analysis of Bad events

Let $\mathcal{P}$ and $\mathcal{E}$ denote the index set of primitive queries and encryption queries respectively.

In ideal world the oracle chooses random functions $P : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ and $Q : T \times \{0,1\}^n \to \{0,1\}^n$ such that for all $K \in \{0,1\}^n$ we have $P(K, \star)$ is a random permutation and for all $tw \in T$ we have $Q(tw, \star)$ is a random permutation.

Primitive Query: In the Ideal world for the i-th primitive query of the form (K^i, X^i) it computes $Y^i = P(K^i, X^i)$ and sends it as a response.

Define $\omega_t = (K^i, X^i, Y^i)_{i \in \mathcal{P}}$ to be the primitive transcript.

Online Query: On receiving the i-th input query of the form $((N^i, j^i), X^i)$ it computes $Y^i = Q((N^i, j^i), X^i)$ and sends it as the response.

Offline Computation : Oracle Chooses $K \in \{0,1\}^n$ uniformly at random. It then chooses a permutation $\Pi : \{0,1\}^n \to \{0,1\}^n$ uniformly at random from the set of all permutations over $\{0,1\}^n$. It then defines $K_{N^i} := \Pi(N^i)$ and $K^i = \alpha^{j^i} \cdot K_{N^i}$.

Define $\omega_d = (K, ((N^i, j^i), X^i, Y^i, K^i)_{i \in \mathcal{E}},)$ to be the online transcript.
Define $\omega = (\omega_t, \omega_d)$ be the transcript for the adversary in the ideal world.

Bad Events: Consider the following events due to ω,

BAD1: For some $i \in \mathcal{E} \cup \mathcal{P}$ we have $K^i = K$.

BAD2: For some $i_1 \neq i_2 \in \mathcal{E}$ we have $(N^{i_1}, j^{i_1}) \neq (N^{i_2}, j^{i_2})$ but $K^{i_1} = K^{i_2}$.

BAD3: For some $i \in \mathcal{E}$ and $i' \in \mathcal{P}$ we have $(K^i, X^i) = (K^{i'}, X^{i'})$.

BAD4: $\exists i_1, \dots i_{\mu+1} \in \mathcal{E}$ s.t. $Y^{i_k} = Y^{i_l} \quad \forall k, l \in [1, \mu+1]$.

BAD5: For some $i \in \mathcal{E}$ and $i' \in \mathcal{P}$ we have $(K^i, Y^i) = (K^{i'}, Y^{i'})$.

Definition 1.

$$\text{BAD} = \cup_{i=1}^{5} \text{BAD}i.$$

Lemma 1.

$$\Pr[\text{BAD}] \leq \frac{t+d}{2^n} + \frac{d^2}{2^{n+1}} + \frac{d}{2^{\frac{n}{2}}} + \frac{2\mu t}{2^n} + \frac{\binom{d}{\mu+1}}{(2^n)^\mu}.$$

Proof. Here we try to bound the distinct bad events defined above.

Bounding BAD1: Fix $i \in \mathcal{P} \cup \mathcal{E}$, since K is chosen uniformly at random we have probabilty that $K^i = K$ is at most $\frac{1}{2^n}$. similar Varying over all i,

$$\Pr[\text{BAD1}] \leq \frac{d+t}{2^n}$$

.

Bounding BAD2: This event can be divided into the following cases.

Case 1: $(N^{i_1} \neq N^{i_2})$ In this case since Π is a random permutation, $K_{N^{i_1}} \neq K_{N^{i_2}}$ are distinct and independent. Hence probability that $K^{i_1} = K^{i_2}$ is atmost $\frac{1}{2^n}$. Varying over all $i_1, i_2 \in \mathcal{E}$ we have,

$$\Pr[\text{Case 1}] \leq \frac{d^2}{2^{n+1}}.$$

Case 2: $(N^{i_1} = N^{i_2}; j^{i_1} \neq j^{i_2})$ In this case we have $K_{N^{i_1}} = K_{N^{i_2}}$.

Hence Case 2 event occurs if and only if, $r_i \mid (j^{i_1} - j^{i_2})$ where r_i is the periodicity of K_{N^i}.

Note that queries of this form arise due to the encryption query of $\mathscr{B}$ with nonce N^i in the privacy game.

Let l_i denote the number of blocks in the encryption query of $\mathscr{B}$ with nonce N^i. Then for all i_i, i_2 such that $N^{i_1} = N^{i_2} = N^i$, we have $|j^{i_1} - j^{i_2}| \leq l_i$.

Hence we have $r_i \leq l_i$ and by Assumption 1 probability that this event holds is at most $\frac{l_i}{2^{\frac{n}{2}}}$.
Now varying over all possible i and from the observation that $\sum_i l_i \leq d$ we have,

$$\Pr[\text{Case 2}] \leq \sum_i \frac{l_i}{2^{\frac{n}{2}}} \leq \frac{d}{2^{\frac{n}{2}}}.$$

Since the above two cases are mutually exclusive we have,

$$\Pr[\text{BAD2}] \leq \frac{d^2}{2^{n+1}} + \frac{d}{2^{\frac{n}{2}}}.$$

Bounding BAD3: For a given $i' \in \mathcal{P}$, let the adversary makes the primitive query $(K^{i'}, X^{i'})$. Then there can be at most μ-many encryption query of the form $((N^{i_k}, j^{i_k}), X^{i'})_{k \in [1,\mu], i_k \in \mathcal{E}}$ and hence at most μ-many $(K^{i_k}, X^{i'})_{k \in [1,\mu], i_k \in \mathcal{E}}$ tuples. now since K^{i_k} are chosen uniformly at random during encryption query we have for a given $i_k \in \mathcal{E}$, probability that $K^{i_k} = K^{i'}$ is at most $\frac{1}{2^n}$. Hence for a given $i' \in \mathcal{P}$ probability that $\exists i \in \mathcal{E}$ s.t. $(K^i, X^i) = (K^{i'}, X^{i'})$ is at most $\frac{\mu}{2^n}$. Varying over all i', we have

$$\Pr[\text{BAD3}] \leq \frac{\mu t}{2^n}.$$

BOUNDING BAD4: Since for each $i \in \mathcal{E}$, Y^i is chosen uniformly at random. given $i_1, \ldots, i_{\mu+1} \in \mathcal{E}$ probability that $Y^{i_j} = Y^{i_j}$ for all $j \in [1, \mu+1]$ is at most $\frac{1}{(2^n)^\mu}$. Hence varying over all choices of $i_1, \ldots, i_{\mu+1}$ we have

$$\Pr[\mathrm{BAD4}] \leq \frac{\binom{d}{\mu+1}}{(2^n)^\mu}.$$

BOUNDING BAD5$|\overline{\mathrm{BAD4}}$: For a given $i' \in \mathcal{P}$, let the adversary's primitive transcript be $(K^{i'}, \star, Y^{i'})$. Then there can be at most μ-many encryption transcript of the form $((N^{i_k}, j^{i_k}), \star, Y^{i'})_{k\in[1,\mu], i_k \in \mathcal{E}}$ and hence at most μ-many $(K^{i_k}, Y^{i'})_{k\in[1,\mu], i_k\in\mathcal{E}}$ tuples. now since K^{i_k} are chosen uniformly at random during encryption query we have for a given $i_k \in \mathcal{E}$, probability that $K^{i_k} = K^{i'}$ is at most $\frac{1}{2^n}$. Hence for a given $i' \in \mathcal{P}$ probability that $\exists i \in \mathcal{E}$ s.t. $(K^i, Y^i) = (K^{i'}, Y^{i'})$ is atmost $\frac{\mu}{2^n}$. Varying over all i', we have

$$\Pr\left[\mathrm{BAD5}|\overline{\mathrm{BAD4}}\right] \leq \frac{\mu t}{2^n}.$$

Adding all the probabilities we get the Lemma. □

7.1.2 Real World and Good transcript analysis

The real world has oracle E_K. All the primitive queries and the encryption queries are responded based on the responses of E_K.

By good transcript we mean any transcript which is not bad. Now consider a good transcript $\omega = (\omega_t, \omega_d)$. Let Θ_0 and Θ_1 be the transcript random variable obtained in the ideal world and real world respectively.

Then we have $\Pr[\Theta_0 = \omega] = \prod\limits_{t_i} \frac{1}{(2^n)_{t_i}} \times \frac{1}{2^n} \times \frac{1}{(2^n)_d} \times \frac{1}{(2^n)_d}$.
Where t_i denotes the number of primitive Queries with the key $K_i' \in \{0,1\}^\kappa$. i.e $\sum_i t_i = t$.

Now note that in the real world the primitive queries and online queries are permutation compatible.

Hence we have $\Pr[\Theta_1 = \omega] = \prod\limits_{k_i} \frac{1}{(2^n)_{k_i}} \times \frac{1}{2^n} \times \frac{1}{(2^n)_d}$. Where $k_i = d_i + t_i$ such that t_i denotes the number of primitive queries with key K_i and d_i denotes the number of encryption queries of the form (N^l, j^l, X) such that $K^l = K_i$. Note that $\sum_i k_i = d + t$.

Hence

$$\begin{aligned}\frac{\Pr[\Theta_1]}{\Pr[\Theta_0]} &= \frac{\prod\limits_{t_i}(2^n)_{t_i} \times 2^n \times (2^n)_d \times (2^n)_d}{\prod\limits_{k_i}(2^n)_{k_i} \times 2^n \times (2^n)_d} \\ &= \frac{\prod\limits_{i}(2^n)_{t_i} \times (2^n)_d}{\prod\limits_{i}(2^n)_{t_i+d_i}} \\ &= \frac{(2^n)_d}{\prod\limits_{i}(2^n - t_i)_{d_i}} > 1.\end{aligned}$$

Hence by H-coefficient technique we have, Theorem 3.

Theorem 3.

$$\mathbf{Adv}_{\tilde{E}}^{\mu\text{-}TPRP}(d, t, \mu) \leq \frac{t+d}{2^n} + \frac{d^2}{2^{n+1}} + \frac{d}{2^{\frac{n}{2}}} + \frac{2\mu t}{2^n} + \frac{\binom{d}{\mu+1}}{(2^n)^\mu}.$$

7.2 Bounding (μ, λ)-*mcp* Security

Here we try to bound the advantage of a μ-respecting adversary $\mathscr{A}$ making t-many primitive queries and d-many online queries playing the (μ, λ)-multi commitment prediction game with a challenger $\mathcal{CH}$.

We assume that the adversary doesn't make repetitive or redundant queries.
PRIMITIVE QUERIES: Whenever $\mathscr{A}$ makes a primitive query of the form (K^i, X^i) for some $i \in \mathcal{P}$ the $\mathcal{CH}$ responds with $Y_i = E_{K^i}(X^i)$.Let $\omega_t = (K^i, X^i, Y^i)_{i\in\mathcal{P}}$ be the primitive transcript of the adversary $\mathscr{A}$.

ONLINE QUERIES: Whenever $\mathscr{A}$ makes an online query of the form $((N^i, j^i), X^i)$ for some $i \in \mathcal{E}$, $\mathcal{CH}$ checks that the query is μ-respecting. If not, then it aborts. Else, $\mathcal{CH}$ computes $K_{N^i} = E_K(N^i)$, $K^i = \alpha^{j^i} \cdot K_{N^i}$ and finally outputs $Y^i = E_{K^i}(X)$ as response.

Let $\omega_d = ((N^i, j^i), K^i, X^i, Y^i)_{i\in\mathcal{E}}$ be the online transcript of the adversary.

Define $\omega = (\omega_t, \omega_d)$ as the transcript of $\mathscr{A}$.
Bad Events
Consider the following events depending on the transcript ω of the adversary $\mathscr{A}$.

Bad Events due to primitive and encryption query.

BAD1: For some $i \in \mathcal{E} \cup \mathcal{P}$ we have $K^i = K$.

BAD2: For some $i_1 \neq i_2 \in \mathcal{E}$ we have $(N^{i_1}, j^{i_1}) \neq (N^{i_2}, j^{i_2})$ but $K^{i_1} = K^{i_2}$.

BAD3: For some $i \in \mathcal{E}$ and $i' \in \mathcal{P}$ we have $(K^i, X^i) = (K^{i'}, X^{i'})$.

BAD4: $\exists i_1, \dots i_{\mu+1} \in \mathcal{E}$ s.t. $Y^{i_k} = Y^{i_l} \quad \forall k, l \in [1, \mu + 1]$.

BAD5: For some $i \in \mathcal{E}$ and $i' \in \mathcal{P}$ we have $(K^i, Y^i) = (K^{i'}, Y^{i'})$.

Bad event due to multi-commitment prediction game.
BAD 6 : For some $i \in [1, \lambda]$ and $k \in \mathcal{E}$, we have a commitment $((N^i, j^i), x^i, y^i)$ is such that, $(N^i, j^i) \neq (N^k, j^k)$ but $K^i = K^k$ where $K^i = \rho^{j^i} \cdot E_K(N^i)$.

BAD7: For some $i \in [1, \lambda]$ and $k \in \mathcal{P}$, we have a commitment $((N^i, j^i), x^i, y^i)$ is such that, $(K^i, x^i) = (K^k, \lfloor X^k \rfloor_{\frac{n}{2}})$ where $K^i = \rho^{j^i} \cdot E_K(N^i)$.

Definition 2.

$$\text{BAD} = \cup_{i=1}^{7} \text{BAD}i.$$

Lemma 2.

$$\Pr[\text{BAD}] \leq \frac{t + d}{2^n} + \frac{d^2}{2^{n+1}} + \frac{2d}{2^{\frac{n}{2}}} + \frac{2\mu t}{2^n} + \frac{\binom{d}{\mu+1}}{(2^n)^\mu} + \frac{\lambda d}{2^{n+1}} + \frac{\lambda t}{2^{\frac{3n}{2}}}.$$

Proof. Here we try to bound the distinct bad events defined above.

Bounds of events BAD1 to BAD5 has been found while bounding μ-TPRP security of $\tilde{E}$.

BOUNDING BAD6: This event can be divided into the following cases.

CASE 1: $(N^i \neq N^k)$ In this case $K_{N^i} \neq K_{N^k}$ are distinct. Hence probability that $K^i = K^k$ is at most $\frac{1}{2^n}$. Varying over all $k \in \mathcal{E}$ and $i \in [1, \lambda]$ we have,

$$\Pr[\text{CASE } 1] \leq \frac{\lambda d}{2^{n+1}}.$$

CASE 2: $(N^i = N^k; j^i \neq j^k)$ In this case we have $K_{N^i} = K_{N^k}$.
Note that, this event occurs if and only if, $r_i \mid (j^i - j^k)$ where r_i is the periodicity of K_{N^i}.

Note that, queries of this form arise due to the encryption query of $\mathscr{B}$ with nonce N^i in the forgery game.

Let l denote the maximum block length in any encryption or decryption query of $\mathscr{B}$ in the forgery game. Then for all i, k such that $N^i = N^k$, we have $|j^i - j^k| \leq l$.

Hence we have $r_i \leq l$ and by Assumption 1 probability that this event holds is at most $\frac{l_i}{2^{\frac{n}{2}}}$.
Now varying over all possible $k \in \mathcal{E}$, $i \in [1, \lambda]$ we have, and the observation that $l \leq d$ we have,

$$\Pr[\textsc{Case } 2] \leq \frac{d}{2^{\frac{n}{2}}}.$$

Hence

$$\Pr[\mathrm{BAD6}] \leq \frac{\lambda d}{2^{n+1}} + \frac{d}{2^{\frac{n}{2}}}.$$

Bounding BAD7: Fix $i \in [1, \lambda]$ and $k \in \mathcal{P}$. Since K_{N^i} is distributed uniformly at random, and there is no primitive query after commitment, we have probability that $(K^i, x^i) = (K^k, \lfloor X^k \rfloor_{\frac{n}{2}})$ is at most $\frac{1}{2^{\frac{3n}{2}}}$. varying over all i, k we have,

$$\Pr[\mathrm{BAD7}\ ] \leq \frac{\lambda t}{2^{\frac{3n}{2}}}.$$

Adding all the probabilities we get the Lemma. □

Good Transcript Analysis
By good transcript we mean any transcript which is not bad. Now consider a good transcript $\omega = (\omega_t, \omega_d)$.

Then we have $(tw^i, X^i)_{i \in [1,\lambda]}$ is fresh. Hence for a fix i, probability that $\lfloor \tilde{E}(tw^i, X^i) \rfloor_{\frac{n}{2}} = y_i$ is bounded by at most $\frac{1}{2^{\frac{n}{2}}}$. Hence varying over all $i \in [1, \lambda]$ we have,

$$\Pr\left[\mathscr{A}^{\tilde{E}} \text{wins the } (\lambda, \mu)\text{-}mcp \text{ game}\right] \leq \frac{\lambda}{2^{\frac{n}{2}}}.$$

Combining the results we have Theorem 4.

Theorem 4.

$$\mathbf{Adv}_{\tilde{E}}^{(\mu,\lambda)\text{-mcp}}(d, t) \leq \frac{\lambda}{2^{\frac{n}{2}}} + \frac{t+d}{2^n} + \frac{d^2}{2^{n+1}} + \frac{2d}{2^{\frac{n}{2}}} + \frac{2\mu t}{2^n} + \frac{\binom{d}{\mu+1}}{(2^n)^\mu} + \frac{\lambda d}{2^{n+1}} + \frac{\lambda t}{2^{\frac{3n}{2}}}.$$

7.3 Bounding μ-multi collision

7.3.1 Bounding μ-multi collision for $\tilde{E}$

We model E as a random permutation. We assume that the adversary doesn't make repetitive or redundant queries.

$\mathcal{O}$ Queries: Whenever $\mathscr{A}$ makes an online query of the form $((N^i, j^i), X^i, C^i)$ for some $i \in \mathcal{E}$, $\mathcal{O}$ computes $K_{N^i} = E_K(N^i)$, $K^i = \alpha^{j^i} \cdot K_{N^i}$ and finally outputs$z^i = \lfloor Y^i \oplus C^i \rfloor_{\frac{n}{2}}$ as response where $Y^i = E_{K^i}(X^i)$.

Let $\omega_d = ((N^i, j^i), K^i, X^i, z^i)_{i \in \mathcal{E}}$ be the online transcript of the adversary.

We have the μ- multi collision occurs if $\exists i_1, \ldots, i_\mu \in [1, d]$ such that $z^{i_k} = z^{i_l}$ for all $k, l \in [1, \mu]$.
Consider the following event

BAD: For some $i_k \neq i_l \in [1, d]$ we have $tw^{i_k} \neq tw^{i_l}$ but $K^{i_k} = K^{i_l}$. As shown earlier, probability of this can be bounded by $\frac{d^2}{2^{n+1}} + \frac{d}{2^{\frac{n}{2}}}$.

Now suppose $\forall k, l \in [1, d]$ we have BAD doesn't occur. Then Note that the probability of μ-multi collision is highest when the tweak is same for all the queries.

In that case for a given $x \in \{0,1\}^{\frac{n}{2}}$, and fixed $i_k \in [1, d]$ number of possible tuples of (Y^{i_k}, C^{i_K}) such that $z^{i_k} = \lfloor Y^{i_k} \oplus C^{i_k} \rfloor_{\frac{n}{2}} = x$ is bounded by $2^{\frac{n}{2}}$. Hence varying over all $i_1, \ldots, i_\mu \in [1, d]$, we have number of possible tuples $(Y^{i_1}, C^{i_1}), \ldots, (Y^{i_\mu}, C^{i_\mu})$ such that $z^{i_k} = \lfloor Y^{i_k} \oplus C^{i_k} \rfloor_{\frac{n}{2}} = x \ \forall k \in [1, \mu]$ is bounded by $2^{\frac{\mu n}{2}}$.

Varying over all $x \in \{0,1\}^{\frac{n}{2}}$ and for all combination of $i_1, \ldots, i_\mu \in [1,d]$ we have number of ways in which μ-multi collision occurs is at most $\binom{d}{\mu} 2^{\frac{(\mu+1)n}{2}}$.

Hence we have

$$\begin{aligned}\Pr\left[\mu\text{-mcoll}|\overline{\mathrm{BAD}}\right] &\leq \frac{\binom{d}{\mu} 2^{\frac{(\mu+1)n}{2}}}{(2^n)_\mu} \\ &\leq d\left(1+\frac{\mu^2}{2^n}\right)\left(\frac{d}{2^{\frac{n}{2}}}\right)^{\mu-1}.\end{aligned}$$

Combining the results of this section we have Theorem 5.

Theorem 5.

$$\mathbf{Adv}_{\tilde{E}}^{\mu\text{-mcoll}}(d) \leq d\left(1+\frac{\mu^2}{2^n}\right)\left(\frac{d}{2^{\frac{n}{2}}}\right)^{\mu-1} + \frac{d^2}{2^{n+1}} + \frac{d}{2^{\frac{n}{2}}}$$

.

7.3.2 Bounding μ-multi collision for P

Let $P : T \times \{0,1\}^n \to \{0,1\}^n$ be a random function such that we have $P(tw, \star)$ is a random permutation for all $tw \in T$.

We assume that the adversary doesn't make repetitive or redundant queries.

$\mathcal{O}$ QUERIES: Whenever $\mathscr{A}$ makes an online query of the form $((N^i, j^i), X^i, C^i)$ for some $i \in \mathcal{E}$, $\mathcal{O}$ computes $z^i = \lfloor Y^i \oplus C^i \rfloor_{\frac{n}{2}}$ as response where $Y^i = P(tw^i, X^i)$.

Let $\omega_d = ((N^i, j^i), X^i, z^i)_{i\in\mathcal{E}}$ be the online transcript of the adversary.

We have the μ- multi collision occurs if $\exists i_1, \ldots, i_\mu \in [1,d]$ such that $z^{i_k} = z^{i_l}$ for all $k, l \in [1,\mu]$.

Note that the probability of μ-multi collision is highest when the tweak is same for all the queries.

In that case for a given $x \in \{0,1\}^{\frac{n}{2}}$ and fixed $i_k \in [1,d]$ number of possible tuples of (Y^{i_k}, C^{i_K}) such that $z^{i_k} = \lfloor Y^{i_k} \oplus C^{i_k} \rfloor_{\frac{n}{2}} = x$ is bounded by $2^{\frac{n}{2}}$. Hence varying over all $i_1, \ldots, i_\mu \in [1,d]$, we have number of possible tuples $(Y^{i_1}, C^{i_1}), \ldots, (Y^{i_\mu}, C^{i_\mu})$ such that $z^{i_k} = \lfloor Y^{i_k} \oplus C^{i_k} \rfloor_{\frac{n}{2}} = x\ \forall k \in [1,\mu]$ is bounded by $2^{\frac{\mu n}{2}}$.

Varying over all $x \in \{0,1\}^{\frac{n}{2}}$ and for all combination of $i_1, \ldots, i_\mu \in [1,d]$ we have number of ways in which μ-multi collision occurs is at most $\binom{d}{\mu} 2^{\frac{(\mu+1)n}{2}}$.

Hence we have

$$\begin{aligned}\Pr\left[\mu\text{-mcoll}\right] &\leq \frac{\binom{d}{\mu} 2^{\frac{(\mu+1)n}{2}}}{(2^n)_\mu} \\ &\leq d\left(1+\frac{\mu^2}{2^n}\right)\left(\frac{d}{2^{\frac{n}{2}}}\right)^{\mu-1}.\end{aligned}$$

Combining the results of this section we have Theorem 6.

Theorem 6.

$$\mathbf{Adv}_{P}^{\mu\text{-mcoll}}(d) \leq d\left(1+\frac{\mu^2}{2^n}\right)\left(\frac{d}{2^{\frac{n}{2}}}\right)^{\mu-1}$$

.

8 Security Bounds for mixFeed

Here we bound the advantages of an adversary playing in different security games as defined in Section 5.1.

For any adversary running in time t and making atmost q many encryption and decryption(in case of forgery) query with total of at most σ many blocks,

Theorem 7.

$$\mathbf{Adv}^{priv}_{\mathsf{mixFeed}} \leq \frac{t+\sigma}{2^n} + \frac{\sigma^2}{2^{n+1}} + \frac{\sigma}{2^{\frac{n}{2}}} + \frac{2\mu t}{2^n} + \frac{\binom{\sigma}{\mu+1}}{(2^n)^\mu} + \sigma\left(1 + \frac{(\mu+1)^2}{2^n}\right)\left(\frac{\sigma}{2^{\frac{n}{2}}}\right)^\mu .$$

Theorem 8.

$$\mathbf{Adv}^{forge}_{\mathsf{mixFeed}} \leq \frac{t+\sigma}{2^n} + \frac{3\sigma^2}{2^{n+1}} + \frac{4\sigma}{2^{\frac{n}{2}}} + \frac{2(\mu-1)t}{2^n} + \frac{\binom{\sigma}{\mu}}{(2^n)^{\mu-1}} + \frac{\sigma t}{2^{\frac{3n}{2}}} + \sigma\left(1 + \frac{(\mu+1)^2}{2^n}\right)\left(\frac{\sigma}{2^{\frac{n}{2}}}\right)^\mu .$$

where n is the state size and μ is the number of multi collisions allowed in the input of the tweakable blockcipher. For all calculation purposes take $\mu \geq 4$.

Note that According to NIST requirement $\sigma \leq 2^{46}$ and $t \leq 2^{112}$. Following our recommendation in Section 2, we take $n = 128$ and $\mu = 4$. Then We have $\sigma\left(1 + \frac{(\mu+1)^2}{2^n}\right)\left(\frac{\sigma}{2^{\frac{n}{2}}}\right)^\mu < 2^{-25}$ and hence the dominating term is $\frac{2\mu t}{2^n}$ Theorem 7 and $\frac{2(\mu-1)t}{2^n}$ in Theorem 8 which are both less than 2^{-12}. Hence we conclude that mixFeed is well secured within the complexity bounds specification of NIST.

Proof. Theorem 7 can be derived from Theorem 1, Theorem 3 and Theorem 6.

Theorem 8 can be derived from Theorem 2 , Theorem 4 and Theorem 5.

□

References

[1] Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. Biclique cryptanalysis of the full AES. In *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, pages 344–371, 2011.

[2] Lorenzo Grassi. Mixture differential cryptanalysis: a new approach to distinguishers and attacks on round-reduced AES. *IACR Trans. Symmetric Cryptol.*, 2018(2):133–160, 2018.

[3] Lorenzo Grassi, Christian Rechberger, and Sondre Rønjom. Subspace trail cryptanalysis and its applications to AES. *IACR Trans. Symmetric Cryptol.*, 2016(2):192–225, 2016.

[4] Lorenzo Grassi, Christian Rechberger, and Sondre Rønjom. A new structural-differential property of 5-round AES. In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*, pages 289–317, 2017.

[5] Khoongming Khoo, Eugene Lee, Thomas Peyrin, and Siang Meng Sim. Human-readable proof of the related-key security of AES-128. *IACR Trans. Symmetric Cryptol.*, 2017(2):59–83, 2017.

[6] Moses Liskov, Ronald L Rivest, and David Wagner. Tweakable block ciphers. In *Annual International Cryptology Conference*, pages 31–46. Springer, 2002.

[7] NIST. Announcing the ADVANCED ENCRYPTION STANDARD (AES). Fedral Information Processing Standards Publication FIPS 197, National Institute of Standards and Technology, U. S. Department of Commerce, 2001.

[8] Sondre Rønjom, Navid Ghaedi Bardeh, and Tor Helleseth. Yoyo tricks with AES. In *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*, pages 217–243, 2017.

Appendix

Relevant figures for **mixFeed**

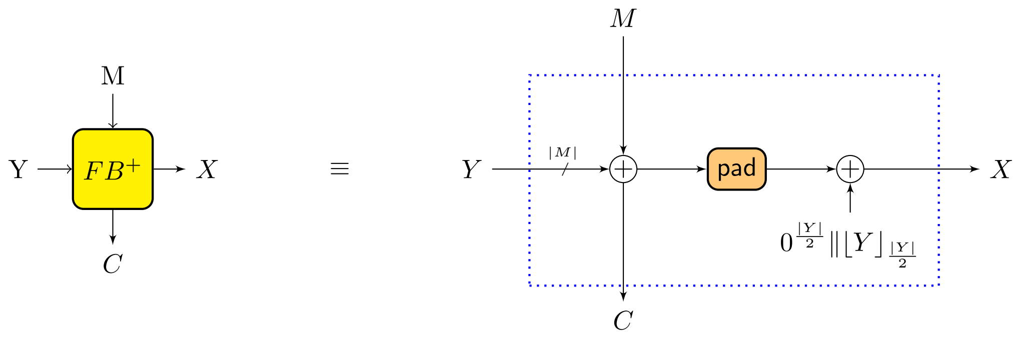

Figure 2: The FB^+ Function in mixFeed. pad is 0^*1 padding.

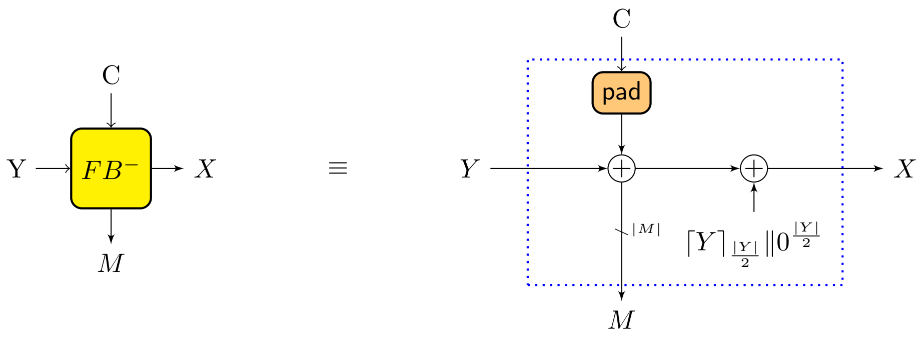

Figure 3: The FB^- Function in mixFeed

.

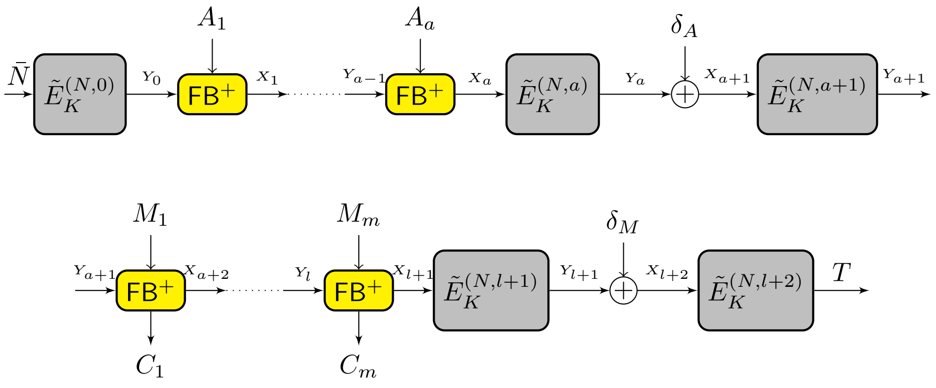

Figure 4: Block diagram for mixFeed encryption. Here, $\bar{N} = N\|x$, where $x = 0^8/0^71/0^610$ depending on the condition that $(a \neq 0)$ or $(a = 0\ \&\ m \neq 0)$ or $(a = 0\ \&\ m = 0)$ respectively. We define $l = a + m$.

Test Vectors

Test Vectors for **AES'128/128**

testvector 1
$Input = 000102030405060708090a0b0c0d0e0f$
$Key = efcb089475ded60586a7d97c64baf3e1$
$Input = 2f22aa67066bf48cdd3cf0808ebc86ed$
$Key = 8cc110aba3f926985eef0262bc0e21dc$

testvector 2

$Input = 000102030405060708090a0b0c0d0e0f$
$Key = efcb089475ded60586a7d97c64baf453$
$Input = 58f6d4eb08a72d19d1fae7e85634a28e$
$Key = 21ee22c7c5e266da384848b306dc549d$

Test Vectors for **mixFeed**

testvector 1
$Key = 000102030405060708090A0B0C0D0E0F$
$Nonce = 000102030405060708090A0B0C0D0E$
$PT =$
$AD = 000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F$
$CT = 6CDB385142B591F8E57D50FC41899B23$

testvector 2
$Key = 000102030405060708090A0B0C0D0E0F$
$Nonce = 000102030405060708090A0B0C0D0E$
$PT = 00$
$AD = 000102030405060708090A0B0C0D$
$CT = E56EDEC0001E1D94074303E6397D238CCF$

testvectors 3
$Key = 000102030405060708090A0B0C0D0E0F$
$Nonce = 000102030405060708090A0B0C0D0E$
$PT = 000102$
$AD = 000102030405060708090A0B0C0D0E$
$CT = 4753140EA6C5D3B01F06BBBC3F55181BB3FFE5$