J. Cryptol. (2008) 21: 547–578
DOI: 10.1007/s00145-008-9024-z

Journal of
CRYPTOLOGY

Encryption Modes with Almost Free Message Integrity

Charanjit S. Jutla
IBM T.J. Watson Research Center, Yorktown Heights, NY 10598-704, USA
csjutla@us.ibm.com

Communicated by Moti Yung

Received 24 December 2003 and revised 8 April 2008
Online publication 24 May 2008

Abstract. We define a new mode of operation for block ciphers which, in addition to providing confidentiality, also ensures message integrity. In contrast, previously for message integrity a separate pass was required to compute a cryptographic message authentication code (MAC). The new mode of operation, called Integrity Aware Parallelizable Mode (IAPM), requires a total of $m + 1$ block cipher evaluations on a plaintext of length m blocks. For comparison, the well-known CBC (cipher block chaining) encryption mode requires m block cipher evaluations, and the second pass of computing the CBC-MAC essentially requires additional $m + 1$ block cipher evaluations. As the name suggests, the new mode is also highly parallelizable.

Key words. Block ciphers, Encryption, Authentication, Pairwise independent, Parallelizable.

1. Introduction

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

There are two primary aspects of any security system: *confidentiality* and *authentication*. In its most prevalent form, confidentiality is attained by encryption of bulk digital data using *block ciphers*. The block ciphers (e.g., DES [26], AES[1]), which are designed to encrypt fixed length data, are used in various chaining modes to encrypt bulk data. One such mode of operation is cipher block chaining (CBC) ([2,27]). The security of CBC has been well studied [5].

Cipher block chaining of block ciphers is also used for authentication between parties sharing a secret key. The CBC-MAC (CBC Message Authentication Code) is an international standard [14]. The security of CBC-MAC was demonstrated in [6]. Authentication in this symmetric key setting is also called *Message Integrity*.

Despite similar names, the two CBC modes, one for encryption and the other for MAC are different, as in the latter the intermediate results of the computation of the

MAC must be kept secret. In fact, in most standards (TLS, IPsec [29,30]), as well as in proprietary security systems, two different passes with two different keys, one each of the two modes is used to achieve confidentiality and message integrity.

Nevertheless, it is enticing to combine the two passes into one so that in a single cipher block chaining pass, both confidentiality and message integrity are ensured. Many such attempts have been made, which essentially use a simple checksum or manipulation detection code (MDC) in the chaining mode [9,23,28]. Unfortunately, all such previous schemes are susceptible to attacks (see, e.g., [32]).

We mention here that there are two alternative approaches to *authenticated encryption* [4], i.e., encryption with message integrity. The first is to generate a MAC using universal hash functions [8] as in UMAC [7]. UMACs on certain architectures can be generated rather fast. However, UMAC suffers from requiring too much key material or a pseudorandom number generator to expand the key. (For comparison sake, on a message of size n, UMAC requires a key of size n for similar efficiency and security.) In another scheme, block numbers are embedded into individual blocks to thwart attacks against message integrity [18]. However, this makes the cipher-text longer.

In this paper, we present a new mode of operation for block ciphers, which in a single pass achieves both confidentiality and message integrity. In one variant, to encrypt a message of length m blocks, the new mode requires a total of $m + 1$ block cipher evaluations. All other operations are simple operations, like exclusive-or. To contrast this with the usual CBC mode, the encryption pass requires m block cipher evaluations, and the CBC-MAC computation on the ciphertext requires another $m + 1$ block cipher evaluations.

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

The efficiency of the new mode comes from proving that the output whitening random sequence need only be pair-wise independent. In other words, if the output whitening sequence is $s_0, s_1, s_2, \ldots, s_m$, then each s_i is required to be random but only pairwise-independent of the other elements. Such a sequence is easily generated by performing only $\log m$ cryptographic operations like block cipher evaluations. A simple algebraic scheme can also generate such a sequence by performing only two cryptographic operations.

In fact, an even weaker condition than pair-wise independence suffices. A sequence of uniformly distributed n-bit random numbers $s_0, s_1, \ldots, s_m$ is called *XOR-universal* [19] if, for every n-bit constant c and every pair $i, j, i \neq j$, the probability that $s_i \oplus s_j$ equals c is 2^{-n}. We show that the output whitening sequence need only be XOR-universal. A simple algebraic scheme can generate such a sequence by performing only one cryptographic operation.

The XOR-universal sequence generated to ensure message integrity can also be used to remove chaining from the encryption mode while still ensuring confidentiality. This results in a mode of operation for authenticated encryption which is highly parallelizable, and we call this mode IAPM (for Integrity Aware Parallelizable Mode).

It is known (see [4,18]) that for symmetric key encryption, confidentiality under chosen plaintext attacks (CPA), along with integrity of ciphertexts, implies confidentiality under chosen ciphertext attacks (CCA). In this paper we prove the schemes secure for confidentiality under CPA and secure for integrity of ciphertexts.

Concurrently to our work, Gligor and Donescu ([10]) have also described a mode of operation similar to CBC (but not the parallelizable mode) which has built-in message integrity, although with a slightly weaker security bound than our construction. Subsequently, and based on these results, a new authenticated encryption mode OCB was described in [31]. The mode OCB was designed to also handle plaintexts of irregular lengths, i.e., bit lengths which are not multiple of the block length.

It has also been shown ([17]) that any scheme to encrypt m blocks of size n bits each, that assures message integrity, is linear in $(\mathrm{GF2})^n$, and uses $m + k$ invocations of random functions (from n bits to n bits) and v blocks of randomness, must have $k + v$ at least $\Omega(\log m)$.

Pairwise independent random number generators (also called universal hash functions [8]) have been used extensively in cryptography. In particular, Goldreich, Krawczyk, and Luby [11] used them to build pseudorandom generators from regular one-way functions, and Naor and Yung [25] used them in the construction of universal one-way hash functions. In the context of symmetric key encryption, pairwise independent random permutations were used to construct pseudo-random permutations from pseudo-random functions [24]. In the context of de-randomization, Luby had demonstrated how the random choices needed in a randomized parallel algorithm for maximal independent set problem need only be pairwise independent [21].

The rest of the paper is organized as follows. Section 2 formalizes the notions of security for both confidentiality and message integrity. Section 3 describes the new mode of operation IAPM. We also formalize the mode in the random permutation model. In Sect. 4 we prove that the new scheme is secure for message integrity. In Sect. 5 we prove the secrecy theorem for IAPM.

2. Authenticated Encryption Schemes

We give definitions of schemes which explicitly define the notion of secrecy of the input message. In addition, we also define the notion of message integrity. Moreover, we allow arbitrary finite length input messages as long as they are multiples of the block size of the underlying block cipher.

Let Coins be the set of infinite binary strings. Let $\mathcal{K} \subseteq \{0, 1\}^*$ be the key space, and $\mathcal{D}$ be a distribution on the key space.

Definition. A (probabilistic, symmetric, stateless) *authenticated encryption* scheme, with block size n, key space $\mathcal{K}$, and distribution $\mathcal{D}$, consists of the following:

- **Initialization:** All parties exchange information over private lines to establish a private key $x \in \mathcal{K}$. All parties store x in their respective private memories.

- **Message sending with integrity:**

$$\text{Let } E : \mathcal{K} \times \text{Coins} \times \left\{\{0,1\}^n\right\}^* \to \left\{\{0,1\}^n\right\}^*$$
$$D : \mathcal{K} \times \left\{\{0,1\}^n\right\}^* \to \left\{\{0,1\}^n\right\}^* \cup \{\perp\}$$

be polynomial time computable function ensembles. The functions E and D must have the property that for all $x \in \mathcal{K}$, $P \in \{\{0,1\}^n\}^*$, and $c \in$ Coins,

$$D_x\big(E_x(c, P)\big) = P.$$

We will usually drop the random argument to E as well and just think of E as a probabilistic function ensemble. The security of such a scheme is given by the following two definitions, the first defining confidentiality under chosen plaintext attacks, and the second defining message integrity.

Definition (Security under Find-then-Guess [5,22]).

Consider an adaptive probabilistic adversary A which runs in two stages: *find* and *guess*. The two stages will be called $A1$ and $A2$. It is given access to the encryption oracle E_x. In the find stage it tries to come up with two equal length messages P^0 and P^1. It also retains a state $C1$ for the next stage. In the guess stage it is given the encryption $C2$ of P^b, where b is chosen randomly to be 0 or 1. The value $C2$ can really be seen as result of another oracle query P^b, except that b is hidden from the adversary. This "oracle call" will also be called the "choice" stage. The adversary's success is reflected in how well it guesses b. Formally,

$$\begin{aligned}\text{Adv}_A = \big|\Pr\big[x \leftarrow_{\mathcal{D}} \mathcal{K};\ \big(P^0, P^1, C1\big) \leftarrow A1^{E_x(\cdot)};\ b \leftarrow_R \{0,1\};\ C2 \leftarrow E_x\big(P^b\big) :\\ A2^{E_x(\cdot)}(C1, C2) = b\big] - 1/2\big|.\end{aligned}$$

An authenticated encryption scheme is said to be (t, q, μ, ϵ)-secure against chosen plaintext attack if, for any adversary A as above which runs in time at most t and asks at most q queries of E_x, these totaling at most μ blocks, and its advantage Adv_A is at most ϵ.

The following notion of security is also called *integrity of ciphertext* [4].

Definition. (*Message Integrity*): Consider an adaptive probabilistic adversary A running in two stages. In the first stage (*find*), A asks r queries of the oracle E_x. Let the oracle replies be $C^1, \ldots, C^r$. Subsequently in the second stage, A produces a ciphertext C', different from each C^i, $i \in [1..r]$. The adversary's success probability is given by

$$\text{Succ} \overset{\text{def}}{=} \Pr\big[D_x(C') \neq \perp\big],$$

where the probability is over the choice of x from $\mathcal{K}$ according to $\mathcal{D}$, other randomness used by E, and the probabilistic choices of A.

An authenticated encryption scheme is (t, q, μ, ϵ)-secure for message integrity if, for any adversary A running in time at most t and making at most q queries totaling μ blocks, its success probability is at most ϵ.

3. The New Modes of Operation

We begin by defining XOR-universal hash function families [19].

3.1. *XOR-Universal Distributions*

Definition. We denote by $\mathcal{F}(m \to n)$ the set of all functions from m bits to n bits. We denote by $\mathcal{P}(n \to n)$ the set of all permutations from n bits to n bits.

Definition (Hash Function Family). An (m, n)-*family of hash functions* H is a collection of functions that map the set of binary strings of length m bits into the set of binary strings of length n, i.e., a subset of $\mathcal{F}(m \to n)$.

Definition (XOR-Universal Hash Function Family [19]). An (m, n)-family of hash functions H is called an XOR-Universal hash function family if, for every m-bit value M and every n-bit value c, $\Pr_h[h(M) = c]$ is 2^{-n}, and further if, for every pair of distinct m-bit values $M1$ and $M2$ and every n-bit value c, $\Pr_h[h(M1) \oplus h(M2) = c]$ is 2^{-n}, where the probabilities are over choosing h uniformly from H.

An (m, n) hash function family H can be given by a single function $\mathcal{H}$ which takes a $\lceil \log |H| \rceil$-bit value, called *seed*, as another argument.

Definition (XOR-Universal Sequence). A probability distribution over sequences of n-bit numbers $s_0, s_1, \ldots, s_{m-1}$ is called *XOR-universal* if each s_i is uniformly distributed and, for every n-bit constant c and every pair $i, j, i \neq j$, the probability that $s_i \oplus s_j$ is c is 2^{-n}.

An XOR-universal sequence of length 2^m can be generated using an XOR-Universal (m, n)-hash function family $\mathcal{H}$ and seed k by $s_i = \mathcal{H}(k, i)$.

3.2. *The New Mode—IAPM*

We now describe the new mode of operation for block ciphers, which along with confidentiality also guarantees message integrity. The new mode is also highly parallelizable, as will be clear from the description. It is called *IAPM* for *integrity aware parallelizable mode*. There are many variants of this mode, depending on how the XOR-universal sequence is generated and even on how the initial vectors are chosen. One variant is described in Fig. 1. We now give more details for IAPM and its many variants.

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

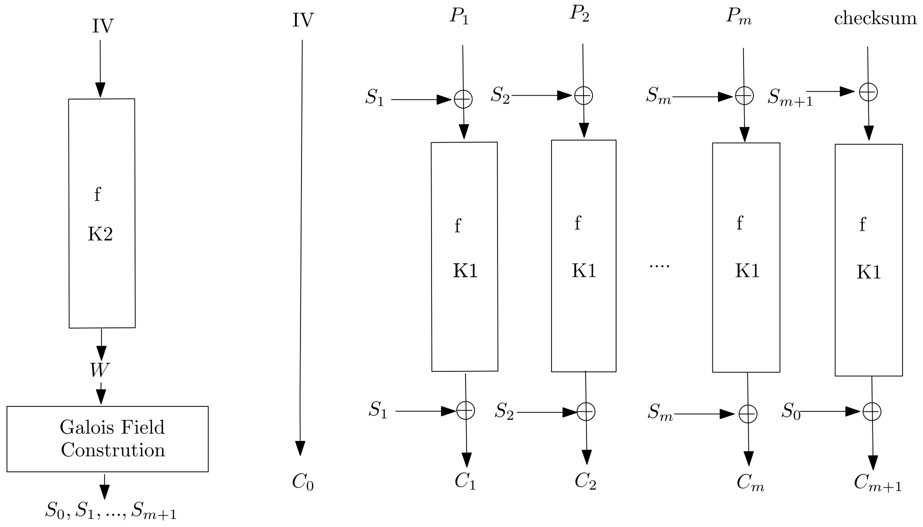

Fig. 1. Parallelizable encryption with message integrity (IAPM).

The message to be encrypted, P, is divided into blocks of length n each. Let these blocks be $P_1, P_2, \ldots, P_m$. A random initial block, also called *initial vector* (IV), of length n (bits) is chosen. As we discuss later, the IV need not be random, as long as it is unique (that is never reused). The IV is expanded using the key $K2$, used as a secret seed, to produce an XOR-Universal sequence $S_0, \ldots, S_{m+1}$. There are various methods to achieve this, which we will discuss shortly. The cipher-text message $C = \langle C_0, C_1, \ldots, C_{m+1} \rangle$ is then generated as follows (see Fig. 1). The encryption pseudocode follows:

```
$C_0 = \mathrm{IV}$
for $j = 1$ to $m$ do
    $M_j = P_j \oplus S_j$
    $N_j = f_{K1}(M_j)$
    $C_j = N_j \oplus S_j$
end for
checksum = $\bigoplus_{j=1}^{m} P_j$
$M_{m+1}$ = checksum $\oplus S_{m+1}$
$N_{m+1} = f_{K1}(M_{m+1})$
$C_{m+1} = N_{m+1} \oplus S_0$
```

Note that S_0 is used in the last step. The xor-ing of S_j with P_j before applying the function f is commonly called *pre-whitening*. Similarly, xor-ing of S_j to N_j to obtain C_j is called *post-whitening*.

It is easy to see that the above scheme is invertible. The inversion process yields blocks $P_1, P_2, \ldots, P_{m+1}$. The decrypted plain-text is $\langle P_1, P_2, \ldots, P_m \rangle$. Message integrity is verified by checking $P_{m+1} = P_1 \oplus P_2 \oplus \cdots \oplus P_m$.

Generation of XOR-Universal Sequences We now focus on how the XOR-universal sequence used above is generated. We first describe methods which employ the block cipher itself. The block IV is first expanded into $t = O(\log m)$ new random blocks $W_1, \ldots, W_t$ by using the block cipher and key $K2$ as follows:

```
$W_1 = f_{K2}(\mathrm{IV})$
for $i = 2$ to $t$ do
    $W_i = f_{K2}(W_1 + i - 2)$
end for
```

The t blocks are then used to produce $m + 2$ new XOR-universal random blocks $S_0, S_1, \ldots, S_{m+1}$. In other words, the t blocks $W_1, \ldots, W_t$ combined serve as the seed into an XOR-Universal Hash Function family $\mathcal{H}$. There are several such XOR-Universal families, some requiring t to be only one. Such a family will be described later. For now, consider the following elementary method using subsets ($t = \lceil \log(m + 2) \rceil$):

```
for $i = 1$ to $2^t$ do
    Let $\langle a_1, a_2, \ldots, a_t \rangle$ be the binary representation of $i$
    $S_{i-1} = \bigoplus_{j=1}^{t} (a_j \cdot W_j)$
end for
```

Galois Field Constructions of XOR-Universal Sequences There are several algebraic XOR-Universal Hash families. Firstly, one could consider a pairwise independent hash function family $\mathcal{H}$ using an algebraic construction in GF(p) as follows: generate two random blocks W_1 and W_2, and then let $S_j = \mathcal{H}(\langle W_1, W_2 \rangle, j) = (W_1 + W_2 * j) \bmod p$, where p is a prime of appropriate size. For example, if the block cipher has block size 128 bits, p could be chosen to be $2^{128} - 159$. This leads to a faster implementation than the subset construction.

A sequence of $2^n - 1$ uniformly distributed n-bit random numbers, which are XOR-universal, can also be generated by viewing the n-bit numbers as elements of GF(2^n). Consider $\mathcal{H}(W, j) = j \cdot W$, where multiplication is in GF(2^n). It is easy to see that $\mathcal{H}$ is an (n, n)-XOR-universal hash family (except for the value $j = 0$). Now, let $S_j = \mathcal{H}(W, e(j))$, where $e(j)$ is any one-to-one function from $\mathbf{Z}_{2^n-1}$ to nonzero elements of GF(2^n). Then, it is easy to see that $S_0, \ldots, S_{2^n-2}$ is an XOR-universal sequence. Note that this requires generation of only a single W, i.e., $t = 1$ (see Fig. 1).

It is worth noting that in a serial implementation of IAPM, and particularly in a resource constrained system, generation of S_j from W or from S_{j-1} may influence the efficiency of the mode. In particular, if $e(j) = g^j$, where g is a generator of the field GF(2^n), it takes at least one multiplication to get S_j from S_{j-1}. If $e(j)$ is the binary representation of $j + 1$, then a basis of the field GF(2^n) over GF2 (times W) maybe initialized in n vectors, and then S_j can be computed by exclusive-or operations of these vectors. In fact, $e(j)$ can be the $(j + 1)$th gray code vector, in which case, only one n-bit exclusive-or operation is required to get S_j from S_{j-1} (see, e.g., [15,31]), as long as the n precomputed vectors are maintained.

In some situations, even maintaining n vectors in active memory maybe too taxing on the system. In such a situation, a GF(p) based solution as described in the next section may be advantageous.

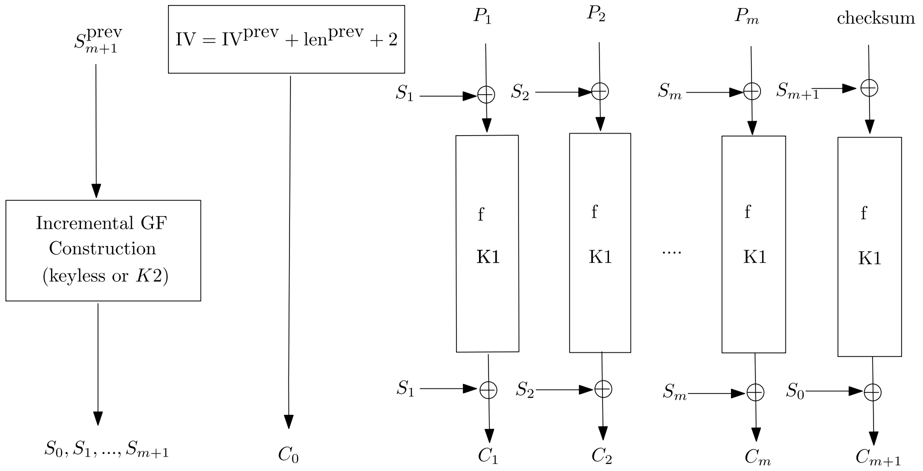

Fig. 2. IAPM with safe initial vectors.

Safe Initial Vectors Till now we have focused on construction of XOR-universal sequences using fresh seeds for each message, e.g., using $W = f_{K2}(\text{IV})$ as seed into an XOR-universal hash family. Halevi [12] has observed that the XOR-universal sequences can be generated using noncryptographic operations, i.e., by avoiding $f_{K2}(\text{IV})$. A careful setup and analysis shows that one can use the same seed for all messages, and hence this "global" seed can just be the independently chosen n-bit key $K2$. To this end, we define a set of initial vectors to be *safe* as follows.

Definition. For a sequence of messages P^i, $i = 1$ to z, each of length m^i n-bit blocks, a sequence of n-bit initial vectors IV^i ($i = 1$ to z) is called *safe* if (a) for all $i \in [1..z]$, $\text{IV}^i + m^i + 1 < 2^n - 1$, where the addition is integer addition, and (b) for all $i, i1 \in [1..z]$, $i \neq i1$, for all $j \in [0..m^i + 1]$, and for all $j1 \in [0..m^{i1} + 1]$, $\text{IV}^i + j \neq \text{IV}^{i1} + j1$.

The jth whitening value for the ith message, S_j^i ($j \in [0..m^i + 1]$), is then generated as $S_j^i = \mathcal{H}(K2, \text{IV}^i + j)$, where $\mathcal{H}$ is any XOR-Universal hash family. We will show the surprising result that if the initial vectors are safe, then regardless of how the adversary choses the n-bit initial vector for its adversarial message, its (success) probability for attaining message integrity is negligible.

There are many ways to implement safe initial vectors, including a random choice for the initial vector. Alternatively, one could require the initial vector to be a multiple of $2^{n/2}$, and assuming that the length of each message is less than $2^{n/2} - 1$, this leads to a sequence of safe initial vectors. However, with this scheme, some of the optimizations mentioned above for computing S_j from S_{j-1} do not work when switching to a new message. For the same optimizations to work even across messages, one can set $\text{IV}^i = \text{IV}^{i-1} + m^{i-1} + 2$ (see Fig. 2), and it is easy to see that this leads to safe initial vectors.

The intuition behind why safe initial vectors are secure is that while encrypting genuine messages, the value $\text{IV}^i + m^i + 1$ is never used for calculating a post-whitening value. Now, suppose that an adversary, attacking the message integrity of the scheme, tries to use an initial vector different from IV^i but one which is close enough, say $\mathrm{IV}^i + s$, where $s \leq m^i$. Then, the above fact and the asymmetry in the last block whitening values forces the adversary to end up using a "wrong" whitening value (either post-whitening or pre-whitening value) for at least one block. We defer complete details of the proof to Sect. 4.

3.3. *Integrity Aware Parallelizable Mode (IAPM) Using a Prime Number*

The GF(p) construction with only a single W, instead of two, is not XOR-universal (as opposed to the previous construction in GF(2^n)). However, it is XOR-universal in GF(p). Such a sequence can be used securely in a slight variant of the mode described above where "whitening" now refers to addition modulo 2^n. We now give more details of this variant.

Let p be a prime close to 2^n. For example, for 128 bit block ciphers p could be $2^{128} - 159$, which is known to be a prime. This prime will be fixed for all invocations of this mode using block ciphers of block size 128 bit. For 64-bit ciphers, $p = 2^{64} - 257$ is a close prime.

Let $K2$ be an additional independently chosen key (in addition to key $K1$ for the block cipher). Now, the sequence $S_0, S_1, \ldots, S_{m+1}$ is generated by the following procedure:

```
*procedure* xor_universal_gfp_sequence(*input* IV, $m$, $K2$; *output* $S$)
{
  $S_0 = \mathrm{IV} * K2 \bmod p$
  *for* $j = 1$ to $m + 1$ do
  $S_j^* = (S_{j-1} + K2) \bmod 2^n$
  *if* $(K2 > S_j^*)$  $S_j = S_j^* + (2^n - p)$ else $S_j = S_j^*$.
  *end for*
}
```

We assume that the initial vectors IV are chosen to be safe, e.g., by requiring them to be multiple of $2^{\lceil n/2 \rceil}$ or by incrementing them appropriately as in the previous section.

In the above code, the condition $(K2 > S_j^*)$ is equivalent to n-bit integer addition overflow in the previous step. Essentially, we are computing S_j to be $(j + 1) * K2 \bmod p$, except that we use a lazy representation. In other words, we do not reduce modulo p if $(S_{j-1} + K2) < 2^n$ but we do compensate by $2^n - p$ if $(S_{j-1} + K2) \geq 2^n$, since in the latter case, $(S_{j-1} + K2) = S_{j-1} + K2 - p = (S_{j-1} + K2 - 2^n) + (2^n - p) \pmod p$. We prove in Lemma 9 that there is no overflow when compensating by $(2^n - p)$.

In this mode, the pre- and post-whitening is done by n-bit integer addition. The ciphertext message $C = \langle C_0, C_1, \ldots, C_{m+1} \rangle$ is generated as follows (see Fig. 3):

```
$C_0 = \mathrm{IV}$
*for* $j = 1$ to $m$ do
        $M_j = (P_j + S_j) \bmod 2^n$
        $N_j = f_{K1}(M_j)$
        $C_j = (N_j + S_j) \bmod 2^n$
*end for*
checksum = $P_1 \oplus P_2 \oplus \cdots \oplus P_m$
```

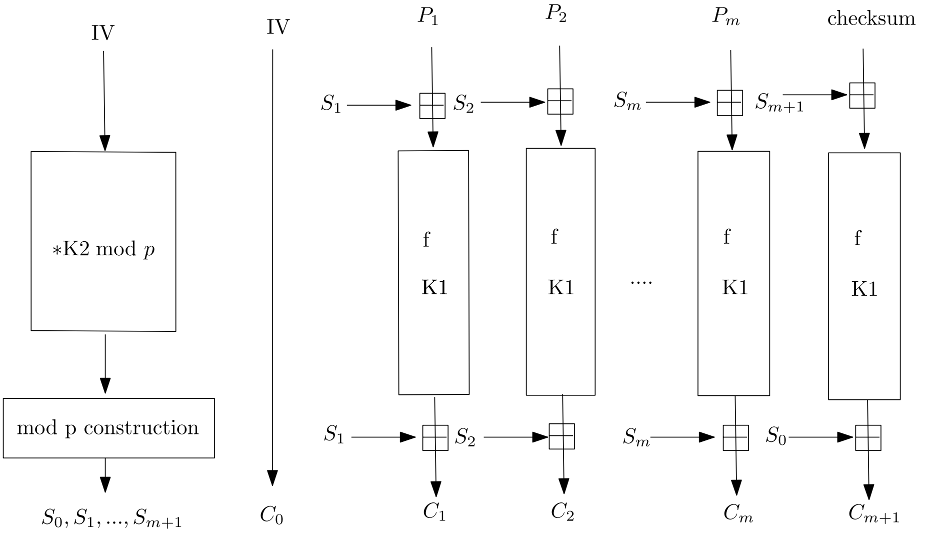

Fig. 3. Integrity aware parallelizable mode (IAPM) in GF(p) using Safe IVs.

M_{m+1} = (checksum $+ S_{m+1}$) mod 2^n
$N_{m+1} = f_{K1}(M_{m+1})$
$C_{m+1} = (N_{m+1} + S_0)$ mod 2^n

Note that for computing the checksum, we use exclusive-or instead of addition modulo 2^n. Note that S_0 is used in the last step. The above scheme is invertible.

3.4. *IAPM in Random Permutation Model*

Since the description of IAPM in Sect. 3.2 was for block ciphers, we formally define the authenticated encryption scheme IAPM in the random permutation model here. In the following, the operator "+" will stand for integer addition, and "$\oplus$" for n-bit exclusive-or.

Definition. Given a permutation f from n bits to n bits and a function g from n bits to n bits, the (deterministic) function E-IAPM$_{f,g}$: $\{0,1\}^n \times \{\{0,1\}^n\}^* \rightarrow \{\{0,1\}^n\}^+$ is defined as follows:

- Let the input to E-IAPM$_{f,g}$ be an n-bit IV (denoting initial vector) and an mn-bit string P ($2^n > m \geq 0$) such that IV$+m+1 < 2^n - 1$, which is divided into m n-bit strings $P_1, P_2, \ldots, P_m$.
- Define $C_0 =$ IV, and checksum $= 0 \oplus \bigoplus_{j=1}^{m} P_j$.
- For notational convenience, we will also refer to checksum as P_{m+1}.
- Define for $j = 1$ to m: $C_j = g(\mathrm{IV} + j) \oplus f(P_j \oplus g(\mathrm{IV} + j))$.
- Define $C_{m+1} = g(\mathrm{IV}) \oplus f(P_{m+1} \oplus g(\mathrm{IV} + m + 1))$.
- The output of the function E-IAPM$_{f,g}$ is the $(m+2)n$-bit string $C_0, C_1, \ldots, C_m, C_{m+1}$.

Definition. Given a permutation f from n bits to n bits and a function g from n bits to n bits, the (deterministic) function $\text{D-IAPM}_{f,g}$: $\{\{0,1\}^n\}^+ \rightarrow \{\{0,1\}^n\}^* \cup \{\perp\}$ is defined as follows:

- Let the input to $\text{D-IAPM}_{f,g}$ be an $(m+2)n$-bit string C $(2^n \geq m \geq 0)$, which is divided into $(m+2)$ n-bit strings $C_0, C_1, \ldots, C_m, C_{m+1}$.
- Define $\text{IV} = C_0$.
- If $\text{IV}+m+1 \geq 2^n - 1$, the output of the function is $\perp$.
- Define for $j = 1$ to m: $P_j = g(\text{IV}+j) \oplus f^{-1}(C_j \oplus g(\text{IV}+j))$.
- Define $P_{m+1} = g(\text{IV}+m+1) \oplus f^{-1}(C_{m+1} \oplus g(\text{IV}))$.
- if $0 \oplus \bigoplus_{j=1}^{m} P_j$ is not same as P_{m+1}, return $\perp$, otherwise the output of the function $\text{D-IAPM}_{f,g}$ is the mn-bit string $P_1, \ldots, P_m$.

Definition (IAPM in random permutation model). Let $\mathcal{G}$ be a (n,n)-family of XOR-universal hash functions. The authenticated encryption scheme IAPM($\mathcal{G}$) with block size n is given by the following key space, distribution, encryption function, and decryption function:

- The set $\mathcal{K}$ of keys is the set of pairs **f** and **g**, where **f** is in $\mathcal{P}(n \rightarrow n)$ (i.e., a permutation), and **g** is in $\mathcal{G}$.
- the distribution $\mathcal{D}$ on $\mathcal{K}$ is given by choosing **f** uniformly from $\mathcal{P}(n \rightarrow n)$ and choosing **g** independently and uniformly from $\mathcal{G}$.
- The encryption function under key (**f**, **g**) is given by $\text{E-IAPM}_{\mathbf{f},\mathbf{g}}$.
- The decryption function under key (**f**, **g**) is given by $\text{D-IAPM}_{\mathbf{f},\mathbf{g}}$.

It is easy to see that $\text{D-IAPM}_{\mathbf{f},\mathbf{g}}(\text{E-IAPM}_{\mathbf{f},\mathbf{g}}(\text{IV}, \langle P_1, \ldots, P_m \rangle)) = \langle P_1, \ldots, P_m \rangle$.

4. Message Integrity of IAPM

In this section we will prove the message integrity of IAPM in the random permutation model. The proof can be extended to strong (super) pseudo-random permutations [22] by standard techniques.[1] For simplicity, we will assume that the initial vectors (IVs) are chosen deterministically as a function of the previous ciphertexts (which includes the previous initial vectors and the lengths of the previous ciphertexts). As shown in Sect. 3.2, there are several deterministic schemes to achieve safe initial vectors, and the following theorem assumes any such scheme. If, on the other hand, the initial vectors are chosen randomly (and completely independent of **f** and **g**), a slight modification of the proof below shows that the adversary's success probability is marginally higher, i.e., by $(z+u)(z+1) * 2^{-n}$ (where z and u are as in the theorem below). In the proof, we will mention the changes required for this random IV case.

Theorem 1. *Let A be an adaptive adversary attacking the message integrity of IAPM($\mathcal{G}$) (in the random permutation model). Let A make at most z queries in the first*

[1] Even if the function **g** was chosen as an application of another random permutation from the same pseudo-random permutation class from which **f** is chosen (as opposed to **g** being chosen independently of **f**), a standard hybrid argument shows that **g** can still be considered independent of **f**.

stage, totaling at most μ blocks. Let $u = \mu + z$. Moreover, assume that the initial vectors for the queries in the first stage are chosen using a deterministic scheme such that they are safe. Let v be the maximum number of blocks in the second stage. If $4u^2 < 2^n$ and $4v^2 < 2^n$, then for adversary A,

$$\mathrm{Succ} < \left(u^2 + 2u + 3v + 4z + 1\right) * 2^{-n}.$$

Proof. We first note that we allow arbitrary functions as adversaries and not just computable functions. Then without loss of generality, we can assume that the adversary is deterministic, as every probabilistic adversary is just a probability distribution over all deterministic adversaries [20].

Note that, in the message integrity attack, the adversary's success probability is measured under the key chosen from $\mathcal{K}$ according to distribution $\mathcal{D}$. Thus by the definition of IAPM($\mathcal{G}$), the space for the probability distribution is the set of pairs $\mathbf{f}$ and $\mathbf{g}$. Any variable which is a function of $\mathbf{f}$ and $\mathbf{g}$ will be called a random variable and for clarity will be in bold-face. We will refer to $\mathbf{f}$ as *the permutation* and $\mathbf{g}$ as *the whitening function*.

Fix an adaptive adversary. Since the adversary is deterministic, the first query's plaintext (say $P^1 = \langle P_1^1, \ldots, P_m^1 \rangle$) is fixed for that adversary. Thus, the first query's output, say C^1, is only a function of $\mathbf{f}$ and $\mathbf{g}$. Note that the IV for the first message (which is the first block of C^1) is also chosen deterministically and is in fact fixed. The adversary being adaptive, its second query is a function of C^1. But, since C^1 is only a function of $\mathbf{f}$ and $\mathbf{g}$, the second query's plaintext and IV can also be considered just as a function of $\mathbf{f}$ and $\mathbf{g}$. Thus, C^2 is only a function of $\mathbf{f}$ and $\mathbf{g}$, and so forth.

For all variables corresponding to a message (query), we will use superscripts to denote the message number and subscripts to denote blocks in a particular message. We will use $\mathbf{C}$ to denote the whole *transcript* of sequence of ciphertext outputs $\mathbf{C}^1, \ldots, \mathbf{C}^z$. Thus, $\mathbf{C}_j^i$ is a variable denoting the jth block in the ith ciphertext message. More precisely, this variable $\mathbf{C}$ should be written $\mathbf{C}(\mathbf{f}, \mathbf{g})$, as it is a function of $\mathbf{f}$ and $\mathbf{g}$, as argued in the previous paragraph.

We will use c^i to denote prospective values for $\mathbf{C}^i$. We will use c to denote the prospective ciphertext transcript $c^1, \ldots, c^z$. The function $|\cdot|$ is used to represent length of a message in n-bit blocks. Let $l()$ be the length of the first ciphertext (determined by the adversary A). Given a sequence of ciphertext messages $c^1, \ldots, c^i$, $i < z$, let $l(c^1, \ldots, c^i)$ be the length of the $(i+1)$th ciphertext (which is determined by the adversary and therefore is a deterministic function of $c^1, \ldots, c^i$). We will use the shorthand l^i for $|c^i|$. If the adversary makes less than z queries in the first stage, say $z' \leq z$, we assume, for convenience, that $l(c^1, \ldots, c^i) = 1$ for all $i \geq z'$, as the ciphertext transcript includes the initial vectors c_0^i. Note that if a query is a null message, then IAPM generates two blocks of ciphertext, the initial vector and the block produced from the checksum. Thus for all $i \leq z'$, $l^i \geq 2$, whereas, for all $i > z'$, $l^i = 1$. We will use the function $Z(c)$ to determine the largest i ($\leq z$) such that $l^i(c) \geq 2$. Similarly, the random variable $\mathbf{Z}$ will denote $Z(\mathbf{C}(\mathbf{f}, \mathbf{g}))$. Note that $\mathbf{Z} \leq z$.

We will also refer to c_0^i as $\mathrm{IV}^i(c)$ or just IV^i when clear from context.

Let the adversary's query in the second stage, the *attempted forgery*, be cipher-text $\mathbf{C}'$, different from all ciphertexts in the first stage. We will refer to $\mathbf{C}'$ as *the forged*

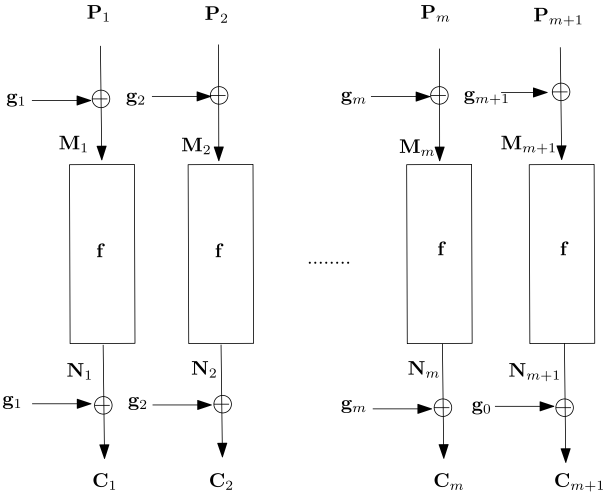

Fig. 4. IAPM in random permutation model.

ciphertext. Since $\mathbf{C}'$ is a deterministic function of $\mathbf{C}$, given $c^1, \ldots, c^z$, let the ciphertext in the second stage be c' with length l', i.e., $c' = \mathbf{C}'(c)$. We will also refer to c'_0 as $\mathrm{IV}'(c)$ or just IV' when clear from context.

Let $\mathbf{L}^i = l(\mathbf{C}^1, \ldots, \mathbf{C}^{i-1})$ be the random variable representing the length of ciphertext $\mathbf{C}^i$ (i.e., the checksum block has index $\mathbf{L}^i - 1$). Similarly, $\mathbf{L}'$ will denote the length of $\mathbf{C}'$.

As per the definition of IAPM in random permutation model (also see Fig. 4), the whitening function $\mathbf{g}$ is applied before and after the application of the permutation $\mathbf{f}$. For each block j in message i, the pre-whitening is done with $\mathbf{g}$ applied to IV^i offset by j. Similarly for the post-whitening, except when j is the last block, in which case the post-whitening is done with $\mathbf{g}$ applied to IV^i offset with zero. Motivated by this, for each i in $[1..z]$, define $\sigma^i_j(c)$ to be the *post-whitening offset* in the jth block of the ith message, namely $\sigma^i_j(c) = j$ if $j < l^i - 1$ and $\sigma^i_j(c) = 0$ if $j = l^i - 1$. Similarly, define $\sigma'_j(c) = j$ if $j < l' - 1$ and $\sigma'_j(c) = 0$ if $j = l' - 1$.

For a fixed ciphertext transcript c, the plaintext block P^i_j (being chosen adaptively) can be *viewed as only a function of* c, and we will write it as $P^i_j(c)$. Thus, instead of writing P^i_j as a function of the permutation $\mathbf{f}$ and the whitening function $\mathbf{g}$, we will consider it as a function of prospective ciphertext transcript c. The random variable $\mathbf{P}^i_j$ can still be expressed as $P^i_j(\mathbf{C}) = P^i_j(\mathbf{C}(\mathbf{f}, \mathbf{g}))$.

For any prospective ciphertext transcript c and whitening function $g \in \mathcal{G}$, for $i \in [1..z]$ and $j \in [1..l^i - 1]$, define $M^i_j(c, g) = P^i_j(c) \oplus g(c^i_0 + j)$. Similarly, define $N^i_j(c, g) = c^i_j \oplus g(c^i_0 + \sigma^i_j(c))$. We will use $\mathbf{M}^i_j$ to denote the random variable $M^i_j(\mathbf{C}, \mathbf{g})$ and use $\mathbf{N}^i_j$ to denote the random variable $N^i_j(\mathbf{C}, \mathbf{g})$. In other words, $\mathbf{M}^i_j$ is the actual input to the permutation $\mathbf{f}$ (for ith message's jth block), and $\mathbf{N}^i_j$ is the output of $\mathbf{f}$ on that input. We will refer to $\mathbf{M}^i_j$s as *the whitened plaintext blocks* and to $\mathbf{N}^i_j$s as *the raw ciphertext blocks*. Just as for $\mathbf{C}$, we will use $P(c)$, $M(c, g)$, and $N(c, g)$ to denote the whole sequence. Note that although $\mathbf{N}^i_j = \mathbf{f}(\mathbf{M}^i_j)$, there is no such relationship between $N^i_j(c, g)$ and $M^i_j(c, g)$. In particular, $N^i_j(c, g) = \mathbf{f}(M^i_j(c, g))$ only if the transcript c and whitening function g are such that $c^i_j = \mathbf{f}(M^i_j(c, g)) \oplus g(c^i_0 + \sigma^i_j(c))$.

Moving on to the forged ciphertext, again for a fixed c, as c' is fixed, for $j \in [1..l' - 1]$, define $N'_j(c, g) = c'_j \oplus g(c'_0 + \sigma'_j(c))$. Note that as c' is picked by the adversary, p' is not just a function of c, and hence M' (as opposed to M^i_j) *cannot* be defined as a function of just c and g. Thus, for $j \in [1..l' - 1]$, any permutation f, and $g \in \mathcal{G}$, define $M'_j(c, g, f) = f^{-1}(N'_j(c, g))$. As before, $\mathbf{N}'_j$ will stand for the random variable $N'_j(\mathbf{C}, \mathbf{g})$, and $\mathbf{M}'_j$ for $M'_j(\mathbf{C}, \mathbf{g}, \mathbf{f})$. We will refer to $\mathbf{N}'_j$s as *the whitened forged ciphertext blocks* and to $\mathbf{M}'_j$s as *the raw forged plaintext blocks*.

Also, for $j \in [1..l' - 1]$, define $P'_j(c, g, f) = M'(c, g, f) \oplus g(c'_0 + j)$. By the definition of IAPM($\mathcal{G}$) (see D-IAPM), the random variable $\mathbf{P}'_j (= P'_j(\mathbf{C}, \mathbf{g}, \mathbf{f}))$ is $\mathbf{M}'_j \oplus \mathbf{g}(\mathbf{C}'_0 + j)$.

For future reference, we list all these definitions and equalities here.

$$\mathbf{P}^i_j = P^i_j(\mathbf{C}), \quad \text{for } j \in \left[1..\mathbf{L}^i - 2\right], \tag{1}$$

$$\mathbf{C}^i_j = \mathbf{g}\big(\mathbf{C}^i_0 + j\big) \oplus \mathbf{f}\big(\mathbf{P}^i_j \oplus \mathbf{g}\big(\mathbf{C}^i_0 + j\big)\big) \quad \text{for } j \in \left[1..\mathbf{L}^i - 2\right], \tag{2}$$

$$\mathbf{P}^i_{\mathbf{L}^i - 1} = 0 \oplus \bigoplus_{j=1}^{\mathbf{L}^i - 2} \mathbf{P}^i_j, \tag{3}$$

$$\mathbf{C}^i_{\mathbf{L}^i - 1} = \mathbf{g}\big(\mathbf{C}^i_0\big), \oplus \mathbf{f}\big(\mathbf{P}^i_{\mathbf{L}^i - 1} \oplus \mathbf{g}\big(\mathbf{C}^i_0 + \mathbf{L}^i - 1\big)\big), \tag{4}$$

$$M^i_j(c, g) = P^i_j(c) \oplus g\big(c^i_0 + j\big), \tag{5}$$

$$N^i_j(c, g) = c^i_j \oplus g\big(c^i_0 + \sigma^i_j(c)\big), \tag{6}$$

$$\mathbf{M}^i_j = M^i_j\big(\mathbf{C}, \mathbf{g}\big), \tag{7}$$

$$\mathbf{N}^i_j = N^i_j(\mathbf{C}, \mathbf{g}) = \mathbf{f}\big(\mathbf{M}^i_j\big), \tag{8}$$

$$N'_j(c, g) = c'_j \oplus g\big(c'_0 + \sigma'_j(c)\big), \tag{9}$$

$$\mathbf{N}'_j = N'_j(\mathbf{C}, \mathbf{g}), \tag{10}$$

$$M'_j(c, g, f) = f^{-1}\big(N'_j(c, g)\big), \tag{11}$$

$$\mathbf{M}'_j = M'_j(\mathbf{C}, \mathbf{g}, \mathbf{f}), \tag{12}$$

$$P'_j(c, g, f) = M'(c, g, f) \oplus g(c'_0 + j), \tag{13}$$

$$\mathbf{P}'_j = \mathbf{M}'_j \oplus \mathbf{g}(\mathbf{C}'_0 + j). \tag{14}$$

Below we define events **E0**, **E1**, and **E2**, which are random variables (being functions of the permutation **f** and the whitening function **g**). We prove that either the adversary forces the events **E0** or **E1**, or the event **E2** happens with high probability. In either case, we show that the checksum validates with low probability. The events **E0** and

E1 describe attacks in which the forged ciphertext is copied from one of the previous legitimate ciphertexts, possibly with re-arrangement and deletion of blocks. The event **E0** is called deletion attempt, as the adversary in this case just truncates an original ciphertext but retains the last block. The event **E1** can be seen as a rotation attempt by the adversary.

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

$$\text{(i)} \quad \forall j \in [0..\mathbf{L}' - 2] : \ \mathbf{C}'_j = \mathbf{C}^i_j \quad \text{and} \quad \text{(ii)} \quad \mathbf{C}'_{\mathbf{L}'-1} = \mathbf{C}^i_{\mathbf{L}^i-1}.$$

Event E1 (*rotation attempt*)[2]: There is an $i \in [1..\mathbf{Z}]$ and a t, $1 \leq t \leq \mathbf{L}^i - \mathbf{L}'$, such that

$$\text{(i)} \quad \mathbf{C}'_0 = \mathbf{C}^i_0 + t, \qquad \text{(ii)} \quad \forall j \in [1..\mathbf{L}' - 1] : \ \mathbf{C}'_j = \mathbf{C}^i_{\sigma'_j(\mathbf{C})+t}.$$

In other words, $\mathbf{C}'_1 = \mathbf{C}^i_{t+1}, \mathbf{C}'_2 = \mathbf{C}^i_{t+2}, \ldots, \mathbf{C}'_{\mathbf{L}'} = \mathbf{C}^i_t$. Also, (i) is same as requiring the initial vector of the forged ciphertext to be same as the initial vector of the ith ciphertext offset by t.

Event **E2** says that there is a block x in the forged ciphertext $\mathbf{C}'$ such that its (whitened forged ciphertext block) $\mathbf{N}'_x$ variable is different from all previous (raw ciphertext variables) **N**s and also different from all other **N**'s.

Event E2: There is an $x \in [1..\mathbf{L}' - 1]$ such that

$$\text{(i)} \quad \forall t \in [1..z]\, \forall j \in [1..\mathbf{L}^t - 1] \ : \mathbf{N}'_x \neq \mathbf{N}^t_j \quad \text{and}$$

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

The adversary's success probability is upper bounded by (a) the probability of event **E0** or **E1** or **E2** not happening, plus (b) the probability of the checksum validating along with events **E0** or **E1** or **E2** happening. Intuitively, when **E0** or **E1** holds, a pre-whitening value will have a discrepancy, whereas if **E2** holds, a post-whitening value will have a discrepancy. These discrepancies lead to a bound on the latter probability (b), though proving the bound requires a few lemmas.

As for bounding the probability (a), the event **E2** not happening translates into a disjunction of events of the type $\mathbf{g}^i_j \oplus \mathbf{g}^{i1}_{j1} = \mathbf{C}^i_j \oplus \mathbf{C}^{i1}_{j1}$, where $\mathbf{g}^i_j$ stands for $\mathbf{g}(\mathrm{IV}^i + j)$. Naively, since $\mathcal{G}$ is XOR-universal, one would think that the probability of each of these events is 2^{-n}. However, it is not guaranteed that the whitening function **g** is independent of the ciphertext **C**, as the ciphertext satisfies $\mathbf{C}^i_j = \mathbf{N}^i_j \oplus \mathbf{g}^i_j$. Intuitively, if all the whitened plaintexts $\mathbf{M}^i_j$ were distinct and $\mathbf{C}^i_j = \mathbf{N}^i_j \oplus \mathbf{g}^i_j$ were the only relations between **C** and **g**, then indeed **g** would be independent of **C** (as **g** and **f** are independently chosen). But, requiring all $\mathbf{M}^i_j$ to be distinct implies another relation between **C** and **g**.

However, it can be shown that, on every fixed outcome of the ciphertext **C** (i.e., $\mathbf{C} = c$ for some constant transcript c), requiring the **M** variables to be distinct (and the **N** variables to be distinct), rules out only a negligible fraction of functions in $\mathcal{G}$ as a potential value for **g** and moreover leaves the remaining functions in $\mathcal{G}$ equiprobable.

[2] If we only consider initial vectors chosen with a nice structure, e.g., with enough zeroes in the least significant bits to unambiguously embed block numbers, then the event **E1** need not be considered. In that case, one can show that either the adversary forces event **E0**, or event **E2** happens with high probability.

So, consider the following predicate PD (*pairwise different*). For any constant c and function $g \in \mathcal{G}$, define PD(c, g) to be

$$\forall i, i1 \in [1..z], \forall j \in \bigl[1..l^i - 1\bigr], \forall j1 \in \bigl[1..l^{i1} - 1\bigr], (i, j) \neq (i1, j1) :$$
$$\bigl(M^i_j(c, g) \neq M^{i1}_{j1}(c, g)\bigr) \wedge \bigl(N^i_j(c, g) \neq N^{i1}_{j1}(c, g)\bigr).$$

Again, we will use **PD** to denote the random variable PD$(\mathbf{C}(\mathbf{f}, \mathbf{g}), \mathbf{g})$. If a random schedule is used to pick the initial vectors, then we must include in this predicate the condition that $\{c^i_0\}_i$ are safe.

The rest of the proof of the theorem is organized as follows. To start with, we will formalize in Lemma 1 the equiprobability of the allowed **g**, given a constant transcript $\mathbf{C} = c$ and conditioned on the event PD$(c, \mathbf{g})$. We use this lemma to prove in Lemma 6 that the probability of the event **PD** and the negation of (**E0** or **E1** or **E2**) is low. We also use Lemma 1 to prove in Lemma 3 that event **PD** itself happens with high probability. Finally, we prove that the checksum validating along with events **E0** or **E1** or **E2** is a small probability event as well (Lemmas 7 and 8), which would lead to the proof of the theorem. We first state all the lemmas and use them to prove the theorem. The proofs of the lemmas follow later.

We need to characterize the set of prospective ciphertexts with safe IVs for this particular adversary A. Before that, recall the (adversarial) ciphertext length function l from above. Also, recall that the predicate "safe" applies to a set of initial vectors. Now, for $0 \le i < z$, define

$$\mathcal{L}\bigl(c^1, \ldots, c^i\bigr) = \bigl\{c^{i+1} \;:\; \bigl|c^{i+1}\bigr| = l\bigl(c^1, \ldots, c^i\bigr) \text{ and } \bigl\{c^j_0\bigr\}_{j \in [1..i+1]} \text{ safe}\bigr\}.$$

Let

$$\mathcal{C} \;=\; \bigl\{c \;:\; \forall i \in [1..z]\; c^i \in \mathcal{L}\bigl(c^1, \ldots, c^{i-1}\bigr)\bigr\}.$$

Thus, $\mathcal{C}$ can be seen as the space of prospective ciphertext transcripts for this particular adversary A. Note that when we sum over c ranging from $\mathcal{C}$, it really means the following telescopic sum:

$$\sum_{c \in \mathcal{C}} \;=\; \sum_{c^1 \in \mathcal{L}()} \cdots \sum_{c^i \in \mathcal{L}(c^1, \ldots, c^{i-1})} \cdots \sum_{c^z \in \mathcal{L}(c^1, \ldots, c^{z-1})} .$$

Remark. If a random schedule is used to choose the IVs, then we exclude the safety condition from $\mathcal{C}$ and include it in the predicate PD. Moreover, in all the lemmas and the proof of the theorem below, the probability will be over choosing $(\mathbf{f}, \mathbf{g})$ according to $\mathcal{D}$, as well as choosing the initial vectors randomly and independently. The only change will be in the analysis of Lemma 2, as the safety condition will incur an additional cost of $(z + u)(z + 1) * 2^{-n}$.

In the following lemmas, the adversary A is fixed to be as in Theorem 1 statement. The quantities n, z, μ, u, and v are as stipulated in Theorem 1 statement.

Lemma 1. *For every prospective ciphertext transcript $c \in \mathcal{C}$ and for any function $g \in \mathcal{G}$ such that* $\mathrm{PD}(c, g)$,

$$\Pr_{(\mathbf{f},\mathbf{g})\in_{\mathcal{D}}\mathcal{K}}\bigl[\mathbf{g} = g | \mathbf{C} = c \ \wedge\ \mathrm{PD}(c, \mathbf{g})\bigr] = \frac{\Pr_{(\mathbf{f},\mathbf{g})\in_{\mathcal{D}}\mathcal{K}}[\mathbf{g} = g]}{\Pr_{(\mathbf{f},\mathbf{g})\in_{\mathcal{D}}\mathcal{K}}[\mathrm{PD}(c, \mathbf{g})]}.$$

Lemma 2. *For every prospective ciphertext transcript $c \in \mathcal{C}$,*

$$\Pr_{(\mathbf{f},\mathbf{g})\in_{\mathcal{D}}\mathcal{K}}\bigl[\neg PD(c, \mathbf{g})\bigr] < u^2 * 2^{-n}.$$

Lemma 3.

$$\Pr_{(\mathbf{f},\mathbf{g})\in_{\mathcal{D}}\mathcal{K}}\bigl[\neg\mathbf{PD}\ \bigr] < u^2 * 2^{-n}.$$

The following lemma follows from Lemmas 1 and 2 and is used to prove Lemmas 6 and 8.

Lemma 4. *For every triple of n-bit constants a, b, and d such that $a \neq b$ and for every prospective ciphertext transcript $c \in \mathcal{C}$,*

$$\Pr_{(\mathbf{f},\mathbf{g})\in_{\mathcal{D}}\mathcal{K}}\bigl[\mathbf{g}(a) \oplus \mathbf{g}(b) = d\ \ \wedge \mathbf{C} = c \wedge \mathrm{PD}(c, \mathbf{g})\bigr] \leq 2^{-n+1} * \Pr_{(\mathbf{f},\mathbf{g})\in_{\mathcal{D}}\mathcal{K}}[\mathbf{C} = c].$$

The following lemma is also used to prove Lemma 6.

Lemma 5. *For every prospective ciphertext transcript $c \in \mathcal{C}$ and its corresponding forged transcript c', either* **E0** *or* **E1** *holds for c, or*

$$\begin{gathered}\exists x \in [1..l' - 1]\, \forall t \in [1..z]\, \forall j \in [1..l^t - 1]\ : \\ \bigl(\mathrm{IV}'(c) + \sigma'_x(c) = \mathrm{IV}^t(c) + \sigma^t_j(c)\bigr) \Rightarrow (c'_x \neq c^t_j).\end{gathered}$$

Lemma 6. *Let events* **E0**, **E1**, *and* **E2** *be as in Theorem 1. Then,*

$$\Pr_{(\mathbf{f},\mathbf{g})\in_{\mathcal{D}}\mathcal{K}}\bigl[\neg(\mathbf{E0}\ \vee \mathbf{E1}\ \vee \mathbf{E2})\ \wedge\ \mathbf{PD}\bigr] < (2u + 2v) * 2^{-n}.$$

Lemma 7. $\Pr_{(\mathbf{f},\mathbf{g})\in_{\mathcal{D}}\mathcal{K}}[\bigoplus_{j=1}^{\mathbf{L}'-1} \mathbf{P}'_j = 0 \,|\, \mathbf{E2}\,] \leq \frac{v}{2^n - (u+v)}$.

Lemma 8. $\Pr_{(\mathbf{f},\mathbf{g})\in_{\mathcal{D}}\mathcal{K}}[\bigoplus_{j=1}^{\mathbf{L}'-1} \mathbf{P}'_j = 0\ \wedge\ (\mathbf{E0} \vee \mathbf{E1}) \wedge \mathbf{PD}\,] \leq z * 2^{-n+2}$.

Proof of Theorem 1 (*continued*):

$$\begin{aligned}&\Pr_{(\mathbf{f},\mathbf{g})\in_{\mathcal{D}}\mathcal{K}}\left[\bigoplus_{j=1}^{\mathbf{L}'-1} \mathbf{P}'_j = 0\right] \\ &\quad\leq \Pr\left[\bigoplus_{j=1}^{\mathbf{L}'-1} \mathbf{P}'_j = 0\ \wedge \mathbf{PD}\right] + \Pr[\neg\mathbf{PD}]\end{aligned}$$

$$
\begin{aligned}
&\le \Pr\left[\bigoplus_{j=1}^{\mathbf{L}'-1} \mathbf{P}'_j = 0 \ \wedge (\mathbf{E0} \vee \mathbf{E1} \vee \mathbf{E2}) \ \wedge \ \mathbf{PD}\right] \\
&\quad + \Pr\left[\bigoplus_{j=1}^{\mathbf{L}'-1} \mathbf{P}'_j = 0 \ \wedge \neg(\mathbf{E0} \vee \mathbf{E1} \vee \mathbf{E2}) \ \wedge \ \mathbf{PD}\right] + \Pr[\neg\mathbf{PD}] \\
&\le \Pr\left[\bigoplus_{j=1}^{\mathbf{L}'-1} \mathbf{P}'_j = 0 \ \wedge (\mathbf{E0} \vee \mathbf{E1}) \ \wedge \ \mathbf{PD}\right] + \Pr\left[\bigoplus_{j=1}^{\mathbf{L}'-1} \mathbf{P}'_j = 0 \ \wedge \mathbf{E2}\right] \\
&\quad + \Pr\big[\neg(\mathbf{E0} \vee \mathbf{E1} \vee \mathbf{E2}) \ \wedge \ \mathbf{PD}\big] + \Pr[\neg\mathbf{PD}] \\
&\le z * 2^{-n+2} + \frac{v}{2^n - (u+v)} \\
&\quad + (u+v) * 2^{-n+1} + u^2 * 2^{-n} \quad \text{(by Lemmas 8, 7, 6, and 3 resp.)} \\
&\le \left(u^2 + 2u + 3v + 4z\right) * 2^{-n} + O(u+v) * v * 2^{-2n}. \qquad \square
\end{aligned}
$$

4.1. *Proofs of the Lemmas*

Lemma 1. *For every prospective ciphertext transcript $c \in \mathcal{C}$ and for any function $g \in \mathcal{G}$ such that $PD(c, g)$,*

$$\Pr_{(\mathbf{f},\mathbf{g}) \in_{\mathcal{D}} \mathcal{K}}\big[\mathbf{g} = g \mid \mathbf{C} = c \ \wedge \ PD(c, \mathbf{g})\big] = \frac{\Pr_{(\mathbf{f},\mathbf{g}) \in_{\mathcal{D}} \mathcal{K}}[\mathbf{g} = g]}{\Pr_{(\mathbf{f},\mathbf{g}) \in_{\mathcal{D}} \mathcal{K}}[\mathrm{PD}(c, \mathbf{g})]}.$$

Proof. Now,

$$
\begin{aligned}
&\Pr_{(\mathbf{f},\mathbf{g}) \in_{\mathcal{D}} \mathcal{K}}\big[\mathbf{g} = g \mid \mathbf{C} = c \ \wedge \ \mathrm{PD}(c, \mathbf{g})\big] \\
&\quad = \frac{\Pr[\mathbf{g} = g \ \wedge \ \mathbf{C} = c \ \wedge \ \mathrm{PD}(c, \mathbf{g})]}{\Pr[\mathbf{C} = c \ \wedge \ \mathrm{PD}(c, \mathbf{g})]}.
\end{aligned}
$$

We first consider the numerator:

$$
\begin{aligned}
&\Pr\big[\mathbf{g} = g \ \wedge \ \mathbf{C} = c \ \wedge \ \mathrm{PD}(c, \mathbf{g})\big] \\
&\quad = \sum_{f'} \Pr\big[\mathbf{g} = g \ \wedge \ \mathbf{f} = f' \ \wedge \ \mathbf{C} = c \ \wedge \ \mathrm{PD}(c, \mathbf{g})\big] \\
&\quad = \ \Pr\big[\mathbf{g} = g \ \wedge \ \mathbf{f} \in F_{c,g} \ \wedge \ \mathbf{C} = c \ \wedge \ \mathrm{PD}(c, \mathbf{g})\big] \\
&\quad = \ \Pr[\mathbf{g} = g \ \wedge \ \mathbf{f} \in F_{c,g}],
\end{aligned}
$$

where $F_{c,g}$ is the set of permutation defined as follows: since $\mathrm{PD}(c, g)$ holds, all the raw ciphertext block variables $N(c, g)$ are distinct. Similarly, all whitened plaintext block variables $M(c, g)$ are distinct. These $M(c, g)$ and $N(c, g)$ values determine a unique permutation $f_{c,g}$ projected on a number of blocks given by c (i.e., $|c| - z$). Thus, for c, g s.t. $\mathrm{PD}(c, g)$, define $F_{c,g}$ to be the set of permutations with the projection on these blocks equal to $f_{c,g}$ and with no other restrictions on other blocks. If c, g are such that $\neg\mathrm{PD}(c, g)$, then we let $F_{c,g}$ to be the empty set.

The last equality above follows as the two events are identical. To see that $\mathbf{g} = g$ and $\mathbf{f} \in F_{c,g}$ implies $\mathbf{C} = c$, first note that since $\mathbf{f}$ is in $F_{c,g}$, the set $F_{c,g}$ is nonempty, and hence $\mathrm{PD}(c, g)$ holds, which implies $\mathrm{PD}(c, \mathbf{g})$. Now note that the first plaintext message p^1 is fixed, and moreover the first initial vector c_0^1 is fixed, which fixes $\mathbf{M}^1$ to $M^1(c, g)$ by (7) and (5). Since $\mathbf{N}^1 = \mathbf{f}(\mathbf{M}^1)$ (by (8)), this fixes $\mathbf{N}^1$ to $\mathbf{f}(M^1(c, g))$, which is $f_{c,g}(M^1(c, g))$ by the definition of $F_{c,g}$ and $f_{c,g}$. But $f_{c,g}(M^1(c, g))$ is same as $N^1(c, g)$ by the definition of $f_{c,g}$. Thus, $\mathbf{C}^1$ is fixed to c^1 by (2) and (6). This, in turn, fixes $\mathbf{P}^2 = P^2(c)$ by (1) and fixes $\mathbf{C}_0^2$ to c_0^2 as the initial vectors are chosen as a deterministic function of the previous ciphertexts, and so forth inductively.

We now consider the denominator:

$$\begin{aligned}
&\Pr\bigl[\mathbf{C} = c \ \wedge \ \mathrm{PD}(c, \mathbf{g})\bigr] \\
&\quad = \sum_{g' \in \mathcal{G}} \sum_{f'} \Pr\bigl[\mathbf{g} = g' \ \wedge \ \mathbf{f} = f' \ \wedge \ \mathbf{C} = c \ \wedge \ \mathrm{PD}(c, \mathbf{g})\bigr] \\
&\quad = \sum_{g' \in \mathcal{G}\,:\mathrm{PD}(c,g')} \Pr\bigl[\mathbf{g} = g' \ \wedge \ \mathbf{f} \in F_{c,g'} \ \wedge \ \mathbf{C} = c \ \wedge \ \mathrm{PD}(c, \mathbf{g})\bigr] \\
&\quad = \sum_{g' \in \mathcal{G}\,:\mathrm{PD}(c,g')} \Pr[\mathbf{g} = g' \ \wedge \ \mathbf{f} \in F_{c,g'}].
\end{aligned}$$

The above follows because, as before, when $\mathrm{PD}(c, g')$ holds, there is a fixed set of permutations $F_{c,g'}$ with a unique projection (on $|c| - z$ blocks) compatible with $\mathbf{g} = g'$ and $\mathbf{C} = c$.

Since $\mathbf{g}$ and $\mathbf{f}$ are independent, we have from the above analysis:

$$\begin{aligned}
&\Pr\bigl[\mathbf{g} = g | \mathbf{C} = c \ \wedge \ \mathrm{PD}(c, \mathbf{g})\bigr] \\
&\quad = \frac{\Pr[\mathbf{g} = g \ \wedge \ \mathbf{f} \in F_{c,g}]}{\sum_{g' \in \mathcal{G}\,:\mathrm{PD}(c,g')} \Pr[\mathbf{g} = g' \ \wedge \ \mathbf{f} \in F_{c,g'}]} \\
&\quad = \frac{\Pr[\mathbf{g} = g] * \Pr[\mathbf{f} \in F_{c,g}]}{\sum_{g' \in \mathcal{G}\,:\mathrm{PD}(c,g')} \Pr[\mathbf{g} = g'] * \Pr[\mathbf{f} \in F_{c,g'}]} = \frac{\Pr[\mathbf{g} = g]}{\Pr[\mathrm{PD}(c, \mathbf{g})]}.
\end{aligned}$$

The last equality follows because in distribution $\mathcal{D}$, $\mathbf{f}$ is chosen uniformly, $F_{c,g}$ is nonempty by the hypothesis of the lemma, and $F_{c,g'}$ is nonempty as $\mathrm{PD}(c, g')$ holds, and further $|F_{c,g}| = |F_{c,g'}|$. □

Lemma 2. *For every prospective ciphertext transcript $c \in \mathcal{C}$,*

$$\Pr_{(\mathbf{f},\mathbf{g}) \in_{\mathcal{D}} \mathcal{K}}\bigl[\neg\mathrm{PD}(c, \mathbf{g})\bigr] < \bigl(u^2\bigr) * 2^{-n}.$$

Proof. Recall that event $\mathrm{PD}(c, \mathbf{g})$ is

$$\forall i, i1 \in [1..z], \forall j \in \left[1..l^i - 1\right], \forall j1 \in \left[1..l^{i1} - 1\right], (i, j) \neq (i1, j1) :$$
$$\left(M^i_j(c, \mathbf{g}) \neq M^{i1}_{j1}(c, \mathbf{g})\right) \wedge \left(N^i_j(c, \mathbf{g}) \neq N^{i1}_{j1}(c, \mathbf{g})\right).$$

Then $\neg$ $\mathrm{PD}(c, \mathbf{g})$ can be written as

$$\exists i, i1 \in [1..z], \exists j \in \left[1..l^i - 1\right], \exists j1 \in \left[1..l^{i1} - 1\right] : (i, j) \neq (i1, j1) \wedge$$
$$\left[\left(M^i_j(c, \mathbf{g}) = M^{i1}_{j1}(c, \mathbf{g})\right) \vee \left(N^i_j(c, \mathbf{g}) = N^{i1}_{j1}(c, \mathbf{g})\right)\right].$$

Since we have a constant ciphertext transcript c and hence a constant plaintext $P(c)$ as well, the probability of any event $M^i_j(c, \mathbf{g}) = M^{i1}_{j1}(c, \mathbf{g})$ is just 2^{-n}, as each $M^i_j(c, \mathbf{g})$ is just a function of $\mathbf{g}$, the latter being chosen from an XOR-universal set $\mathcal{G}$, and given that the initial vectors are safe. Similarly for $N^i_j(c, \mathbf{g}) = N^{i1}_{j1}(c, \mathbf{g})$. The lemma follows by union bound. □

Lemma 3.

$$\Pr_{(\mathbf{f},\mathbf{g})\in_{\mathcal{D}}\mathcal{K}}[\neg\ \mathbf{PD}] < \left(u^2\right) * 2^{-n}.$$

Proof. For $c = c^1, c^2, \ldots, c^i$, $i \leq z$, define $\#(c)$ to be $(2^n)!/(2^n - \sum_{j=1}^{i}(|c^j| - 1))!$. In other words, $\#(c)$ is the ratio of the number of permutations on 2^n blocks and $|F_{c,g}|$ (as defined in Lemma 1, and which is same irrespective of g, as long as $\mathrm{PD}(c, g)$ holds).

Recall from the proof of Lemma 1 that, for $c \in \mathcal{C}$,

$$\Pr\left[\mathbf{C} = c \ \wedge \ \mathrm{PD}(c, \mathbf{g})\right] = \sum_{g' \in \mathcal{G} : PD(c, g')} \Pr[\mathbf{g} = g' \ \wedge \ \mathbf{f} \in F_{c,g'}].$$

We use this to get

$$\begin{aligned}
\Pr\left[\mathrm{PD}(\mathbf{C}, \mathbf{g})\right] &= \sum_{c\in\mathcal{C}} \Pr\left[\mathbf{C} = c \ \wedge \ \mathrm{PD}(c, \mathbf{g})\right] \\
&= \sum_{c\in\mathcal{C}} \sum_{g'\in\mathcal{G} : \mathrm{PD}(c,g')} \Pr[\mathbf{g} = g' \ \wedge \ \mathbf{f} \in F_{c,g'}] \\
&= \sum_{c\in\mathcal{C}} \sum_{g'\in\mathcal{G} : \mathrm{PD}(c,g')} \Pr[\mathbf{g} = g'] * \Pr[\mathbf{f} \in F_{c,g'}] \\
&= \sum_{c\in\mathcal{C}} \sum_{g'\in\mathcal{G} : \mathrm{PD}(c,g')} \Pr[\mathbf{g} = g'] * \frac{1}{\#(c)} \\
&= \sum_{c\in\mathcal{C}} \left(\frac{1}{\#(c)} * \ \Pr[\mathrm{PD}(c, \mathbf{g}]\right) \\
&\geq \min_{c\in\mathcal{C}}\left\{\Pr\left[\mathrm{PD}(c, \mathbf{g})\right]\right\} \ * \sum_{c\in\mathcal{C}} \frac{1}{\#(c)}
\end{aligned}$$

$$\geq \left(1-u^2 * 2^{-n}\right) * \sum_{c\in\mathcal{C}} \frac{1}{\#(c)} \quad \text{(by Lemma 2)}$$
$$\geq \left(1-u^2 * 2^{-n}\right).$$

The last inequality follows by

$$\begin{aligned}\sum_{c\in\mathcal{C}} \frac{1}{\#(c)} &= \sum_{c^1\in\mathcal{L}()} \cdots \sum_{c^i\in\mathcal{L}(c^1,\ldots,c^{i-1})} \cdots \sum_{c^z\in\mathcal{L}(c^1,\ldots,c^{z-1})} \frac{1}{\#(c^1;\ldots;c^z)} \\ &\geq \sum_{c^1\in\mathcal{L}()} \cdots \sum_{c^i\in\mathcal{L}(c^1,\ldots,c^{i-1})} \cdots \sum_{c^{z-1}\in\mathcal{L}(c^1,\ldots,c^{z-2})} \frac{1}{\#(c^1;\ldots;c^{z-1})} \\ &\geq \cdots \geq \sum_{c^1\in\mathcal{L}()} \frac{1}{\#(c^1)} \geq 1,\end{aligned}$$

where we used the fact that the number of c^z in $\mathcal{L}(c^1,\ldots,c^{z-1})$ is $2^{n(|c^z|-1)}$, and so on. □

We will need the following lemma to prove Lemmas 6 and 8.

Lemma 4. *For every triple of n-bit constants a, b, and d such that $a \neq b$ and for every prospective ciphertext transcript $c \in \mathcal{C}$,*

$$\mathrm{Pr}_{(\mathbf{f},\mathbf{g})\in_{\mathcal{D}}\mathcal{K}}\left[\mathbf{g}(a)\oplus\mathbf{g}(b)=d \ \wedge \mathbf{C}=c\wedge \mathrm{PD}(c,\mathbf{g})\right] \leq 2^{-n+1} * \mathrm{Pr}_{(\mathbf{f},\mathbf{g})\in_{\mathcal{D}}\mathcal{K}}[\mathbf{C}=c].$$

Proof.

$$\begin{aligned}&\Pr\left[\mathbf{g}(a)\oplus\mathbf{g}(b)=d\wedge\mathbf{C}=c\wedge\ \mathrm{PD}(c,\mathbf{g})\right]\\ &\quad=\Pr\left[\mathbf{g}(a)\oplus\mathbf{g}(b)=d\mid\mathbf{C}=c\wedge\ \mathrm{PD}(c,\mathbf{g})\right]*\Pr\left[\mathbf{C}=c\wedge\ \mathrm{PD}(c,\mathbf{g})\right]\\ &\quad\leq\Pr\left[\mathbf{g}(a)\oplus\mathbf{g}(b)=d\mid\mathbf{C}=c\wedge\ \mathrm{PD}(c,\mathbf{g})\right]\ *\Pr[\mathbf{C}=c].\end{aligned}$$

The first factor is upper bounded by $2^{-n}/\Pr[\mathrm{PD}(c,\mathbf{g})]$ by using Lemma 1. To see this,

$$\begin{aligned}&\mathrm{Pr}_{(\mathbf{f},\mathbf{g})\in_{\mathcal{D}}\mathcal{K}}\left[\mathbf{g}(a)\oplus\mathbf{g}(b)=d\mid\mathbf{C}=c\wedge\ \mathrm{PD}(c,\mathbf{g})\right]\\ &\quad=\sum_{g\in\mathcal{G}}\Pr\left[\mathbf{g}=g\ \wedge\ \mathbf{g}(a)\oplus\mathbf{g}(b)=d\mid\mathbf{C}=c\wedge\ \mathrm{PD}(c,\mathbf{g})\right]\\ &\quad=\sum_{g\in\mathcal{G}}\Pr\left[\mathbf{g}=g\mid\mathbf{C}=c\wedge\ \mathrm{PD}(c,\mathbf{g})\right]\\ &\qquad\ *\ \Pr\left[\mathbf{g}(a)\oplus\mathbf{g}(b)=d\mid\mathbf{g}=g\ \wedge\ \mathbf{C}=c\wedge\ \mathrm{PD}(c,\mathbf{g})\right]\\ &\quad=\sum_{g\in\mathcal{G}:g(a)\oplus g(b)=d}\Pr\left[\mathbf{g}=g\mid\mathbf{C}=c\wedge\mathrm{PD}(c,\mathbf{g})\right]\end{aligned}$$

$$\begin{aligned}
&\le \sum_{g\in\mathcal{G}:g(a)\oplus g(b)=d} \frac{\Pr_{(\mathbf{f},\mathbf{g})\in_{\mathcal{D}}\mathcal{K}}[\mathbf{g}=g]}{\Pr[\mathrm{PD}(c,\mathbf{g})]} \quad \text{(by Lemma 1)}\\
&= \frac{1}{\Pr[\mathrm{PD}(c,\mathbf{g})]} * \sum_{g\in\mathcal{G}:g(a)\oplus g(b)=d} \frac{1}{|\mathcal{G}|}\\
&= \frac{1}{\Pr[\mathrm{PD}(c,\mathbf{g})]} * \Pr_{\mathbf{g}\in\mathcal{G}}\bigl[\mathbf{g}(a)\oplus\mathbf{g}(b)=d\bigr]\\
&= \frac{2^{-n}}{\Pr[\mathrm{PD}(c,\mathbf{g})]}.
\end{aligned}$$

Now by Lemma 2 and the hypothesis of Theorem 1 that $4u^2 < 2^n$, we have $\Pr[\mathrm{PD}(c,\mathbf{g})] > 1/2$, and hence the lemma follows. □

Lemma 5. *For every prospective ciphertext transcript $c \in \mathcal{C}$ and its corresponding forged transcript c', either* **E0** *or* **E1** *holds for c, or*

$$\exists x \in [1..l'-1]\,\forall t \in [1..z]\,\forall j \in [1..l^t-1] :$$
$$\bigl(\mathrm{IV}'(c)+\sigma'_x(c)=\mathrm{IV}^t(c)+\sigma^t_j(c)\bigr) \Rightarrow \bigl(c'_x \ne c^t_j\bigr)$$

Proof. Since the initial vectors are safe, by definition, for all $t\in[1..z]$, $\mathrm{IV}^t+l^t-1 < 2^n-1$. Also, $\mathrm{IV}'+l'-1 < 2^n-1$ (see Step 3 of D-IAPM).

Recall that $\sigma^t_j(c)$ is the post-whitening offset for block j in message t. As it is clear from context, we will drop the argument c from σ and σ'.

If for all message indices $t\in[1..z]$, the forged initial vector IV' is not equal to IV^t (along with their offsets), i.e., for all t: $\mathrm{IV}' \notin \mathrm{IV}^t + [0..l^t-2]$, then we can take $x = l'-1$, in which case $\sigma'_x = 0$, and hence $\mathrm{IV}'+\sigma'_x = \mathrm{IV}'$. Now, note that σ^t_j ranges from 0 to l^t-2, and hence this x satisfies the lemma vacuously.

Next, consider the case where there exists a $t\in[1..z]$ such that IV' equals IV^t with some offset, i.e., $\mathrm{IV}' \in \mathrm{IV}^t+[0..l^t-2]$. As the initial vectors are safe, there can be at most one such t. Also, note that $t \le Z(c)$, as for $i > Z(c)$, $l^i = 1$. There are two main sub-cases.

(a) For every $x\in[1..l'-1]$, $\mathrm{IV}'+\sigma'_x \in \mathrm{IV}^t+[0..l^t-2]$ (i.e., the set IV' along with its offsets is contained in the set IV^t along with its offsets). Again, as the initial vectors are safe, $\mathrm{IV}'+\sigma'_x$ cannot equal $\mathrm{IV}^{t'}+\sigma^{t'}_{j'}$ for some other $t'\ne t$. Also, since σ'_x ranges over values from 0 to $l'-2$, we have $\mathrm{IV}'+l'-2 \le \mathrm{IV}^t+l^t-2$. There are two further sub-cases.

(a1) ($\mathrm{IV}'=\mathrm{IV}^t$: *truncation attempt*). Here $l' \le l^t$. If c' is a (strict) prefix of c^t, then we pick the last block of c', i.e., we let $x = l'-1$. Since it is the last block, the post-whitening offset is zero, i.e., $\sigma'_x = 0$. Since $\mathrm{IV}'=\mathrm{IV}^t$, the value $\mathrm{IV}'+\sigma'_x$ will be same as $\mathrm{IV}^t+\sigma^t_j$ (for some j) only if $\sigma^t_j=\sigma'_x=0$, or in other words only if $j = l^t-1$. Now, c' being a prefix of c^t, if $c'_x = c^t_{l^t-1}$, then it forces event **E0** (the deletion attempt) for c (note $t \le Z(c)$).

Otherwise, if c' is not a prefix of c^t, let x be the least index in which c' and c^t differ. If for some j, $\sigma^t_j = \sigma'_x$, then either $\sigma^t_j = \sigma'_x = 0$ or $j = x$. In the latter case, $c'_x \oplus c^t_j = c'_x \oplus c^t_x$, which is nonzero as x is the index in which c' and c^t differ. In the former case, $j = l^t - 1$ and $x = l' - 1$. In this case, $c'_x \oplus c^t_j = c'_{l'-1} \oplus c^t_{l^t-1}$. If this quantity is zero, then since x $(= l' - 1)$ was the least index in which c^t and c' differed, event **E0** would hold for c.

(a2) ($\mathrm{IV}' \neq \mathrm{IV}^t$: *rotation attempt*) Note that, if instead of general safe initial vectors we had required the initial vectors to have enough least significant bits to be zero, so that the offsets could be embedded unambiguously, then this case would not arise. In other words, with this restriction, $\mathrm{IV}' + \sigma'_x \in \mathrm{IV}^t + [0..l^t - 2]$ could only happen if $\mathrm{IV}' = \mathrm{IV}^t$. However, in the case of general safe initial vectors, this case could certainly arise. We will show that for each x, there is a unique $j_x \in [1..l^t - 1]$ such that $\mathrm{IV}' + \sigma'_x = \mathrm{IV}^t + \sigma^t_{j_x}$. Recall that $\sigma'_x = x$ except for $x = l' - 1$, in which case it drops to zero, i.e., $x - (l' - 1)$. Hence, for the above j_x to exist, $j_{l'-1}$ must drop by $(l' - 1)$ as well (we formalize this in the next paragraph). Next, we will show that either for some x, $c'_x \neq c^t_{j_x}$ or event **E1** holds (i.e., c' is a rotation of a portion of c^t).

To be more precise, we first note that $\mathrm{IV}' \geq \mathrm{IV}^t + 1$, as IV' is in $\mathrm{IV}^t + [0..l^t - 2]$. Thus from $\mathrm{IV}' \leq \mathrm{IV}^t + l^t - l'$ it follows that $\mathrm{IV}' = \mathrm{IV}^t + s$ for some s such that $1 \leq s \leq l^t - l'$ (thus satisfying **E1**(i)). Thus, for every $x \in [1..l' - 1]$, $\mathrm{IV}' + \sigma'_x = \mathrm{IV}^t + \sigma'_x + s$. Note that $1 \leq \sigma'_x + s \leq l^t - 2$, as $0 \leq \sigma'_x \leq l' - 2$ and $1 \leq s \leq l^t - l'$. Hence, for each $x \in [1..l' - 1]$, $\sigma^t_{\sigma'_x+s} = \sigma'_x + s$, and hence $\mathrm{IV}' + \sigma'_x = \mathrm{IV}^t + \sigma'_x + s = \mathrm{IV}^t + \sigma^t_{\sigma'_x+s}$. Thus, for each x, there is a unique j_x, namely $\sigma'_x + s$, such that $\mathrm{IV}' + \sigma'_x = \mathrm{IV}^t + \sigma^t_{j_x}$. Now, suppose that for all $x \in [1..l' - 1]$, $c'_x \oplus c^t_{j_x} = 0$, i.e., $c'_x = c^t_{\sigma'_x+s}$. But this implies that **E1** holds for c. Otherwise, we have an x such that $c'_x \oplus c^t_{j_x} \neq 0$, and the lemma follows as t and j_x are the only values for which $\mathrm{IV}' + \sigma'_x = \mathrm{IV}^t + \sigma^t_{j_x}$.

(b) (*extension attempt*) There exists an $x \in [1..l' - 1]$ such that $\mathrm{IV}' + \sigma'_x \notin \mathrm{IV}^t + [0..l^t - 2]$. Since $\mathrm{IV}' \in \mathrm{IV}^t + [0..l^t - 2]$, it follows that there exists an $x \in [1..l' - 2]$ such that $\mathrm{IV}' + \sigma'_x \notin \mathrm{IV}^t + [0..l^t - 2]$. For the least such x (and note $\sigma'_x = x$), we have $\mathrm{IV}' + x = \mathrm{IV}^t + l^t - 1$. Since the initial vectors are safe, there is no other t', j' such that $\mathrm{IV}' + \sigma'_x = \mathrm{IV}^{t'} + \sigma^{t'}_{j'}$, j' in $[1..l^{t'} - 1]$. Thus this x satisfies the lemma vacuously. The key observation here is that for every $t \in [1..Z(c)]$, the value $\mathrm{IV}^t + l^t - 1$ is never used as a post-whitening index in the first stage of the attack.

□

Lemma 6. *Let events* **E0**, **E1**, *and* **E2** *be as in Theorem* 1. *Then*,

$$\Pr_{(\mathbf{f},\mathbf{g}) \in_{\mathcal{D}} \mathcal{K}}\left[\neg(\mathbf{E0} \vee \mathbf{E1} \vee \mathbf{E2}) \wedge \mathbf{PD}\right] < (2u + 2v) * 2^{-n}.$$

Proof. To begin with, we have

$$\Pr\bigl[\neg(\mathbf{E0} \vee \mathbf{E1} \vee \mathbf{E2}) \,\wedge\, \mathbf{PD}\bigr] = \sum_{c \in \mathcal{C}} \Pr\bigl[\neg(\mathbf{E0} \vee \mathbf{E1} \vee \mathbf{E2}) \,\wedge \mathbf{C} = c \,\wedge\, \mathbf{PD}\bigr]. \quad (15)$$

Focusing on the negation of **E2**, the inside expression above (see the definition of **E2**) is the probability of the conjunction (one for each x) of disjunctions. Hence, it is upper bounded by the least (over x) of the probabilities of the disjunctions, which in turn is upper bounded by the sum of the probability of each disjunct. Thus, for any fixed ciphertext transcript c,

$$\begin{aligned}
&\Pr\bigl[\neg(\mathbf{E0} \vee \mathbf{E1} \vee \mathbf{E2}) \wedge \mathbf{C} = c \wedge \mathbf{PD}\bigr] \\
&\quad \le \min_{x \in [1..l'-1]} \Biggl\{ \sum_{t \in [1..z],\, j \in [1..|c^t|-1]} \Pr\bigl[(\mathbf{N}'_x = \mathbf{N}^t_j) \wedge \mathbf{C} = c \wedge \neg(\mathbf{E0} \vee \mathbf{E1}) \wedge \mathbf{PD}\bigr] \\
&\qquad + \sum_{j \in [1..l'-1],\, j \neq x} \Pr\bigl[(\mathbf{N}'_x = \mathbf{N}'_j) \wedge \mathbf{C} = c \wedge \mathbf{PD}\bigr] \Biggr\}.
\end{aligned}$$

Since each of the summands in the expression above has a conjunct $\mathbf{C} = c$ for some constant string c, it follows that $\mathbf{N}^t_j = N^t_j(c, \mathbf{g})$ and $\mathbf{N}'_x = N'_x(c, \mathbf{g})$. Thus, each of the summands in the first sum can be written (by equation (6)) as

$$\Pr\bigl[(\mathbf{g}'_{\sigma'_x} \oplus \mathbf{g}^t_{\sigma^t_j} = c'_x \oplus c^t_j) \;\wedge \mathbf{C} = c \wedge \neg(\mathbf{E0} \vee \mathbf{E1}) \wedge\; \mathrm{PD}(c, \mathbf{g})\bigr],$$

where $\mathbf{g}'_j$ is shorthand for $\mathbf{g}(\mathrm{IV}'(c) + j)$, and $\mathbf{g}^t_j$ is shorthand for $\mathbf{g}(\mathrm{IV}^t(c) + j)$. Now, by Lemma 4, each of these probabilities is upper bounded by $2^{-n+1} * \Pr[\mathbf{C} = c]$ as long as $\mathrm{IV}' + \sigma'_x(c) \neq \mathrm{IV}^t + \sigma^t_j(c)$. However, if $\mathrm{IV}' + \sigma'_x(c) = \mathrm{IV}^t + \sigma^t_j(c)$, then by Lemma 5 either (**E0** or **E1**) holds for c, or $c'_x \oplus c^t_j \neq 0$, which would make this probability zero. For the summands in the second sum, Lemma 4 is unconditionally applicable as $\sigma'_x(c) \neq \sigma'_j(c)$.

From (15), we then get

$$\Pr\bigl[\neg(\mathbf{E0} \;\vee \mathbf{E1} \;\vee \mathbf{E2}) \wedge \mathbf{PD}\bigr] \le (u + v) * 2^{-n+1}. \qquad \square$$

Lemma 7. $\Pr_{(\mathbf{f},\mathbf{g}) \in_{\mathcal{D}} \mathcal{K}}[\bigoplus_{j=1}^{\mathbf{L}'-1} \mathbf{P}'_j = 0 \mid \mathbf{E2}] \le \frac{v}{2^n - (u+v)}$.

Proof. For each x in $[1..v - 1]$, let $\mathbf{E2}(x)$ denote the event **E2** holding with this x. First note that

$$\Pr\left[\bigoplus_{j=1}^{\mathbf{L}'-1} \mathbf{P}'_j = 0 \mid \mathbf{E2}\right] \le \sum_x \Pr\left[\bigoplus_{j=1}^{\mathbf{L}'-1} \mathbf{P}'_j = 0 \mid \mathbf{E2}(x)\right], \quad (16)$$

which follows from $\Pr[A|B \vee C] \le \Pr[A|B] + \Pr[A|C]$ for arbitrary events A, B, and C.

Now, for any x in $[1..\mathbf{L}'-1]$, we have $\bigoplus_{j=1}^{\mathbf{L}'-1}\mathbf{P}'_j = 0$ iff

$$\mathbf{f}^{-1}(\mathbf{N}'_x) = \mathbf{M}'_x = \bigoplus_{j=1, j\neq x}^{\mathbf{L}'-1} \big(\mathbf{M}'_j \oplus \mathbf{g}(\mathbf{C}'_0 + j)\big) \oplus \mathbf{g}(\mathbf{C}'_0 + x).$$

The first equation follows from (12), (11), and (10), and the "iff" claim follows by (14). Under the condition $\mathbf{E2}(x)$, given any value of the RHS of (16), we will show that the LHS of (16) can take (at least) $2^n - (\mu + v - 2)$ values, each with equal probability, and hence the probability of LHS being equal to RHS is at most $1/(2^n - (u+v))$.

To this end, we calculate the above probability by fixing $\mathbf{g}$, each $\mathbf{N}^t_j$, and each $\mathbf{M}'_j$ $(j \neq x)$, and summing the probability over all the fixings:

$$\begin{aligned}
&\Pr\left[\mathbf{f}^{-1}(\mathbf{N}'_x) = \bigoplus_{j=1, j\neq x}^{\mathbf{L}'-1} \big(\mathbf{M}'_j \oplus \mathbf{g}(\mathbf{C}'_0 + j)\big) \oplus \mathbf{g}(\mathbf{C}'_0 + x) \mid \mathbf{E2}(x)\right] \\
&= \sum_{g, n^t_j, m'_j (j\neq x)} \Pr\Big[\mathbf{g} = g \wedge \bigwedge\big(\mathbf{N}^t_j = n^t_j\big) \wedge \bigwedge_{j\neq x}(\mathbf{M}'_j = m'_j) \wedge \mathbf{f}^{-1}(\mathbf{N}'_x) \\
&\qquad = \bigoplus \ldots \mid \mathbf{E2}(x)\Big] \\
&= \sum \Pr\Big[\mathbf{f}^{-1}\big(N'_x(\mathbf{C}, g)\big) = \bigoplus \ldots \mid \mathbf{E2}(x) \ \wedge \mathbf{g} \\
&\qquad = g \wedge \bigwedge\big(\mathbf{N}^t_j = n^t_j\big) \wedge \bigwedge_{j\neq x}(\mathbf{M}'_j = m'_j)\Big] \\
&\qquad * \Pr\Big[\mathbf{g} = g \wedge \bigwedge\big(\mathbf{N}^t_j = n^t_j\big) \wedge \bigwedge_{j\neq x}(\mathbf{M}'_j = m'_j) \mid \mathbf{E2}(x)\Big].
\end{aligned} \tag{17}$$

We now show that event $\mathbf{E2}(x)$ and $\mathbf{C}'_0$ are completely determined by (i) the whitening function $\mathbf{g}$, and (ii) $\mathbf{N}^t_j$ ($t \in [1..z]$, $j \in [1..\mathbf{L}^t - 1]$). First, by (2), (4), (5), (7), and (8), $\mathbf{N}^t_j$ and $\mathbf{g}$ completely determine $\mathbf{C}$. Hence, the adversarial choice of $\mathbf{C}'_0$, $\mathbf{L}'$, and in fact the whole of $\mathbf{C}'$ is determined by these quantities. On fixing $\mathbf{g}$ to g, and fixing $\mathbf{N}^t_j$ to n^t_j, say the ciphertext $\mathbf{C}$ fixes to c, the plaintext $\mathbf{P}$ fixes to p, and the whitened plaintext $\mathbf{M}^t_j$ fixes to m^t_j. Further, say $\mathbf{L}'$ fixes to l', and $\mathbf{C}'_j$ fixes to c'_j , $j \in [0..l'-1]$.

Further, note that for all $j \in [1..\mathbf{L}'-1]$, $\mathbf{N}'_j = \mathbf{C}'_j \oplus \mathbf{g}(\mathbf{C}'_0 + \sigma'_j(\mathbf{C}))$ (by (10) and (9)). Thus, for each j (including x), $\mathbf{N}'_j$ fixes to a constant value, say n'_j. Thus, the conjunction of the conditions $(\mathbf{g} = g)$, $(\mathbf{N}^t_j = n^t_j)$, and $\mathbf{E2}(x)$ is equivalent to the conjunction of $(\mathbf{g} = g)$, $(\mathbf{N}^t_j = n^t_j)$, and the condition that n'_x is different from all other n'_j and from all n^t_j.

The first factor in the above summation (17) now simplifies to

$$\Pr\Bigl[\mathbf{f}^{-1}(n'_x) = \bigoplus\bigl(m'_j \oplus g(c'_0 + j)\bigr) \oplus g(c'_0 + x) \mid$$

$$(\mathbf{g} = g) \wedge \bigwedge(\mathbf{N}^t_j = n^t_j) \wedge \bigwedge_{j\neq x}(\mathbf{M}'_j = m'_j) \wedge \bigwedge_{t,j}(n'_x \neq n^t_j) \wedge \bigwedge_{j:\, j\neq x}(n'_x \neq n'_j)\Bigr]. \quad (18)$$

Now note that $(\mathbf{g} = g) \wedge \bigwedge(\mathbf{N}^t_j = n^t_j)$ is implied by $(\mathbf{g} = g) \wedge \bigwedge(\mathbf{f}^{-1}(n^t_j) = m^t_j)$, where m^t_j is as fixed above. This follows by induction, noting that m^1 is determined by g, the fixed adversarial value P^1, and $\mathbf{C}^1_0$ (also see the second paragraph of the proof of Lemma 1 for a similar argument). The conditioning in the above probability 18 is then same as (by (8), (12), and (11))

$$(\mathbf{g} = g) \wedge \bigwedge\bigl(\mathbf{f}^{-1}(n^t_j) = m^t_j\bigr) \wedge \bigwedge_{j\neq x}\bigl(\mathbf{f}^{-1}(n'_j) = m'_j\bigr) \wedge \bigwedge_{t,j}(n'_x \neq n^t_j) \wedge \bigwedge_{j:\, j\neq x}(n'_x \neq n'_j).$$

Since the permutation **f** is independent of the whitening function **g**, the above probability (18) (i.e., the first factor of summation (17)) is at most $1/(2^n - (\mu + v))$. The lemma follows by summing over all x. □

Lemma 8. $\Pr_{(\mathbf{f},\mathbf{g})\in_{\mathcal{D}}\mathcal{K}}[\bigoplus_{j=1}^{\mathbf{L}'-1} \mathbf{P}'_j = 0 \wedge (\mathbf{E0} \vee \mathbf{E1}) \wedge \mathbf{PD}] \leq z * 2^{-n+2}$.

Proof. In the following, we will drop the argument **C** from σ and σ', as it will be clear from context. We will also use $\mathbf{g}^i_j$ as shorthand for $\mathbf{g}(\mathbf{C}^i_0 + j)$.

We first consider the event **E0** happening. Since **E0**(i) implies that for some message $i : \mathbf{C}'_0 = \mathbf{C}^i_0$, it also implies, along with **E0**(ii) and $\sigma'_{\mathbf{L}'-1} = \sigma^i_{\mathbf{L}^i-1} = 0$, that $\mathbf{N}'_{\mathbf{L}'-1} = \mathbf{N}^i_{\mathbf{L}^i-1}$, and hence $\mathbf{M}'_{\mathbf{L}'-1} = \mathbf{M}^i_{\mathbf{L}^i-1}$. Further, **E0**(i) also implies $\mathbf{N}'_j = \mathbf{N}^i_j$ (for $j = 1$ to $\mathbf{L}' - 2$), which in turn implies $\mathbf{M}'_j = \mathbf{M}^i_j$ and hence also $\mathbf{P}'_j = \mathbf{P}^i_j$. Thus, we have

$$\left(\bigoplus_{j=1}^{\mathbf{L}'-1} \mathbf{P}'_j = 0\right) \wedge \mathbf{E0}$$

$$\Rightarrow \left(\bigoplus_{j=1}^{\mathbf{L}'-1} \mathbf{P}'_j = 0\right) \wedge \mathbf{E0} \wedge \exists i\, \bigl(\mathbf{M}'_{\mathbf{L}'-1} = \mathbf{M}^i_{\mathbf{L}^i-1}\bigr)$$

$$\Rightarrow \left(\bigoplus_{j=1}^{\mathbf{L}'-1} \mathbf{P}'_j = 0\right) \wedge \mathbf{E0} \wedge \exists i\, \bigl(\mathbf{P}'_{\mathbf{L}'-1} \oplus \mathbf{g}^i_{\mathbf{L}'-1} = \mathbf{M}^i_{\mathbf{L}^i-1}\bigr)$$

$$\Rightarrow \exists i \left(\bigoplus_{j=1}^{\mathbf{L}'-2}\bigl(\mathbf{P}^i_j\bigr) \oplus \mathbf{M}^i_{\mathbf{L}^i-1} \oplus \mathbf{g}^i_{\mathbf{L}'-1} = 0\right)$$

$$\equiv \exists i \left(\bigoplus_{j=1}^{\mathbf{L}'-2} \left(\mathbf{P}_j^i\right) \oplus \bigoplus_{j=1}^{\mathbf{L}^i-2} \left(\mathbf{P}_j^i\right) \oplus \mathbf{g}_{\mathbf{L}^i-1}^i \oplus \mathbf{g}_{\mathbf{L}'-1}^i = 0 \right).$$

Since $\mathcal{G}$ is XOR-universal and $\mathbf{L}' < \mathbf{L}^i$ and since the initial vectors are safe, we have

$$\begin{aligned}
&\Pr\left[\bigoplus_{j=1}^{\mathbf{L}'-1} \mathbf{P}'_j = 0 \ \wedge \ \mathbf{E0} \wedge \mathbf{PD} \right] \\
&\quad \le \Pr\left[\mathbf{PD} \ \wedge \ \exists i \left(\bigoplus_{j=1}^{\mathbf{L}'-2} \left(\mathbf{P}_j^i\right) \oplus \bigoplus_{j=1}^{\mathbf{L}^i-2} \left(\mathbf{P}_j^i\right) \oplus \mathbf{g}_{\mathbf{L}^i-1}^i \oplus \mathbf{g}_{\mathbf{L}'-1}^i = 0 \right) \right] \\
&\quad = \sum_{c \in \mathcal{C}} \Pr\left[\mathbf{C} = c \ \wedge \ \mathbf{PD} \wedge \ \exists i \left(\bigoplus_{j=1}^{\mathbf{L}'-2} \left(\mathbf{P}_j^i\right) \oplus \bigoplus_{j=1}^{\mathbf{L}^i-2} \left(\mathbf{P}_j^i\right) \oplus \mathbf{g}_{\mathbf{L}^i-1}^i \oplus \mathbf{g}_{\mathbf{L}'-1}^i = 0 \right) \right] \\
&\quad = \sum_{c \in \mathcal{C}} \Pr\left[\mathbf{C} = c \ \wedge \ \mathbf{PD} \wedge \ \exists i \left(\bigoplus_{j=1}^{\mathbf{L}'-2} \left(P_j^i(c)\right) \oplus \bigoplus_{j=1}^{\mathbf{L}^i-2} \left(P_j^i(c)\right) \oplus \mathbf{g}_{\mathbf{L}^i-1}^i \oplus \mathbf{g}_{\mathbf{L}'-1}^i = 0 \right) \right] \\
&\quad \le z \cdot 2^{-n+1} \cdot \sum_{c \in \mathcal{C}} \Pr[\mathbf{C} = c],
\end{aligned}$$

where the last inequality follows by Lemma 4, and the union bound.

Now, consider the event **E1** happening. We have that for some message i : $\mathbf{C}'_0 = \mathbf{C}_0^i + t$ with $1 \le t \le \mathbf{L}^i - \mathbf{L}'$. Note that, for $j \in [1..\mathbf{L}' - 1]$, $\sigma'_j + t \le \mathbf{L}^i - 2$. For $j \in [1..\mathbf{L}' - 1]$, from **E1**(ii) we then have

$$\mathbf{N}'_j = \mathbf{C}'_j \oplus \mathbf{g}(\mathbf{C}'_0 + \sigma'_j) = \mathbf{C}_{\sigma'_j+t}^i \oplus \mathbf{g}(\mathbf{C}_0^i + t + \sigma'_j).$$

Since $\sigma'_j + t \le \mathbf{L}^i - 2$, we get $\mathbf{N}'_j = \mathbf{N}_{\sigma'_j+t}^i$, and hence $\mathbf{M}'_j = \mathbf{M}_{\sigma'_j+t}^i$. Now, for $j \in [1..\mathbf{L}' - 2]$, since $\sigma'_j = j$, we have

$$\mathbf{M}'_j = \mathbf{P}'_j \oplus \mathbf{g}(\mathbf{C}'_0 + j) = \mathbf{P}'_j \oplus \mathbf{g}\left(\mathbf{C}_0^i + t + j\right).$$

Since $\mathbf{M}_{j+t}^i = \mathbf{P}_{j+t}^i \oplus \mathbf{g}(\mathbf{C}_0^i + t + j)$, we have $\mathbf{P}'_j = \mathbf{P}_{j+t}^i$.

Also, $\mathbf{M}'_{\mathbf{L}'-1} = \mathbf{M}_t^i$, as $\sigma'_{\mathbf{L}'-1} = 0$. Thus, $\mathbf{P}'_{\mathbf{L}'-1} \oplus \mathbf{g}(\mathbf{C}_0^i + t + \mathbf{L}' - 1) = \mathbf{M}_t^i$. Hence,

$$\begin{aligned}
&\left(\bigoplus_{j=1}^{\mathbf{L}'-1} \mathbf{P}'_j = 0 \right) \wedge \mathbf{E1} \\
&\quad \Rightarrow \left(\bigoplus_{j=1}^{\mathbf{L}'-1} \mathbf{P}'_j = 0 \right) \wedge \mathbf{E1} \ \wedge \exists i \ \left(\mathbf{P}'_{\mathbf{L}'-1} \oplus \mathbf{g}_{t+\mathbf{L}'-1}^i = \mathbf{M}_t^i \right)
\end{aligned}$$

$$\Rightarrow \exists i \left(\bigoplus_{j=1}^{\mathbf{L}'-2} (\mathbf{P}^i_{j+t}) \oplus \mathbf{M}^i_t \oplus \mathbf{g}^i_{t+\mathbf{L}'-1} = 0 \right)$$

$$\equiv \exists i \left(\bigoplus_{j=1}^{\mathbf{L}'-2} (\mathbf{P}^i_{j+t}) \oplus \mathbf{P}^i_t \oplus \mathbf{g}^i_t \oplus \mathbf{g}^i_{t+\mathbf{L}'-1} = 0 \right).$$

As $\mathbf{L}' \geq 2$, as before, using Lemma 4, we get an upper bound of $z \cdot 2^{-n+1}$. □

4.2. *Modes Using GF(p)*

We now prove Theorem 1 for the IAPM scheme as in Fig. 3 (Sect. 3.3), i.e., using the mod p construction. We first show that for each i, j, S^i_j (as defined in Sect. 3.3) is uniformly distributed in GF(p).

When it is clear from context, we will drop i from the superscript.

Lemma 9. *For every j, S_j is uniformly distributed in GF(p).*

Proof. First we prove that there is no overflow in the last step of the for-loop, i.e., while adding $(2^n - p)$.

If $S_0 < (2^n - p)$, then let t be the least j such that $S_j \geq (2^n - p)$, otherwise $t = 0$. Clearly, for $j \leq t$, the condition $(K2 > S^*_j)$ cannot be satisfied, as $K2 < p$.

We next show by induction that for $j \geq t$, $S_j \geq (2^n - p)$. Clearly, for $j = t$, it is true by the definition of t. If for some $j > t$, $(K2 \leq S^*_j)$, then $S_j = S_{j-1} + K2$, and there was no overflow in this addition, hence by induction $S_j \geq (2^n - p)$. If for some $j > t$, $(K2 > S^*_j)$, then $S^*_j < p$, as $K2 < p$. Thus, there is no overflow while adding $(2^n - p)$, and hence $S_j \geq (2^n - p)$.

Finally, we show that $S_j = K2 * (j + \mathrm{IV}) \bmod p$, which proves the lemma. Clearly, this is true for $j = 0$. Suppose that it is true for $j - 1$, then we show that $S_j = K2 * (j + \mathrm{IV}) \bmod p$. If $(S_{j-1} + K2) < 2^n$, then $S_j = S_{j-1} + K2$, and hence $S_j = K2 * (j + \mathrm{IV}) \bmod p$, by induction. If $(S_{j-1} + K2) \geq 2^n$, then $S_j = (S_{j-1} + K2) - 2^n + (2^n - p)$, since there is no overflow while adding $(2^n - p)$ as shown in the previous paragraph, and the lemma follows. □

Lemma 10. *For any constant $c \in [0..2^n - 1]$ and for every $i, j, i1, j1$ such that $j + \mathrm{IV}^i \neq j1 + \mathrm{IV}^{i1}$,*

$$\Pr_{K2 \in \mathrm{GF}p} \left[S^i_j - S^{i1}_{j1} = c \bmod p \right] \leq 1/p.$$

Proof. Since, by the proof of the previous lemma, $S^i_j = K2 * (j + \mathrm{IV}^i) \bmod p$, the lemma follows. □

In the following theorem, $\alpha(n)$ denotes the smallest t such that $2^n - t$ is a prime. For modes of practical interest, the quantity $\alpha(n)$ in the following theorem is less than $2n$. For example, for 128 bit block ciphers, if we let $p = 2^{128} - 159$, this quantity is 159.

Theorem 2. *Let A be an adversary attacking the message integrity of* IAPM ($t = 1$) *with the* GF(p) *construction* (*Fig.* 3), *with f chosen uniformly from set of permutations and $K2$ chosen uniformly from* GF(p). *Let A make at most z queries in the first stage, totaling at most μ blocks. Let $u = \mu + z$. Let v be the maximum number of blocks in the second stage. Also, assume that the initial vectors are chosen safe. If $2v < 2^n$ and $4u^2 < 2^n$, then for adversary A,*

$$\text{Succ} < 2 * \big(u^2 + z^2 + 2(u + v + z) + 1 + o(1) + (z + 1) * \alpha(n)\big) * 2^{-n}.$$

The proof is similar to that of Theorem 1, except that Lemma 10 is used in probability calculations.

5. Message Secrecy

We now prove security in the find-then-guess model, which implies that the IAPM scheme (Figs. 1 and 2) is secure for message secrecy. A similar theorem holds for the mod p version of IAPM (Fig. 3). We will again prove our theorem for the IAPM mode in the random permutation model as in Sect. 3.4.

Theorem 3. *Let A be a chosen plaintext attack adversary of the encryption scheme* IAPM($\mathcal{G}$) *making at most z queries, these totaling at most u blocks. Assume that the initial vectors are chosen to be safe. If $2u^2 < 2^n$, then*

$$\text{Adv}_A \le \frac{3 * u^2 + z(2z + u)}{2} \cdot 2^{-n}.$$

Proof. As in the proof of Theorem 1, we will use subscripts to denote particular blocks in a message. Let the z queries be divided into y queries in the find stage, one query in the "choice" stage, and y' queries in the guess stage. By assuming y' to be large enough (i.e., by repetition of queries) we can assume that the adversary makes exactly z queries. Thus, $z = y + 1 + y'$. As in Theorem 1, if IAPM chooses initial vectors randomly (uniformly and independently), then the adversary's success probability increases by at most $(2z + u) * z * 2^{-n}$.

As in Theorem 1, we consider the event PD (pairwise different) under which all the M (and respectively N) variables are different. However, there is a small difference here. The event PD in Theorem 1 was defined as a function of c and g, as there c completely determined the plaintexts for all the blocks. Here, we have two variants (corresponding to $\mathbf{b}$ being 0 or 1). Thus, we define two variants of the predicate **PD**, namely $\mathbf{PD}^0$ and $\mathbf{PD}^1$, where the predicate $\mathbf{PD}^0$ ($\mathbf{PD}^1$) uses the $(y + 1)$th plaintext block according to $\mathbf{b} = 0$ ($\mathbf{b} = 1$ respectively).

We will bound

$$\left| \sum_{b \in [0..1]} \Pr_{\mathbf{b}, (\mathbf{f}, \mathbf{g}) \in_{\mathcal{D}} \mathcal{K}} \big[A(\mathbf{C}) = \mathbf{b} \mid \mathbf{b} = b\big] - \Pr_{\mathbf{b}, (\mathbf{f}, \mathbf{g}) \in_{\mathcal{D}} \mathcal{K}} \big[A(\mathbf{C}) \ne \mathbf{b} \mid \mathbf{b} = b\big] \right|.$$

This is upper bounded by

$$\left|\sum_{b\in[0..1]} \Pr\left[A(\mathbf{C})=\mathbf{b}\ \wedge \mathbf{PD}^{\mathbf{b}} \mid \mathbf{b}=b\right] - \Pr\left[A(\mathbf{C})\neq \mathbf{b}\ \wedge \mathbf{PD}^{\mathbf{b}} \mid \mathbf{b}=b\right]\right|$$
$$+ \Pr\left[\neg\mathbf{PD}^{0}\right] + \Pr\left[\neg\mathbf{PD}^{1}\right]. \tag{19}$$

For the first term, note that for any constant binary bit t, we have

$$\begin{aligned}&\sum_{b\in[0..1]} \Pr_{\mathbf{b},(\mathbf{f},\mathbf{g})\in_{\mathcal{D}}\mathcal{K}}\left[A(\mathbf{C})=b\oplus t\ \wedge\ \mathbf{PD}^{\mathbf{b}} |\ \mathbf{b}=b\ \right]\\ &= \sum_{c^1\in\mathcal{L}()} \cdots \sum_{c^i\in\mathcal{L}(c^1,\ldots,c^{i-1})}\\ &\quad \cdots \sum_{c^z\in\mathcal{L}(c^1,\ldots,c^{z-1})} \sum_{b\in[0..1]} \Pr\left[A(c)=b\oplus t\ \wedge \mathbf{C}=c\ \wedge\ \mathrm{PD}^{\mathbf{b}}(c,\mathbf{g})|\ \mathbf{b}=b\ \right].\end{aligned}$$

We note that for any constant c, the probability of $A(c)$ being $b\oplus t$ is independent of $\mathbf{b}$, $\mathbf{f}$, and $\mathbf{g}$. Thus, we have

$$\begin{aligned}&\left|\sum_{b\in[0..1]} \Pr\left[A(\mathbf{C})=b\ \wedge \mathbf{PD}^{\mathbf{b}} \mid \mathbf{b}=b\right] - \Pr\left[A(\mathbf{C})\neq b\ \wedge \mathbf{PD}^{\mathbf{b}} \mid \mathbf{b}=b\right]\right|\\
&= \Bigg|\sum_{c\in\mathcal{C}}\sum_{b\in[0..1]} \Big(\Pr\left[A(c)=b\ \wedge\ \mathbf{C}=c\ \wedge \mathrm{PD}^{\mathbf{b}}(c,\mathbf{g}) \mid \mathbf{b}=b\right]\\
&\qquad - \Pr\left[A(c)\neq b\ \wedge\ \mathbf{C}=c\ \wedge \mathrm{PD}^{\mathbf{b}}(c,\mathbf{g}) \mid \mathbf{b}=b\right]\Big)\Bigg|\\
&= \Bigg|\sum_{c\in\mathcal{C}}\sum_{b\in[0..1]} \Big(\Pr\left[A(c)=b\right] * \Pr\left[\mathbf{C}=c\ \wedge \mathrm{PD}^{\mathbf{b}}(c,\mathbf{g}) \mid \mathbf{b}=b\right]\\
&\qquad - \Pr\left[A(c)\neq b\right] * \Pr\left[\mathbf{C}=c\ \wedge \mathrm{PD}^{\mathbf{b}}(c,\mathbf{g}) \mid \mathbf{b}=b\right]\Big)\Bigg|\\
&= \Bigg|\sum_{b\in[0..1]}\sum_{c\in\mathcal{C}} \Pr\left[A(c)=b\right] * \Big(\Pr\left[\mathbf{C}=c\ \wedge \mathrm{PD}^{\mathbf{b}}(c,\mathbf{g}) \mid \mathbf{b}=b\right]\\
&\qquad - \Pr\left[\mathbf{C}=c\ \wedge \mathrm{PD}^{\mathbf{b}}(c,\mathbf{g}) \mid \mathbf{b}=b\oplus 1\right]\Big)\Bigg|\\
&\leq \sum_{b\in[0..1]} \Bigg|\sum_{c\in\mathcal{C}} \Big(\Pr\left[\mathbf{C}=c\ \wedge \mathrm{PD}^{\mathbf{b}}(c,\mathbf{g}) \mid \mathbf{b}=b\right]\\
&\qquad - \Pr\left[\mathbf{C}=c\ \wedge \mathrm{PD}^{\mathbf{b}}(c,\mathbf{g}) \mid \mathbf{b}=b\oplus 1\right]\Big)\Bigg|\end{aligned}$$

$$= \sum_{b\in[0..1]} \left|\Pr\left[\mathbf{PD}^{b}\right] - \Pr\left[\mathbf{PD}^{b\oplus 1}\right]\right|$$

$$\leq 2 * \left|\Pr\left[\neg\mathbf{PD}^{0}\right] - \Pr\left[\neg\mathbf{PD}^{1}\right]\right|.$$

The theorem then follows from (19) and Lemma 3. □

Acknowledgements

The author is extremely grateful to Shai Halevi and Pankaj Rohatgi for help with the proof of message integrity. The author thanks J. Håstad for suggesting Lemma 1 and the current form of the event PD, which made the proofs more transparent.

The author also thanks Don Coppersmith, Nick Howgrave-Graham, J.R. Rao, Ron Rivest, Phil Rogaway, referees of Eurocrypt 2001, and reviewers of this journal for helpful suggestions. Finally, the author thanks Pau-Chen Cheng for introducing him to the problem.

References

[1] Advanced Encryption Standard, National Institute of Standards and Technology, U.S. Department of Commerce, *FIPS 197* (2001)

[2] ANSI X3.106, American national standard for information systems—data encryption algorithm—modes of operation. In *American National Standards Institute* (1983)

[3] M. Bellare, C. Namprempre, Authenticated encryption: relations among notions and analysis of the generic composition paradigm. In *Proc. Asiacrypt*. LNCS, vol. 1976 (2000)

[4] M. Bellare, C. Namprempre, Authenticated encryption: relations among notions and analysis of the generic composition paradigm. In *Proc. Asiacrypt 2000*, ed. by T. Okamoto (Springer, Berlin, 2000)

[5] M. Bellare, A. Desai, E. Jokipii, P. Rogaway, A concrete security treatment of symmetric encryption: analysis of the DES modes of operation. In *Proc. 38th IEEE FOCS* (1997)

[6] M. Bellare, J. Kilian, P. Rogaway, The security of cipher block chaining. *JCSS* **61**(3), 362–399 (2000)

[7] J. Black, S. Halevi, H. Krawczyk, T. Krovetz, P. Rogaway, UMAC: Fast and secure message authentication. In *Proc. Advances in Cryptology-CRYPTO 99*. LNCS, vol. 1666 (1999)

[8] J. Carter, M. Wegman, Universal classes of hash functions. *JCSS* **18**, 143–154 (1979)

[9] V.D. Gligor, P. Donescu, Integrity aware PCBC encryption schemes. In *Proc. 7th Intl. Work. on Security Protocols*. LNCS, vol. 1796 (Cambridge, 1999), pp. 153–171

[10] V.D. Gligor, P. Donescu, Fast encryption authentication: XCBC encryption and XECB authentication modes. http://csrc.nist.gov/encryption/modes/workshop1

[11] O. Goldreich, H. Krawczyk, M. Luby, On the existence of pseudorandom generators. In *Proc. FOCS* (1988), pp. 12–14. Also in SIAM J. Comput. **22**(6), 1163–1175

[12] S. Halevi, An observation regarding Jutla's modes of operation. http://eprint.iacr.org/2001/015/

[13] J. Håstad, Message integrity of IAPM and IACBC. http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/iapm/integrityproofs.pdf

[14] ISO/IEC 9797, Data cryptographic techniques—data integrity mechanism using a cryptographic check function employing a block cipher algorithm. In *International Organization for Standardization*, Geneva, Switzerland (1989)

[15] C.S. Jutla, Encryption modes with almost free message integrity. http://csrc.nist.gov/groups/ST/toolkit/BCM/workshops.html

[16] C.S. Jutla, Encryption modes with almost free message integrity. In *Proc. Eurocrypt 2001*. LNCS, vol. 2045 (2001)

[17] C.S. Jutla, Tight lower bound on linear authenticated encryption. In *Proc. Selected Areas in Cryptography 2003*. LNCS, vol. 3006 (2003)

[18] J. Katz, M. Yung, Unforgeable encryption and adaptively secure modes of operation. In *Proc. Fast Software Encryption*. LNCS, vol. 1978 (2000)

[19] H. Krawczyk, LFSR-based hashing and authentication. In *Proc. Crypto 94*. LNCS, vol. 839 (1994)

[20] H.W. Kuhn, Extensive games and the problem of information. In *Contributions to the Theory of Games II*, ed. by H.W. Kuhn, A.W. Tucker. Annals of Mathematical Studies, vol. 28 (Princeton Univ. Press, Princeton, 1950)

[21] M. Luby, A simple parallel algorithm for the maximal independent set problem. *SIAM J. Comput.* **15**(4), 1036–55 (1986)

[22] M. Luby, Pseudorandomness and cryptographic applications. In *Princeton Computer Science Notes* (Princeton Univ. Press, Princeton, 1996)

[23] C.H. Meyer, S.M. Matyas, *Cryptography: A New Dimension in Computer Data Security* (Wiley, New York, 1982)

[24] M. Naor, O. Reingold, On the construction of pseudo-random permutations: Luby–Rackoff revisited. In *Proc. 29th ACM STOC* (1997), pp. 189–199

[25] M. Naor, M. Yung, Universal Hash functions and their cryptographic applications. In *Proc. STOC*, (1989), pp. 33–43

[26] National Bureau of Standards, Data encryption standard, U.S. Department of Commerce, *FIPS 46* (1977)

[27] National Bureau of Standards, DES modes of operation, U.S. Department of Commerce, *FIPS 81* (1980)

[28] RFC 1510, The Kerberos network authentication service (V5), *J. Kohl and B.C. Neuman* (Sept. 1993)

[29] RFC 2401, Security architecture for the Internet protocol. http://www.ietf.org/rfc/rfc2401.txt

[30] RFC 2246, The TLS protocol. http://www.ietf.org/rfc/rfc2246.txt

[31] P. Rogaway, M. Bellare, J. Black, T. Krovetz, OCB: A block-cipher mode of operation for efficient authenticated encryption. In *Proc. 8th ACM Conf. Comp. and Comm. Security* (CCS), ACM (2001)

[32] S.G. Stubblebine, V.D. Gligor, On message integrity in cryptographic protocols. In *Proc. 1992 IEEE Comp. Soc. Symp. on Research in Security and Privacy* (1992)