Fast Encryption and Authentication: XCBC Encryption and XECB Authentication Modes

Virgil D. Gligor* Pompiliu Donescu

VDG Inc
6009 Brookside Drive
Chevy Chase, Maryland 20815
{gligor, pompiliu}@eng.umd.edu

March 30, 2001
April 20, 2001 (revision)

Abstract

We present the eXtended Ciphertext Block Chaining (XCBC) schemes or modes of encryption that can detect encrypted-message forgeries with high probability even when used with typical non-cryptographic Manipulation Detection Code (MDC) functions (e.g., bitwise exclusive-or and cyclic redundancy code (CRC) functions). These modes detect encrypted-message forgeries at low cost in performance, power, and implementation, and preserve both message secrecy and integrity in a single pass over the message data. Their performance and security scale directly with those of the underlying block cipher function. We also present the XECB message authentication modes. These modes have all the operational properties of the XOR-MAC modes (e.g., fully parallel and pipelined operation, incremental updates, and out-of-order verification), and have better performance. They are intended for use either stand-alone or with encryption modes that have similar properties (e.g., counter-based XOR encryption). However, the XECB-MAC modes have higher upper bounds on the probability of adversary's success in producing a forgery than the XOR-MAC modes.

1 Introduction

No one said this was an easy game !
Paul van Oorschot, March 1999.

A long-standing goal in the design of block encryption modes has been the ability to provide message-integrity protection with simple Manipulation Detection Code (MDC) functions, such as the exclusive-or, cyclic redundancy code (CRC), or even constant functions [5, 7, 9]. Most attempts to achieve this goal in the face of chosen-plaintext attacks focused on different variations of the Cipher Block Chaining (CBC) mode of encryption, which is the most common block-encryption mode in use. To date, most attempts, including one of our own, failed [8].

*This work was performed while this author was on sabbatical leave from the University of Maryland, Department of Electrical and Computer Engineering, College Park, Maryland 20742.

In this paper, we define the eXtended Ciphertext Block Chaining (XCBC) modes that can be used with an exclusive-or function to provide the authentication of encrypted messages in a single pass over the data. These modes detect integrity violations at a low cost in performance, power, and implementation, and can be executed in a parallel or pipelined manner. They provide authentication of encrypted messages in real-time, without the need for an additional processing path over the input data. The performance and security of these modes scale directly with the performance and security of the underlying block cipher function since separate cryptographic primitives, such as hash functions, are unnecessary.

We also present the XECB modes for message authentication (i.e., XECB-MAC modes) and their salient properties. These message authentication modes have all the operational properties of the XOR message authentication (XOR-MAC) modes (e.g., they can operate in a fully parallel and pipelined manner, and support incremental updates and out-of-order verification [2]), and have better performance; i.e., they use only about half the number of block-cipher invocations required by the XOR-MAC modes. However, the XECB-MAC modes have higher bounds on the adversary's success of producing a forgery than those of the XOR-MAC modes. The XECB modes are intended for use either stand-alone to protect the integrity of plaintext messages, or with encryption modes that have similar properties (e.g., counter-based XOR encryption [1] a.k.a "counter mode") whenever it is desired that separate keys be used for secrecy and integrity modes.

2 Integrity Modes for Encryption

Preliminaries and Notation. In defining the encryption modes we adopt the approach of Bellare *et al.* (viz., [1]), who show that an encryption mode can be viewed as the triple (E, D, KG), where E is the encryption function, D is the decryption function, and KG is the probabilistic key-generation algorithm. (Similarly, a message authentication (MAC) mode can be viewed as the triple (S, V, KG), where S is the message signing function, V is the message verification function, and KG is the probabilistic key-generation algorithm.) Our encryption (and authentication) modes are implemented with block ciphers, which are modeled with finite families of pseudorandom functions (PRFs) or pseudorandom permutations (PRPs).

In this context, we use the concepts of pseudorandom functions (PRFs), pseudorandom permutations (PRPs), and super-pseudorandom permutations (SPRPs) ([1], [14]). Let $R^{l,L}$ the set of all functions $\{0,1\}^l \rightarrow \{0,1\}^L$. We use F to denote either a family of pseudorandom functions or a family of super-pseudorandom permutations, as appropriate (e.g., for the encryption schemes, F will be a family of super-pseudorandom permutations, while for our MAC schemes, F can be a family of pseudorandom functions).

Given encryption scheme $\Pi = (E, D, KG)$ that is implemented with SPRP F, we denote the use of the key $K \stackrel{\mathcal{R}}{\leftarrow} KG$ in the encryption of a plaintext string x by $E^{F_K}(x)$, and in the decryption of ciphertext string y by $D^{F_K}(y)$. The most common method used to detect modifications of encrypted messages applies a MDC function g (e.g., a non-keyed hash, cyclic redundancy code (CRC), bitwise exclusive-or function [15]) to a plaintext message and concatenates the result with the plaintext before encryption with $E^{F_K}(x)$. A message thus encrypted can be decrypted and accepted as valid only after the integrity check is passed; i.e., after decryption with $D^{F_K}(y)$, the concatenated value of function g is removed from the plaintext, and the check passes only if this value matches that obtained by applying the MDC function to the remaining plaintext [5, 7, 15]. If the integrity check is not passed, a special failure indicator, denoted by $Null$ herein, is returned. This method[1] has been used in commercial systems such as Kerberos V5 [17, 21] and DCE

[1]Note that other methods for protecting the integrity of encrypted messages exist; e.g., encrypting the message with a secret key and then taking the separately keyed MAC of the ciphertext [15, 3]. These methods require two passes over the

[6, 21], among others. The encryption scheme obtained by using this method is denoted by Π-g = (E-g,D-g,KG), where Π is said to be *composed* with MDC function g. In this mode, we denote the use of the key K in the encryption of a plaintext string x by (E^{F_K}-g)(x), and in the decryption of ciphertext string y by (D^{F_K}-g)(y).

A design goal for Π-g = (E-g, D-g, KG) modes is to find the simplest encryption mode Π = (E,D,KG) (e.g., comparable to the CBC modes) such that, when this mode is composed with a simple, non-cryptographic MDC function g (e.g., as simple as a bitwise exclusive-or function), message encryption is protected against *existential forgeries*. For any key K, a forgery is any ciphertext message that is not the output of E^{F_K}-g. An existential forgery (EF) is a forgery that passes the integrity check of D^{F_K}-g upon decryption; i.e., for forgery y', (D^{F_K}-g)(y') $\neq Null$, where $Null$ is a failure indicator. Note that the plaintext outcome of an existential forgery need not be known to the forgerer. It is sufficient that the receiver of a forged ciphertext decrypt the forgery correctly.

Message Integrity Attack: Existential Forgery in a Chosen-Plaintext Attack. The attack is defined by a protocol between an adversary A and an oracle O[2] as follows.

1. A and O select encryption mode Π-g = (E-g,D-g,KG), and O selects, uniformly at random, a key K of KG.
2. A sends encryption queries (i.e., plaintext messages to be encrypted) x^p, $p = 1, \cdot\ \cdot, q_e$, to the encryption function of O. Oracle O responds to A by returning $y^p = (E^{F_K}\text{-}g)(x^p)$, $p = 1, \cdot\ \cdot, q_e$, where x^p are A's chosen plaintext messages. A records both its encryption queries and O's responses to them.
3. After receiving O's encryption responses, A forges a collection of ciphertexts $y'^i, 1 \leq i \leq q_v$ where $y'^i \neq y^p, \forall p = 1, \cdot\ \cdot, q_e$, and sends each decryption query y'^i to the decryption function of O. O returns a success or failure indicator to A, depending on whether of $(D^{F_K}\text{-}g)(y'^i) \neq Null$.

Adversary A is successful if at least one decryption query y'^i such that $(D^{F_K}\text{-}g)(y'^i) \neq Null$ for $1 \leq i \leq q_v$; i.e., y'^i is an existential forgery. The mode Π-g = (E-g,D-g,KG) is said to be secure in a message-integrity attack if the probability of an existential forgery in a chosen-plaintext attack is negligible. (We use the notion of negligible probability in the same sense as that of Naor and Reingold [16].)

Attack Parameters. A is allowed q_e encryption queries (i.e., queries to E^{F_K}-g), and q_v decryption queries (i.e., queries to D^{F_K}-g) totaling $\mu_e + \mu_v$ bits, and taking time $t_e + t_v$.

Parameters q_e, μ_e, t_e are bound by the parameters $(q', \mu', t', \epsilon')$ which define the chosen-plaintext security of Π = (E,D,KG) in a secrecy attack (e.g., in the left-or-right sense [1], for instance), and a constant c' determined by the speed of the function g. Since parameters $(q', \mu', t', \epsilon')$ are expressed in terms of the given parameters (t, q, ϵ) of the SPRP family F, the attack parameters can be related directly to those of the SPRP family F.

Parameters $q_e, \mu_e, t_e, q_v, \mu_v, t_v$ are also bound by the parameters (t, q, ϵ) of the SPRP family F, namely $\mu_e + \mu_v \leq ql$, and $t_e + t_v \leq t$. (The parameters q_e, q_v are determined by μ_e, μ_v.) These parameters can be set to specific values determined by the desired probability of adversary's success. Note that $q_v > 0$ since A must be allowed verification queries. Otherwise, A cannot test whether his forgeries are correct, since A does not know key K.

---

message data, require more power, and are more complex to implement than the modes we envision for most common use. Nevertheless these methods are useful whenever key separation is desired for secrecy and integrity.

[2] O can be viewed as two oracles, the first for the encryption function of O and the second for the decryption function of O.

The message-integrity attack defined above is not weaker than an adaptive one in the sense that the success probability of adversary A bounds from above the success probability of another adversary A' that intersperses the q_e encryption and q_v verification queries; i.e., the adversary is allowed to make his choice of forgery after seeing the result of legitimate encryptions and other forgeries. (This has been shown for chosen-message attacks against MAC functions [2], but the same argument holds here.) To date, this is the strongest of the known goal-attack combinations against the integrity (authentication) of encrypted messages [3, 10].

3 Definition of the XCBC and XCBC-XOR Modes

We present three XCBC modes, namely (1) stateless, (2) stateful-sender, and (3) stateful modes, and some implementation options. In general, the fewer state variables the more robust the mode is in the face of failures (or disconnections) and intrusion. This might suggest that, in practice, stateless modes are preferable. However, this may not always be the case because a good, high-performance, source of randomness that can be used for each message may be unavailable or may be hard to protect in terms of confidentiality, integrity and availability. Further, the new random number used in each message encryption by the sender must be securely transmitted to the receiver, which usually costs at least an additional block-cipher invocation. The stateful-sender mode (e.g., a counter-based mode) eliminates the need for a good source of randomness but does not always eliminate the extra block-cipher invocation and the need to protect the extra sender state variables; e.g., the source of randomness is replaced by the enciphering of a message counter but the counter must be maintained and its integrity must be protected by the sender across multiple message authentications. (The other advantage of counter-based modes, namely the ability to go beyond the "birthday barrier" when used with pseudo-random functions, does not materialize in the context of the Advanced Encryption Standard (AES) since AES is modeled as a family of pseudo-random permutations.)

Maintaining secret shared-state variables, as opposed to just sender-state, helps eliminate the extra block-cipher invocations. Extending the shared keying state with extra, per-key, random variables shared by senders and receivers is a fairly straight-forward matter; e.g., these shared variables can be generated and distributed in the same way as the shared secret key, or can be generated using the shared key (at some marginal extra cost per message) by encrypting constants with the shared key. However, maintaining the shared state in the face of failures (or disconnections), and intrusion presents an extra challenge for the mode user; e.g., enlarging the shared state beyond that of a shared secret key may increase the exposure of the mode to physical attacks. The above discussion suggests that none of the three types of operational modes is superior to the others in all environments, and hence all of them should be supported in a general mode definition.

In the encryption modes presented below, the key generation algorithm, KG, outputs a random, uniformly distributed, k-bit string or key K for the underlying $SPRP$ family F, thereby specifying $f = F_K$ and $f^{-1} = F_K^{-1}$ of l-bits to l-bits. If a separate second key is needed in a mode, then a new string or key K' is generated by KG identifying $f' = F_{K'}$ and $f'^{-1} = F_{K'}^{-1}$. The plaintext message to be encrypted is partitioned into a sequence of l-bit blocks (padding is done first, if necessary), $x = x_1 \quad x_n$. Throughout this paper, $\oplus$ is the *exclusive-or* operator and $+$ represents *modulo* 2^l *addition.*

Stateless XCBC Mode (XCBC$)
The encryption and decryption functions of the stateless mode,
$\mathcal{E}\Leftrightarrow XCBC\$^{F_K}(x)$ and $\mathcal{D}\Leftrightarrow XCBC\$^{F_K}(y)$, are defined as follows.

function $\mathcal{E}\Leftrightarrow$XCBC\$$^f(x)$	**function** $\mathcal{D}\Leftrightarrow$XCBC\$$^f(y)$
$r_0 \leftarrow \{0,1\}^l$	Parse y as $y_0 \| y_1 \quad y_n$
$y_0 = f(r_0);\ z_0 = f'(r_0)$	$r_0 = f^{-1}(y_0);\ z_0 = f'(r_0)$
for $i = 1, \quad , n$ **do** $\{$	**for** $i = 1, \quad , n$ **do** $\{$
$z_i = f(x_i \oplus z_{i-1})$	$z_i = y_i \Leftrightarrow i \times r_0$
$y_i = z_i + i \times r_0\ \}$	$x_i = f^{-1}(z_i) \oplus z_{i-1}\ \}$
return $y = y_0 \| y_1 y_2 \quad y_n$	**return** $x = x_1 x_2 \quad x_n$

Stateful-Sender XCBC Mode (XCBCC)
The encryption and decryption functions of the stateful-sender mode, $\mathcal{E}\Leftrightarrow XCBCC^{F_K}(x, ctr)$ and $\mathcal{D}\Leftrightarrow XCBCC^{F_K}(y)$, are defined as follows.

function $\mathcal{E}\Leftrightarrow$XCBCC$^f(x, ctr)$	**function** $\mathcal{D}\Leftrightarrow$XCBCC$^f(y)$
$r_0 = f(ctr);\ z_0 = f'(r_0)$	Parse y as $ctr \| y_1 \quad y_n$
for $i = 1, \quad , n$ **do** $\{$	$r_0 = f(ctr);\ z_0 = f'(r_0)$
$z_i = f(x_i \oplus z_{i-1})$	**for** $i = 1, \quad , n$ **do** $\{$
$y_i = z_i + i \times r_0\ \}$	$z_i = y_i \Leftrightarrow i \times r_0$
$ctr' \leftarrow ctr + 1$	$x_i = f^{-1}(z_i) \oplus z_{i-1}\ \}$
$y = ctr \| y_1 y_2 \quad y_n$	**return** $x = x_1 x_2 \quad x_n$
return y	

Note that in the XCBCC mode the counter ctr can be initialized to a known constant such as $\Leftrightarrow$1 by the sender. ctr' represents the updated ctr value. In both of the above modes the complexity is $n + 2$ block-cipher invocations, where n in the length of input string x in blocks.

Stateful XCBC Mode (XCBCS)
Let IV be a random and uniformly distributed variable that is part of the keying state shared by the sender and receiver.
$\mathcal{E}\Leftrightarrow XCBCS\$^{F_K}(x)$ and $\mathcal{D}\Leftrightarrow XCBCS\$^{F_K}(y)$, are defined as follows.

function $\mathcal{E}\Leftrightarrow$XCBCS\$$^f(x)$	**function** $\mathcal{D}\Leftrightarrow$XCBCS\$$^f(y)$
$r_0 \leftarrow \{0,1\}^l$	Parse y as $y_0 \| y_1 \quad y_n$
$y_0 = f(r_0);\ z_0 = IV + r_0$	$r_0 = f^{-1}(y_0);\ z_0 = IV + r_0$
for $i = 1, \quad , n$ **do** $\{$	**for** $i = 1, \quad , n$ **do** $\{$
$z_i = f(x_i \oplus z_{i-1})$	$z_i = y_i \Leftrightarrow i \times r_0$
$y_i = z_i + i \times r_0\ \}$	$x_i = f^{-1}(z_i) \oplus z_{i-1}\ \}$
return $y = y_0 \| y_1 y_2 \quad y_n$	**return** $x = x_1 x_2 \quad x_n$

Note that in the XCBCS mode the shared IV value can be generated randomly by KG and distributed to the sender and receiver along with key K thereby saving one block cipher invocation, or can be can be generated using key K by standard key-separation techniques thereby requiring an additional block encryption operation per key. In the former case, the complexity of the mode is exactly $n + 1$ block-cipher invocations and, in the latter, is *asymptotically* $n + 1$ block-cipher invocations.

Chaining Sequence. The *block chaining sequence* is that used for the traditional CBC mode, namely $z_i = f(x_i \oplus z_{i-1})$, where z_0 is the initialization vector, x_i is the plaintext and z_i is the ciphertext of

block $i, i = 1, \quad , n$. In contrast with the traditional CBC mode, the value of z_i is not revealed outside the encryption modes, and, for this reason, z_i is called a *hidden* ciphertext block. The actual ciphertext output, y_i, of the XCBC modes is defined using extra randomization, namely $y_i = z_i + i \times r_0$, where $i \times r_0$ is the *modulo* 2^l *addition* of the random, uniformly distributed, variable r_0, i times to itself; i.e., $i \times r_0 \stackrel{def}{=} \underbrace{r_0 + \quad + r_0}_{i\ times}$.

Examples for why the randomization is necessary include those which show that, without randomization, the swapping of two z_i blocks of a ciphertext message, or the insertion of two arbitrary but identical blocks into two adjacent positions of a ciphertext message, would cause the decryption of the resulting forgery with probability one whenever an bitwise exclusive-or function is used as the MDC (which is what we intend to use, since these functions are among the fastest known). Correct randomization sequences, such as $i \times r_0$, ensure that, among other things, collisions between any two z_i values is negligible regardless of whether these values are obtained during message encryption, forgery decryption, or both. Note that this probability is negligible even though the randomization sequence $i \times r_0$ allows low-order bits of some z_i's to become known. (A detailed account of why such collisions contribute to an adversary's success in breaking message integrity is provided in the proof of the XCBC$ mode; viz., Appendix A.) Examples of incorrect randomization sequences can be readily found; e.g., the sequence whereby each element i is computed as an bitwise exclusive-or of i instances of r_0.

Initialization. In *stateless* implementations of the XCBC modes $r_0 \leftarrow \{0,1\}^l$; i.e., r_0 is initialized to a random, uniformly distributed, l-bit value for every message. The value of r_0 is sent by the sender to the receiver as $y_0 = f(r_0)$. In contrast, in *stateful-sender* implementations, which avoid the use of a random number generator, a counter, *ctr*, is initialized to a new l-bit constant (e.g., -1) for every key K, and incremented on every message encryption. In *stateful* implementations, a random initialization-vector value IV that is shared by the sender and receiver is generated for every key K, and used to create a per-message random initialization vector z_0.

In all XCBC modes, the initialization vector z_0 is independent of r_0. While non-independent z_0 and r_0 values might yield secure initialization, simple relationships between these values can lead to the discovery of r_0 with non-negligible probability, and integrity can be easily broken.[3] Since we use z_0 in the definition of function $g(x)$ (see below), z_0 should also be unpredictable so that $g(x)$ has a per-message unpredictable value.

The choice of encrypting r_0 with a second key K' to obtain z_0 (i.e., $z_0 = f'(r_0)$) is made exclusively to simplify the both the secrecy [1] and the integrity proofs; e.g., such a z_0 is independent of r_0 and is unpredictable. To eliminate the use of the second key and still satisfy the requirements for z_0 suggested above, we can compute $z_0 = f(r_0 + 1)$ in stateless and stateful sender implementations, whereas in stateful implementations we compute $z_0 = IV + r_0$, where the per-message r_0 can be generated as a random value, or as an encryption of *ctr* in the XORC mode. This eliminates the additional block-cipher invocations necessary in the stateless and stateful-sender modes at the cost of maintaining an extra shared state variable (IV). This choice still satisfies the requirements for z_0.

Generalization. The above method for protecting message integrity against existential forgeries in chosen-plaintext attacks can be generalized as follows. Let the output ciphertext y_i be computed as $y_i = z_i\ op\ E_i$, where *op* is the randomization operation, E_i are the elements of the randomization sequence, and z_i the hidden ciphertext. The encryption mode Π (1) must be secure in adaptive chosen-plaintext attacks with respect to secrecy, and (2) must use the input plaintext blocks x_i to generate the input to f. The PCBC

[3] As a simple example illustrating why this is the case, let $z_0 = r_0 + 1$, choose x_1 such that $z_0 \oplus x_1 = r_0$ with non-negligible probability, and then compute $y_1 - y_0 = r_0$. With a known r_0, one can cause collisions in the values of z_i and break integrity.

[12, 15], and the "infinite garble extension" [5] modes are suitable, but counter-mode/XORC and XOR$ are not (since they fail condition (2)). Operation *op* must be invertible, so $\oplus$, modular 2^l addition and subtraction are appropriate. Elements E_i must be unpredictable such that collisions among z_i's (discussed above) could only occur with negligible probability. Other sequences can be used. For example $E_i = a^i \times r_0$ can be used, where E_i is a linear congruence sequence with multiplier a, where a can be chosen so that the sequence passes spectral tests to whatever degree of accuracy is deemed necessary. (Examples of good multipliers are readily available in the literature [11].)

XCBC-XOR Modes. To illustrate the properties of the XCBC modes in integrity attacks, we choose $g(x) = z_0 \oplus x_1 \oplus \quad \oplus x_n$ for plaintext $x = x_1 \quad x_n$, where z_0 is defined as the initialization vector of the mode. In this example, block $g(x)$ is appended to the *end* of a n-block message plaintext x, and hence block $x_{n+1} = z_0 \oplus x_1 \oplus \quad \oplus x_n$. For this choice of $g(x)$, the integrity check performed at decryption becomes $z_0 \oplus x_1 \oplus \quad \oplus x_n = f^{-1}(z_{n+1}) \oplus z_n$, where $z_{n+1} = y_{n+1} \Leftrightarrow (n+1) \times r_0$, and $z_n = y_n \Leftrightarrow n \times r_0$.

Message Padding. Standard padding methods (e.g., ASN.1), which typically require that a bit pattern and its length be added to the last block of a message to obtain an integer number of (padded) plaintext blocks, have the undesirable consequence that an additional block cipher invocation is required for the extra block of padding added for plaintexts of an integer number of blocks. Alternatives that avoid standard padding are known [4], but they require use of an extra (shared secret) key – a somewhat less desirable alternative when maintaining the unpredictability of the redundant padding information added by a mode is not an operational goal.

Known Padding Pattern. The goal of the first padding option for the XCBC modes is two-fold: (1) avoid extra block-cipher invocations, and (2) avoid the use of extra keying material. Padding with a known pattern is performed as follows: (1) use a pattern that always starts with a "1" bit followed by the minimum number of "0" bits necessary to fill the last block of plaintext [4]; (2) if the last block of a message is unpadded, use block $g'(x) = \overline{z_0} \oplus x_1 \oplus \quad \oplus x_n$ as the x_{n+1} plaintext block, where $\overline{z_0}$ is the bitwise complement of z_0; otherwise, use $g(x) = z_0 \oplus x_1 \oplus \quad \oplus x_n$.

At decryption, the integrity check performs the exclusive-or of $f^{-1}(z_{n+1}) \oplus z_n$ with $x'_1 \oplus \quad \oplus x'_n$, where $x'_1, \quad , x'_n$ are the plaintext blocks obtained at decryption, and then compares the result with the z_0 computed during decryption; if this check fails, the result is compared with $\overline{z_0}$, the complement of z_0 computed at decryption, and only if this second comparison for equality fails the ciphertext-message decryption returns failure. If the the comparison check with z_0 passes, meaning that the message was padded at encryption, the padding pattern is checked, extracted (providing some extra confidence, if found) and removed. It follows that the decryption of unpadded (but unforged) messages would fail first the first equality check but not the second. Of course, the extra check would be required only for unpadded messages and forgeries. This padding scheme satisfies our goals at a modest cost; i.e., that of including padding bits in the ciphertext and an extra check for equality.

Unpredictable Padding Pattern. The goal of the second padding option for the XCBC modes, in addition to (1) above, is to retain the unpredictability of the redundant information added by these modes to user input. This goal is set for pragmatic reasons, since these modes are secure with respect to chosen-plaintext attacks. It stems from the long-standing belief that a mode of encryption should avoid adding redundant information that provides an adversary additional conditions to verify the success of his attacks (e.g., key guessing) beyond those already available to him from knowledge of user input; e.g., in a ciphertext-only attack, the adversary who knows nothing about the plaintext would benefit from added predictable redundancy by padding and integrity checks.

In the XCBC modes, padding with an unpredictable pattern is performed as follows. Let Mask be a

random and uniformly distributed block that that is part of the keying state shared by the sender and receiver. The Mask can be generated and distributed along with the key or is can be generated by any of the available standard methods (e.g., encrypt a constant with the shared secret key to initialize Mask). For each plaintext input whose last block is incomplete, fill the last block with the known bit pattern used in the Known-Padding-Pattern option above (i.e., the pattern that always starts with a "1" bit followed by the minimum number of "0" bits necessary to fill the last block of plaintext) Perform the bitwise exclusive-or operation between the Mask and the filled last plaintext block. Use the result as the plaintext block x_n in the computation of the $x_{n+1} = g(x)$ block. Use z_0 to compute $g(x)$ for padded messages and $\overline{z_0}$ for unpadded ones as in the Known-Padding-Pattern option above. At decryption, use the same integrity check as that used in the Known-Padding-Pattern option (defined above), and if the check for padded messages passes, perform the bitwise exclusive-or of the Mask and the recovered block x'_n, and check and extract the known padding bit pattern from x'_n before returning the plaintext to the user.

The stateless and stateful encryption modes Π-g obtained by the use of schemes Π = XCBC\$, Π = XCBCC, or Π = XCBCS with function $g(x) = z_0 \oplus x_1 \oplus \quad x_n$ are denoted by XCBC\$-XOR, XCBCC-XOR, and XCBCS-XOR respectively.

Examples of Other Encryption Modes that Preserve Message Integrity.

Recently, C.S. Jutla [13] proposed an interesting scheme in which the output blocks z_i of CBC encryption are modified by (i.e., bitwise exclusive-or operations) with a sequence E_i of pairwise independent elements. In this model, $E_i = (i \times IV_1 + IV_2) mod\ p$, where IV_1, IV_2 are random values generated from an initial random value r, and p is prime, and the complexity is $n + 3$, where n is the length of the plaintext input in blocks. In contrast with C.S. Jutla's scheme, the elements of the XCBC sequence, $E_i = (i \times r_0) mod\ 2^l$, are not pairwise independent, and the complexity is $n + 2$ for the stateless and stateful-sender cases, and $n + 1$ for the stateful case. Also, the performance of the required modular 2^l additions is slightly better than that of $mod\ p$ additions, where p is prime. However, the pairwise independence of C.S. Jutla's E_i sequence should yield a slightly tighter bound on the probability of successful forgery illustrating, yet again, a fundamental tradeoff between performance and security. (The bound is tighter by a fraction of a log_2 factor depending on the value of p, which would mean that the attack complexity is within the same order of magnitude of the XCBC bound viz., Section 5).

More recently, P. Rogaway [19] has proposed other schemes that use interesting variations of non-independent and pairwise-independent elements for the E_i sequence, similar to the sequences presented in this paper and C.S. Jutla's, to achieve $n+1$ complexity. Under the same assumptions regarding stateful and stateless implementations, C.S. Jutla's modes require an extra block enciphering over the XCBC and P. Rogaway's modes. We note that all modes for authenticated encryption include an extra block cipher operation for the enciphering of the exclusive-or MDC.

Interleaved-Parallel or Pipelined Encryption. The choice of $g(x) = z_0 \oplus x_1 \oplus \quad \oplus x_n$, allows the interleaved-parallel or pipelined implementation of the XCBC modes. Other non-cryptographic MDC functions $g(x)$ would also allow such implementation, since they be executed in a parallel or a pipelined manner (by definition). This mode is useful when the number of processors available for encryption and decryption in parallel is a priori known or negotiated. For example, for interleaved-parallel execution using $g(x)$, each plaintext message x is partitioned into L segments, $x^{(1)} \quad x^{(L)}$ each of length n_s, $s = 1, \quad , L$, after customary block-level padding (n.b., this L should not be confused with the output length of a PRF, which is typically denoted by L, also). Each segment, $x^{(s)}, s = 1, \quad , L$, consists of one or more l-bit blocks, and if $g(x^{(s)}) = z_0^{(s)} \oplus x_1^{(s)} \oplus \quad \oplus x_{n_s}^{(s)}$ is used, then an additional l-bit block is included in each segment. Each segment is encrypted/decrypted in parallel on a separate processor.

In interleaved-parallel or pipelined implementations of the XCBC modes, the initialization and computation of the block chaining sequence is performed on a per-segment basis starting with a common value of r_0, which is a random, uniformly distributed, l-bit value for every message. Also, the per-message value y_0 is computed as $y_0 \leftarrow f(r_0)$ in stateless implementations. The initialization of the block chaining sequence for message segment s can be $r_0^{(s)} = r_0 + s,\ z_0^{(s)} = f'(r_0^{(s)})$, and the encryption sequence can be $z_i^{(s)} = f(x_i^{(s)} \oplus z_{i-1}^{(s)}), y_i^{(s)} = z_i^{(s)} + i \times r_0^{(s)}$. In stateful implementations ctr is updated to $ctr + L$ after the encryption of each message. (Other functions, not just addition modulo 2^l, can be used for the computation of the per-segment, block chaining sequence, and initialization sequence can be used for $r_0^{(s)}$ and $z_0^{(s)}$.)

The encrypted segments of a message are assembled to form the message ciphertext. Segment assembly encodes the number of segments L, the length of each segment n_s and, implicitly, the segment sequence in the message (e.g., all can be found in the ASN.1 encoding). If the segments of a message have different lengths, segment assembly is also synchronized with the end of each segment encryption or decryption within a message.

At decryption, the parsing of the message ciphertext yields the message length, L, segment sequence number, s, and the length of each segment, n_s. Message integrity is maintained both on a per segment and per message basis by performing the per-segment integrity check; if $g(x) = z_0 \oplus x_1 \oplus \quad \oplus x_n$, the per-segment check is $z_0^{(s)} \oplus x_1^{(s)} \oplus \quad \oplus x_{n_s}^{(s)} = f^{-1}(z_{n_s+1}^{(s)}) \oplus z_{n_s}^{(s)}$ where $z_{n_s+1}^{(s)} = y_{n_s+1}^{(s)} \Leftrightarrow (n_s + 1) \times r_0^{(s)}$ and $z_{n_s}^{(s)} = y_{n_s}^{(s)} \Leftrightarrow n_s \times r_0^{(s)}$. Failure of any per-segment integrity check, which also detects out-of-sequence segments and message-length modifications, signals a message integrity violation.

We illustrate an interleaved- parallel implementation of the stateless XCBC mode below. Stateful parallel schemes can be implemented in a similar manner, using the same methods as those illustrated for the sequential implementation.

Stateless Parallel XCBC Mode (ipXCBC$)
The encryption and decryption functions of the stateless mode,
$\mathcal{E}\Leftrightarrow ipXCBC\$^{F_K}(x)$ and $\mathcal{D}\Leftrightarrow ipXCBC\$^{F_K}(y)$, are defined as follows.

function $\mathcal{E}\Leftrightarrow$ipXCBC$\$^f(x)$
partition x into L segments $x^{(s)}$
each of length n_s;
$r_0 \leftarrow \{0,1\}^l$; $y_0 = f(r_0)$;
for segment $s, s = 1, \quad , L$, **do** {
$r_0^{(s)} = r_0 + s,\ z_0^{(s)} = f'(r_0^{(s)})$
for $i = 1, \quad , n_s$ **do** {
$z_i^{(s)} = f(x_i^{(s)} \oplus z_{i-1}^{(s)})$
$y_i^{(s)} = z_i^{(s)} + i \times r_0^{(s)}$ }
$y^{(s)} = y_1^{(s)} \quad y_{n_s}^{(s)}$ }
assemble $y = y_0 || y^{(1)} \quad y^{(L)}$;
return y.

function $\mathcal{D}\Leftrightarrow$ipXCBC$\$^f(y)$
parse y into y_0 and L segments $y^{(s)}$
each of length n_s;
$r_0 = f^{-1}(y_0)$
for segment $s, s = 1, \quad , L$ **do** {
Parse $y^{(s)}$ as $y_1^{(s)} \quad y_{n_s}^{(s)}$
$r_0^{(s)} = r_0 + s;\ z_0^{(s)} = f'(r_0^{(s)})$
for $i = 1, \quad , n_s$ **do** {
$z_i^{(s)} = y_i^{(s)} \Leftrightarrow i \times r_0^{(s)}$
$x_i^{(s)} = f^{-1}(z_i^{(s)}) \oplus z_{i-1}^{(s)}$ }
$x^{(s)} = x_1^{(s)} \quad x_{n_s}^{(s)}$ }
assemble $x = x^{(1)} \quad x^{(L)}$;
return x.

Incremental Updates of Encrypted Data. The segmentation of a message used for parallel and pipelined implementation of the XCBC modes can also be used in sequential encryption of data structures (e.g., a file, a message) whenever incremental updates of data structures are anticipated. Such segmentation enables the localization of the decryption, plaintext update, and encryption to single segments saving the

decryption and encryption of other segments unaffected by the updates. Note that message integrity is retained after such incremental updates.

Architecture-Independent Parallel Encryption. C.S. Jutla's recent parallel mode [13] requires that both the input to and output of the block cipher are randomized using a sequence of *pairwise-independent* random blocks. Our fully parallel modes achieve the same effect *without* using a sequence pairwise-independent random blocks. For these modes, it is sufficient to randomize the input and output blocks of f using the same type of sequence. In this case, the probability of input or output collisions, which would be necessary to break security and integrity respectively, would remain negligible. An example is the *stateful Extended Electronic Codebook-XOR* encryption (XECBS-XOR) mode, in which for index $i, 1 \leq i \leq n+1, n = |x|$, the ciphertext block y_i is obtained through the formulae:

$$\begin{aligned} y_i &= f(x_i + ctr \times R + i \times R^*) + ctr \times R + i \times R^*, \quad \forall i, 1 \leq i \leq n, ctr \leq q_e \\ y_{n+1} &= f(x_{n+1} + ctr \times R) + ctr \times R + (n+1) \times R^*, \end{aligned}$$

where R, R^* are two random, uniformly distributed and independent blocks each of l bits in length that are part of the keying state shared by the sender and receiver, and ctr is the counter that serves as message identifier. The counter ctr is initialized to 1 and increased by 1 on every message encryption up to q_e, which is the bound of the number of allowable message encryptions (viz., Theorem 5 below). Note that the sequence of elements $E_i = ctr \times R + i \times R^*$ can be precomputed for multiple messages, can be computed incrementally, and in an out-of-order manner.

To provide authentication, the last block is computed using the following formula for the function g:

$$x_{n+1} = g(x) = x_1 \oplus \qquad \oplus x_n.$$

This authenticated encryption mode achieves optimal performance, i.e., $n+1$ parallel block cipher invocations, and has a throughput of a single block cipher invocation. The security of the XECBS-XOR mode with respect to confidentiality in an adaptive chosen-plaintext attack can be demonstrated in the same manner as that used for the CBC mode [1].

For the XECBS-XOR encryption scheme proposed above, padding follows the similar conventions as those the XCBC-XOR modes to distinguish between padded and unpadded messages; i.e., use the following formula for the enciphering of the last block.

$$y_{n+1} = f(x_{n+1} + ctr \times Z) + ctr \times R + (n+1) \times R^*,$$

where $Z = \overline{R}$ is the bitwise complement of R and is used for unpadded messages and $Z = R$ for padded messages.

Stateless architecture-independent parallel modes and stateful-sender architecture-independent parallel modes can be specified in the same manner as those for the XCBC modes; for example, R and R^* can be derived from the l-bit random number number r_0 (e.g., $R = f(r_0 + 1)$ and $R^* = f(r_0 + 2)$), and, in the stateful-sender $r_0 = f(ctr)$, where ctr is an l-bit counter initialized to a constant such as $\Leftrightarrow$1.

In the modes thus obtained (and other related variants), there would not be any ciphertext chaining, and a priori knowledge of the number of processors would be unnecessary.

As noted earlier, the sequence $E_i = ctr \times R + i \times R^*$ does not completely hide the low order bits of x_i thereby enabling verification of key guesses by an adversary. Resistance to such attacks can be implemented in a similar manner as that of DESX [18], if deemed necessary. However, adoption of modern block ciphers with long keys should reduce the need for this.

4 Definition of the XECB Authentication Modes

In this section, we introduce new Message Authentication Modes (MACs) that counter adaptive chosen-message attacks [2]. We call these MACs the eXtended Electronic Codebook MACs, or XECB-MACs. The XECB-MAC modes have all the properties of the XOR MACs [2], but they do not waste half of the block size for recording the block identifier thereby avoid doubling the number of block cipher invocations. Many variants of XECB-MACs are possible, and here we present stateless version, XECB$-MAC, a stateful-sender version XECBC-MAC, and a stateful version, the XECBS-MAC.

Message Signing. In both the stateless and stateful-sender implementation, we generate a per-message random value y_0 that is used to randomize each plaintext block of a message x, namely $x_i, 1 \leq i \leq n, n = |x|$, before it is fed to the block cipher function f, where $f = F_K$ is selected from a PRF family F by a key K, which is random and uniform. The result of the randomization is $x_i + i \times y_0$, and the result of block enciphering with f is $y_i = f(x_i + i \times y_0)$. The stateless mode initialization requires a random number generator to create the random block r_0; i.e., $r_0 \leftarrow \{0,1\}^l$. Then $y_0 = f(r_0)$. Stateful-sender implementations avoid the use of the random number generator, and instead, uses a counter ctr, to create y_0 directly, namely $y_0 = f(ctr)$. The counter ctr is initialized by the sender on a per-key basis to a constant, such as $\Leftrightarrow 1$, and is maintained across consecutive signing requests for the same key K.

For the purposes of simplifying the proofs, we made the following choices for the generation and use of random vector z_0 in both implementations: (1) an additional per-message unpredictable block z_0 is generated and treated as an additional last block of the message plaintext before it is also randomized and enciphered by f, namely $x_{n+1} = z_0$ and $y_{n+1} = f(z_0 + (n+1) \times y_0)$; and (2) we set $z_0 = f'(r_0)$, where $f' = F_{K'}$ is a PRF selected with the second key K'. Clearly, the generation of z_0 can be performed with the same key, K, by block enciphering a simple function of r_0 (e.g., $f(r_0 + 1)$), and use of K' becomes unnecessary.

The block cipher outputs, $y_1, \quad , y_n, y_{n+1}$, are exclusive-or-ed to generate the authentication tag $w = y_1 \oplus \quad \oplus y_n \oplus y_{n+1}$. Alternative implementation options include the ones whereby the block cipher outputs, $y_1, \quad , y_n, y_{n+1}$, are added modulo $2^l \Leftrightarrow 1$, or subtracted modulo $2^l \Leftrightarrow 1$, to generate the authentication tag. The modes output the pair (r_0, w) in the stateless mode, and (ctr, w) in the stateful-sender mode.

We include the stateless version and the stateful-sender version of the XECB modes below.

Stateless XECB-MAC Mode (XECB$-MAC)

function Sign-XECB$-MAC$^f(x)$	**function** Verify-XECB$-MAC$^f(x, r_0, w)$
$r_0 \leftarrow \{0,1\}^l$	$y_0 = f(r_0), z_0 = f'(r_0)$
$y_0 = f(r_0), z_0 = f'(r_0)$	$x_{n+1} = z_0$
$x_{n+1} = z_0$	**for** $i = 1, \quad , n+1$ **do** $\{$
for $i = 1, \quad , n+1$ **do** $\{$	$y_i = f(x_i + i \times y_0)$ $\}$
$y_i = f(x_i + i \times y_0)$ $\}$	$w' = y_1 \oplus \quad \oplus y_n \oplus y_{n+1}$
$w = y_1 \oplus \quad \oplus y_n \oplus y_{n+1}$	**if** $w = w'$ **then return** 1
return (r_0, w)	**else return** 0.

Stateful-Sender XECB-MAC Mode (XECBC-MAC)

function Sign-XECBC-MAC$^f(ctr, x)$
$y_0 = f(ctr), z_0 = f'(y_0)$
$x_{n+1} = z_0$
for $i = 1, \quad , n+1$ **do** $\{$
$y_i = f(x_i + i \times y_0)\ \}$
$w = y_1 \oplus \quad \oplus y_n \oplus y_{n+1}$
$ctr' \leftarrow ctr + 1$
return (ctr, w)

function Verify-XECBC-MAC$^f(x, ctr, w)$
$y_0 = f(ctr), z_0 = f'(y_0)$
$x_{n+1} = z_0$
for $i = 1, \quad , n+1$ **do** $\{$
$y_i = f(x_i + i \times y_0)\ \}$
$w' = y_1 \oplus \quad \oplus y_n \oplus y_{n+1}$
if $w = w'$ **then return** 1
else return 0.

Note that ctr' represents the updated ctr value.

The following stateful variant of the XECB modes (whose proof is presented in Appendix C) comes close to the optimal performance of any parallel MAC, namely n parallel block-cipher invocations and throughput equivalent of a single block-cipher invocation.

Stateful XECB-MAC Mode (XECBS-MAC)
Let R, R^* be two random, uniformly distributed and independent blocks that are part of the keying state shared by the sender and receiver.

function Sign-XECBS-MAC$^f(ctr, x)$
for $i = 1, \quad , n$ **do** $\{$
$y_i = f(x_i + ctr \times R + i \times R^*)\ \}$
$w = y_1 \oplus \quad \oplus y_n$
$ctr' \leftarrow ctr + 1$
return (ctr, w)

function Verify-XECBS-MAC$^f(x, ctr, w)$
if $ctr > q_s$ **then return** 0
for $i = 1, \quad , n$ **do** $\{$
$y_i = f(x_i + ctr \times R + i \times R^*)\ \}$
$w' = y_1 \oplus \quad \oplus y_n$
if $w = w'$ **then return** 1
else return 0.

Note that ctr is initialized to 1, and ctr' represents the updated ctr value.

Message Tag Verification. For verification, an adversary submits a forgery $x = x_1 \quad x_n$ and a forged pair (r_0, w) or (ctr, w) depending upon the mode.[4] Message x is then signed and an authentication tag $w' = y_1 \oplus \quad \oplus y_n \oplus y_{n+1}$ is generated. The algorithm outputs a bit that is either 1, if the forged authentication tag is correct, namely $w = w'$, or 0, otherwise.

Message Padding. For the stateless and stateful-sender XECB-MAC schemes, padding follows the same conventions as those the XCBC-XOR modes to distinguish between padded and unpadded messages; i.e., the authentication tag generation and verification use $\overline{z_0}$ for unpadded messages and z_0 for padded messages. For the stateful XECB-MAC scheme, padding follows the similar conventions as those the XCBC-XOR modes to distinguish between padded and unpadded messages; i.e., the authentication tag generation and verification use $\overline{R}$ for unpadded messages and R for padded messages. For all schemes, the padding pattern is the typical one; i.e., the pattern that always starts with a "1" bit followed by the minimum number of "0" bits necessary to fill the last block of plaintext.

Properties of the XECB Authentication Modes

1. Security. The XECB authentication modes are intended to be secure in adaptive chosen-message attacks [2], and Theorems 3 and 4 below show the security bounds for the stateful-sender mode. The

[4] The forgery (x, r_0, w) or (x, ctr, w) are not a previously signed queries. Note also that the length n of the forged message need not be equal to the length of any signed message.

XECB modes, as well as all the other modes that use similar types of randomization sequences, have higher, but still negligible, upper bounds on the adversary's success in producing a forgery than those of the XOR-MAC modes.

2. Concurrent Block-Cipher Invocations and Mode Throughput. The goal of the XECB-MAC modes is to allow the block-cipher (e.g., AES) computations on different blocks to be made in a *fully parallel* or *pipelined* manner; i.e., to exploit any degree of parallelism or pipelining available at the sender or receiver without apriori knowledge of the number of processors available.

We note that despite the fact that the throughput of a mode depends on the number of block cipher invocations, and hence on the availability of enough parallel processing units, throughput also depends on how a mode uses those units. For example, the number of block-cipher invocations in the stateless and stateful-sender XECB modes can be reduced from $n+3$ to $n+2$ simply by eliminating the enciphering of block x_{n+1}; e.g., the enciphering of the last plaintext block (i.e., n-th block) can be changed to $y_n = f(x_n \oplus z_0 + n \times y_0)$ (without affecting the proofs significantly). Nevertheless, the throughput of these modes is close to that of *two sequential* block cipher invocations, since the enciphering of y_0 precedes the parallel enciphering of the input plaintext blocks. In contast, in the stateful XECB mode, the number of block-cipher invocations is n, just as in the case of the PMAC [20] which is also a stateful mode. However, the throughput of the XECB modes is close to that of *a single* block-cipher invocation, as opposed to that of PMAC, which corresponds to that of two sequential block-cipher invocations since the tag is computed after $n \Leftrightarrow 1$ block cipher invocations regardless of the number of processors available. The performance goal of n block-cipher parallel invocations and a throughput equivalent of a single block-cipher invocation appears to be a achievable with stateful MAC modes.

3. Incremental Updates. The XECB-MAC modes are incremental with respect to block replacement; e.g., a block x_i of a long message is replaced with a new value x'_i. For instance, let us consider the stateful-sender mode. Let the two messages have the same counter ctr; hence, the authentication tag of the new message, w', is obtained from the authentication tag of the previous message, w, by the following formula: $w' = w \oplus f(x_i + i \times y_0) \oplus f(x'_i + i \times y_0)$. The replacement property can be easily extended to insertion and deletion of blocks, and to the modes that use modular $2^l \Leftrightarrow 1$ addition or subtraction in the place of the exclusive-or of the block cipher outputs.

4. Out-of-order Verification. The verification of the authentication tag can proceed even if the blocks of the message arrive out of order as long as each block is accompanied by its index and the first block has been retrieved.

5 Security Considerations

In this section, we provide evidence for the security of the XCBC modes against both adaptive chosen-plaintext and message-integrity attacks. We also present the security of the XECB modes in adaptive chosen-message attacks.

We first address the security (i.e., secrecy) of the XCBC\$ mode against adaptive chosen-plaintext attacks. The theorems and proofs that demonstrate that the stateful mode (XCBC) and the two-key variations are secure in a left-or-right sense [1] are similar to that for the XCBC\$ mode and, therefore, will be omitted.

The Lemma and Theorem below, which establish the security (i.e., secrecy) of the XCBC\$ mode are restatements of Lemma 16 and Theorem 17 respectively, which are presented for the CBC mode in the full version of the Bellare *et al.* paper ([1]). The proof of the Lemma and Theorem are similar to those for the

CBC mode and hence are omitted.

Lemma 1 [Upper bound on the security of the XCBC\$ mode in random function model] Let $XCBC\$^R$ be the implementation of the XCBC\$ mode with the family of random functions $R(l,l)$. Let A be any adversary attacking $XCBC\$^R$ in the left-or-right sense, making at most q' queries, totaling at most μ' bits. Then, the adversary's advantage is

$$Adv_A^{lr} \leq \delta_{XCBC\$} \stackrel{def}{=} \left(\frac{\mu'^2}{l^2} \Leftrightarrow \frac{\mu'}{l}\right)\frac{1}{2^l}.$$

The following theorem defines the security of the XCBC\$ mode against an adaptive chosen-plaintext attack when the XCBC\$ mode is implemented with a (q,t,ϵ)-pseudorandom function family F. F is (q,t,ϵ)-pseudorandom, or (q,t,ϵ)-secure, if an adversary (1) spends time t to evaluate $f = F_K$ at q input points via adaptively chosen queries, and (2) has a negligible advantage bounded by ϵ over simple guessing in distinguishing the output of f from that of a function chosen at random from R.

Theorem 1 [Security of XCBC\$ in Adaptive Chosen-Plaintext Attacks] Suppose F is a (t,q,ϵ)-secure PRF family with block length l. There is a constant $c > 0$ such that for any number of queries q_e totaling μ' bits of memory and taking time t', the $XCBC\$(F)$ is (t',q',μ',ϵ')-secure in the left-or-right sense, for $\mu' = q'l$, $t' = t \Leftrightarrow c\mu'$, and $\epsilon' = 2\epsilon + \delta_{XCBC\$}$ where $\delta_{XCBC\$} \stackrel{def}{=} \left(\frac{\mu'^2}{l^2} \Leftrightarrow \frac{\mu'}{l}\right)\frac{1}{2^l}$.

The XCBC\$ and XCBC modes can easily be analyzed assuming F is a SPRP family (not a PRF family), since AES is an intended block cipher for these modes. Hence only needs to apply the results of Proposition 8 of Bellare *et al.* [1] to the result of Theorem 1. A similar lemma and theorem hold for chosen-plaintext attacks in a real-or-random sense, as defined by Bellare *et al.* [1].

In establishing the security of the XCBC\$ mode against the message-integrity attack, let the parameters used in the attack be bound as follows: $q_e \leq q'$, since the XCBC\$ mode is also chosen-plaintext secure, $t_e + t_v \leq t$, and $\mu'' = \mu_e + \mu_v \leq ql$. Let the forgery verification parameters q_v, μ_v, t_v be chosen within the constraints of these bounds and to obtain the desired $Pr_{f \stackrel{\mathcal{R}}{\leftarrow} F}[Succ]$.

Theorem 2 [Security of XCBC\$-XOR in a Message-Integrity Attack] Suppose F is a (t,q,ϵ)-secure SPRP family with block length l. The mode XCBC\$-XOR is secure against a message-integrity attack consisting of $q_e + q_v$ queries, totaling $\mu_e + \mu_v \leq ql$ bits, and taking at most $t_e + t_v \leq t$ time; i.e., the probability of adversary's success is

$$Pr_{f \stackrel{\mathcal{R}}{\leftarrow} F}[\text{Succ}] \quad \leq \quad \epsilon + \frac{\mu_v(\mu_v \Leftrightarrow l)}{l^2 2^{l+1}} + \frac{q_e(q_e \Leftrightarrow 1)}{2^{l+1}} + \frac{(q_e+1)\mu_v}{l2^l} + \frac{\mu_v}{l2^{l+1}}(\log_2 \frac{\mu_v}{l} + 3) + \frac{q_v\mu_e}{l2^l}(\log_2 \frac{\mu_e}{l} + 3).$$

(The proof of Theorem 2 can be found in Appendix A.) Note that parameters q_e, μ_e, t_e can be easily stated in terms of secrecy parameters $(t', q', \mu', \epsilon')$ above by introducing a constant c' defining the speed of the XOR function.

Theorem 2 above allows us to estimate the complexity of a message-integrity attack.[5] In a successful attack, $Pr_{f \stackrel{\mathcal{R}}{\leftarrow} F}[\text{Succ}] \in (\text{negligible}, 1]$. To estimate complexity, we set the probability of success when $f \stackrel{\mathcal{R}}{\leftarrow} P^l$ to

[5] Technically, the complexity of a successful integrity attack, and the bound of Theorem 2, should account for the success of a secrecy attack; i.e., the secrecy bound shown of Lemma 1 above (adjusted for the use of PRPs) should be added to the bound in Theorem 2. This is the case because, in general, in modes using the same key for both secrecy and integrity, a successful secrecy attack can break integrity and, vice-versa, a successful integrity attack can break secrecy. (This can be shown using the secrecy and integrity properties of the *IGE* mode; viz., *http://csrc.nist.gov/encryption/modes/proposedmodes.*) As suggested below, the addition of the secrecy bound would not affect the complexity of a successful integrity attack.

the customary 1/2, and assume that the attack parameters used in the above bound, namely $\frac{\mu_e}{l}$, $\frac{\mu_v}{l}$, are of the same order or magnitude, namely $2^{\alpha l}$, where $0 < \alpha < 1$. Also, since the shortest message has at least three blocks, $q_e, q_v \leq \lfloor \frac{2^{\alpha l}}{3} \rfloor$.

In this case, by setting

$$\frac{q_e(q_e \Leftrightarrow 1)}{2^{l+1}} + \frac{\mu_v(\mu_v \Leftrightarrow l)}{l^2 2^{l+1}} + \frac{(q_e+1)\mu_v}{l2^l} + \frac{\mu_v}{l2^{l+1}}(\log_2 \frac{\mu_v}{l} + 3) + \frac{q_v \mu_e}{l2^l}(\log_2 \frac{\mu_e}{l} + 3) = 1/2,$$

we obtain (by ignoring the $\lfloor . \rfloor$ function) the equation $2^{2\alpha l \frac{6\alpha l + 34}{9}} + 2^{\alpha l \frac{3\alpha l + 11}{3}} = 2^l$, which allows us to estimate α for different values of l. (In this estimate, we can ignore the term in $2^{\alpha l}$ since it is insignificant compared to the other term of the sum.) For example, for $l = 64, \alpha \approx \frac{29}{64}$, for $l = 128, \alpha \approx \frac{61}{128}$, and for $l = 256, \alpha \approx \frac{124}{256}$. Hence, this attack is very close to a square-root attack (i.e., $\alpha \to \frac{1}{2}$ as l increases), and remains this way even is the secrecy bound of Lemma 1 (adjusted for PRPs) is added to the integrity bound. Thus the security payoff of improved bounds is limited when using families of SPRPs.

A variant of Theorem 2 can be proved for the stateful modes. Furthermore, similar theorems hold for single-key stateless modes. The statement and proof for such theorems are similar to the statement and proof for the integrity theorem for the stateless mode, and hence, are omitted.

The XECB-MAC modes are intended to be secure against adaptive chosen-message attacks [2] consisting of up to q_s signature queries totaling at most μ_s bits and using time up to t_s, and q_v verification queries totaling at most μ_v bits and using time at most t_v. The security of the XECBC-MAC mode, when implemented with a PRF family, is established by the following theorem. (The restatement of this theorem in terms of a family of PRPs, such as AES, and the corresponding proof modifications are pretty much standard.)

Theorem 3 [Security of XECBC-MAC in an Adaptive Chosen-Message Attack]
Suppose F is a (t, q, ϵ)-secure PRF family with block length l. The message authentication mode (Sign-XECBC^f, Verify-XECBC^f, KG) is secure against adaptive chosen-message (q_s, q_v) attacks consisting of $q_s + q_v$ queries totaling $\mu_s + \mu_v \leq ql$ bits and taking at most $t_s + t_v \leq t$ time; i.e., the probability of adversary's success is

$$Pr_{f \overset{\mathcal{R}}{\leftarrow} F}[\text{Succ}] \leq \epsilon + \frac{\mu_v}{l2^l}(\log_2 \frac{\mu_v}{l} + 3) + \frac{q_s \mu_v}{l2^l} + \left(q_s + 2q_v + \frac{\mu_s}{2l}\right) \frac{\mu_s}{l2^{l+1}}(\log_2 \frac{\mu_s}{l} + 3).$$

The proof of this theorem is similar to that of Theorem 2 and is presented in Appendix B.

We also present a theorem for the security of the XECBS-MAC mode. (The restatement of this theorem in terms of a family of PRPs, such as AES, and the corresponding proof modifications are pretty much standard.)

Theorem 4 [Security of XECBS-MAC in an Adaptive Chosen-Message Attack]
Suppose F is a (t, q, ϵ)-secure PRF family with block length l. The message authentication mode (Sign-XECBC^f, Verify-XECBC^f, KG) is secure against adaptive chosen-message (q_s, q_v) attacks consisting of $q_s + q_v$ queries $(q_v \leq q_s)$ totaling $\mu_s + \mu_v \leq ql$ bits and taking at most $t_s + t_v \leq t$ time; i.e., the probability of adversary's success is

$$\begin{aligned} Pr_{f \overset{\mathcal{R}}{\leftarrow} F}[\text{Succ}] \quad \leq \quad & \epsilon + \frac{q_v}{2^l} + \frac{\mu_v}{l2^{l+1}}(\log_2 \frac{\mu_v}{l} + 3) + \left(q_v + \frac{\mu_s}{l}\right) \frac{q_s}{2^{l+1}}(\log_2 q_s + 3) + \\ & \left(q_v + \frac{\mu_s}{l}\right) \frac{\mu_s}{l2^{l+1}}(\log_2 \frac{\mu_s}{l} + 3). \end{aligned}$$

The proof of this theorem is similar to that of Theorems 2 and 3 and is presented in Appendix C.

A similar theorem can be provided for the stateless message authentication mode. The complexity of an attack against XECB-MAC modes can be determined in a similar manner as that of an attack against the XCBC$-XOR mode.

The security of the XECBS-XOR mode in a message-integrity attack is shown by the theorem bellow.

Theorem 5 [Security of XECBS-XOR in a Message-Integrity Attack]
Suppose F is a (t, q, ϵ)-secure SPRP family with block length l. The mode XECBS-XOR is secure against a message-integrity attack consisting of $q_e + q_v$ queries ($q_v \leq q_e$), totaling $\mu_e + \mu_v \leq ql$ bits, and taking at most $t_e + t_v \leq t$ time; i.e., the probability of adversary's success is

$$\begin{aligned} Pr_{f \stackrel{\mathcal{R}}{\leftarrow} F}[\text{Succ}] \quad \leq \quad & \epsilon + \frac{\mu_v(\mu_v \Leftrightarrow l)}{l^2 2^{l+1}} + \frac{q_v}{2^l} + \frac{\mu_v}{l2^{l+1}}(\log_2 \frac{\mu_v}{l} + 3) + \\ & q_v + \frac{\mu_e}{l} \quad \frac{q_e}{2^{l+1}}(\log_2 q_e + 3) + \quad q_v + \frac{\mu_e}{l} \quad \frac{\mu_e}{l2^{l+1}}(\log_2 \frac{\mu_e}{l} + 3) + \frac{\mu_e(\mu_e \Leftrightarrow l)}{l^2 2^{l+1}}. \end{aligned}$$

(The proof of Theorem 5 can be found in Appendix D). Note that maximum allowable values for q_s and q_e in Theorems 4 and 5 can be determined by setting the probability of successful forgery to a desired value.

Acknowledgments

We thank David Wagner for pointing out an oversight in an earlier version of Theorem 2, Tal Malkin for her thoughtful comments and suggestions, Omer Horvitz and Radostina Koleva for their careful reading of earlier versions of this paper.

References

[1] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway, "A Concrete Security Treatment of Symmetric Encryption," Proceedings of the 38th Symposium on Foundations of Computer Science, IEEE, 1997, (394-403). A full version of this paper is available at *http://www-cse.ucsd.edu/users/mihir*.

[2] M. Bellare, R. Guerin, and P.Rogaway, "XOR MACs: New methods for message authentication using finite pseudo-random functions", Advances in Cryptology- CRYPTO '95 (LNCS 963), 15-28, 1995. (Also U.S. Patent No. 5,757,913, May 1998, and U.S. Patent No. 5,673,318, Sept. 1997.)

[3] J. Black and P. Rogaway, "CBC MACs for Arbitrary-Length Messages: The Three-key Constructions," Advances in Cryptology - CRYPTO '00 Springer Verlag (LNCS 1880), pp. 197-215, Aug. 2000.

[4] M. Bellare and C. Namprempre, "Authenticated Encryption: Relations among notions and analysis of the generic composition paradigm," manuscript, May 26, 2000. *http://eprint.iacr.org/2000.025.ps*.

[5] C.M. Campbell, "Design and Specification of Cryptographic Capabilities," in *Computer Security and the Data Encryption Standard*, (D.K. Brandstad (ed.)) National Bureau of Standards Special Publications 500-27, U.S. Department of Commerce, February 1978, pp. 54-66.

[6] Open Software Foundation, "OSF - Distributed Computing Environment (DCE), Remote Procedure Call Mechanisms," Code Snapshot 3, Release, 1.0, March 17, 1991.

[7] V.D. Gligor and B. G. Lindsay,"Object Migration and Authentication," *IEEE-Transactions on Software Engineering*, SE-5 Vol. 6, November 1979. (Also IBM-Research Report RJ 2298 (3l04), August 1978.)

[8] V.D. Gligor, and P. Donescu, "Integrity-Aware PCBC Schemes," in Proc. of the 7th Int'l Workshop on *Security Protocols*, (B. Christianson, B.Crispo, and M. Roe (eds.)), Cambridge, U.K., LNCS 1796, April 2000.

[9] R.R. Juneman, S.M. Mathias, and C.H. Meyer, "Message Authentication with Manipulation Detection Codes," Proc. of the IEEE Symp. on Security and Privacy, Oakland, CA., April 1983, pp. 33-54.

[10] J. Katz and M. Yung, "Complete characterization of security notions for probabilistic private-key encryption," Proc. of the 32nd Annual Symp. on the Theory of Computing, ACM 2000.

[11] D.E. Knuth, "The Art of Computer Programming - Volume 2: Seminumerical Algorithms," Addison-Wesley, 1981 (second edition), Chapter 3.

[12] J. T. Kohl, "The use of encryption in Kerberos for network authentication", *Advances in Cryptology-CRYPTO '89* (LNCS 435), 35-43, 1990.

[13] C.S. Jutla, "Encryption Modes with Almost Free Message Integrity," IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, manuscript, August 1, 2000. *http://eprint.iacr.org/2000/039.*

[14] M Luby and C. Rackoff, "How to construct pseudorandom permutations from pseudorandom functions", *SIAM J. Computing*, Vol. 17, No. 2, April 1988.

[15] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, 1997.

[16] M. Naor and O. Reingold, "From Unpredictability to Indistinguishability: A Simple Construction of Pseudo-Random Functions from MACs," *Advances in Cryptology - CRYPTO '98* (LNCS 1462), 267-282, 1998.

[17] RFC 1510, "The Kerberos network authentication service (V5)", Internet Request for Comments 1510, J. Kohl and B.C. Neuman, September 1993.

[18] P. Rogaway, "The Security of DESX," RSA Laboratories *Cryptobytes*, Vol. 2, No. 2, Summer 1996.

[19] P. Rogaway, "OCB Mode: Parallelizable Authenticated Encryption", Preliminary Draft, October 16, 2000, available at *http://csrc.nist.gov/encryption/aes/modes/rogaway-ocb1.pdf.*

[20] P. Rogaway, "PMAC: A Parallelizable Message Authentication Mode," Preliminary Draft, October 16, 2000, available at *http://csrc.nist.gov/encryption/aes/modes/rogaway-pmac1.pdf.*

[21] S. G. Stubblebine and V. D. Gligor, "On message integrity in cryptographic protocols", Proceedings of the 1992 IEEE Computer Society Symposium on Research in Security and Privacy, 85-104, 1992.

Appendix A - Proof [Security of the XCBC$-$XOR$ in a Message-Integrity Attack]

Notation: Throughout this proof, the superscripts of variables x^p, z^p, y^p, and r_0^p denote the plaintext, hidden ciphertext, ciphertext, and initial random value of a queried message $p, 1 \leq p \leq q_e$, whereas the (primed) variables x'^i, z'^i, y'^i, and $r_0'^i$ denote the plaintext, hidden ciphertext, ciphertext, and the initial random value of the i-th forged (i.e., unqueried) message, $1 \leq i \leq q_v$. The length of the plaintext of

message p is denoted by $n_p = |x^p|$ and that of forgery y'^i by $n'_i = |x'^i|$ blocks. (These lengths do not include the last plaintext block that holds the value of the XOR function.)

To find an upper bound on the probability of an adversary's success we (1) define four types of events on which we condition the adversary's success, (2) express the upper bound in terms of the conditional probabilities obtained, and (3) compute upper bounds on these probabilities. Our choice and number of conditioning events is motivated exclusively by the need to obtain a (good) upper bound for the probability of the adversary's success. Undoubtedly, other events could be used for deriving alternate upper bounds.

To provide some intuition for the choice of conditioning events defined, we give examples of events that cause an adversary's success. (The reader can skip these examples without loss of continuity.)

Examples of Adversary's Success. A way for the adversary to find a forgery y' that passes the integrity check $g(x') = x'_{n+1}$, is to look for collisions in the input of f^{-1}, namely collisions of the (1) hidden ciphertext blocks generated during the decryption of a forgery, $z'_s, 1 \leq s \leq n+1$, and (2) initialization block y'_0 (i.e., block 0 of the forged ciphertext). These blocks could collide either with blocks $y^p_0, z^p_k, 1 \leq p \leq q_e, 1 \leq k \leq n_i + 1$ obtained at encryption or among themselves. The following four examples illustrate why such collisions cause an adversary's success. Other such examples, and other ways to find forgeries, exist.

Example 1 – Collisions between blocks z'_s and z^p_k

Suppose that all hidden ciphertext blocks z'_s obtained during the decryption of forgery y' collide with some hidden ciphertext blocks z^p_k obtained at encryption. If this event occurs during forgery decryption, we declare pessimistically that the adversary is successful. Why is the adversary successful? Among the forgeries that make this event true, some will decrypt correctly with probability one. For example, if any two of the hidden ciphertext blocks between position 1 and n_p of a queried message p are swapped, the decryption of the resulting hidden ciphertext will pass the integrity check $g(x') = x'_{n+1}$ with probability one (viz., [15], Example 9.89, pp. 367-368, for a similar example). Thus, any forgery that generates such hidden ciphertext at decryption will pass this integrity check with probability one.

Why is our criterion for adversary's success based on such a collision event pessimistic? Among the forgeries that make this event true, some will decrypt correctly with negligible probability. These forgeries include truncations of the ciphertext of already queried messages.[6] For truncations, the integrity check cannot pass with probability greater than $1/2^l$ (and for this reason we can focus on other types of forgeries for the rest of this proof).

Example 2 – Collisions among the z'_s blocks

Suppose that two hidden ciphertext blocks z'_s and z'_t obtained during forgery decryption do not collide with any hidden ciphertext blocks obtained during encryption, but collide with each other. If this event occurs during forgery decryption, we declare pessimistically that the adversary is successful. Why is the adversary successful? Among the forgeries that make event true, some will decrypt correctly with probability one. For

[6] Let the forged ciphertext y' be a truncation of ciphertext y^p obtained at encryption; i.e., $y'_s = y^p_s, \forall s, 0 \leq s \leq n'+1, |y'| = n'+1$ and $n' < n_p$, i.e., $n'+1 \leq n_p$. The condition $n'+1 \leq n_p$ (due to truncation) implies that all the plaintext blocks $x^p_1, \cdot\ \cdot, x^p_{n'+1}$ are constants. In this case, $z'_s = z^p_s, \forall s, 0 \leq s \leq n'+1$ and thus $x'_s = x^p_s, \forall s, 0 \leq s \leq n'+1$. The integrity check, $z^p_0 \oplus x^p_1 \oplus \cdot\ \cdot \oplus x^p_{n'} \oplus x^p_{n'+1} = 0$, is the exclusive-or of a random and uniformly distributed variable $z'_0 = f'(r'_0) = f'(r^p_0) = z^p_0$, where $f' \overset{\mathcal{R}}{\leftarrow} R^{l,l}$, and constant plaintexts $x^p_1, \cdot\ \cdot, x^p_{n'+1}$. Hence, $Pr[z^p_0 \oplus x^p_1 \oplus \cdot\ \cdot \oplus x^p_{n'} \oplus x^p_{n'+1} = 0] = \frac{1}{2^l}$.

example, if any two identical blocks never seen among the hidden ciphertext blocks obtained at encryption are inserted into two adjacent positions between 1 and n_p of the hidden ciphertext of message p (i.e., $z'_s = z'_{s+1}, 1 \leq s < n_p \Leftrightarrow 1$), the decryption of the resulting hidden ciphertext will pass the integrity check $g(x') = x'_{n+1}$ with probability one (viz., [15], Example 9.89, pp. 367-368, for a similar example). Thus, any forgery that generates such hidden ciphertext blocks at decryption will pass this integrity check with probability one.

Why is our criterion for adversary's success based such a collision event pessimistic? Among the forgeries that make this event true, some will decrypt correctly with negligible probability. For example, consider forgeries that cause an odd number of identical hidden ciphertext blocks to be generated during decryption. Suppose these blocks have the following properties: (1) they do not collide with any hidden blocks obtained at encryption, (2) they do not collide with any initialization blocks $y_0^i, 1 \leq i \leq q_e$, obtained at encryption, (3) they do not collide with the initialization block y'_0 of the forgery, and (4) they appear between positions 1 and $n_p + 1$ of the hidden ciphertext of queried message p obtained at encryption. Forgeries that produce such blocks during decryption cannot pass the integrity check with probability greater than $1/2^l$. This is the case because the decryption of these identical hidden blocks produces random, uniformly distributed plaintext blocks that are independent of any other plaintext blocks in $g(x') = x'_{n+1}$ and can only cancel each other out in pairs under the exclusive-or operation.

The next two examples refer to collision events of the initialization block y'_0. These can lead to forgeries that satisfy the conditions of the events defined in Examples 1 and 2 above, and hence such collisions contribute to an adversary's success.

Example 3 Collisions between blocks y'_0 and z^p_{k+1}

Suppose that, during the decryption of forgery y', block y'_0 collides with some hidden ciphertext block obtained during encryption. Let $y'_0 = z^p_{k+1}, 1 \leq p \leq q_e, 1 \leq k \leq n_p$. This means that the lower order bits of $r'_0 = f^{-1}(y'_0) = x^p_{k+1} \oplus z^p_k$ can be predicted (at least) to the same extent as those of z^p_k, since x^p_{k+1} is chosen. In (pessimistic) case the entire r'_0 is predicted, the adversary's forgeries can satisfy the collision events of Examples 1 and 2 above.

Example 4 Collisions between blocks y^i_0 and y^p_0

Suppose that an adversary finds a collision between the initialization blocks of two ciphertext messages i and p obtained at encryption, namely y^i_0 and y^p, and chooses the initialization block of the forgery y' to be $y'_0 = y^i_0$. If the adversary can find such a collision event at encryption, the adversary can also find forgeries that satisfy the collision events of Example 1 at decryption. For example, the adversary can create a ciphertext message that has not been seen before (i.e., a forgery) by mixing the blocks of two ciphertext messages obtained at encryption whose initial ciphertext blocks collide; e.g., ciphertext block y^i_k of messages i replaces ciphertext $y^p_k \neq y^i_k$ of message p, where $y'_0 = y^i_0 = y^p_0, i \neq p, n_i \leq n_p, 1 \leq i, p \leq q_e$, $1 \leq k \leq n_i$.

Conditioning Events. To compute an upper bound on the probability of successful forgery, we choose four conditioning events based on collisions in the input of f^{-1}. Intuition for the choice of events is provided by Examples 1 4 above.

For each verification query (or forgery) y'^i, $1 \leq i \leq q_v$, we define two types of collision events, C_i and D_i, that refer to the hidden ciphertext blocks z'^i_s obtained during forgery decryption.

Event C_i includes all the instances when the hidden blocks $z_s'^i$ of forgery y'^i collide either with initialization blocks y_0^p or with some hidden ciphertext blocks z_k^p generated during encryption, where $1 \leq p \leq q_e, 1 \leq k \leq n_p + 1$. To define event C_i formally, let S be the the union of all the y_0^p blocks and all the hidden ciphertext blocks z_k^p produced at encryption:

$$S = \{y_0^p, 1 \leq p \leq q_e\} \cup \{z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_p + 1\}.$$

Also let Z_i be the collection of hidden ciphertext blocks z'^i generated during the decryption of the arbitrary forgery $y'^i, 1 \leq i \leq q_v$, that do not collide with blocks of S:

$$Z_i = \{z_s'^i, 1 \leq s \leq n_i' + 1, z_s'^i \notin S\}.$$

Hence, event C_i (*Collision*) is defined by:

$$C_i : Z_i = \emptyset;$$

i.e., Z_i is empty; or, equivalently, $C_i : Z_i \subseteq S$.

The second type of collision event defined for the arbitrary forgery $y'^i, 1 \leq i \leq q_v$, refers to collisions among blocks $y_0'^i, z_s'^i, 1 \leq s \leq n_i' + 1$ where $z_s'^i \in Z_i$, and is denoted by $\overline{D_i}$ (*not distinct*) below. This event is defined in terms of its complementary event D_i (*distinct*), which states that there is at least a hidden block $z_s'^i \in Z_i$ that does not collide with any other hidden block $z_t'^i \in Z_i$ or with $y_0'^i$.[7] It is clear that this definition makes sense only when $Z_i \neq \emptyset$. Formally, if $Z_i \neq \emptyset$,

$$D_i : \exists z_s'^i \in Z_i, 1 \leq s \leq n_i' + 1 : z_s'^i \neq z_t'^i, \forall z_t'^i \in Z_i, t \neq s, 1 \leq t \leq n_i' + 1 \text{ and } z_s'^i \neq y_0'^i.$$

The third type of collision event for the arbitrary forgery $y'^i, 1 \leq i \leq q_v$, which is denoted by I_i below, includes all the instances when the initialization block $y_0'^i$ collides with some hidden ciphertext blocks generated during encryption (i.e., $z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_i + 1$). Formally, event I_i is defined by:

$$I_i : y_0'^i \in S \Leftrightarrow \{y_0^p, 1 \leq p \leq q_e\},$$

or, equivalently,

$$I_i : y_0'^i \in \{z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_p + 1\},$$

The fourth type of collision event, denoted by E below, defines collisions among the initialization blocks (i.e., block 0 of the ciphertext) generated at encryption. (Hence, this collision event is independent of the forgery y'^i.) Formally, this event is defined as

$$E : y_0^i = y_0^p,$$

where $i \neq p, 1 \leq i, p \leq q_e$.

Note 0: Events denoting collisions in the inputs to f during encryption, such as those used in the proofs of Lemma 1 and Theorem 1, can also allow an adversary to produce a successful forgery. For example, collisions in the input to f during the encryption of a message $p, 1 \leq p \leq q_e$, cause hidden ciphertext blocks generated during encryption to collide, thereby leading to the discovery of $r_0^p, 1 \leq p \leq q_e$. This would break both integrity and secrecy. To account for these events, we could condition on them (in a similar manner as that used for event $\overline{E}$ below) and add the bound provided by Lemma 1 (adjusted for the use of $f \overset{\mathcal{R}}{\leftarrow} P^l$) to the final bound. Technically, this would enable us to assume that an adversary could

[7] Recall that hidden ciphertext blocks $z_s'^i, z_t'^i \in Z_i$ do not collide with any z_k^p or with any y_0^p obtained during encryption, where $1 \leq p \leq q_e, 1 \leq k \leq n_p + 1$.

not discover $r_0^p, 1 \leq p \leq q_e$, and that r_0^p are random, uniformly distributed and independent of each other. For the sake of brevity, we make this assumption below without actually conditioning on collision events in the input to f at encryption (for the reasons discussed in the estimation of the complexity of a successful integrity attack following the statement of Theorem 2).

Note 1: Other events than the four defined above could cause an adversary's forgery y'^i to pass the integrity check $g(x'^i) = x'^i_{n_i+1}$. However, Claim 1 below makes it clear that the success of such a forgery could only occur with probability no greater than $1/2^l$.

Note 2: Another collision event in the input of f^{-1}, $y'^i_0 = y^p_0, 1 \leq i \leq q_v, 1 \leq p \leq q_e$, can be caused simply by the adversary's choice of the initial forgery block. Unlike the four events defined above (and illustrated by Examples 1 4), the occurrence of this collision event cannot cause an adversary's success in the absence of other collision events. Nevertheless, the occurrence of this event is accounted for in the proof; viz., Proof of Claim 3 below.

Upper bound on the Probability of Successful Forgery. Let F be a SPRP family, P^l be the set of all permutations on $\{0,1\}^l$, and $f \overset{\mathcal{R}}{\leftarrow} P^l$ denote the random selection of f and f^{-1} from P^l. Let $S_{f \overset{\mathcal{R}}{\leftarrow} P^l}$ represent all the ciphertext blocks produced at the encryption of the q_e queries (viz., the definition of S used for collision events above) when the XCBC\$-XOR scheme is implemented with $f \overset{\mathcal{R}}{\leftarrow} P^l$; i.e.,

$$S_{f \overset{\mathcal{R}}{\leftarrow} P^l} = \{f(r_0^p), 1 \leq p \leq q_e\} \cup \{f(x_k^p \oplus z_{k-1}^p), 1 \leq p \leq q_e, 1 \leq k \leq n_p + 1\}.$$

For any $f \overset{\mathcal{R}}{\leftarrow} P^l$ and $S_{f \overset{\mathcal{R}}{\leftarrow} P^l}$, we define the finite family of random functions $G_S : \{0,1\}^k \times \{0,1\}^l \to \{0,1\}^l$ whose members are $f, \overline{f}$, with $\overline{f}$ defined as:

$$\overline{f} = \begin{cases} f^{-1}(t), & t \in S_{f \overset{\mathcal{R}}{\leftarrow} P^l} \\ v(t), & t \in \{0,1\}^l \Leftrightarrow S_{f \overset{\mathcal{R}}{\leftarrow} P^l}, v \overset{\mathcal{R}}{\leftarrow} R^{l,l} \end{cases},$$

where $R^{l,l}$ is the set of all functions from $\{0,1\}^l$ to $\{0,1\}^l$. We denote by $f \overset{\mathcal{R}}{\leftarrow} G_S$ the random selection of f and $\overline{f}$ from G_S.

The family of functions G_S behaves exactly like P^l when the plaintext blocks input to f and ciphertext blocks input to f^{-1} are those generated during the encryption of any adversary's q_e chosen-plaintext queries, and behaves exactly like $R^{l,l}$ during the decryption of any ciphertext block *not* in $S_{f \overset{\mathcal{R}}{\leftarrow} P^l}$.

Note that the family G_S is well-defined for any message-integrity attack because, by definition (viz., Section 2), in any such attack, all q_e encryption queries precede all q_v forgery verification queries. Thus $S_{f \overset{\mathcal{R}}{\leftarrow} P^l}$ and $\overline{f}$ are completely determined before any of the q_v forgery verification queries are possible, whose processing would require block decryption with $\overline{f}$. (Also note that we allow $q_e = 0$ and, in this case, $S_{f \overset{\mathcal{R}}{\leftarrow} P^l} = \emptyset$ and $\overline{f} = v$.)

For the balance of this proof, we use the result of Fact 1 below (whose proof can be found at the end of this appendix) that provides the reduction from $f \overset{\mathcal{R}}{\leftarrow} F$ to $f \overset{\mathcal{R}}{\leftarrow} G_S$.

Fact 1
(a)

$$Pr_{f \overset{\mathcal{R}}{\leftarrow} F}[\text{Succ}] \leq \epsilon + Pr_{f \overset{\mathcal{R}}{\leftarrow} P^l}[\text{Succ}].$$

(b)

$$Pr_{f \stackrel{\mathcal{R}}{\leftarrow} P^l}[\text{Succ}] \leq Pr_{f \stackrel{\mathcal{R}}{\leftarrow} G_S}[\text{Succ}] + \frac{\mu_v(\mu_v \Leftrightarrow l)}{l^2 2^{l+1}}.$$

Fact 1 reduces the problem to finding an upper bound for $Pr_{f \stackrel{\mathcal{R}}{\leftarrow} G_S}[\text{Succ}]$. Unless we state otherwise, assume that $f \stackrel{\mathcal{R}}{\leftarrow} G_S$ (and drop this subscript from $Pr_{f \stackrel{\mathcal{R}}{\leftarrow} G_S}[\text{Succ}]$.)

To compute an upper bound for the probability of successful forgery, $Pr[\text{Succ}]$, we condition on event E first, since this event does not depend on the forgery y'^i. Using standard conditioning, we obtain

$$Pr[\text{Succ}] \leq Pr[E] + Pr[\text{Succ} \mid \overline{E}].$$

Since event E is equivalent to the event that at least a collision happens when q_e balls are thrown at random in 2^l buckets [2],

$$Pr[E] \leq \frac{q_e(q_e \Leftrightarrow 1)}{2^{l+1}}.$$

To find an upper bound for $Pr[\text{Succ} \mid \overline{E}]$, we use the definition of adversary's success (viz., the attack definition), which states that at least one forgery (and verification query) y'^i succeeds; i.e., there exists an index $i, 1 \leq i \leq q_v$ such that $g(x'^i) = x'^i_{n'_i+1}$. Hence, by union bound,

$$Pr[\text{Succ} \mid \overline{E}] \leq \sum_{i=1}^{q_v} Pr[g(x'^i) = x'^i_{n'_i+1} \mid \overline{E}].$$

To find an upper bound for the probability of decrypting a single, arbitrary (non-truncation) forgery y'^i correctly given $\overline{E}$, namely for $Pr[g(x'^i) = x'^i_{n'_i+1} \mid \overline{E}]$, we condition on event (C_i or $\overline{D_i}$). Using the total probability formula we obtain:

$$\begin{aligned} Pr[g(x'^i) = x'^i_{n'_i+1} \mid \overline{E}] &= Pr[g(x'^i) = x'^i_{n'_i+1} \mid \overline{E} \text{ and } (C_i \text{ or } \overline{D_i})]Pr[C_i \text{ or } \overline{D_i} \mid \overline{E}] + \\ &\quad Pr[g(x'^i) = x'^i_{n'_i+1} \mid \overline{E} \text{ and } (\overline{C_i} \text{ and } D_i)]Pr[\overline{C_i} \text{ and } D_i \mid \overline{E}]. \end{aligned}$$

Hence,[8]

$$Pr[g(x'^i) = x'^i_{n'_i+1} \mid \overline{E}] \leq Pr[C_i \text{ or } \overline{D_i} \mid \overline{E}] + Pr[g(x'^i) = x'^i_{n'_i+1} \mid \overline{E} \text{ and } \overline{C_i} \text{ and } D_i].$$

However, both event C_i and event $\overline{D_i}$ depend on the event I_i (viz., Example 3 above). Hence, to compute $Pr[C_i \text{ or } \overline{D_i} \mid \overline{E}]$ we condition on event I_i and, using the total probability formula, we obtain:

$$\begin{aligned} Pr[C_i \text{ or } \overline{D_i} \mid \overline{E}] &= Pr[C_i \text{ or } \overline{D_i} \mid \overline{E} \text{ and } I_i]Pr[I_i \mid \overline{E}] + Pr[C_i \text{ or } \overline{D_i} \mid \overline{E} \text{ and } \overline{I_i}]Pr[\overline{I_i} \mid \overline{E}] \\ &\leq Pr[I_i \mid \overline{E}] + Pr[C_i \text{ or } \overline{D_i} \mid \overline{E} \text{ and } \overline{I_i}]. \end{aligned}$$

Furthermore,

$$\begin{aligned} Pr[C_i \text{ or } \overline{D_i} \mid \overline{E} \text{ and } \overline{I_i}] &= Pr[C_i \text{ or } \overline{D_i} \mid \overline{C_i} \text{ and } \overline{E} \text{ and } \overline{I_i}]Pr[\overline{C_i} \mid \overline{E} \text{ and } \overline{I_i}] \\ &\quad + Pr[C_i \text{ or } \overline{D_i} \mid C_i \text{ and } \overline{E} \text{ and } \overline{I_i}]Pr[C_i \mid \overline{E} \text{ and } \overline{I_i}] \\ &\leq Pr[C_i \text{ or } \overline{D_i} \mid \overline{C_i} \text{ and } \overline{E} \text{ and } \overline{I_i}] + Pr[C_i \mid \overline{E} \text{ and } \overline{I_i}] \\ &= Pr[C_i \mid \overline{E} \text{ and } \overline{I_i}] + Pr[\overline{D_i} \mid \overline{C_i} \text{ and } \overline{E} \text{ and } \overline{I_i}], \end{aligned}$$

[8] This also follows from our pessimistic assumption that if event (C_i or $\overline{D_i}$) is true, then the adversary has broken integrity.

since event $[C_i \text{ or } \overline{D_i} \mid \overline{C_i} \text{ and } \overline{E} \text{ and } \overline{I_i}]$ is equivalent to event $[\overline{D_i} \mid \overline{C_i} \text{ and } \overline{E} \text{ and } \overline{I_i}]$.

Combining the results of the last three inequalities, we obtain:

$$\begin{aligned} Pr[g(x'^i) = x'^i_{n'_i+1} \mid \overline{E}] \quad \leq \quad & Pr[g(x'^i) = x'^i_{n'_i+1} \mid \overline{E} \text{ and } \overline{C_i} \text{ and } D_i] + \\ & Pr[I_i \mid \overline{E}] + Pr[C_i | \overline{E} \text{ and } \overline{I_i}] + Pr[\overline{D_i} \mid \overline{C_i} \text{ and } \overline{E} \text{ and } \overline{I_i}]. \end{aligned}$$

The probabilities that appear at the right side of this inequality are bounded as shown in the following four claims whose proofs are included below. (Note again that forgeries based on truncations of ciphertext messages obtained at encryption are *not* included in any of the claims below. All these claims refer to a single, arbitrary (non-truncation) forgery $y'^i, 1 \leq i \leq q_v$.)

Claim 1

$$Pr[g(x'^i) = x'^i_{n'_i+1} \mid \overline{E} \text{ and } \overline{C_i} \text{ and } D_i] \leq \frac{1}{2^l}.$$

Claim 2

$$Pr[I_i \mid \overline{E}] \leq \frac{1}{2^l} \frac{\mu_e}{2l} \quad \log_2 \frac{\mu_e}{l} + 3 \quad .$$

Claim 3

$$Pr[C_i | \overline{E} \text{ and } \overline{I_i}] \leq \frac{(n'_i + 1) q_e}{2^l} + \frac{1}{2^l} \frac{\mu_e}{2l} \quad \log_2 \frac{\mu_e}{l} + 3 \quad .$$

Claim 4

$$Pr[\overline{D_i} \mid \overline{C_i} \text{ and } \overline{E} \text{ and } \overline{I_i}] \leq \frac{n'_i}{2^{l+1}} (\log_2 n'_i + 3) + \frac{n'_i + 1}{2^l}.$$

Note that if the maximum length m of the encrypted messages is known, the $\log_2 \frac{\mu_e}{l}$ term of Claims 2 and 3 can be replaced with $\log_2 m$.

Further in this proof as well as in the proofs of Claims 2 4, we use the following three facts, whose proofs can be found at the end of this appendix.

Fact 2
For any $1 \leq i \leq 2^l \Leftrightarrow 1$, let m be defined by $i = d \times 2^m$, where d is odd. If r_0 is random and uniformly distributed, then for any constant a,

$$Pr[i \times r_0 = a] \leq \frac{1}{2^{l-m}}.$$

Fact 3
For any $N > 1$, let m be defined by $a = d \times 2^m$, where $1 \leq a \leq N \Leftrightarrow 1$ and d is odd. Then

$$\sum_{a=1}^{N-1} 2^m \leq \frac{N \Leftrightarrow 1}{2} (\log_2 (N \Leftrightarrow 1) + 3).$$

Fact 4
If for any $p, 1 \leq p \leq q_e, n_p > 0$, and if $\sum_{p=1}^{q_e} (n_p + 1) \leq \frac{\mu_e}{l}$, then,

$$\sum_{p=1}^{q_e} (n_p + 1) \log_2 (n_p + 1) \leq \frac{\mu_e}{l} \log_2 \frac{\mu_e}{l};$$

and, further, if $m = \max(n_p + 1)$, then

$$\sum_{p=1}^{q_e} (n_p + 1) \log_2 (n_p + 1) \leq \frac{\mu_e}{l} \log_2 m.$$

Note that a similar relation is obtained if the summation is done for the verification queries, i.e.,

$$\sum_{i=1}^{q_v} (n'_i + 1) \log_2 (n'_i + 1) \leq \frac{\mu_v}{l} \log_2 \frac{\mu_v}{l};$$

and, further, if $m' = \max(n'_i + 1)$, then

$$\sum_{i=1}^{q_v} (n'_i + 1) \log_2 (n'_i + 1) \leq \frac{\mu_v}{l} \log_2 m'.$$

By Claims 1–4, the probability of success given $\overline{E}$ for a single, arbitrary (non-truncation) forgery is

$$\begin{aligned} Pr[g(x'^i) = x'^i_{n'_i+1} \mid \overline{E}] &\leq \frac{1}{2^l} + \frac{(n'_i + 1)q_e}{2^l} + \frac{1}{2^l} \quad \frac{\mu_e}{l} \log_2 \frac{\mu_e}{l} + \frac{3\mu_e}{l} \quad + \frac{n'_i}{2^{l+1}} (\log_2 n'_i + 3) + \frac{n'_i + 1}{2^l} \\ &= \frac{(n'_i + 1)(q_e + 1)}{2^l} + \frac{n'_i}{2^{l+1}} (\log_2 n'_i + 3) + \frac{1}{2^l} \quad \frac{\mu_e}{l} \log_2 \frac{\mu_e}{l} + \frac{3\mu_e}{l} \quad . \end{aligned}$$

Hence, the probability of adversary's success when he has up to q_v verification queries totaling at most μ_v bits and using up to t_v time is bounded by

$$\begin{aligned} Pr[\text{Succ}] &\leq Pr[E] + \sum_{i=1}^{q_v} Pr[g(x'^i) = x'^i_{n'_i+1} \mid \overline{E}] \\ &\leq \frac{q_e(q_e \Leftrightarrow 1)}{2^{l+1}} + \sum_{i=1}^{q_v} \quad \frac{(n'_i + 1)(q_e + 1)}{2^l} + \frac{n'_i}{2^{l+1}} (\log_2 n'_i + 3) + \frac{1}{2^l} \quad \frac{\mu_e}{l} \log_2 \frac{\mu_e}{l} + \frac{3\mu_e}{l} \\ &\leq \frac{q_e(q_e \Leftrightarrow 1)}{2^{l+1}} + \frac{\mu_v(q_e + 1)}{l 2^l} + \frac{\mu_v}{l 2^{l+1}} (\log_2 \frac{\mu_v}{l} + 3) + \frac{q_v}{2^l} \quad \frac{\mu_e}{l} \log_2 \frac{\mu_e}{l} + \frac{3\mu_e}{l} \end{aligned}$$

because $\sum_{i=1}^{q_v} (n'_i + 1) \leq \frac{\mu_v}{l}$ and

$$\sum_{i=1}^{q_v} \frac{n'_i}{2^{l+1}} (\log_2 n'_i + 3) \sum_{i=1}^{q_v} \frac{n'_i + 1}{2^{l+1}} (\log_2 (n'_i + 1) + 3) \leq \frac{\mu_v}{l 2^{l+1}} (\log_2 \frac{\mu_v}{l} + 3)$$

by Fact 4.

Furthermore, by using Fact 1, the probability of adversary's success when $f \overset{\mathcal{R}}{\leftarrow} F$ is bounded by:

$$Pr_{f \overset{\mathcal{R}}{\leftarrow} F}[\text{Succ}] \leq \epsilon + \frac{\mu_v(\mu_v \Leftrightarrow l)}{l^2 2^{l+1}} + \frac{q_e(q_e \Leftrightarrow 1)}{2^{l+1}} + \frac{\mu_v(q_e + 1)}{l 2^l} + \frac{\mu_v}{l 2^{l+1}} (\log_2 \frac{\mu_v}{l} + 3) + \frac{q_v}{2^l} \quad \frac{\mu_e}{l} \log_2 \frac{\mu_e}{l} + \frac{3\mu_e}{l} \quad .$$

Also, if the maximum length m of the encrypted messages is known, the last term of the above bounds can be replaced with $\frac{q_v}{2^l} \quad \frac{\mu_e}{l} \log_2 m + \frac{3\mu_e}{l}$, and if the maximum length m' of the decryption queries is known, the next to the last term of the above bounds can be replaced with $\frac{\mu_v}{l 2^{l+1}} (\log_2 m' + 3)$.

The parameters of the attack are bounded as follows: $q_e \leq q'$, since the scheme is also supposed to be chosen-plaintext secure, $t_e + t_v \leq t$, and $\mu'' = \mu_e + \mu_v \leq ql$. The forgery verification parameters q_v, μ_v, t_v can be chosen within the constraints of these bounds and the desired $Pr_{f \stackrel{\mathcal{R}}{\leftarrow} F}[Succ]$. □

Proofs of Claims 1–4

Notation: Recall that Claims 1 4 above refer to a *single*, arbitrary (non-truncation) forgery $y'^i, 1 \leq i \leq q_v$. Hence, to simplify notation in the proof of these claims, we drop the forgery index i from the events D_i, C_i, I_i, and simply use D, C, I for these events. We also drop the forgery index i from the collection Z_i and use Z instead. Furthermore, we drop the prime and forgery index i from the ciphertext y'^i, hidden ciphertext, z'^i, plaintext x'^i, $r_0'^i$, and the length n_i'. Hence, when we refer to the (single) forgery, we use the variables y, for forgery ciphertext, x for forgery plaintext, z for the hidden blocks of forgery y, y_0 for the initialization block of forgery y (and r_0 for the decryption of the initialization block y_0), and n for the length of x. Superscripts continue to identify encryption queries. In the proof of Claims 1 4, we use the notation $Pr_A[\ .\ ] = Pr[\ .\ |A]$, where A is an arbitrary event.

Proof of Claim 1

If $\overline{C}$ is true, then Z is not empty. For any $z_s \in Z$,

$$x_s = \overline{f}(z_s) \oplus z_{s-1}$$

Since z_s does not collide with any hidden blocks obtained at encryption, and event ($\overline{C}$ and D) is true (i.e., there is at least one hidden block $z_s \in Z$ by event $\overline{C}$ that does not collide with another hidden ciphertext block $z_t \in Z, s \neq t$ or with y_0 by event D), then $\overline{f}(z_s) = v(z_s)$ is uniformly distributed and independent of anything else (since $v \stackrel{\mathcal{R}}{\leftarrow} R^{l,l}$); i.e., independent of any other $\overline{f}(z_k), z_k \in Z$, $k \neq s$, and independent of any $z_k, 0 \leq k \leq n+1$. Hence, the corresponding plaintext block x_s is uniformly distributed and independent of anything else. Thus,

$$g(x) \oplus x_{n+1} = z_0 \oplus x_1 \oplus \qquad \oplus x_n \oplus x_{n+1}$$

is random and uniformly distributed, and hence:

$$Pr[g(x) \oplus x_{n+1} = 0 \mid \overline{E} \text{ and } \overline{C} \text{ and } D] = Pr[g(x) = x_{n+1} \mid \overline{E} \text{ and } \overline{C} \text{ and } D] \leq \frac{1}{2^l}.$$

□

Proof of Claim 2

Event $I : y_0 \in S \Leftrightarrow \{y_0^p, 1 \leq p \leq q_e\} = \{z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_p + 1\}$ is equivalent to the union of all possible events $y_0 = z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_p + 1$. Hence, by union bound,

$$Pr[I \mid \overline{E}] \leq \sum_{p=1}^{q_e} \sum_{k=1}^{n_p+1} Pr[y_0 = z_k^p \mid \overline{E}].$$

We determine an upper bound for $Pr[y_0 = z_k^p \mid \overline{E}]$ based on

$$y_0 = z_k^p \ \Leftrightarrow\ y_0 = y_k^p \Leftrightarrow k \times r_0^p \ \Leftrightarrow\ k \times r_0^p = y_k^p \Leftrightarrow y_0.$$

In this expression, r_0^p is random and uniformly distributed, and from the definition of event E, if $\overline{E}$ is true, then r_0^p is random and uniformly distributed. Hence, since $y_k^p \Leftrightarrow y_0$ is a known constant, by Fact 2,

$$Pr[y_0 = z_k^p \mid \overline{E}] = Pr[k \times r_0^p = y_k^p \Leftrightarrow y_0 \mid \overline{E}] \leq \frac{1}{2^{l-m}},$$

where the exponent m is defined by $k = d \times 2^m$ and d is odd. Hence, for each $p, 1 \leq p \leq q_e$, from this and Fact 3 with $N \Leftrightarrow 1 = n_p + 1$ and $a = k$,

$$\sum_{k=1}^{n_p+1} Pr[y_0 = z_k^p \mid \overline{E}] \leq \frac{1}{2^l} \sum_{k=1}^{n_p+1} 2^m \leq \frac{1}{2^l} \frac{n_p+1}{2} (\log_2(n_p+1) + 3).$$

Since $\sum_{p=1}^{q_e}(n_p+1) \leq \frac{\mu_e}{l}$ by the definition of $n+p$ and of the attack, we obtain

$$Pr[I \mid \overline{E}] \quad \leq \quad \sum_{p=1}^{q_e} \sum_{k=1}^{n_p+1} Pr[y_0 = z_k^p \mid \overline{E}] \leq \frac{1}{2^l} \sum_{p=1}^{q_e} \frac{n_p+1}{2} (\log_2(n_p+1) + 3) \leq \frac{1}{2^l} \frac{\mu_e}{2l} \quad \log_2 \frac{\mu_e}{l} + 3 \quad ,$$

by Fact 4. Further, if $m = \max(n_p + 1)$, then $Pr[I \mid \overline{E}] \leq \frac{1}{2^l} \frac{\mu_e}{2l} (\log_2 m + 3)$, also by Fact 4. □

Proof of Claim 3

Below we use the notation that $Pr_A[\ .\ ] = Pr[\ .\ |A]$, where A is an arbitrary event.

C is equivalent to the event that every hidden ciphertext block obtained during decryption is found among the hidden ciphertext blocks obtained during encryption or among the y_0^p blocks obtained at encryption. This implies that *for any* $s, 1 \leq s \leq n+1$: $Pr_{\overline{I} \text{ and } \overline{E}}[C] \leq Pr_{\overline{I} \text{ and } \overline{E}}[z_s \in S]$ by union bound. Since, $S = \{y_0^p, 1 \leq p \leq q_e\} \cup \{z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_p + 1\}$, it follows that, by union bound,

$$\begin{aligned} Pr_{\overline{I} \text{ and } \overline{E}}[z_s \in S] \quad &\leq \quad Pr_{\overline{I} \text{ and } \overline{E}}[z_s \in \{y_0^p, 1 \leq p \leq q_e\}] \\ &+ \quad Pr_{\overline{I} \text{ and } \overline{E}}[z_s \in \{z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_p + 1\}]. \end{aligned}$$

For the first term, for any $s, 1 \leq s \leq n+1$, the event $z_s \in \{y_0^p, 1 \leq p \leq q_e\}$ is the union of all collision events $z_s = y_0^p, 1 \leq p \leq q_e$. Hence,

$$Pr_{\overline{I} \text{ and } \overline{E}}[z_s \in \{y_0^p, 1 \leq p \leq q_e\}] \leq \sum_{p=1}^{q_e} Pr_{\overline{I} \text{ and } \overline{E}}[z_s = y_0^p].$$

But $z_s = y_s \Leftrightarrow s \times r_0$ by the scheme definition, and hence $s \times r_0 = y_s \Leftrightarrow y_0^p$. To compute $Pr_{\overline{I} \text{ and } \overline{E}}[s \times r_0 = y_s \Leftrightarrow y_0^p]$, we use the following claim, whose proof can be found at the end of this appendix:

Claim 3.1
Let $y_0^p y_1^p \quad y_{n_p+1}^p$ be a queried message, and $y = y_0 y_1 \quad y_{n+1}$ be a forged ciphertext. If event $\overline{I}$ is true, then r_0 is random and uniformly distributed. Furthermore, if $y_0 \neq y_0^p$, then r_0 is also independent of r_0^p.

Since event $\overline{I}$ is true, it follows that r_0 is random and uniformly distributed (by Claim 3.1 above). Also, event $\overline{I}$ and $\overline{E}$ implies that r_0 is random and uniformly distributed by the definition of event E. Hence, by Fact 2,

$$Pr_{\overline{I} \text{ and } \overline{E}}[s \times r_0 = y_s \Leftrightarrow y_0^p] \leq \frac{1}{2^{l-m}},$$

where m is defined by $s = d \times 2^m$ and d is odd. Furthermore, $m \leq \log_2 s \leq \log_2(n+1)$, since $s \leq n+1$. Hence, $2^m \leq n+1$, and

$$Pr_{\overline{I} \text{ and } \overline{E}}[s \times r_0 = y_s \Leftrightarrow y_0^p] \leq \frac{n+1}{2^l}.$$

Hence, for any $s, 1 \leq s \leq n+1$:

$$Pr_{\overline{I} \text{ and } \overline{E}}[z_s \in \{y_0^p, 1 \leq p \leq q_e\}] \leq \sum_{p=1}^{q_e} \frac{n+1}{2^l} = \frac{(n+1)q_e}{2^l}.$$

To compute an upper bound for the second term, namely on $Pr_{\overline{I} \text{ and } \overline{E}}[z_s \in \{z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_p+1\}]$, we are free to choose a hidden ciphertext block at index j of forgery y, namely z_j, and then we only need to show that $Pr_{\overline{I} \text{ and } \overline{E}}[z_j \in \{z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_p+1\}]$, is bounded. (This is the case because the bound must be true *for any* $s, 1 \leq s \leq n+1$.)

Thus, the balance of the proof of Claim 3 consists of two parts. In the first part, we partition the space of forgeries that are not truncations into three complementary types and choose a z_j (and hence, index j) for each type. In the second part, we find an upper bound for the probability $Pr_{\overline{I} \text{ and } \overline{E}}[z_j \in \{z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_p+1\}]$ for each of the chosen z_j's. Hence, the maximum of these three upper bounds represents the upper bound for $Pr_{\overline{I} \text{ and } \overline{E}}[z_j \in \{z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_p+1\}]$ for all forgeries that are not truncations.

Part 1. Finding index j depends on the type of forgery. A forgery can be such that a ciphertext obtained at encryption is the prefix of the forgery; we call this the *prefix* case. The complementary case for the prefix case, which we call *non-prefix*, includes two separate subcases, namely when y_0 is different from any y_0^i of any ciphertext obtained at encryption, or when there is an index i such that $y_0 = y_0^i$. Hence, in the latter case, there must be at least a block in the forged ciphertext y that is different from the corresponding block of the ciphertext of a queried message i, namely y^i. Further, the length of the forged ciphertext y, denoted by n, may be different from the length of the message plaintext defined by n_i.

This partition of forgery types shows that a forged ciphertext $y = y_0 y_1 \quad y_{n+1}$, which is not a truncation, can be in one of the following three complementary types:
(a) $\exists i, 1 \leq i \leq q_e : n > n_i, \forall k, 0 \leq k \leq n_i + 1 : y_k = y_k^i$; i.e., the forged ciphertext is an extension of the ciphertext y^i (the prefix case). The non-prefix case consists of the following two forgery types:
(b1) $y_0 \neq y_0^i, \forall i, 1 \leq i \leq q_e$; i.e., the forged ciphertext and all queried-message ciphertexts differ in the first block.
(b2) $\exists i, 1 \leq i \leq q_e : y_0 = y_0^i, \exists k, 1 \leq k \leq \min(n_i+1, n+1) : y_k \neq y_k^i$; i.e., the forged ciphertext is obtained by modifying a queried message ciphertext starting with some block between the second and last block of that queried-message ciphertext. In this case, let j be the smallest index such that $y_j \neq y_j^i$ (i.e., $\forall k, 0 \leq k \leq j \Leftrightarrow 1 : y_k = y_k^i$).

Let us choose index j (and hence z_j) as follows. For forgeries of type (a), $j = n_i + 2$ (or $j > n_i + 1$); for forgeries of type (b1), $j = 1$; and for forgeries of type (b2), j is the smallest index such that $y_j \neq y_j^i, 1 \leq j \leq \min\{n_i+1, n+1\}$. In all cases $j \geq 1$, and hence, the chosen ciphertext block z_j is well defined.

Part 2. For the index j chosen in Part 1, we find an upper bound for $Pr_{\overline{I} \text{ and } \overline{E}}[z_j \in \{z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_p+1\}]$. Event $z_j \in \{z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_p+1\}$ is the union of all possible events

$z_j = z_k^p, 1 \le p \le q_e, 1 \le k \le n_p + 1$. Hence, union bound leads to:

$$Pr_{\overline{I} \text{ and } \overline{E}}[z_j \in \{z_k^p, 1 \le p \le q_e, 1 \le k \le n_p + 1\}] \le \sum_{p=1}^{q_e} \sum_{k=1}^{n_p+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p].$$

Now we find an upper bound for $Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p]$ for each of the three forgery types. In determining this upper bound, we use the following claim, whose proof can be found at the end of this appendix:

Claim 3.2
Let $z_k^p, 1 \le p \le q_e$, be the hidden ciphertext blocks generated at the encryption of a queried message $y_0^p y_1^p \quad y_{n_p+1}^p$, and z_j be the chosen hidden ciphertext block generated during the decryption of forgery $y = y_0, y_1, \quad y_{n+1}$. Then $\forall k, 1 \le k \le n_p + 1$,

$$Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p] \le \frac{1}{2^{l-m}},$$

where
(a) if $y_0 \ne y_0^p$, then $m = \min(m_1, m_2)$, with m_1 and m_2 being defined by $j = d_1 \times 2^{m_1}$, $k = d_2 \times 2^{m_2}$, where d_1, d_2 are odd; and
(b) if $y_0 = y_0^p$, where m is defined by $k \Leftrightarrow j = d \times 2^m$ if $k > j$, or by $j \Leftrightarrow k = d \times 2^m$ if $j < k$, and d is odd.

Claim 3.2 provides upper bounds for $Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p]$, where p, k are arbitrary values that satisfy the hypotheses of parts (a) or (b) and z_j is the chosen hidden ciphertext block defined in Part 1. These hypotheses are verified for the chosen j of each forgery type as shown below.

Upper bound for forgeries of type (a).
Let the ciphertext of queried message i be the prefix of forgery y. To find the upper bound in this case, we partition the sum $\sum_{p=1}^{q_e} \sum_{k=1}^{n_p+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p]$ into two sums, for $p \ne i$ and $p = i$, respectively. For $p \ne i$, we use Claim 3.2(a), and for $p = i$ we use Claim 3.2(b), to find an upper bound for $Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p]$. Then we find individual upper bounds for each of these two sums, and add these upper bounds.

$$\sum_{p=1}^{q_e} \sum_{k=1}^{n_p+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p] \quad = \quad \sum_{p=1, p \ne i}^{q_e} \sum_{k=1}^{n_p+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p] + \sum_{k=1}^{n_i+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^i].$$

For the first sum, note that $p \ne i$, and recall that for forgeries of type (a) $y_0 = y_0^i$. Since $\overline{E}$ is true, $y_0 = y_0^i \ne y_0^p$. Hence, by Claim 3.2(a), $Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p] \le \frac{1}{2^{l-m}}$, where $m \le m_2$ with m_2 being defined by $k = d_2 \times 2^{m_2}$ and d_2 is odd. Thus,

$$\sum_{p=1, p \ne i}^{q_e} \sum_{k=1}^{n_p+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p] \le \frac{1}{2^l} \sum_{p=1, p \ne i}^{q_e} \sum_{k=1}^{n_p+1} 2^{m_2}.$$

But, by Fact 3 with $N \Leftrightarrow 1 = n_p + 1$ and $a = k$,

$$\sum_{k=1}^{n_p+1} 2^{m_2} \le \frac{n_p + 1}{2} (\log_2(n_p + 1) + 3).$$

Hence,

$$\sum_{p=1, p \ne i}^{q_e} \sum_{k=1}^{n_p+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p] \le \frac{1}{2^l} \sum_{p=1, p \ne i}^{q_e} \frac{n_p + 1}{2} (\log_2(n_p + 1) + 3).$$

For the second sum, we note that $p = i$, which means that $y_0 = y_0^i = y_0^p$, and that $j = n_i + 2 > k, \forall k, 1 \leq k \leq n_i + 1$. Hence, by Claim 3.2(b) $Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p] \leq \frac{1}{2^{l-m}}$, where $j \Leftrightarrow k = d \times 2^m$ and d is odd. Since $j = n_i + 2$, in follows that $j \Leftrightarrow k = n_i + 1, \quad , 1$, and thus,

$$\sum_{k=1}^{n_i+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^i] \leq \sum_{j-k=1}^{n_i+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^i] \frac{1}{2^l} \sum_{j-k=1}^{n_i+1} 2^m.$$

But, by Fact 3 with $N \Leftrightarrow 1 = n_i + 1$ and $a = j \Leftrightarrow k$,

$$\sum_{j-k=1}^{n_i+1} 2^m \leq \frac{n_i + 1}{2}(\log_2(n_i + 1) + 3),$$

and hence,

$$\sum_{k=1}^{n_i+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^i] \leq \frac{1}{2^l} \frac{n_i + 1}{2}(\log_2(n_i + 1) + 3).$$

Adding the two upper bounds, we obtain

$$\begin{aligned} \sum_{p=1}^{q_e} \sum_{k=1}^{n_p+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p] &\leq \frac{1}{2^l} \frac{n_i + 1}{2}(\log_2(n_i + 1) + 3) + \frac{1}{2^l} \sum_{p=1, p \neq i}^{q_e} \frac{n_p + 1}{2}(\log_2(n_p + 1) + 3) \\ &= \frac{1}{2^l} \sum_{p=1}^{q_e} \frac{n_p + 1}{2}(\log_2(n_p + 1) + 3). \end{aligned}$$

Since $\sum_{p=1}^{q_e}(n_p + 1) \leq \frac{\mu_e}{l}$, by Fact 4, it follows that

$$\begin{aligned} &Pr_{\overline{I} \text{ and } \overline{E}}[z_j \in \{z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_p + 1\}] \leq \\ &\sum_{p=1}^{q_e} \sum_{k=1}^{n_p+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p] \leq \frac{1}{2^l} \frac{\mu_e}{2l} \quad \log_2 \frac{\mu_e}{l} + 3 \quad . \end{aligned}$$

Further, if $m = \max(n_p + 1)$, then

$$Pr_{\overline{I} \text{ and } \overline{E}}[z_j \in \{z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_p + 1\}] \leq \frac{1}{2^l} \frac{\mu_e}{2l} (\log_2 m + 3),$$

also by Fact 4.

Upper bound for forgeries of type (b1).
For this type of forgery, $y_0 \neq y_0^p, \forall p, 1 \leq p \leq q_e$. Hence, by Claim 3.2(a), $Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p] \leq \frac{1}{2^{l-m}}$, where $m \leq m_2$ with m_2 being defined by $k = d_2 \times 2^{m_2}$ and d_2 is odd. By following the same derivation as that for forgeries of type (a), we obtain:

$$\begin{aligned} &Pr_{\overline{I} \text{ and } \overline{E}}[z_j \in \{z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_p + 1\}] \leq \sum_{p=1}^{q_e} \sum_{k=1}^{n_p+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p] \leq \\ &\frac{1}{2^l} \sum_{p=1}^{q_e} \sum_{k=1}^{n_p+1} 2^{m_2} \leq \frac{1}{2^l} \sum_{p=1}^{q_e} \frac{n_p + 1}{2}(\log_2(n_p + 1) + 3) \leq \frac{1}{2^l} \frac{\mu_e}{2l} \quad \log_2 \frac{\mu_e}{l} + 3 \quad . \end{aligned}$$

Also, if $m = \max(n_p + 1)$, then

$$Pr_{\overline{I} \text{ and } \overline{E}}[z_j \in \{z_k^p, 1 \le p \le q_e, 1 \le k \le n_p + 1\}] \le \frac{1}{2^l} \frac{\mu_e}{2l} (\log_2 m + 3).$$

Upper bound for forgeries of type (b2).
Let the first $j \Leftrightarrow 1$ ciphertext blocks of queried message i provide the first $j \Leftrightarrow 1$ ciphertext blocks of forgery y. To find the upper bound in this case, we partition the sum $\sum_{p=1}^{q_e} \sum_{k=1}^{n_p+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p]$ into four terms, find individual upper bounds for each term, and then add these upper bounds. The first term is a sum taken for $p \neq i$ and in this case we use Claim 3.2(a) to find an upper bound for $Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p]$. The last three terms are for the case $p = i$, and two of these terms are sums taken for k and $k > j$, respectively. For these sums, we apply Claim 3.2(b) to find an upper bound for $Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p]$. For the remaining term corresponding to $i = p$ and $k = j$, we show that the event $z_j = z_k^p$ is impossible.

$$\begin{aligned}\sum_{p=1}^{q_e} \sum_{k=1}^{n_p+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p] &= \sum_{p=1, p\neq i}^{q_e} \sum_{k=1}^{n_p+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p] + \sum_{k=1}^{j-1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^i] + \\ &\quad Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_j^i] + \sum_{k=j+1}^{n_i+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^i].\end{aligned}$$

For the first of the four terms above, we have the same bound as that of the first of the two sums in the case of forgeries of type (a) above, namely,

$$\sum_{p=1, p\neq i}^{q_e} \sum_{k=1}^{n_p+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p] \le \frac{1}{2^l} \sum_{p=1, p\neq i}^{q_e} \frac{n_p + 1}{2} (\log_2(n_p + 1) + 3).$$

For the second term, namely $\sum_{k=1}^{j-1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^i]$, we note that $i = p$, which means that $y_0 = y_0^i = y_0^p$, and $k < j$. Hence, by Claim 3.2(b), $Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^i] \le \frac{1}{2^{l-m}}$, where $j \Leftrightarrow k = d \times 2^m$ and d is odd. Since $k = 1, \quad , j \Leftrightarrow 1$, it follows that $j \Leftrightarrow k = j \Leftrightarrow 1, \quad , 1$, and by Fact 3 with $N \Leftrightarrow 1 = j \Leftrightarrow 1$ and $a = j \Leftrightarrow k$,

$$\sum_{k=1}^{j-1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^i] = \sum_{j-k=1}^{j-1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^i] \le \frac{1}{2^l} \sum_{j-k=1}^{j-1} 2^m \le \frac{1}{2^l} \frac{j \Leftrightarrow 1}{2} (\log_2(j \Leftrightarrow 1) + 3).$$

For the third term, $Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_j^i] = 0$. This is the case because $z_j = z_j^i \Leftrightarrow y_j \Leftrightarrow j \times r_0 = y_j^i \Leftrightarrow j \times r_0^i$ and, since $y_0 = y_0^i \Leftrightarrow r_0 = r_0^i$, it follows that $z_j = z_j^i \Leftrightarrow y_j = y_j^i$, which is impossible by the definition of j. (Recall that for forgeries of type (b2), j is the smallest index such that $y_j \neq y_j^i, 1 \le j \le \min\{n_i+1, n+1\}$.)

For the fourth term, namely $\sum_{k=j+1}^{n_i+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^i]$, we note that $i = p$, which means that $y_0 = y_0^i = y_0^p$, and $j < k$. Hence, by Claim 3.2(b), $Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^i] \le \frac{1}{2^{l-m}}$, where $k \Leftrightarrow j = d \times 2^m$ and d is odd. Since $k = j + 1, \quad , n_i + 1$, it follows that $k \Leftrightarrow j = 1, \quad , n_i \Leftrightarrow j + 1$, and by Fact 3 with $N \Leftrightarrow 1 = n_i + 1 \Leftrightarrow j$ and $a = k \Leftrightarrow j$,

$$\begin{aligned}\sum_{k=j+1}^{n_i+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^i] &= \sum_{k-j=1}^{n_i-j+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^i] \le \frac{1}{2^l} \sum_{k-j=1}^{n_i-j+1} 2^m \\ &\le \frac{1}{2^l} \frac{n_i \Leftrightarrow j + 1}{2} (\log_2(n_i \Leftrightarrow j + 1) + 3).\end{aligned}$$

Now, we add the bounds of the last three of the individual upper bounds, and then we add the first upper bound to obtain the total upper bound for forgeries of type (b2).

$$\sum_{k=1}^{j-1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^i] + Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_j^i] + \sum_{k=j+1}^{n_i+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^i] \leq$$
$$\frac{1}{2^l}\frac{j \Leftrightarrow 1}{2}(\log_2(j \Leftrightarrow 1) + 3) + \frac{1}{2^l}\frac{n_i \Leftrightarrow j+1}{2}(\log_2(n_i \Leftrightarrow j+1) + 3).$$

Since for this type of forgeries $1 \leq j \leq n_i + 1$, the terms under $\log_2$ are $j \Leftrightarrow 1 \leq n_i, n_i \Leftrightarrow j+1 \leq n_i$. Thus, the sum of the last three terms is bounded as follows:

$$\sum_{k=1}^{j-1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^i] + Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_j^i] + \sum_{k=j+1}^{n_i-j+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^i] \leq$$
$$\frac{1}{2^l}\frac{j \Leftrightarrow 1}{2}(\log_2 n_i + 3) + \frac{1}{2^l}\frac{n_i \Leftrightarrow j+1}{2}(\log_2 n_i + 3) = \frac{1}{2^l}\frac{n_i}{2}(\log_2 n_i + 3) \leq$$
$$\frac{1}{2^l}\frac{n_i+1}{2}(\log_2(n_i+1) + 3).$$

Hence, by adding the first of the individual upper bounds to this above sum, we obtain:

$$\begin{aligned}\sum_{p=1}^{q_e}\sum_{k=1}^{n_p+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p] &\leq \frac{1}{2^l}\frac{n_i+1}{2}(\log_2(n_i+1) + 3) + \\ &\quad \frac{1}{2^l}\sum_{p=1, p\neq i}^{q_e}\frac{n_p+1}{2}(\log_2(n_p+1) + 3) \\ &= \frac{1}{2^l}\sum_{p=1}^{q_e}\frac{n_p+1}{2}(\log_2(n_p+1) + 3).\end{aligned}$$

Since $\sum_{p=1}^{q_e}(n_p+1) \leq \frac{\mu_e}{l}$, by Fact 4, it follows that

$$Pr_{\overline{I} \text{ and } \overline{E}}[z_j \in \{z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_p+1\}] \leq$$
$$\sum_{p=1}^{q_e}\sum_{k=1}^{n_p+1} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p] \leq \frac{1}{2^l}\frac{\mu_e}{2l} \quad \log_2\frac{\mu_e}{l} + 3 \quad .$$

Further, if $m = \max(n_p+1)$, then $Pr_{\overline{I} \text{ and } \overline{E}}[z_j \in \{z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_p+1\}] \leq \frac{1}{2^l}\frac{\mu_e}{2l}(\log_2 m + 3)$.

Finally, for any forgery that is not a truncation, $Pr_{\overline{I} \text{ and } \overline{E}}[z_j \in \{z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_p+1\}]$ is bounded by the maximum of the bounds for the types (a), (b1) and (b2), namely

$$Pr_{\overline{I} \text{ and } \overline{E}}[z_j \in \{z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_p+1\}] \leq \frac{1}{2^l}\frac{\mu_e}{2l} \quad \log_2\frac{\mu_e}{l} + 3 \quad ;$$

or, if $m = \max(n_p+1)$, then $Pr_{\overline{I} \text{ and } \overline{E}}[z_j \in \{z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_p+1\}] \leq \frac{1}{2^l}\frac{\mu_e}{2l}(\log_2 m + 3)$. Hence, returning to the probability of event C conditioned by ($\overline{I}$ and $\overline{E}$),

$$Pr_{\overline{I} \text{ and } \overline{E}}[C] = Pr[C \mid \overline{I} \text{ and } \overline{E}] \leq \frac{(n+1)q_e}{2^l} + \frac{1}{2^l}\frac{\mu_e}{2l} \quad \log_2\frac{\mu_e}{l} + 3 \quad .$$

Also, if the maximum length m of the encrypted messages is known, the last term of the above bound can be replaced with $\frac{1}{2^l}\frac{\mu_e}{2l}(\log_2 m + 3)$. $\square$

Proof of Claim 4

Event $\overline{C}$ is true implies that there is at least one element $z_s \in Z$. Event $\overline{D}$ states that any hidden ciphertext block $z_s \in Z$ collides with another hidden block $z_t \in Z, t \neq s$, or z_s collides with y_0. Let s be the smallest index of the element $z_s \in Z$; hence, event $\overline{D}$ implies that z_s collides with some other element $z_t \in Z, t > s$ or $z_s = y_0$, or, alternatively, $z_s \in Z \Leftrightarrow \{z_s\}$ or $z_s = y_0$. Hence,

$$Pr[\overline{D} \mid \overline{C} \text{ and } \overline{E} \text{ and } \overline{I}] \leq Pr[z_s \in Z \Leftrightarrow \{z_s\} \text{ or } z_s = y_0 \mid \overline{C} \text{ and } \overline{E} \text{ and } \overline{I}]$$

Union bound leads to:

$$\begin{aligned} Pr[\overline{D} \mid \overline{C} \text{ and } \overline{E} \text{ and } \overline{I}] &\leq Pr[z_s \in Z \Leftrightarrow \{z_s\} \mid \overline{C} \text{ and } \overline{E} \text{ and } \overline{I}] + Pr[z_s = y_0 \mid \overline{C} \text{ and } \overline{E} \text{ and } \overline{I}] \\ &\leq \sum_{t>s, z_t \in Z} Pr[z_s = z_t \mid \overline{C} \text{ and } \overline{E} \text{ and } \overline{I}] + Pr[z_s = y_0 \mid \overline{C} \text{ and } \overline{E} \text{ and } \overline{I}]. \end{aligned}$$

To compute the upper bound of the first probability of the sum, $Pr[z_s = z_t, z_s, z_t \in Z, t \neq s \mid \overline{C} \text{ and } \overline{E} \text{ and } \overline{I}]$, recall that Z must have at least one element (since $\overline{C}$ is true). If Z has only one element, then this probability is zero. If Z has at least two elements, z_s, z_t, we use the following claim, whose proof can be found at the end of this Appendix:

Claim 4.1
(a) For any $z_s, z_t \in Z, 1 \leq s \quad t \leq n+1$:

$$Pr_{\overline{C} \text{ and } \overline{E} \text{ and } \overline{I}}[z_s = z_t] \leq \frac{1}{2^{l-m}},$$

where the exponent m is defined by $t \Leftrightarrow s = d \times 2^m$ and d is odd.
(b) For any $z_s \in Z, 1 \leq s \leq n+1$, and for any y_0:

$$Pr_{\overline{C} \text{ and } \overline{E} \text{ and } \overline{I}}[z_s = y_0] \leq \frac{1}{2^{l-m}},$$

where the exponent m is defined by $s = d \times 2^m$ and d is odd.

Then, by Claim 4.1(a)

$$\sum_{t>s, z_t \in Z} Pr[z_s = z_t \mid \overline{C} \text{ and } \overline{E} \text{ and } \overline{I}] \leq \sum_{t>s, z_t \in Z} \frac{2^m}{2^l}$$

where $t \Leftrightarrow s = d \times 2^m$ and d is odd. Let $a = t \Leftrightarrow s, z_s, z_t \in Z, s \quad t$. Then, by using this notation, the fact that the differences $t \Leftrightarrow s$ represent a subset of set $\{1, \quad, n\}$, and Fact 3, we obtain

$$\begin{aligned} \sum_{t>s, z_t \in Z} \frac{2^m}{2^l} &= \sum_{a=t-s, t>s, z_s, z_t \in Z} \frac{2^m}{2^l} \\ &\leq \sum_{a=1}^{n} \frac{2^m}{2^l} \leq \frac{n}{2^{l+1}}(\log_2 n + 3). \end{aligned}$$

For the term $Pr[z_s = y_0 \mid \overline{C} \text{ and } \overline{E} \text{ and } \overline{I}]$, we use Claim 4.1(b) and obtain:

$$Pr[z_s = y_0 \mid \overline{C} \text{ and } \overline{E} \text{ and } \overline{I}] \leq \frac{1}{2^{l-m}},$$

where m is defined by $s = d \times 2^m$ and d is odd. By definition, $m \leq \log_2 s \leq \log_2(n+1)$, and hence $2^m \leq n+1$. Thus,

$$Pr[z_s = y_0 \mid \overline{C} \text{ and } \overline{E} \text{ and } \overline{I}] \leq \frac{n+1}{2^l}.$$

By adding the two upper bounds, it follows that

$$Pr[\overline{D} \mid \overline{C} \text{ and } \overline{E} \text{ and } \overline{I}] \leq \frac{n}{2^{l+1}}(\log_2 n + 3) + \frac{n+1}{2^l}.$$

□

Proof of Fact 1

(a) Let A be an adversary attacking the $XCBC\$ \Leftrightarrow XOR$ mode using $q_e + q_v$ queries, $\mu_e + \mu_v$ total memory for these queries, and time $t_e + t_v$. The probability of success is related directly to the security of the underlying encryption mode XCBC\$ and F. To find an upper bound for this probability, we introduce a distinguisher D for F, which is given two oracles f and f^{-1}, where f is a permutation used by the $XCBC\$ \Leftrightarrow XOR$ mode. D runs A, simulates an oracle for $XCBC\$ \Leftrightarrow XOR$ via queries for its own oracles f and f^{-1}, responds to A's q_e encryption queries, and verifies A's choices of ciphertext forgeries $y'^i = y'^i_0 y'^i_1 \quad y'^i_n, y'^i_{n+1}, 1 \leq i \leq q_v$. D returns the result of each y'^i's verification to A; i.e., D returns either *Success* or *Failure* to A. D outputs 1 if A's forgery decrypts successfully, and 0, otherwise.

Distinguisher D's advantage, $Adv_D(F, P^l) \leq \epsilon$, is defined as:

$$Adv_D^{sprp}(F, P^l) = Pr_{f \overset{\mathcal{R}}{\leftarrow} F}[D^f = 1] \Leftrightarrow Pr_{f \overset{\mathcal{R}}{\leftarrow} P^l}[D^f = 1].$$

where $f \overset{\mathcal{R}}{\leftarrow} F$ denotes the selection of function f from the SPRP family F by the random key K, and $f \overset{\mathcal{R}}{\leftarrow} P^l$ denotes the random selection of f from the set of all permutations P^l.

By the definition of the distinguisher algorithm:

$$Pr_{f \overset{\mathcal{R}}{\leftarrow} F}[D^f = 1] = Pr_{f \overset{\mathcal{R}}{\leftarrow} F}[\mathcal{D} \Leftrightarrow XCBC\$ \Leftrightarrow XOR(y) \neq Null] = Pr_{f \overset{\mathcal{R}}{\leftarrow} F}[Succ]$$

and

$$Pr_{f \overset{\mathcal{R}}{\leftarrow} P^l}[D^f = 1] = Pr_{f \overset{\mathcal{R}}{\leftarrow} P^l}[\mathcal{D} \Leftrightarrow XCBC\$ \Leftrightarrow XOR(y) \neq Null] = Pr_{f \overset{\mathcal{R}}{\leftarrow} P^l}[\text{Succ}].$$

The above probabilities are over the random choice of r_0, $f \overset{\mathcal{R}}{\leftarrow} F$, $f \overset{\mathcal{R}}{\leftarrow} P^l$, and D's guesses. Hence,

$$\begin{aligned} Pr_{f \overset{\mathcal{R}}{\leftarrow} F}[\text{Succ}] &= Pr_{f \overset{\mathcal{R}}{\leftarrow} F}[\text{Succ}] \Leftrightarrow Pr_{f \overset{\mathcal{R}}{\leftarrow} P^l}[\text{Succ}] + Pr_{f \overset{\mathcal{R}}{\leftarrow} P^l}[\text{Succ}] \\ &= Adv_D^{sprp}(F, P^l) + Pr_{f \overset{\mathcal{R}}{\leftarrow} P^l}[\text{Succ}] \leq \epsilon + Pr_{f \overset{\mathcal{R}}{\leftarrow} P^l}[\text{Succ}]. \end{aligned}$$

(b) This proof is based on constructing a polynomial-time algorithm D that distinguishes between $f^{-1} \overset{\mathcal{R}}{\leftarrow} P^l$ and $\overline{f} \overset{\mathcal{R}}{\leftarrow} G_S$ using an adversary A for the $XCBC\$ \Leftrightarrow XOR$ mode.

In a similar manner to the one used in part (a) (repeated here for completeness), let A be an adversary attacking the $XCBC\$ \Leftrightarrow XOR$ mode using $q_e + q_v$ queries, $\mu_e + \mu_v$ total memory for these queries, and

time $t_e + t_v$. To find an upper bound for $Pr_{f \overset{\mathcal{R}}{\leftarrow} P^l}[Succ]$, we introduce a distinguisher D for P^l which is given two oracles $\mathcal{O}$, $\mathcal{O}^{-1}$. These oracles simulate the block encryption and decryption operations needed by D to simulate the $XCBC\$ \Leftrightarrow XOR$ mode for adversary A. Oracle $\mathcal{O}$ simply uses $f \overset{\mathcal{R}}{\leftarrow} P^l$ to respond to D's block encryption requests. In contrast, oracle $\mathcal{O}^{-1}$ flips a coin $b \in \{0,1\}$ and responds to D's block decryption requests by using either $f^{-1} \overset{\mathcal{R}}{\leftarrow} P^l$ or $\overline{f} \overset{\mathcal{R}}{\leftarrow} G_S$. D runs A, responds to A's q_e encryption queries, and then verifies A's choices of ciphertext forgeries $y'^i = y'^i_0 y'^i_1 \quad y'^i_n, y'^i_{n+1}, 1 \leq i \leq q_v$. [As a consequence, D issues all its requests for block encryption to $\mathcal{O}$, if any, before all the requests for block decryption to $\mathcal{O}^{-1}$.] D returns the result of each y'^i's decryption to A; i.e., D returns either *Success* or *Failure* to A. D outputs 1 if A's forgery decrypts successfully, and 0, otherwise.

Distinguisher D's advantage, $Adv_D(P^l, G_S)$, is defined as:

$$Adv_D(P^l, G_S) = Pr_{f \overset{\mathcal{R}}{\leftarrow} P^l}[D^f = 1] \Leftrightarrow Pr_{f \overset{\mathcal{R}}{\leftarrow} G_S}[D^f = 1].$$

where $f \overset{\mathcal{R}}{\leftarrow} P^l$ denotes the selection of function f, and its inverse f^{-1}, from the set of all permutations P^l by the random key K, and $f \overset{\mathcal{R}}{\leftarrow} G_S$ denotes the random selection of f from P^l to encrypt and the associated function $\overline{f} \overset{\mathcal{R}}{\leftarrow} G_S$ to decrypt.

By the definition of the distinguisher algorithm:

$$Pr_{f \overset{\mathcal{R}}{\leftarrow} P^l}[D^f = 1] = Pr_{f \overset{\mathcal{R}}{\leftarrow} P^l}[\mathcal{D} \Leftrightarrow XCBC\$ \Leftrightarrow XOR(y) \neq Null] = Pr_{f \overset{\mathcal{R}}{\leftarrow} P^l}[Succ]$$

and

$$Pr_{f \overset{\mathcal{R}}{\leftarrow} G_S}[D^f = 1] = Pr_{f \overset{\mathcal{R}}{\leftarrow} G_S}[\mathcal{D} \Leftrightarrow XCBC\$ \Leftrightarrow XOR(y) \neq Null] = Pr_{f \overset{\mathcal{R}}{\leftarrow} G_S}[\text{Succ}].$$

The above probabilities are over the random choice of r_0, $f \overset{\mathcal{R}}{\leftarrow} P^l$, $f \overset{\mathcal{R}}{\leftarrow} G_S$, and D's guesses. Hence,

$$\begin{aligned} Pr_{f \overset{\mathcal{R}}{\leftarrow} P^l}[\text{Succ}] = Pr_{f \overset{\mathcal{R}}{\leftarrow} P^l}[\text{Succ}] &\Leftrightarrow Pr_{f \overset{\mathcal{R}}{\leftarrow} G_S}[\text{Succ}] + Pr_{f \overset{\mathcal{R}}{\leftarrow} G_S}[\text{Succ}] \\ &= Adv_D(P^l, G_S) + Pr_{f \overset{\mathcal{R}}{\leftarrow} G_S}[\text{Succ}]. \end{aligned}$$

Now we find an upper bound for D's advantage in distinguishing between P^l and G_S. By the definition of the two oracles $\mathcal{O}$ and $\mathcal{O}^{-1}$, only oracle $\mathcal{O}^{-1}$ can be used by D to distinguish between P^l and G_S. Furthermore, whenever a block decryption request to oracle $\mathcal{O}^{-1}$ is a ciphertext block that was generated during the encryption of A's q_e queries, the output of oracle $\mathcal{O}^{-1}$ is the same for both $f \overset{\mathcal{R}}{\leftarrow} P^l$ and $f \overset{\mathcal{R}}{\leftarrow} G_S$ (by the definition of $\overline{f}$), and a distinction between P^l and G_S cannot be made. Hence, D can make a distinction between P^l and G_S only when the ciphertext blocks of the decryption requests to oracle $\mathcal{O}^{-1}$ (i.e., the inputs to f^{-1} or $\overline{f}$) have never been generated during the encryption of A's q_e queries; i.e., the ciphertext blocks are not in $S_{f \overset{\mathcal{R}}{\leftarrow} P^l}$.

To make the distinction between $f^{-1} \overset{\mathcal{R}}{\leftarrow} P^l$ and $\overline{f} \overset{\mathcal{R}}{\leftarrow} G_S$, D needs to send only ciphertext blocks that are not in $S_{f \overset{\mathcal{R}}{\leftarrow} P^l}$ to oracle $\mathcal{O}^{-1}$, since D already has the plaintext blocks corresponding to all the ciphertext blocks in $S_{f \overset{\mathcal{R}}{\leftarrow} P^l}$. In this case, $\overline{f} = v$, where $v \overset{\mathcal{R}}{\leftarrow} R^{l,l}$, and the advantage of distinguisher D cannot be higher than the advantage of any polynomial-time algorithm D' that distinguishes a random permutation from a random function using the same block decryption requests from $\{0,1\}^l \Leftrightarrow S_{f \overset{\mathcal{R}}{\leftarrow} P^l}$ to oracle $\mathcal{O}^{-1}$ as those made by distinguisher $\mathcal{D}$; i.e., $Adv_D(P^l, G_S) \leq Adv_{D'}(P^l, R^{l,l})$. However, by the bound of the birthday

attack, $Adv_{D'}(P^l, R^{l,l}) \leq \frac{q(q-1)}{2^{l+1}}$ where q is the number of the block decryption requests to oracle $\mathcal{O}^{-1}$; i.e., $q \leq \frac{\mu_v}{l}$ Hence,

$$Adv_D(P^l, G_S) \leq Adv_{D'}(P^l, R^{l,l}) \leq \frac{\mu_v(\mu_v \Leftrightarrow l)}{l^2 2^{l+1}}.$$

Hence,

$$Pr_{f \xleftarrow{\mathcal{R}} P^l}[\text{Succ}] \leq Pr_{f \xleftarrow{\mathcal{R}} G_S}[\text{Succ}] + \frac{\mu_v(\mu_v \Leftrightarrow l)}{l^2 2^{l+1}}.$$

□

Proof of Fact 2

If $i = d \times 2^m$, then $i \times r_0 = d \times 2^m \times r_0$ has (at least) the first (i.e., least significant) m bits zero. Also, since $i \quad 2^l$, it follows that $d \quad 2^{l-m}$. Let $r_{0m} = r_0[1 \quad l \Leftrightarrow m]$ be the least significant $l \Leftrightarrow m$ bits of r_0. (These bits will be shifted in the most significant $l \Leftrightarrow m$ bit positions of a block by multiplication with 2^m.)

First, we note that

$$i \times r_0 = (dr_{0m})||\underbrace{0 \quad 0}_{m}$$

where $dr_{0m} = \underbrace{r_{0m} + \quad + r_{0m}}_{d\ times} \bmod 2^{l-m}$ and $||$ is the concatenation operator. To see this:

$$\begin{aligned} i \times r_0 &= (d \times 2^m) \times r_0 = d \times (r_0 \times 2^m) = \underbrace{(r_0 \times 2^m) + \quad + (r_0 \times 2^m)}_{d\ times} \\ &= \underbrace{(r_{0m}||\underbrace{0 \quad 0}_{m}) + \quad + (r_{0m}||\underbrace{0 \quad 0}_{m})}_{d\ times} = (\underbrace{r_{0m} + \quad + r_{0m}}_{d\ times})||\underbrace{0 \quad 0}_{m} \\ &= (dr_{0m})||\underbrace{0 \quad 0}_{m} \end{aligned}$$

where $dr_{0m} = \underbrace{r_{0m} + \quad + r_{0m}}_{d\ times} \bmod 2^{l-m}$.

Second, we divide all values of an arbitrary constant a into two complementary classes based on whether the first (i.e., least significant) m bits of a are all zero, compute $Pr[i \times r_0 = a]$ for each class separately, and then take the maximum of the two probabilities as the overall bound.

Let $a[1 \quad m] = 0$ denote the values of a for which the first m bits are zero, and $a[1 \quad m] \neq 0$ those for which at least one of the the first m bits is not zero. Since $i \times r_0 = (dr_{0m})||\underbrace{0 \quad 0}_{m}$, it follows that, if $a[1 \quad m] \neq 0$, $Pr[\ i \times r_0 = a\ ] = 0$. However, if $a[1 \quad m] = 0$, then $[\ i \times r_0 = a\ ] \Leftrightarrow [\ dr_{0m} = b\ ]$, where $b = a[m+1 \quad l]$ represents bits $m+1, \quad l$ of a, i.e., the $l \Leftrightarrow m$ most significant bits of a. Hence, in this case,

$$Pr[\ i \times r_0 = a\ ] = Pr[\ dr_{0m} = b\ ],$$

where $d, r_{0m}, b \in \{0,1\}^{l-m}$. However, d and 2^{l-m} are relatively prime because d is odd. Hence, d has a left inverse,[9] e, and $dr_{0m} = b \Leftrightarrow edr_{0m} = eb \Leftrightarrow r_{0m} = eb \pmod{2^{l-m}}$, which happens with probability $1/2^{l-m}$ because $r_{0m} = r[1 \quad l \Leftrightarrow m]$ is random and uniformly distributed in $\{0,1\}^{l-m}$. Thus, if $a[1 \quad m] = 0$,

$$Pr[i \times r_0 = a] = \frac{1}{2^{l-m}}.$$

[9] A way to see that d has a left inverse, e, is to label $2^{l-m} = f$, and to note that, if d and f are relatively prime, then, by Euclid's gcd algorithm, there exists e and h such that $ed + hf = 1$; i.e., $ed = 1 - hf$ or $ed = 1(\text{mod} f)$.

Hence, for any value of constant a, $Pr[i \times r_0 = a] \leq \frac{1}{2^{l-m}}$. $\square$

Proof of Fact 3

Since any a can be expressed as $a = d \times 2^m$, where d is odd, there are multiple values of a that have the same exponent m. (For example, for all odd values of a, $m = 0$, and for all even values of a that are not a multiple of 4, $m = 1$.) Hence, when computing the sum $\sum_{a=1}^{N-1} 2^m$, we can group together the terms 2^m that have the same exponent m (i.e., we group the terms 2^m that are equal).

For a given exponent m, we find the number of distinct values of a that have the same exponent m when represented as $d \times 2^m$. To find this number, we note that $1 \leq a \leq N \Leftrightarrow 1$ and, hence, $1 \leq d \leq \lfloor \frac{N-1}{2^m} \rfloor$. Hence, the number of distinct values of a that yield the same exponent m is $\lfloor \frac{1}{2} \ \lfloor \frac{N-1}{2^m} \rfloor + 1 \ \rfloor$, since this number is bounded by the number of distinct values of d odd.

¿From the definition of exponent m, $2^m \leq N \Leftrightarrow 1$ (i.e., $0 \leq m \leq \log_2(N \Leftrightarrow 1)$). Hence,

$$\begin{aligned}
\sum_{a=1}^{N-1} 2^m &= \sum_{m=0}^{\lfloor \log_2(N-1) \rfloor} \lfloor \frac{1}{2} \ \lfloor \frac{N \Leftrightarrow 1}{2^m} \rfloor + 1 \ \rfloor 2^m \leq \sum_{m=0}^{\lfloor \log_2(N-1) \rfloor} \frac{N \Leftrightarrow 1}{2} + \frac{2^m}{2} \\
&= \frac{N \Leftrightarrow 1}{2} (\lfloor \log_2(N \Leftrightarrow 1) \rfloor + 1) + \frac{2^{\lfloor \log_2(N-1) \rfloor + 1} \Leftrightarrow 1}{2}
\end{aligned}$$

because, for any $M > 0$, $\sum_{m=0}^{M} 2^m = 2^{M+1} \Leftrightarrow 1$. Hence,

$$\sum_{a=1}^{N-1} 2^m \leq \frac{N \Leftrightarrow 1}{2} (\log_2(N \Leftrightarrow 1) + 1) + (N \Leftrightarrow 1) = \frac{N \Leftrightarrow 1}{2} (\log_2(N \Leftrightarrow 1) + 3).$$

$\square$

Proof of Fact 4

Since, by hypothesis, $\sum_{p=1}^{q_e} (n_p + 1) \leq \frac{\mu_e}{l}$, the term under the $\log_2$ is $n_p + 1 \leq \frac{\mu_e}{l}$. Hence, we obtain:

$$\sum_{p=1}^{q_e} (n_p + 1) \log_2(n_p + 1) \leq \log_2 \frac{\mu_e}{l} \sum_{p=1}^{q_e} (n_p + 1),$$

and thus,

$$\sum_{p=1}^{q_e} (n_p + 1) \log_2(n_p + 1) \leq \frac{\mu_e}{l} \log_2 \frac{\mu_e}{l}.$$

Further, if $m = \max(n_p + 1)$, then $\log_2(n_p + 1) \leq \log_2 m$. Hence,

$$\sum_{p=1}^{q_e} (n_p + 1) \log_2(n_p + 1) \leq \frac{\mu_e}{l} \log_2 m.$$

$\square$

Proof of Claim 3.1

There are three possible complementary cases to consider:
(1) $y_0 = y_0^i$, for some queried message $i, 1 \leq i \leq q_e$. Then $r_0 = \overline{f} = f^{-1}(y_0^i) = r_0^i$ is random and uniformly distributed, by definition. Furthermore, if $r_0 = r_0^i \neq r_0^p$ (i.e., $y_0 = y_0^i \neq y_0^p$), then $i \neq p$ and r_0 is also independent of r_0^p, by definition.
(2) $y_0 = z_j^i$, for some queried message $i, 1 \leq i \leq q_e, 1 \leq j \leq n_i + 1$; i.e., y_0 collides with some hidden ciphertext block, z_j^i, generated during the encryption of message i. But this is exactly the event prohibited by $\overline{I}$.
(3) $y_0 \neq y_0^i$ and $y_0 \neq z_k^i$, for all queried messages $i, 1 \leq i \leq q_e, k \geq 1$. Then $r_0 = \overline{f}(y_0) = v(y_0) \neq r_0^i, \forall i, 1 \leq i \leq q_e$ is random, uniformly distributed and independent of anything else because $v \overset{\mathcal{R}}{\leftarrow} R^{l,l}$ and $\overline{f}$ has never been invoked with argument y_0. Hence, r_0 is random, uniformly distributed and independent of r_0^p. □

Proof of Claim 3.2

The event $z_j = z_k^p$ is equivalent to $y_j \Leftrightarrow j \times r_0 = y_k^p \Leftrightarrow k \times r_0^p \;\Leftrightarrow\; j \times r_0 = k \times r_0^p \Leftrightarrow y_k^p + y_j \;\Leftrightarrow\; k \times r_0^p = j \times r_0 \Leftrightarrow y_j + y_k^p$.
(a) If $y_0 \neq y_0^p$, and since event $\overline{I}$ is true, it follows that r_0 is random, uniformly distributed, and independent of r_0^p, by Claim 3.1 above. Also, event $\overline{I}$ and $\overline{E}$ implies that r_0 is random, uniformly distributed, and independent of r_0^p by the definition of event E. Thus, $j \times r_0$ is independent of $k \times r_0^p \Leftrightarrow y_k^p + y_j$ and $k \times r_0^p$ is independent of $j \times r_0 \Leftrightarrow y_j + y_k^p$, since $j, k > 0$, and y_j, y_k^p, j, k are known constants. Furthermore, event $[z_j = z_k^p] \equiv [j \times r_0 = k \times r_0^p \Leftrightarrow y_k^p + y_j] \equiv [k \times r_0^p = j \times r_0 \Leftrightarrow y_j + y_k^p]$. Hence,

$$\begin{aligned} Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p] &= Pr_{\overline{I} \text{ and } \overline{E}}[j \times r_0 = k \times r_0^p \Leftrightarrow y_k^p + y_j] \\ &= Pr_{\overline{I} \text{ and } \overline{E}}[k \times r_0^p = j \times r_0 \Leftrightarrow y_j + y_k^p]. \end{aligned}$$

However, $Pr_{\overline{I} \text{ and } \overline{E}}[j \times r_0 = k \times r_0^p \Leftrightarrow y_k^p + y_j] \leq \frac{1}{2^{l-m_1}}$, where $j = d_1 \times 2^{m_1}$ and d_1 is odd, by Fact 2. Also, $Pr_{\overline{I} \text{ and } \overline{E}}[k \times r_0^p = j \times r_0 \Leftrightarrow y_j + y_k^p] \leq \frac{1}{2^{l-m_2}}$, where $k = d_2 \times 2^{m_2}$ and d_2 is odd. Hence,

$$Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p] \leq \min \left(\frac{1}{2^{l-m_1}}, \frac{1}{2^{l-m_2}} \right) = \frac{1}{2^{l-m}},$$

where $m = \min(m_1, m_2)$.

(b) If $y_0 = y_0^p$, then $r_0 = r_0^p$. Hence,

$$z_j = z_k^p \;\Leftrightarrow\; y_j \Leftrightarrow j \times r_0 = y_k^p \Leftrightarrow k \times r_0^p \;\Leftrightarrow\; (k \Leftrightarrow j) \times r_0 = y_k^p \Leftrightarrow y_j.$$

Thus,

$$Pr_{\overline{I} \text{ and } \overline{E}}[z_j = z_k^p] = Pr_{\overline{I} \text{ and } \overline{E}}[(k \Leftrightarrow j) \times r_0 = y_k^p \Leftrightarrow y_j].$$

However, since event $\overline{I}$ is true, it follows that r_0 is random and uniformly distributed, by Claim 3.1 above. Also, event $\overline{I}$ and $\overline{E}$ implies that r_0 is random and uniformly distributed, by the definition of event E. Since $j, k > 0, j \neq k$, and y_j, y_k^p, j, k are known constants, and $k \neq j$, Fact 2 implies that

$$Pr_{\overline{I} \text{ and } \overline{E}}[(k \Leftrightarrow j) \times r_0 = y_k^p \Leftrightarrow y_j] \leq \frac{1}{2^{l-m}}$$

where m is defined by $k \Leftrightarrow j = d \times 2^m, k > j$ or $j \Leftrightarrow k = d \times 2^m, j > k$, and d is odd. □

Proof of Claim 4.1

(a) One can write the event $z_t = z_s \;\Leftrightarrow\; (t \Leftrightarrow s) \times r_0 = y_t \Leftrightarrow y_s$. Hence,

$$Pr_{\overline{C} \text{ and } \overline{E} \text{ and } \overline{I}}[z_s = z_t] = Pr_{\overline{C} \text{ and } \overline{E} \text{ and } \overline{I}}[(t \Leftrightarrow s) \times r_0 = y_t \Leftrightarrow y_s].$$

Since event $\overline{I}$ is true, r_0 is random and uniformly distributed, by Claim 3.1. Furthermore, by the definition of events $\overline{E}$ and $\overline{C}$, event $\overline{C}$ and $\overline{I}$ and $\overline{E}$ implies that r_0 is random and uniformly distributed. Using the definition of m and the facts that (1) r_0 is random and uniformly distributed, (2) y_t, y_s are constants, and (3) $1 \leq t \Leftrightarrow s \leq 2^l \Leftrightarrow 1$, we obtain (by Fact 2) that

$$Pr_{\overline{C} \text{ and } \overline{E} \text{ and } \overline{I}}[(t \Leftrightarrow s) \times r_0 = y_t \Leftrightarrow y_s] \leq \frac{1}{2^{l-m}}$$

where m is defined by $t \Leftrightarrow s = d \times 2^m$ and d is odd. Hence,

$$Pr_{\overline{C} \text{ and } \overline{E} \text{ and } \overline{I}}[z_s = z_t] \leq \frac{1}{2^{l-m}}.$$

(b) The proof of this part is similar to that of part (a) and is included here for completeness.

Note that, since $z_s = y_s \Leftrightarrow s \times r_0$, event $z_s = y_0 \;\Leftrightarrow\; s \times r_0 = y_s \Leftrightarrow y_0$, where y_s and y_0 are constants. However, since event $\overline{I}$ is true, r_0 is random and uniformly distributed, by Claim 3.1. Furthermore, event $\overline{C}$ and $\overline{I}$ and $\overline{E}$ implies that r_0 is random and uniformly distributed. Hence, by Fact 2,

$$Pr_{\overline{C} \text{ and } \overline{E} \text{ and } \overline{I}}[z_s = y_0] = Pr_{\overline{C} \text{ and } \overline{E} \text{ and } \overline{I}}[s \times r_0 = y_s \Leftrightarrow y_0] \leq \frac{1}{2^{l-m}}$$

where m is defined by $s = d \times 2^m$ and d is odd. $\square$

Appendix B – Proof [Security of the Stateful-Sender XEBC-MAC (XECBC-MAC) in an Adaptive Chosen-Message Attack]

Throughout this proof, we use the same notation as in the Proof for Security of the XCBC\$-XOR in a Message-Integrity Attack, Appendix A, and the same facts (i.e., Facts 1 4). Unless mentioned otherwise, we focus on the probability for adversary's success when $f \stackrel{\mathcal{R}}{\leftarrow} R^{l,l}$, and, for simplicity, we will drop the $f \stackrel{\mathcal{R}}{\leftarrow} R^{l,l}$ subscript from the probability equations.

To find an upper bound on the probability of an adversary's success we use the same proof technique as for the XCBC\$-XOR scheme. That is, we (1) define several types of events on which we condition the adversary's success, (2) express the upper bound in terms of the conditional probabilities obtained, and (3) compute upper bounds on these probabilities. As before, our choice and number of conditioning events is motivated exclusively by the need to obtain a (good) upper bound for the probability of the adversary's success. Undoubtedly, other events could be used for deriving alternate upper bounds.

We provide some intuition for the choice of conditioning events defined, by giving the following examples of events that cause an adversary's success. (The reader can skip these examples without loss of continuity.)

Examples of Adversary's Success. A way for the adversary to find a forgery x' that passes the integrity check $w' = w$, is to look for collisions in the input of f, at forgery verification. The following three examples illustrate why such collisions cause an adversary's success. Other such examples, and other ways to find forgeries, exist.

Example 1 Collisions between inputs of f at forgery verification with those at message signing

Suppose that all inputs to f at forgery verification collide with inputs to f at signing. We pessimistically declare the adversary to be successful. For example, suppose that two of the block inputs to f at the verification of forgery (x', ctr', w') represent two swapped inputs to f at the signing of message x using counter ctr and obtaining the authentication tag w. Also suppose that all other inputs to f at forgery verification are the same as those of message x at signing. Hence, $x' \neq x$. In this case, the authentication check for forgery $(x', ctr' = ctr, w' = w)$ will pass the integrity check.

It should be noted that this criterion for adversary's success is pessimistic because, among the forgeries that make this event true some will decrypt correctly with negligible probability. For instance, if a forgery x' is a truncation of a signed message, the collision of the last forgery block $x'_{n'+1} = z'_0 + (n'+1) \times r'_0$ with any of the inputs to f or f' at message signing is a negligible-probability event and hence truncation would have a negligible chance of success (viz., Claim 1 below provides some intuition for this statement).

Example 2 Collisions among inputs of f at forgery verification

Suppose that two inputs of f obtained during forgery verification, x'_{n+1} and x'_{n+2}, do not collide with any of the inputs to f obtained during message signing, but collide with each other; $x'_{n+1} = x'_{n+2}$. Also suppose that the adversary's forgery (x', ctr', w') is obtained as follows: $x' = x||x'_{n+1}||x'_{n+2}$, $ctr' = ctr$, and $w' = w$. Clearly, $x' \neq x$ and the forgery (x', ctr', w') passes verification under the pessimistic assumption that $f(z_0 + (n+3) \times r_0) = f(z_0 + (n+1) \times r_0)$.

Example 3 Collisions among the inputs of f that cause discovery of r_0

Suppose that the forgery counter ctr'^i collides with an input to f, $x^p_k + k \times r^p_0, 1 \leq p \leq q_s, 1 \leq k \leq n_p$,

obtained during message signing, or with $x'^i_j + j \times r'^i_0, 1 \leq i \leq q_v, 1 \leq j \leq n'_i$, during the verification of forgery (x', ctr', w'). Suppose that the adversary finds that $x^p_k + k \times r^p_0 = ctr'^i$, for some message p, known plaintext block x^p_k and known counter ctr'^i, $1 \leq i \leq q_v$. Hence, the adversary can determine r^p_0 and thus the adversary's forgeries can satisfy collisions of Examples 1 and 2 above. A similar collision event between ctr'^i and an input to f during forgery verification has a similar effect.

Conditioning Events. To compute an upper bound on the probability of successful forgery, we choose three conditioning events based on collisions in the inputs of f. Intuition for the choice of events is provided by Examples 1 3 above. To define the conditioning events, we use the following notation for the last block that is enciphered

$$\begin{aligned} x^p_{n_p+1} &= z^p_0 \\ x'^i_{n'_i+1} &= z'^i_0. \end{aligned}$$

Next, we introduce the sets:

$$\begin{aligned} I^s &: \{ctr^1, \quad , ctr^{q_s}\} \\ S &: \{x^p_k + k \times r^p_0, 1 \leq p \leq q_s, 1 \leq k \leq n_p + 1\}, \\ V_i &: \{x'^i_s + s \times r'^s_0, x'^i_s + s \times r'^s_0 \notin (I^s \quad S), 1 \leq s \leq n'_i + 1\}, \end{aligned}$$

where I^s is the set of all the counters used at signing, S is the set of all the inputs to function f (aside from the counters) at signing, and V_i is the set of all the inputs to function f (aside from the counters) at verification of query i. Based on sets I^s, S, V_i, we introduce the following collision events that arise at the verification of forgery (x'^i, ctr'^i, w'^i):

$$C^i \quad : \quad V_i = \emptyset$$

Event C^i includes all instances when inputs of f at forgery verification (aside from counters) collide with either counters or inputs to function f at message signing. Next we define event D^i as follows:

$$\begin{aligned} D^i \quad : \quad & \exists s, 1 \leq s \leq n'_i + 1 : x'^i_s + s \times r'^i_0 \in V_i \\ & \text{and } x'^i_s + s \times r'^i_0 \neq x'^i_t + t \times r'^i_0, \forall x'^i_t + t \times r'^i_0 \in V_i, t \neq s, 1 \leq t \leq n'_i + 1 \\ & \text{and } x'^i_s + s \times r'^i_0 \neq ctr'^i \end{aligned}$$

Event D^i states that there is at least one input block of forgery i that does not collide with any other block and counter of forgery i. It is clear that the definition for D^i makes sense only when event C^i is false.

The rationale for introducing events C^i (or, actually, $\overline{C^i}$) and D^i is similar to the one used in the proof of Theorem 2. That is, we want to find a desirable event which states that there exists a forgery block that does not collide with any other input to f at either message signing or verification of forgery i (as suggested by Examples 1 and 2). Clearly, if this event is true, then the probability of verification passing is $1/2^l$. To find this event, however, we must ensure that all other collisions that that may lead to the discovery of r_0 are also ruled out for this block (as suggested by Example 3). For this reason, we must introduce two events beside $\overline{C^i}$ and D^i, namely events R^v_i and R^s defined below. (Note that these events need not cover the last block or a signed message or of forgery i since such a collision cannot be used to solve for either r'_0 or r_0 since random variables z'_0 and z_0 remain unknown to the adversary.) After we find the desired event for forgery i, we show that the complement of this event has a negligible probability (viz., the section on *Non-truncation Forgeries* below).

$$R^v_i \quad : \quad ctr'^i \neq x'^i_j + j \times r'^i_0, \forall j, 1 \leq j \leq n'_i$$

Event R_i^v states that all inputs to f during the verification of forgery i (aside from counters and last block) do not collide with forgery counters.

$$R^s \quad : \quad P^s \text{ and } P^v \text{ and } Q^s,$$

where

$$\begin{aligned} P^s &: \; ctr^a \neq x_k^p + k \times r_0^p, \forall a, p, k, 1 \leq a, p \leq q_s, 1 \leq k \leq n_p \\ P^v &: \; ctr'^a \neq x_k^p + k \times r_0^p, \forall a, p, k, 1 \leq a \leq q_v, 1 \leq p \leq q_s, 1 \leq k \leq n_p \\ Q^s &: \; x_j^p + j \times r_0^p \neq x_k^p + k \times r_0^p, \forall p, j, k, 1 \leq p \leq q_s, 1 \leq j, k \leq n_p, j \neq k \end{aligned}$$

and j is the index of a block in forgery i; i.e., $x_j'^i$. Event R^s states that all inputs to f at message signing (aside from counters and last block) do not collide with any other such inputs and with any of the counters used at message signing and forgery verification. Note that event R^s is independent of any forgery i.

Upper bound on the Probability of Successful Forgery. By standard conditioning,

$$Pr[\text{Succ}] \leq Pr[\text{Succ} \mid R^s] + Pr[\overline{R^s}] \leq Pr[\text{Succ} \mid R^s] + Pr[\overline{P^s}] + Pr[\overline{P^v}] + Pr[\overline{Q^s}],$$

since $\overline{R^s} = \overline{P^s}$ or $\overline{P^v}$ or $\overline{Q^s}$. The second, third and fourth terms in the sum are bounded as in the following Claim:

Claim 1
(a)

$$Pr[\overline{P^s}] \leq \frac{q_s \mu_s}{l 2^{l+1}} \quad \log_2 \frac{\mu_s}{l} + 3 \quad .$$

(b)

$$Pr[\overline{P^v}] \leq \frac{q_v \mu_s}{l 2^{l+1}} \quad \log_2 \frac{\mu_s}{l} + 3 \quad .$$

(c)

$$Pr[\overline{Q^s}] \quad \frac{1}{2^l} \frac{(\mu_s^2}{4l^2} (\log_2 \frac{\mu_s}{l} + 3).$$

To compute an upper bound for the probability of successful forgery, when event R^s is true, we note that the adversary is successful if one of his q_v forgeries is successful. Let the i-th adversary's forgery be: (ctr'^i, x'^i, w'^i), where $x'^i = x_1'^i \quad x_{n_i'}'^i$. Hence, by union bound, the probability of adversary's success for all q_v verification queries (when $f \overset{\mathcal{R}}{\leftarrow} R^{l,l}$) is:

$$Pr[\text{Succ} \mid R^s] \leq \sum_{i=1}^{q_v} Pr[w'^i = y_1'^i \oplus \quad \oplus y_{n_i'+1}'^i \mid R^s].$$

Hence, we first compute the probability of adversary's success when a single forgery verification is allowed; i.e., we compute $Pr[w'^i = y_1'^i \oplus \quad \oplus y_{n_i'+1}'^i \mid R^s]$. For this computation, we partition the space of all possible forgeries into (1) truncation and (2) non-truncation forgeries.

Truncation Forgeries. For truncation forgeries, we introduce the events:

$$\begin{aligned} Z_{I^s} &: \; z_0'^i + (n_i' + 1) \times r_0'^i \in I^s \\ Z_S &: \; z_0'^i + (n_i' + 1) \times r_0'^i \in S. \end{aligned}$$

Using these events, we show that the probability of adversary's success in creating a successful forgery i is negligible. If forgery i is a truncation, then there exists $p, 1 \leq p \leq q_s : ctr'^i = ctr^p$ and $x'^i_k = x^p_k, \forall k, 1 \leq k \leq n'_i \quad n_p$, hence $z'^i_0 = z^p_0$. If the input to f at block $n'_i + 1$, namely $z'^i_0 + (n'_i + 1) \times r'^i_0$, does not collide with any counter (i.e., event $\overline{Z_{I^s}}$ is true) and any input to function f (aside from the counters) at signing (i.e., event $\overline{Z_S}$ is true), then $y'^i_{n'_i+1} = f(z'^i_0 + (n'_i + 1) \times r'^i_0)$ is random, uniformly distributed and independent of any other block y' in the formula for w'^i. Hence, in this case, the probability of the event that $y'^i_1 \oplus \quad \oplus y'^i_{n'_i+1} = w'^i$ during the verification of forgery i is $1/2^l$. Summarizing, by standard conditioning and union bound,

$$\begin{aligned} Pr[w'^i = y'^i_1 \oplus \quad \oplus y'^i_{n'_i+1} \mid R^s] &\leq Pr[w'^i = y'^i_1 \oplus \quad \oplus y'^i_{n'_i+1} \mid (\overline{Z_{I^s} \text{ or } Z_S}) \text{ and } R^s] + Pr[Z_{I^s} \text{ or } Z_S \mid R^s] \\ &\leq \frac{1}{2^l} + Pr[Z_{I^s} \text{ or } Z_S] \leq \frac{1}{2^l} + Pr[Z_{I^s} \mid R^s] + Pr[Z_S \mid R^s]. \end{aligned}$$

Upper bounds for the probabilities of events $Z_{I^s} \mid R^s$ and $Z_S \mid R^s$ are given by the following Claim:

Claim 2
(a)

$$Pr[Z_{I^s} \mid R^s] \leq \frac{q_s}{2^l}.$$

(b)

$$Pr[Z_S \mid R^s] \leq \frac{\mu_s}{l2^l} + \frac{n_p}{2^l}.$$

Hence, for any truncation forgery,

$$Pr[w'^i = y'^i_1 \oplus \quad \oplus y'^i_{n'_i+1} \mid R^s] \leq \frac{1}{2^l} + \frac{q_s}{2^l} + \frac{\mu_s}{l2^l} + \frac{n_p}{2^l} \leq \frac{\mu_s}{l2^l} + \frac{q_s + (n_p + 1)}{2^l}.$$

Non-truncation Forgeries. Now, we find an upper bound for $Pr[w'^i = y'^i_1 \oplus \quad \oplus y'^i_{n'_i+1} \mid R^s]$ for non-truncation forgeries. To compute this upper bound, we define an event such that (1) the probability of successful forgery is $1/2^l$ when this event occurs, and (2) the probability of the complement of this event has a negligible upper bound.

Using the events defined above and by standard conditioning, we obtain:

$$\begin{aligned} Pr[w'^i = y'^i_1 \oplus \quad \oplus y'^i_{n'_i+1} \mid R^s] &\leq Pr[w'^i = y'^i_1 \oplus \quad \oplus y'^i_{n'_i+1} \mid \overline{C^i} \text{ and } D^i \text{ and } R^v_i \text{ and } R^s] + \\ &\quad Pr[C^i \text{ or } \overline{D^i} \text{ or } \overline{R^v_i} \mid R^s] \\ &\leq Pr[w'^i = y'^i_1 \oplus \quad \oplus y'^i_{n'_i+1} \mid \overline{C^i} \text{ and } D^i \text{ and } R^v_i \text{ and } R^s] + \\ &\quad Pr[C^i \text{ or } \overline{D^i} \text{ or } \overline{R^v_i} \mid R^v_i \text{ and } R^s] + Pr[\overline{R^v_i} \mid R^s] \\ &= Pr[w'^i = y'^i_1 \oplus \quad \oplus y'^i_{n'_i+1} \mid \overline{C^i} \text{ and } D^i \text{ and } R^v_i \text{ and } R^s] \\ &\quad + Pr[C^i \text{ or } \overline{D^i} \mid R^v_i \text{ and } R^s] + Pr[\overline{R^v_i} \mid R^s] \\ &\leq Pr[w'^i = y'^i_1 \oplus \quad \oplus y'^i_{n'_i+1} \mid \overline{C^i} \text{ and } D^i \text{ and } R^v_i \text{ and } R^s] + \\ &\quad Pr[C^i \text{ or } \overline{D^i} \mid \overline{C^i} \text{ and } R^v_i \text{ and } R^s] + Pr[C^i \mid R^v_i \text{ and } R^s] + Pr[\overline{R^v_i} \mid R^s] \\ &= Pr[w'^i = y'^i_1 \oplus \quad \oplus y'^i_{n'_i+1} \mid \overline{C^i} \text{ and } D^i \text{ and } R^v_i \text{ and } R^s] + \\ &\quad Pr[\overline{D^i} \mid \overline{C^i} \text{ and } R^v_i \text{ and } R^s] + Pr[C^i \mid R^v_i \text{ and } R^s] + Pr[\overline{R^v_i} \mid R^s], \end{aligned}$$

since the following events are equivalent:

$$\begin{aligned}(C^i \text{ or } \overline{D^i} \text{ or } \overline{R_i^v} \mid R_i^v \text{ and } R^s) &\equiv (C^i \text{ or } \overline{D^i} \mid R_i^v \text{ and } R^s)\\ (C^i \text{ or } \overline{D^i} \mid \overline{C^i} \text{ and } R_i^v \text{ and } R^s) &\equiv (\overline{D^i} \mid \overline{C^i} \text{ and } R_i^v \text{ and } R^s).\end{aligned}$$

Event $(\overline{C^i}$ and D^i and R_i^v and $R^s)$ is the desired event mentioned earlier in this proof. If this event happens, then there must exist an index $j, 1 \leq j \leq n_i' + 1$ such that $x_j'^i + j \times r_0'^i$ does not collide with any other input to f, at either message signing or verification of forgery i, and hence $y_j'^i = f(x_j'^i + j \times r_0'^i)$ is random, uniformly distributed and independent of any other terms in the expression $y_1'^i \oplus \quad \oplus y_{n_i'+1}'^i$. Hence, $y_1'^i \oplus \quad \oplus y_{n_i'+1}'^i$ is random and uniformly distributed and hence,

$$Pr[w'^i = y_1'^i \oplus \quad \oplus y_{n_i'+1}'^i \mid \overline{C^i} \text{ and } D^i \text{ and } R_i^v \text{ and } R^s] \leq \frac{1}{2^l}.$$

The other probabilities that appear in the expression for the total probability $Pr[w'^i = y_1'^i \oplus \quad \oplus y_{n_i'+1}'^i \mid R^s]$ are bounded as in Claim 3, whose proof can be found below:

Claim 3

(a)

$$Pr[\overline{R_i^v} \mid R^s] \leq \frac{n_i'}{2^{l+1}}(\log_2 n_i' + 3).$$

(b)

$$Pr[C^i \mid R_i^v \text{ and } R^s] \leq \frac{q_s n_i'}{2^l} + \frac{\mu_s}{l2^{l+1}} \quad \log_2 \frac{\mu_s}{l} + 3 \quad .$$

(c)

$$Pr[\overline{D^i} \mid \overline{C^i} \text{ and } R_i^v \text{ and } R^s] \leq \frac{n_i'}{2^{l+1}}(\log_2 n_i' + 3).$$

Based on this claim, for an arbitrary forgery i that is not a truncation, we obtain:

$$\begin{aligned}Pr[w'^i = y_1'^i \oplus \quad \oplus y_{n_i'+1}'^i \mid R^s] \quad \leq \quad & \frac{1}{2^l} + \frac{n_i'}{2^{l+1}}(\log_2 n_i' + 3) + \\ & \frac{q_s n_i'}{2^l} + \frac{\mu_s}{l2^{l+1}} \quad \log_2 \frac{\mu_s}{l} + 3 \quad + \frac{n_i'}{2^{l+1}}(\log_2 n_i' + 3) \\ & \frac{n_i'}{2^l}(\log_2 n_i' + 3) + \frac{q_s n_i'}{2^l} + \frac{\mu_s}{l2^{l+1}} \quad \log_2 \frac{\mu_s}{l} + 3 \quad .\end{aligned}$$

For any forgery, the upper bound is the maximum from the upper bounds for truncation and non-truncation forgeries, hence,

$$Pr[w'^i = y_1'^i \oplus \quad \oplus y_{n_i'+1}'^i \mid R^s] \quad \leq \quad \frac{n_i'}{2^l}(\log_2 n_i' + 3) + \frac{q_s n_i'}{2^l} + \frac{\mu_s}{l2^{l+1}} \quad \log_2 \frac{\mu_s}{l} + 3 \quad .$$

Hence, for all q_v verification queries, we obtain by union bound,

$$\begin{aligned}Pr[\text{Succ} \mid R^s] \quad &\leq \quad \sum_{i=1}^{q_v} Pr[w'^i = y_1'^i \oplus \quad \oplus y_{n_i'+1}'^i \mid R^s] \\ &\leq \quad \sum_{i=1}^{q_v} \frac{n_i'}{2^l}(\log_2 n_i' + 3) + \frac{q_s n_i'}{2^l} + \frac{\mu_s}{l2^{l+1}} \quad \log_2 \frac{\mu_s}{l} + 3 \\ &\leq \quad \frac{\mu_v}{l2^l}(\log_2 \frac{\mu_v}{l} + 3) + \frac{q_s \mu_v}{l2^l} + \frac{q_v \mu_s}{l2^{l+1}} \quad \log_2 \frac{\mu_s}{l} + 3 \\ &= \quad \frac{\mu_v}{l2^l}(\log_2 \frac{\mu_v}{l} + 3) + \frac{q_s \mu_v}{l2^l} + \frac{q_v \mu_s}{l2^{l+1}} \quad \log_2 \frac{\mu_s}{l} + 3 \quad .\end{aligned}$$

Hence, by Claim 1,

$$\begin{aligned} Pr[\text{Succ}] \quad \leq \quad & \frac{\mu_v}{l2^l}(\log_2 \frac{\mu_v}{l} + 3) + \frac{q_s \mu_v}{l2^l} + \frac{q_v \mu_s}{l2^{l+1}} \quad \log_2 \frac{\mu_s}{l} + 3 \quad + \\ & \frac{(q_s + q_v)\mu_s}{l2^{l+1}} \quad \log_2 \frac{\mu_s}{l} + 3 \quad + \frac{\mu_s^2}{l^2 2^{l+2}}(\log_2 \frac{\mu_s}{l} + 3). \end{aligned}$$

Finally, when $f \overset{\mathcal{R}}{\leftarrow} F$, the probability for adversary's success is bounded as follows:

$$\begin{aligned} Pr_{f \overset{\mathcal{R}}{\leftarrow} F}[\text{Succ}] \quad \leq \quad & \epsilon + \frac{\mu_v}{l2^l}(\log_2 \frac{\mu_v}{l} + 3) + \frac{q_s \mu_v}{l2^l} + \frac{q_v \mu_s}{l2^{l+1}} \quad \log_2 \frac{\mu_s}{l} + 3 \quad + \\ & \frac{(q_s + q_v)\mu_s}{l2^{l+1}} \quad \log_2 \frac{\mu_s}{l} + 3 \quad + \frac{\mu_s^2}{l^2 2^{l+2}}(\log_2 \frac{\mu_s}{l} + 3) \\ = \quad & \epsilon + \frac{\mu_v}{l2^l}(\log_2 \frac{\mu_v}{l} + 3) + \frac{q_s \mu_v}{l2^l} + \quad q_s + 2q_v + \frac{\mu_s}{2l} \quad \frac{\mu_s}{l2^{l+1}}(\log_2 \frac{\mu_s}{l} + 3). \end{aligned}$$

□

Proofs of Claims 1 – 3

For the proof of Claims 1 3 we use the following Fact, which is very similar to Fact 3 in Appendix A:

Fact 1

For any $N > 1$, let m be defined by $b \Leftrightarrow a = d \times 2^m$, where $1 \leq a \quad b \leq N \Leftrightarrow 1$ and d is odd. Then

$$\sum_{1 \leq a < b \leq N-1} 2^m \leq \frac{(N \Leftrightarrow 1)(N \Leftrightarrow 2)}{4}(\log_2(N \Leftrightarrow 2) + 3).$$

Fact 2

If for any $p, 1 \leq p \leq q_s, n_p > 0$, and if $\sum_{p=1}^{q_s}(n_p + 1) \leq \frac{\mu_s}{l}$, then,

$$\sum_{p=1}^{q_s}(n_p + 1)^2 \log_2(n_p + 1) \leq \frac{\mu_s^2}{l^2} \log_2 \frac{\mu_s}{l};$$

and, further, if $m = \max(n_p + 1)$, then

$$\sum_{p=1}^{q_s}(n_p + 1)^2 \log_2(n_p + 1) \leq \frac{\mu_s^2}{l^2} \log_2 m.$$

Proof of Claim 1

(a) Event $\overline{P^s}$ deals with collisions between inputs to f at signing, namely $x_k^p + k \times r_0^p, 1 \leq p \leq q_s, 1 \leq k \leq n_p$ and constant counters at signing, namely $ctr^a, 1 \leq a \leq q_s$. Since $\overline{P^s} \equiv \exists a, p, k, 1 \leq a, p \leq q_s, 1 \leq k \leq n_p$: $ctr^a = x_k^p + k \times r_0^p$, it follows by union bound that

$$Pr[\overline{P^s}] \leq \sum_{a=1}^{q_s} \sum_{p=1}^{q_s} \sum_{k=1}^{n_p} Pr[ctr^a = x_k^p + k \times r_0^p].$$

In this event, ctr^a and x_k^p are constants. Since r_0^p is random and uniformly distributed, and the event of interest can be written as $k \times r_0^p = ctr^a \Leftrightarrow x_k^p$, then, by Fact 2 (Appendix A),

$$Pr[ctr^a = x_k^p + k \times r_0^p] \leq \frac{2^m}{2^l},$$

where $k = d \times 2^m$ and d is odd. Hence, by Fact 3 (Appendix A) we have

$$\sum_{k=1}^{n_p} Pr[ctr^a = x_k^p + k \times r_0^p] \leq \frac{n_p}{2^{l+1}}(\log_2 n_p + 3).$$

Furthermore, by Fact 4 (Appendix A) we have

$$\sum_{p=1}^{q_s}\sum_{k=1}^{n_p} Pr[ctr^a = x_k^p + k \times r_0^p] \leq \sum_{p=1}^{q_s} \frac{n_p}{2^{l+1}}(\log_2 n_p + 3) \leq \frac{1}{2^{l+1}} \quad \frac{\mu_s}{l} \Leftrightarrow q_s \quad \log_2 \quad \frac{\mu_s}{l} \Leftrightarrow q_s + 3$$

since $\sum_{p=1}^{q_s}(n_p + 1) \leq \frac{\mu_s}{l}$, or, $\sum_{p=1}^{q_s} n_p \leq \frac{\mu_s}{l} \Leftrightarrow q_s$. Thus,

$$\begin{aligned}\sum_{a=1}^{q_s}\sum_{p=1}^{q_s}\sum_{k=1}^{n_p} Pr[ctr^a = x_k^p + k \times r_0^p] &\leq \sum_{a=1}^{q_s} \frac{1}{2^{l+1}} \quad \frac{\mu_s}{l} \Leftrightarrow q_s \quad \log_2 \quad \frac{\mu_s}{l} \Leftrightarrow q_s + 3 \\ &= \frac{1}{2^{l+1}} \quad \frac{q_s\mu_s}{l} \Leftrightarrow q_s^2 \quad \log_2 \quad \frac{\mu_s}{l} \Leftrightarrow q_s + 3 \quad .\end{aligned}$$

Hence,

$$Pr[\overline{P^s}] \leq \frac{1}{2^{l+1}} \quad \frac{q_s\mu_s}{l} \Leftrightarrow q_s^2 \quad \log_2 \quad \frac{\mu_s}{l} \Leftrightarrow q_s \quad + 3 \quad .$$

A simple (albeit higher) upper bound is then

$$Pr[\overline{P^s}] \leq \frac{1}{2^{l+1}} \frac{q_s\mu_s}{l} \quad \log_2 \frac{\mu_s}{l} + 3 \quad .$$

(b) Event $\overline{P^v}$ is very similar with event $\overline{P^s}$, i.e., it deals with collisions between inputs to f at signing, namely $x_k^p + k \times r_0^p, 1 \leq p \leq q_s, 1 \leq k \leq n_p$ and constant counters at verification, namely $ctr'^a, 1 \leq a \leq q_v$. In a manner similar to the one used in the proof of (a), since ctr'^a are also constants,

$$\begin{aligned}&Pr[\overline{P^v}] \\ &\leq \sum_{a=1}^{q_v}\sum_{p=1}^{q_s}\sum_{k=1}^{n_p} Pr[ctr'^a = x_k^p + k \times r_0^p] \leq \sum_{a=1}^{q_v} \frac{1}{2^{l+1}} \quad \frac{\mu_s}{l} \Leftrightarrow q_s \quad \log_2 \quad \frac{\mu_s}{l} \Leftrightarrow q_s \quad + 3 \\ &= \frac{1}{2^{l+1}} \quad \frac{q_v\mu_s}{l} \Leftrightarrow q_s q_v \quad \log_2 \quad \frac{\mu_s}{l} \Leftrightarrow q_s \quad + 3\end{aligned}$$

A simple (albeit higher) upper bound is then

$$Pr[\overline{P^v}] \leq \frac{1}{2^{l+1}} \frac{q_v\mu_s}{l} \quad \log_2 \frac{\mu_s}{l} + 3 \quad .$$

(c) Event $\overline{Q^s}$, deals with collisions between inputs to f at signing within the same message, namely $x_j^p + j \times r_0^p \neq x_k^p + k \times r_0^p$ where $1 \leq p \leq q_s, 1 \leq j, k \leq n_p, j \neq k$. Since $\overline{Q^s} \equiv \exists p, j, k, 1 \leq p \leq q_s, 1 \leq j, k \leq n_p, j \neq k : x_j^p + j \times r_0^p \neq x_k^p + k \times r_0^p$. Without loss of generality, let $k > j$. Then, by union bound,

$$Pr[\overline{Q^s}] \leq \sum_{p=1}^{q_s} \sum_{1 \quad j<k \quad n_p} Pr[x_j^p + j \times r_0^p = x_k^p + k \times r_0^p].$$

Event $x_j^p + j \times r_0^p = x_k^p + k \times r_0^p$ is equivalent to $(k \Leftrightarrow j) \times r_0^p = x_j^p \Leftrightarrow x_k^p$. Since r_0^p is random and uniformly distributed, by Fact 2 (Appendix A), this event happens with probability $\frac{2^m}{2^l}$ where $k \Leftrightarrow j = d \times 2^m$ and d is odd. Then, by Fact 1 (Appendix B), we have

$$\sum_{1 \quad j<k \quad n_p} Pr[x_j^p + j \times r_0^p = x_k^p + k \times r_0^p] \leq \sum_{1 \quad j<k \quad n_p} \frac{2^m}{2^l} \leq \frac{1}{2^l}\frac{n_p(n_p \Leftrightarrow 1)}{4}(\log_2(n_p \Leftrightarrow 1) + 3).$$

Furthermore,

$$\begin{aligned} Pr[\overline{Q^s}] &\leq \sum_{p=1}^{q_s} \sum_{j<k \ \ n_p} Pr[x_j^p + j \times r_0^p = x_k^p + k \times r_0^p] \\ &\leq \sum_{p=1}^{q_s} \frac{1}{2^l} \frac{n_p(n_p \Leftrightarrow 1)}{4} (\log_2(n_p \Leftrightarrow 1) + 3) \quad \sum_{p=1}^{q_s} \frac{1}{2^l} \frac{(n_p+1)^2}{4} (\log_2(n_p+1) + 3), \end{aligned}$$

and using Fact 2 (Appendix B), we have

$$Pr[\overline{Q^s}] \quad \frac{1}{2^l} \sum_{p=1}^{q_s} \frac{(n_p+1)^2}{4} (\log_2(n_p+1) + 3) \leq \frac{1}{2^l} \frac{(\mu_s^2}{4l^2} (\log_2 \frac{\mu_s}{l} + 3).$$

□

Proof of Claim 2

(a) Event Z_{I^s} refers to collisions between the last input to f at verification of forgery i, namely $z_0'^i + (n_i' + 1) \times r_0'^i$, and any counter at signing, namely $ctr^a, 1 \leq a \leq q_s$. By union bound,

$$Pr[Z_{I^s} \mid R^s] \leq \sum_{a=1}^{q_s} Pr[z_0'^i + (n_i' + 1) \times r_0'^i = ctr^a \mid R^s].$$

$z_0'^i = z_0^p = f'(r_0^p)$ is random, uniformly distributed and independent of r_0^i and of the counter since it is obtained by enciphering with a different key. Hence, since ctr^a is a constant,

$$Pr[z_0'^i + (n_i' + 1) \times r_0'^i = ctr^a \mid R^s] = \frac{1}{2^l}$$

and

$$Pr[Z_{I^s} \mid R^s] \leq \frac{q_s}{2^l}.$$

□

(b) Event Z_S refers to collisions between the last input to f at verification of forgery i, namely $z_0'^i + (n_i' + 1) \times r_0'^i$, and any input to f at signing (other than counters), i.e., $x_b^a + b \times r_0^a, 1 \leq a \leq q_s, 1 \leq b \leq n_a + 1$. By union bound,

$$Pr[Z_S \mid R^s] \leq \sum_{a=1}^{q_s} \sum_{b=1}^{n_a+1} Pr[z_0'^i + (n_i' + 1) \times r_0'^i = x_b^a + b \times r_0^a \mid R^s].$$

If $b \leq n_a$, then x_b^a is a constant in the equation $z_0'^i + (n_i' + 1) \times r_0'^i = x_b^a + b \times r_0^a$. Then, since $z_0'^i = z_0^p$ is obtained using a different key, $z_0'^i$ is random, uniformly distributed and independent of $r_0'^i = r_0^p, r_0^a$ and of the constant x_b^a. Hence,

$$Pr[z_0'^i + (n_i' + 1) \times r_0'^i = x_b^a + b \times r_0^a \mid R^s] = \frac{1}{2^l}.$$

If $b = n_a + 1$, then $x_b^a = z_0^a$. In this case, if $p \neq a$, then $z_0'^i = z_0^p$ and z_0^a are random, uniformly distributed and independent; they are also independent of $r_0'^i = r_0^p$ and r_0^a. Hence,

$$Pr[z_0'^i + (n_i' + 1) \times r_0'^i = x_b^a + b \times r_0^a \mid R^s] = \frac{1}{2^l}.$$

In the complementary case, namely when $b = n_a + 1, p = a$, then $z_0'^i = z_0^p = z_0^a = x_b^a$ and $r_0'^i = r_0^p = r_0^a$. Since, in this case, $b = n_a + 1 = n_p + 1$, it follows that

$$z_0'^i + (n_i' + 1) \times r_0'^i = x_b^a + b \times r_0^a \;\Leftrightarrow\; (n_p \Leftrightarrow n_i') \times r_0^p = 0,$$

where, $n_p > n_i'$ (since the forgery is a truncation of message p). Event R^s is true, hence r_0^p is unknown, random and uniformly distributed. Hence, by Fact 2 (Appendix A), the probability of this event is $\frac{2^m}{2^l}$ where $n_p \Leftrightarrow n_i' = d \times 2^m$ and d is odd. Hence, $2^m \leq n_p \Leftrightarrow n_i' \leq n_p$. Hence,

$$Pr[z_0'^i + (n_i' + 1) \times r_0'^i = x_b^a + b \times r_0^a \mid R^s] = Pr[(n_p \Leftrightarrow n_i') \times r_0^p = 0 \mid R^s] \leq \frac{2^m}{2^l} \leq \frac{n_p}{2^l}.$$

Hence,

$$\begin{aligned}
Pr[Z_S \mid R^s] &\leq \sum_{a=1}^{q_s} \sum_{b=1}^{n_a+1} Pr[z_0'^i + (n_i' + 1) \times r_0'^i = x_b^a + b \times r_0^a \mid R^s] \\
&= \sum_{a=1}^{q_s} \sum_{b=1}^{n_a} Pr[z_0'^i + (n_i' + 1) \times r_0'^i = x_b^a + b \times r_0^a \mid R^s] + \\
&\quad \sum_{a=1, a\neq p}^{q_s} Pr[z_0'^i + (n_i' + 1) \times r_0'^i = z_0^a + (n_a + 1) \times r_0^a \mid R^s] + \\
&\quad Pr[z_0'^i + (n_i' + 1) \times r_0'^i = z_0^p + (n_p + 1) \times r_0^a \mid R^s] \\
&\leq \sum_{a=1}^{q_s} \sum_{b=1}^{n_a} \frac{1}{2^l} + \sum_{a=1, a\neq p}^{q_s} \frac{1}{2^l} + \frac{n_p}{2^l} \quad \sum_{a=1}^{q_s} \sum_{b=1}^{n_a+1} \frac{1}{2^l} + \frac{n_p}{2^l} \leq \frac{\mu_s}{l2^l} + \frac{n_p}{2^l}.
\end{aligned}$$

□

Proof of Claim 3

(a) Event R_i^v deals with collisions between inputs to f at verification of forgery i and the counter corresponding to forgery i. Hence, in a manner similar to the one used in the Proof of Claim 1(a)

$$Pr[\overline{R_i^v} \mid R^s] \leq \sum_{j=1}^{n_i'} Pr[ctr'^i = x_j'^i + j \times r_0'^i \mid R^s] \leq \sum_{j=1}^{n_i'} \frac{2^m}{2^l} \leq \frac{n_i'}{2^{l+1}}(\log_2 n_i' + 3).$$

□

(b) The proof of this Claim is very similar to the proof of Claim 3 in the Proof of Theorem 2. First, we choose an index j such that for any type of possible non-truncation forgery i, the input to f at the verification of forgery i, namely $x_j'^i + j \times r_0'^i$, does collide with any input to f during message signing with low probability. Next, we compute an upper bound for these collisions.

All non-truncation forgeries can be partitioned in a similar manner as that used in the proof of Claim 3 of Theorem 2. That is, we define extensions of the plaintext of a signed message, which we call the *prefix* case, and the complementary case, which we call *non-prefix* case. The non-prefix case includes two separate subcases, namely when ctr'^i is different from any ctr^p of any message p obtained at signing (i.e., message (x^p, ctr^p, w^p)), or when there is a signed message p such that $ctr'^i = ctr^p$. Hence, in the latter subcase, there must be at least a block position j in the forged message x'^i that is different from the corresponding block of the signed message p. This partition of all possible forgery types shows that a forged message $x'^i = x_1'^i \quad x_{n_i'}'^i$ which is not a truncation, can be in one of the following three complementary types:

(a) $\exists p, 1 \leq p \leq q_s : n'_i > n_p, ctr'^i = ctr^p$ and $\forall k, 1 \leq k \leq n_p : x'^i_k = x^p_k$; i.e., the forged message is an extension of message x^p (the prefix case). The non-prefix case consists of the following two forgery types:
(b1) $ctr'^i \neq ctr^p, \forall p, 1 \leq p \leq q_s$; and
(b2) $\exists p, 1 \leq p \leq q_s : ctr'^i = ctr^p, \exists k, 1 \leq k \leq \min(n'_i, n_p) : x'^i_k \neq x^p_k$; i.e., the forged message is obtained by modifying a queried message starting with some block between the second and last block.

Now we choose index j mentioned above for each type of possible non-truncation forgeries, as follows: for forgeries of type (a), $j = n_p + 1$; for forgeries of type (b1), $j = 1$; and for forgeries of type (b2), j is the smallest index such that $x'^i_j \neq x^p_j, 1 \leq j \leq \min\{n_p, n'_i\}$. In all cases $1 \leq j \leq n'_i$, and hence, the chosen block x'^i_j is well defined.

Event C^i implies that $x'^i_j + j \times r'^i_0 \in I^s$ or $x'^i_j + j \times r'^i_0 \in S$. Hence, by union bound

$$Pr[C^i \mid R^v_i \text{ and } R^s] \leq Pr[x'^i_j + j \times r'^i_0 \in I^s \mid R^v_i \text{ and } R^s] + Pr[x'^i_j + j \times r'^i_0 \in S \mid R^v_i \text{ and } R^s].$$

Let us define the following events:

$$\begin{aligned} E_{I^s} &: x'^i_j + j \times r'^i_0 \in I^s \\ E_S &: x'^i_j + j \times r'^i_0 \in S. \end{aligned}$$

Hence,

$$Pr[C^i \mid R^v_i \text{ and } R^s] \leq Pr[E_{I^s} \mid R^v_i \text{ and } R^s] + Pr[E_S \mid R^v_i \text{ and } R^s].$$

We determine upper bounds for events $E_{I^s} \mid \overline{R^v_i}, E_S \mid \overline{R^v_i}$ using the following Claim, whose proof is found at the end of this appendix:

Claim 3.1
(a)

$$Pr[E_{I^s} \mid R^v_i \text{ and } R^s] \leq \frac{q_s n'_i}{2^l}.$$

(b)

$$Pr[E_S \mid R^v_i \text{ and } R^s] \leq \frac{\mu_s}{l 2^{l+1}} \quad \log_2 \frac{\mu_s}{l} + 3 \quad .$$

Based on Claim 3.1,

$$Pr[C^i \mid R^v_i \text{ and } R^s] \leq Pr[E_{I^s} \mid R^v_i \text{ and } R^s] + Pr[E_S \mid R^v_i \text{ and } R^s] \leq \frac{q_s n'_i}{2^l} + \frac{\mu_s}{l 2^{l+1}} \quad \log_2 \frac{\mu_s}{l} + 3 \quad .$$

□

(c) We find an upper bound for $Pr[\overline{D^i} \mid \overline{C^i} \text{ and } R^v_i \text{ and } R^s]$ in a manner very similar to the one used in Claim 4 of the Proof of Theorem 2.

Event $\overline{C^i}$ implies that there is at least one element $x'^i_s + s \times r'^i_0 \in V_i$. Event $\overline{D^i}$ is true if and only if for any index $s, 1 \leq s \leq n'_i + 1$, the block $x'^i_s + s \times r'^i_0 \in V_i$ collides with another block $x'^i_t + t \times r'^i_0 \in V_i, 1 \leq t \leq n'_i + 1, s \neq t$, or with ctr'^i. But the latter collisions, namely $x'^i_s + s \times r'^i_0 = ctr'^i$, where $x'^i_s + s \times r'^i_0 \in V_i$, is already precluded by event R^v_i. For the former collisions, let s be the smallest index of the element $x'^i_s + s \times r'^i_0 \in V_i$. Hence, event $\overline{D^i}$ implies that $x'^i_s + s \times r'^i_0 \in V_i \Leftrightarrow \{x'^i_s + s \times r'^i_0\}$, and

$$Pr[\overline{D^i} \mid \overline{C^i} \text{ and } R^v_i \text{ and } R^s] \leq Pr[x'^i_s + s \times r'^i_0 \in V_i \Leftrightarrow \{x'^i_s + s \times r'^i_0\} \mid \overline{C^i} \text{ and } R^v_i \text{ and } R^s].$$

Furthermore, by union bound we have

$$Pr[\overline{D^i} \mid \overline{C^i} \text{ and } R_i^v \text{ and } R^s] \leq \sum_{t>s, x_t'^i + t \times r_0'^i \in V_i} Pr[x_s'^i + s \times r_0'^i = x_t'^i + t \times r_0'^i \mid \overline{C^i} \text{ and } R_i^v \text{ and } R^s].$$

In this expression, $r_0'^i$ is unknown, random and uniformly distributed since events R_i^v and R^s are true. Furthermore, $x_s'^i, x_t'^i$ are constants, or $x_s'^i$ is a constant and $x_t'^i = z_0'^i$, since $t > s$; if $x_t'^i = z_0'^i$, then $x_t'^i$ is independent of $r_0'^i$ because $z_0'^i$ was obtained by enciphering with a different key. Hence, by Fact 2 (Appendix A), the probability is at most $\frac{2^m}{2^l}$, where $t \Leftrightarrow s = d \times 2^m$ and d is odd. Hence,

$$Pr[x_s'^i + s \times r_0'^i = x_t'^i + t \times r_0'^i \mid \overline{C^i} \text{ and } R_i^v \text{ and } R^s] \leq \frac{2^m}{2^l}.$$

Furthermore, proceeding in the same manner as for Claim 4 in the proof of Theorem 2 (viz., Appendix A) we have

$$\sum_{t>s, x_t'^i + t \times r_0'^i \in V_i} Pr[x_s'^i + s \times r_0'^i = x_t'^i + t \times r_0'^i \mid \overline{C^i} \text{ and } R_i^v \text{ and } R^s] \leq \frac{n_i'}{2^{l+1}}(\log_2 n_i' + 3),$$

and hence,

$$Pr[\overline{D^i} \mid \overline{C^i} \text{ and } R_i^v \text{ and } R^s] \leq \frac{n_i'}{2^{l+1}}(\log_2 n_i' + 3).$$

□

Proof of Claim 3.1

(a) Event E_{I^s} refers to collisions between the chosen block $x_j'^i + j \times r_0'^i$ and counters at signing, namely $ctr^p, 1 \leq p \leq q_s$. Hence, by union bound, and Fact 2 (Appendix A)

$$Pr[E_{I^s} \mid R_i^v \text{ and } R^s] \leq \sum_{p=1}^{q_s} Pr[x_j'^i + j \times r_0'^i = ctr^p \mid R_i^v \text{ and } R^s] \leq \sum_{p=1}^{q_s} \frac{2^m}{2^l} = \frac{q_s 2^m}{2^l},$$

where $j = d \times 2^m$ and d is odd, since, by events R_i^v and R^s $r_0'^i$ is unknown, random and uniformly distributed, $x_j'^i$ is a constant, and ctr^p is a constant. Furthermore, since $2^m \leq j \leq n_i'$, it follows that

$$Pr[E_{I^s} \mid R_i^v \text{ and } R^s] \leq \frac{q_s j}{2^l} \leq \frac{q_s n_i'}{2^l}.$$

□

(b) Event E_{I^s} refers to collisions between the chosen block $x_j'^i + j \times r_0'^i$ and inputs to f at signing other than counters, namely blocks $x_k^p + k \times r_0^p, 1 \leq p \leq q_s, 1 \leq k \leq n_p + 1$. Hence, by union bound,

$$Pr[E_S \mid R_i^v \text{ and } R^s] \leq \sum_{p=1}^{q_s} \sum_{k=1}^{n_p} Pr[x_j'^i + j \times r_0'^i = x_k^p + k \times r_0^p \mid R_i^v \text{ and } R^s]$$

In a manner similar to the one used for Claim 3 (Part 2) in the proof of Theorem 2, we can show that

$$Pr[E_S \mid R_i^v \text{ and } R^s] \leq \frac{\mu_s}{l 2^{l+1}} \quad \log_2 \frac{\mu_s}{l} + 3 \quad .$$

□

Proof of Fact 1
We will use Fact 3 from Appendix A, but first we rewrite the sum as :

$$\sum_{1 \quad a<b \quad N-1} 2^m = \sum_{a=1}^{N-2} \sum_{b=a+1}^{N-1} 2^m \sum_{a=1}^{N-2} \sum_{c=1}^{N-1-a} 2^m,$$

where $c \stackrel{def}{=} b \Leftrightarrow a$. By Fact 3 from Appendix A, we have

$$\sum_{c=1}^{N-1-a} 2^m \leq \frac{N \Leftrightarrow 1 \Leftrightarrow a}{2} (\log_2(N \Leftrightarrow 1 \Leftrightarrow a) + 3).$$

Hence,

$$\sum_{1 \quad a<b \quad N-1} 2^m \leq \sum_{a=1}^{N-2} \frac{N \Leftrightarrow 1 \Leftrightarrow a}{2} (\log_2(N \Leftrightarrow 1 \Leftrightarrow a) + 3) = \sum_{e=1}^{N-2} \frac{e}{2} (\log_2 e + 3),$$

where the index $e \stackrel{def}{=} N \Leftrightarrow 1 \Leftrightarrow a$. Furthermore, since $e \leq N \Leftrightarrow 2$, we have

$$\sum_{1 \quad a<b \quad N-1} 2^m \leq \sum_{e=1}^{N-2} \frac{e}{2} (\log_2 e + 3) \leq \sum_{e=1}^{N-2} \frac{e}{2} (\log_2(N \Leftrightarrow 2) + 3) \leq \frac{(N \Leftrightarrow 1)(N \Leftrightarrow 2)}{4} (\log_2(N \Leftrightarrow 2) + 3).$$

□

Proof of Fact 2
Since $n_p + 1 \leq \frac{\mu_s}{l}$ and $\sum_{p=1}^{q_s} (n_p + 1)^2 \leq \frac{\mu_s^2}{l^2}$, it follows that

$$\sum_{p=1}^{q_s} (n_p + 1)^2 \log_2(n_p + 1) \leq \sum_{p=1}^{q_s} (n_p + 1)^2 \log_2 \frac{\mu_s}{l} \leq \frac{\mu_s^2}{l^2} \log_2 \frac{\mu_s}{l}.$$

□

Appendix C – Proof [Security of stateful XEBC-MAC (XECBS-MAC) in an Adaptive Chosen-Message Attack]

Throughout this proof, we use the same notation as in the Proof for Security of the XCBC\$-XOR in a Message-Integrity Attack, Appendix A, and the same facts (i.e., Facts 1 4). Unless mentioned otherwise, we focus on the probability for adversary's success when $f \overset{\mathcal{R}}{\leftarrow} R^{l,l}$, and, for simplicity, we will drop the $f \overset{\mathcal{R}}{\leftarrow} R^{l,l}$ subscript from the probability equations.

Notation. Let $z_k^p, 1 \leq p \leq q_s, 1 \leq k \leq n_p$ be the *hidden inputs* of function f at the signing of message p; i.e., for signed message $x^p = x_1^p \quad x_{n_p}^p$, we have

$$z_k^p = x_k^p + p \times R + k \times R^*.$$

Let $z_j'^i, 1 \leq i \leq q_v, 1 \leq j \leq n_i'$ be the *hidden inputs* to function f at the verification of forgery i; i.e., for the forgery $x'^i = x_1'^i \quad x_{n_i'}'^i$ using the message identifier (ID) s'^i ($s'^i \leq q_s$), we have

$$z_j'^i = x_j'^i + s'^i \times R + j \times R^*.$$

To find an upper bound on the probability of an adversary's success we use the same proof technique as for the XCBC\$-XOR scheme. That is, we (1) define several types of events on which we condition the adversary's success, (2) express the upper bound in terms of the conditional probabilities obtained, and (3) compute upper bounds on these probabilities.

We provide some intuition for the choice of conditioning events defined, by giving the following examples of events that cause an adversary's success. (The reader can skip these examples without loss of continuity.)

Examples of Adversary's Success. A way for the adversary to find a forgery x' that passes the integrity check $w' = w$, is to look for collisions in the input of f, at forgery verification. The following three examples illustrate why such collisions cause an adversary's success. Other such examples, and other ways to find forgeries, exist.

Example 1 Collisions between inputs of f at forgery verification with those at message signing

Suppose that all inputs of f at forgery verification collide with inputs of f at signing. We pessimistically declare the adversary to be successful. For example, suppose that two of the block inputs of f at the verification of forgery $(x' \neq x, s', w')$ represent two swapped inputs of f at the signing of message x using message ID s' and obtaining the authentication tag w. Also suppose that all other inputs of f at forgery verification are the same as those of message x at signing. In this case, the authentication check for forgery $(x', s', w' = w)$ will pass the integrity check.

It should be noted that this criterion for adversary's success is pessimistic because, among the forgeries that make this event true some will decrypt correctly with negligible probability. For instance, if a forgery x' is a truncation of a signed message and the message ID s'^i is equal to the identifier of the signed message, then, despite collisions between the inputs of f at forgery verification with inputs of f at signing, the truncation forgery has only negligible chance of success (viz., Claim 1 below provides some intuition for this statement).

Example 2 Collisions among inputs of f at forgery verification

Suppose that two (hidden) inputs of f obtained during forgery verification, namely $z_1' = x_1' + s' \times R + R^*$ and $z_2' = x_2' + s' \times R + 2 \times R^*$, for forgery $x' = x_1' x_2'$ using message ID s', do not collide with any of the

inputs of f obtained during signing of any message x but collide with each other; also assume that $x' \neq x$. Then the forgery $(x', s', w' = 0)$ passes verification.

Example 3 Collisions among the inputs of f that cause discovery of R or R^*

Suppose that, at message signing, two (hidden) inputs of function f collide; i.e., $z_k^p = z_t^s, 1 \leq p, s \leq q_s, 1 \leq k \leq n_p, 1 \leq t \leq n_s$, where $(p, k) \neq (s, t)$. This can lead to the discovery of some, and possibly all, of the bits of R or R^*. For example, suppose that $x_1^p + p \times R + R^* = x_2^p + p \times R + 2 \times R^*$, or $R^* = x_1^p \Leftrightarrow x_2^p$. Knowing R^*, an adversary can choose i and the forgery $x' = x_1' x_2'$ with message ID s' such that $x_1' + s' \times R + R^* = x_2' + s' \times R + 2 \times R^*$, i.e., $x_2' \Leftrightarrow x_1' = R^* = x_2 \Leftrightarrow x_1$. Then the adversary can let the tag $w' = 0$. Similar examples which illustrate collisions that pessimistically lead to the discovery of R can be found; e.g., collision $x_1^p + p \times R + R^* = x_1^r + r \times R + R^*$, where $p \neq r$. (R^* is completely determined if $p \Leftrightarrow r$ is odd.)

Conditioning Events. To compute an upper bound on the probability of successful forgery, we choose three conditioning events based on collisions in the inputs of f. Intuition for the choice of events is provided by Examples 1 3 above. We introduce the sets:

$$\begin{aligned} S &: \{z_k^p, 1 \leq p \leq q_s, 1 \leq k \leq n_p\}, \\ V_i &: \{z_j'^i, z_j'^i \notin S, 1 \leq j \leq n_i'\}, \end{aligned}$$

where S is the set of all the inputs of function f at signing, and V_i is the set of all the inputs of function f at verification of query i. Based on sets S and V_i, we introduce the following collision events that arise at the verification of forgery (x'^i, s'^i, w'^i):

$$C^i \quad : \quad V_i = \emptyset.$$

Event C^i includes all instances when inputs of f at forgery verification collide with inputs of function f at message signing. Next we define event D^i as follows:

$$\begin{aligned} D^i \quad : \quad & \exists j, 1 \leq j \leq n_i' : z_j'^i \in V_i \\ & \text{and } z_j'^i \neq z_m'^i, \forall z_m'^i \in V_i, j \neq m, 1 \leq m \leq n_i'. \end{aligned}$$

Event D^i states that there is at least one "new" input block of forgery i that does not collide with any other "new" block of forgery i, where here "new" input blocks refers to input blocks that are not in the set of input blocks at signing, namely S. It is clear that the definition for D^i makes sense only when event C^i is false.

The rationale for introducing events C^i (or, actually, $\overline{C^i}$) and D^i is similar to the one used in the proof of Theorem 2 (Appendix A). That is, we want to find a desirable event which states that there exists a forgery block that does not collide with any other input to f at either message signing or verification of forgery i (as suggested by Examples 1 and 2). Clearly, if this event is true, then the probability of verification passing is $1/2^l$. To find this event, however, we must ensure that all other collisions that that may lead to the discovery of R or R^*, are also ruled out for this block (as suggested by Example 3). For this reason, we introduce event R^s defined below.

$$R^s \quad : \quad z_k^p \neq z_t^s, 1 \leq p, s \leq q_s, 1 \leq k \leq n_p, 1 \leq t \leq n_s, (p, k) \neq (s, t)$$

Event R^s states that the set S is collision-free. Note that event R^s is independent of any forgery i.

Upper bound on the Probability of Successful Forgery. By standard conditioning, we have

$$Pr[\text{Succ}] \leq Pr[\text{Succ} \mid R^s] + Pr[\overline{R^s}].$$

The second term in the sum is bounded as in the following Claim:

Claim 1

$$Pr[\overline{R^s}] \leq \frac{q_s \mu_s}{l 2^{l+1}}(\log_2 q_s + 3) + \frac{\mu_s^2}{l^2 2^{l+1}}(\log_2 \frac{\mu_s}{l} + 3).$$

To compute an upper bound for the probability of successful forgery, when event R^s is true, we note that the adversary is successful if one of his q_v forgeries (x'^i, s'^i, w'^i) is successful, where $x'^i = x'^i_1 \quad x'^i_{n'_i}$. Hence, by union bound, the probability of adversary's success for all q_v verification queries (when $f \xleftarrow{\mathcal{R}} R^{l,l}$) is:

$$Pr[\text{Succ} \mid R^s] \leq \sum_{i=1}^{q_v} Pr[w'^i = y'^i_1 \oplus \quad \oplus y'^i_{n'_i} \mid R^s].$$

Hence, we first compute the probability of adversary's success when a single forgery verification is allowed; i.e., we compute $Pr[w'^i = y'^i_1 \oplus \quad \oplus y'^i_{n'_i} \mid R^s]$. For this computation, we partition the space of all possible forgeries into (1) truncation and (2) non-truncation forgeries.

Truncation Forgeries. We call truncation a forgery $x'^i = x'^i_1 \quad x'^i_{n'_i}$ together with a value of s'^i such that there exists a signed message $x^p = x^p_1 \quad x^p_{n_p}$ such that $s'^i = p$ and $x'^i_k = x^p_k, \forall k, 1 \leq k \leq n'_i \quad {}_p$.

In this case, for any $1 \leq j \leq n'_i$ we have:

$$z'^i_j = z^p_j, \forall j, 1 \leq j \leq n'_i,$$

and thus

$$y'^i_j = f(z'^i_j) = f(z^p_j) = y^p_j,$$

and the computed tag becomes

$$y'^i_1 \oplus \quad \oplus y'^i_{n'_i} = y^p_1 \oplus \quad \oplus y^p_{n'_i} = w^p \oplus y^p_{n'_i+1} \oplus \quad \oplus y^p_{n_p}.$$

where the exclusive-or sum $y^p_{n'_i+1} \oplus \quad \oplus y^p_{n_p}$ contains at least one term since $n'_i \quad {}_p$.

$$\begin{aligned} Pr[w'^i = y'^i_1 \oplus \quad \oplus y'^i_{n'_i} \mid R^s] \;=\; & Pr[w'^i = w^p \oplus y^p_{n'_i+1} \oplus \quad \oplus y^p_{n_p} \mid R^s] = \\ & Pr[y^p_{n'_i+1} \oplus \quad \oplus y^p_{n_p} = w'^i \oplus w^p \mid R^s]. \end{aligned}$$

In this expression, when there are no collisions in the inputs of f at signing, the values $y^p_{n'_i+1}, \quad , y^p_{n_p}$ are random, uniformly distributed and mutually independent. Since $n_p > n'_i$ there is at least one of these values. These values appear only in the signing of message p and the tag w^p contains other outputs of function f, namely $y^p_1, \quad , y^p_{n'_i}$ which, due to event R^s being true, are also random, uniformly distributed, mutually independent and independent of all the other outputs of function f at signing. (Intuitively, we show that the exclusive-or sum $y^p_{n'_i+1} \oplus \quad \oplus y^p_{n_p}$ is random, uniformly distributed and unknown.) Hence, the exclusive-or sum $y^p_{n'_i+1} \oplus \quad \oplus y^p_{n_p}$ is random and uniformly distributed, and hence

$$Pr[w'^i = y'^i_1 \oplus \quad \oplus y'^i_{n'_i} \mid R^s] = Pr[y^p_{n'_i+1} \oplus \quad \oplus y^p_{n_p} = w'^i \oplus w^p \mid R^s] = \frac{1}{2^l}.$$

Non-Truncation Forgeries. Now, we find an upper bound for $Pr[w'^i = y_1'^i \oplus \quad \oplus y_{n_i'}'^i \mid R^s]$ for non-truncation forgeries. To compute this upper bound, we define an event such that (1) the probability of successful forgery is $1/2^l$ when this event occurs, and (2) the probability of the complement of this event has a negligible upper bound.

Using the events defined above and by standard conditioning, we obtain:

$$\begin{aligned} Pr[w'^i = y_1'^i \oplus \quad \oplus y_{n_i'}'^i \mid R^s] &\leq Pr[w'^i = y_1'^i \oplus \quad \oplus y_{n_i'}'^i \mid \overline{C^i} \text{ and } D^i \text{ and } R^s] + \\ &\quad Pr[C^i \text{ or } \overline{D^i} \mid R^s] \\ &\leq Pr[w'^i = y_1'^i \oplus \quad \oplus y_{n_i'}'^i \mid \overline{C^i} \text{ and } D^i \text{ and } R^s] + \\ &\quad Pr[C^i \text{ or } \overline{D^i} \mid \overline{C^i} \text{ and } R^s] + Pr[C^i \mid R^s] \\ &= Pr[w'^i = y_1'^i \oplus \quad \oplus y_{n_i'}'^i \mid \overline{C^i} \text{ and } D^i \text{ and } R^s] + \\ &\quad Pr[\overline{D^i} \mid \overline{C^i} \text{ and } R^s] + Pr[C^i \mid R^s], \end{aligned}$$

since the following events are equivalent:

$$(C^i \text{ or } \overline{D^i} \mid \overline{C^i} \text{ and } R^s) \equiv (\overline{D^i} \mid \overline{C^i} \text{ and } R^s).$$

Event $(\overline{C^i}$ and D^i and $R^s)$ is the desired event mentioned earlier in this proof. If this event happens, then there must exist an index $j, 1 \leq j \leq n_i'$ such that $z_j'^i$ does not collide with any other input to f, at either message signing or verification of forgery i, and hence $y_j'^i = f(z_j'^i)$ is random, uniformly distributed and independent of any other terms in the expression $y_1'^i \oplus \quad \oplus y_{n_i'}'^i$. Hence, $y_1'^i \oplus \quad \oplus y_{n_i'}'^i$ is random and uniformly distributed and hence,

$$Pr[w'^i = y_1'^i \oplus \quad \oplus y_{n_i'}'^i \mid \overline{C^i} \text{ and } D^i \text{ and } R^s] = \frac{1}{2^l}.$$

The other probabilities that appear in the expression for the total probability $Pr[w'^i = y_1'^i \oplus \cdot \quad \oplus y_{n_i'}'^i \mid R^s]$ are bounded as in Claim 2, whose proof can be found below:

Claim 2

(a)

$$Pr[C^i \mid R^s] \leq \frac{q_s}{2^{l+1}}(\log_2 q_s + 3) + \frac{\mu_s}{l2^{l+1}}(\log_2 \frac{\mu_s}{l} + 3).$$

(b)

$$Pr[\overline{D^i} \mid \overline{C^i} \text{ and } R^s] \leq \frac{n_i'}{2^{l+1}}(\log_2 n_i' + 3).$$

Based on this claim, for an arbitrary forgery i that is not a truncation, we obtain:

$$Pr[w'^i = y_1'^i \oplus \quad \oplus y_{n_i'}'^i \mid R^s] \leq \frac{1}{2^l} + \frac{q_s}{2^{l+1}}(\log_2 q_s + 3) + \frac{\mu_s}{l2^{l+1}}(\log_2 \frac{\mu_s}{l} + 3) + \frac{n_i'}{2^{l+1}}(\log_2 n_i' + 3).$$

For any forgery, the upper bound is the maximum from the upper bounds for truncation and non-truncation forgeries, hence,

$$Pr[w'^i = y_1'^i \oplus \quad \oplus y_{n_i'}'^i \mid R^s] \leq \frac{1}{2^l} + \frac{q_s}{2^{l+1}}(\log_2 q_s + 3) + \frac{\mu_s}{l2^{l+1}}(\log_2 \frac{\mu_s}{l} + 3) + \frac{n_i'}{2^{l+1}}(\log_2 n_i' + 3).$$

Hence, for all q_v verification queries, we obtain by union bound and using Fact 4 from the proof of Theorem 2:

$$\begin{aligned}
Pr[\text{Succ} \mid R^s] &\leq \sum_{i=1}^{q_v} Pr[w'^i = y_1'^i \oplus \quad \oplus y_{n_i'}'^i \mid R^s] \\
&\leq \sum_{i=1}^{q_v} \frac{1}{2^l} + \frac{q_s}{2^{l+1}}(\log_2 q_s + 3) + \frac{\mu_s}{l2^{l+1}}(\log_2 \frac{\mu_s}{l} + 3) + \frac{n_i'}{2^{l+1}}(\log_2 n_i' + 3) \\
&= \frac{q_v}{2^l} + \frac{q_v q_s}{2^{l+1}}(\log_2 q_s + 3) + \frac{q_v \mu_s}{l2^{l+1}}(\log_2 \frac{\mu_s}{l} + 3) + \frac{\mu_v}{l2^{l+1}}(\log_2 \frac{\mu_v}{l} + 3).
\end{aligned}$$

Hence, by Claim 1,

$$\begin{aligned}
Pr[\text{Succ}] &\leq \frac{q_v}{2^l} + \frac{q_v q_s}{2^{l+1}}(\log_2 q_s + 3) + \frac{q_v \mu_s}{l2^{l+1}}(\log_2 \frac{\mu_s}{l} + 3) + \frac{\mu_v}{l2^{l+1}}(\log_2 \frac{\mu_v}{l} + 3) + \\
&\quad \frac{q_s \mu_s}{l2^{l+1}}(\log_2 q_s + 3) + \frac{\mu_s^2}{l^2 2^{l+1}}(\log_2 \frac{\mu_s}{l} + 3).
\end{aligned}$$

Finally, when $f \overset{\mathcal{R}}{\leftarrow} F$, the probability for adversary's success is bounded as follows:

$$\begin{aligned}
Pr_{f \overset{\mathcal{R}}{\leftarrow} F}[\text{Succ}] &\leq \epsilon + \\
&\quad \frac{q_v}{2^l} + \frac{q_v q_s}{2^{l+1}}(\log_2 q_s + 3) + \frac{q_v \mu_s}{l2^{l+1}}(\log_2 \frac{\mu_s}{l} + 3) + \frac{\mu_v}{l2^{l+1}}(\log_2 \frac{\mu_v}{l} + 3) + \\
&\quad \frac{q_s \mu_s}{l2^{l+1}}(\log_2 q_s + 3) + \frac{\mu_s^2}{l^2 2^{l+1}}(\log_2 \frac{\mu_s}{l} + 3) \\
&= \epsilon + \frac{q_v}{2^l} + \frac{\mu_v}{l2^{l+1}}(\log_2 \frac{\mu_v}{l} + 3) + \quad q_v + \frac{\mu_s}{l} \quad \frac{q_s}{2^{l+1}}(\log_2 q_s + 3) + \\
&\quad q_v + \frac{\mu_s}{l} \quad \frac{\mu_s}{l2^{l+1}}(\log_2 \frac{\mu_s}{l} + 3).
\end{aligned}$$

□

Proofs of Claims 1 and 2

Proof of Claim 1

To find an upper bound for $Pr[\overline{R^s}]$, we define the following set (which enables us to define event R^s):

$$S_{p,k} = \{z_v^u, 1 \leq u \leq p \Leftrightarrow 1, 1 \leq v \leq n_u\} \quad \{z_v^p, 1 \leq v \leq k\},$$

and events:

$$R_{p,k}^s : S_{p,k} \text{ is collision-free,}$$

and

$$R_{p,n_p+1}^s = R_{p+1,1}^s, \text{ if } p \quad q_s.$$

Based on these definitions, $R_{1,1}^s$ is the true event, and $R^s = R_{q_s,n_{q_s}}^s$. By convention, $R^s = R_{q_s,n_{q_s+1}}^s$.

Using standard conditioning, we have the recurrence relation:

$$Pr[\overline{R_{p,k+1}^s}] \leq Pr[\overline{R_{p,k+1}^s} \mid R_{p,k}^s] + Pr[\overline{R_{p,k}^s}].$$

Hence,

$$\begin{aligned} Pr[\overline{R^s}] &= Pr[\overline{R^s_{q_s,n_{q_s+1}}}] \\ &\leq \sum_{p=1}^{q_s}\sum_{k=1}^{n_p} Pr[\overline{R^s_{p,k+1}} \mid R^s_{p,k}] + Pr[\overline{R^s_{1,1}}] \\ &= \sum_{p=1}^{q_s}\sum_{k=1}^{n_p} Pr[\overline{R^s_{p,k+1}} \mid R^s_{p,k}] \end{aligned}$$

because $Pr[\overline{R^s_{1,1}}] = 0$, since event $R^s_{1,1}$ is always true. In here, we also have that $Pr[\overline{R^s_{q_s,n_{q_s}+1}} \mid R^s_{q_s,n_{q_s}}] = Pr[\overline{R^s} \mid R^s] = 0$.

When event $R^s_{p,k}$ is true, event $\overline{R^s_{p,k+1}}$ is true only when the collisions $z^p_{k+1} = z^u_v$ happened, where either $u < p$ or $u = p$ and $v \leq k$. By convention, $z^p_{n_p+1} = z^{p+1}_1$, if $p \quad q_s$. Hence, by union bound:

$$Pr[\overline{R^s_{p,k+1}} \mid R^s_{p,k}] \leq \sum_{u=1}^{p-1}\sum_{v=1}^{n_u} Pr[z^p_{k+1} = z^u_v \mid R^s_{p,k}] + \sum_{v=1}^{k} Pr[z^p_{k+1} = z^p_v \mid R^s_{p,k}].$$

To compute a bound for the second sum, we note that

$$z^p_{k+1} = z^p_v \;\Leftrightarrow\; (k+1 \Leftrightarrow v) \times R^* = x^p_v \Leftrightarrow x^p_{k+1},$$

and by using Facts 2 and 3 (Appendix A), we obtain

$$\sum_{v=1}^{k} Pr[z^p_{k+1} = z^p_v \mid R^s_{p,k}] \leq \frac{k}{2^{l+1}}(\log_2 k + 3).$$

To compute a bound for the first sum, we split it into three terms based on the different values of $v \leq n_u$ relative to $k+1$, and obtain

$$\begin{aligned} \sum_{v=1}^{n_u} Pr[z^p_{k+1} = z^u_v \mid R^s_{p,k}] &= \sum_{v=1}^{k} Pr[z^p_{k+1} = z^u_v \mid R^s_{p,k}] + Pr[z^p_{k+1} = z^u_{k+1} \mid R^s_{p,k}] + \\ &\quad \sum_{v=k+2}^{n_u} Pr[z^p_{k+1} = z^u_v \mid R^s_{p,k}]. \end{aligned}$$

(By convention, the probabilities for undefined collisions are set to zero. For instance, if $k+1 > n_u$, then $Pr[z^p_{k+1} = z^u_{k+1} \mid R^s_{p,k}] = 0$ and the last sum is zero, since it does not have any terms.)

For the first term of the first sum, $v \leq k$, and

$$z^p_{k+1} = z^u_v \;\Leftrightarrow\; (k+1 \Leftrightarrow v) \times R^* = x^u_v \Leftrightarrow x^p_{k+1} + (u \Leftrightarrow p) \times R.$$

Here, let m be defined as $k+1 \Leftrightarrow v = d \times 2^m$ and d odd; hence, by Facts 2 and 3 (Appendix A), one can show that

$$\sum_{v=1}^{k} Pr[z^p_{k+1} = z^u_v \mid R^s_{p,k}] \leq \frac{k}{2^{l+1}}(\log_2 k + 3).$$

Similarly, for the last term of the first sum, $v \geq k+2$,

$$\sum_{v=k+2}^{n_u} Pr[z^p_{k+1} = z^u_v \mid R^s_{p,k}] \leq \frac{n_u \Leftrightarrow k \Leftrightarrow 1}{2^{l+1}}(\log_2(n_u \Leftrightarrow k \Leftrightarrow 1) + 3).$$

(Note that if $n_u \Leftrightarrow k \Leftrightarrow 1 \leq 0$, the sum is set to zero, which is consistent with the convention for such sums).

Also, for the middle term of the first sum, $v = k+1$

$$z^p_{k+1} = z^u_{k+1} \;\; \Leftrightarrow \;\; (p \Leftrightarrow u) \times R = x^u_{k+1} \Leftrightarrow x^p_{k+1}$$

and, hence, using Fact 2, we have

$$Pr[z^p_{k+1} = z^u_{k+1} \mid R^s_{p,k}] \leq \frac{2^m}{2^l}$$

where $0 \quad \Leftrightarrow u = d \times 2^m$ and d is odd. Hence, $\quad < p$

$$\sum_{v=1}^{n_u} Pr[z^p_{k+1} = z^u_v \mid R^s_{p,k}] \leq \frac{k}{2^{l+1}}(\log_2 k + 3) + \frac{2^m}{2^l} + \frac{n_u \Leftrightarrow k \Leftrightarrow 1}{2^{l+1}}(\log_2(n_u \Leftrightarrow k \Leftrightarrow 1) + 3),$$

where $0 \quad \Leftrightarrow u = d \times 2^m$ and d is odd. Furthermore, using Fact 4 (Appendix A), we have: $\quad < p$

$$\sum_{v=1}^{n_u} Pr[z^p_{k+1} = z^u_v \mid R^s_{p,k}] \leq \frac{n_u \Leftrightarrow 1}{2^{l+1}}(\log_2(n_u \Leftrightarrow 1) + 3) + \frac{2^m}{2^l} \leq \frac{n_u}{2^{l+1}}(\log_2 n_u + 3) + \frac{2^m}{2^l}.$$

Hence, the first sum becomes

$$\sum_{u=1}^{p-1}\sum_{v=1}^{n_u} Pr[z^p_{k+1} = z^u_v \mid R^s_{p,k}] \leq \sum_{u=1}^{p-1} \frac{n_u}{2^{l+1}}(\log_2 n_u + 3) + \frac{2^m}{2^l} \quad .$$

Using Fact 3 (Appendix A), and $0 \quad \Leftrightarrow u = d \times 2^m$ and d odd, we obtain $\quad < p$

$$\sum_{u=1}^{p-1} \frac{2^m}{2^l} \leq \frac{p \Leftrightarrow 1}{2^{l+1}}(\log_2(p \Leftrightarrow 1) + 3).$$

by using Fact 4 (Appendix A). Hence, the first sum is bounded as follows:

$$\sum_{u=1}^{p-1}\sum_{v=1}^{n_u} Pr[z^p_{k+1} = z^u_v \mid R^s_{p,k}] \leq \frac{p \Leftrightarrow 1}{2^{l+1}}(\log_2(p \Leftrightarrow 1) + 3) + \sum_{u=1}^{p-1} \frac{n_u}{2^{l+1}}(\log_2 n_u + 3).$$

Hence the bound of $Pr[\overline{R^s_{p,k+1}} \mid R^s_{p,k}]$ becomes

$$\begin{aligned} Pr[\overline{R^s_{p,k+1}} \mid R^s_{p,k}] &\leq \sum_{u=1}^{p-1}\sum_{v=1}^{n_u} Pr[z^p_{k+1} = z^u_v \mid R^s_{p,k}] + \sum_{v=1}^{k} Pr[z^p_{k+1} = z^p_v \mid R^s_{p,k}] \\ &\leq \frac{p \Leftrightarrow 1}{2^{l+1}}(\log_2(p \Leftrightarrow 1) + 3) + \sum_{u=1}^{p-1} \frac{n_u}{2^{l+1}}(\log_2 n_u + 3) + \frac{k}{2^{l+1}}(\log_2 k + 3). \end{aligned}$$

Returning to the computation of the bound for $Pr[\overline{R^s}]$, we obtain

$$\begin{aligned} Pr[\overline{R^s}] &= Pr[\overline{R^s_{q_s,n_{q_s}}}] \leq \sum_{p=1}^{q_s}\sum_{k=1}^{n_p} Pr[\overline{R^s_{p,k+1}} \mid R^s_{p,k}] \\ &\leq \sum_{p=1}^{q_s}\sum_{k=1}^{n_p} \left(\frac{p \Leftrightarrow 1}{2^{l+1}}(\log_2(p \Leftrightarrow 1) + 3) + \sum_{u=1}^{p-1} \frac{n_u}{2^{l+1}}(\log_2 n_u + 3) + \frac{k}{2^{l+1}}(\log_2 k + 3) \right) \\ &= \sum_{p=1}^{q_s}\sum_{k=1}^{n_p} \frac{p \Leftrightarrow 1}{2^{l+1}}(\log_2(p \Leftrightarrow 1) + 3) + \sum_{p=1}^{q_s}\sum_{k=1}^{n_p}\sum_{u=1}^{p-1} \frac{n_u}{2^{l+1}}(\log_2 n_u + 3) + \\ &\quad \sum_{p=1}^{q_s}\sum_{k=1}^{n_p} \frac{k}{2^{l+1}}(\log_2 k + 3) \end{aligned}$$

In the first sum, since $p \Leftrightarrow 1 \leq q_s$, it follows that

$$\sum_{p=1}^{q_s}\sum_{k=1}^{n_p}\frac{p \Leftrightarrow 1}{2^{l+1}}(\log_2(p \Leftrightarrow 1)+3) \leq \sum_{p=1}^{q_s}\sum_{k=1}^{n_p}\frac{q_s}{2^{l+1}}(\log_2 q_s+3) \leq \frac{q_s\mu_s}{l2^{l+1}}(\log_2 q_s+3).$$

The second sum yields:

$$\sum_{p=1}^{q_s}\sum_{k=1}^{n_p}\sum_{u=1}^{p-1}\frac{n_u}{2^{l+1}}(\log_2 n_u+3) \leq \frac{1}{2^{l+1}}(\log_2\frac{\mu_s}{l}+3)\sum_{p=1}^{q_s}\sum_{k=1}^{n_p}\sum_{u=1}^{p-1}n_u$$

since $n_u \leq \frac{\mu_s}{l}$. One can also see that for: $p=1, \sum_{u=1}^{p-1} n_u = 0$ since it has no terms, for $p=2, \sum_{u=1}^{p-1} n_u = n_1$, etc. Hence,

$$\sum_{p=1}^{q_s}\sum_{k=1}^{n_p}\sum_{u=1}^{p-1}\frac{n_u}{2^{l+1}} = n_2n_1 + n_3(n_1+n_2) + \quad + n_{q_s}(n_1 + \quad + n_{q_s-1}) \leq \frac{1}{2}(n_1 + \quad + n_{q_s})^2 = \frac{\mu_s^2}{2l}.$$

Hence, the second sum is bounded as follows:

$$\sum_{p=1}^{q_s}\sum_{k=1}^{n_p}\sum_{u=1}^{p-1}\frac{n_u}{2^{l+1}}(\log_2 n_u+3) \leq \frac{\mu_s^2}{l2^{l+2}}(\log_2\frac{\mu_s}{l}+3).$$

In the third sum, we have $k \leq n_p$ and, using Fact 4 (Appendix A), we obtain:

$$\begin{aligned}\sum_{p=1}^{q_s}\sum_{k=1}^{n_p}\frac{k}{2^{l+1}}(\log_2 k+3) &\leq \sum_{p=1}^{q_s}\frac{n_p(n_p \Leftrightarrow 1)}{2^{l+2}}(\log_2 n_p+3) \leq \sum_{p=1}^{q_s}\frac{n_p\mu_s}{l2^{l+2}}(\log_2 n_p+3)\\ &\leq \frac{\mu_s^2}{l^2 2^{l+2}}(\log_2\frac{\mu_s}{l}+3).\end{aligned}$$

Hence,

$$\begin{aligned}Pr[\overline{R^s}] &\leq \frac{q_s\mu_s}{l2^{l+1}}(\log_2 q_s+3) + \frac{\mu_s^2}{l^2 2^{l+2}}(\log_2\frac{\mu_s}{l}+3) + \frac{\mu_s^2}{l^2 2^{l+2}}(\log_2\frac{\mu_s}{l}+3)\\ &\leq \frac{q_s\mu_s}{l2^{l+1}}(\log_2 q_s+3) + \frac{\mu_s^2}{l^2 2^{l+1}}(\log_2\frac{\mu_s}{l}+3).\end{aligned}$$

Remark: With more care one can show that the sum $\sum_{p=1}^{q_s}\sum_{k=1}^{n_p}\frac{k}{2^{l+1}}(\log_2 k+3)$ is actually order $\frac{\mu_s^2}{l^2 2^l}\sqrt{log_2\frac{\mu_s}{l}}$, and, hence, for very large $\frac{\mu_s}{l}$, the dominant term in the upper bound is $\frac{\mu_s^2}{l^2 2^{l+2}}(\log_2\frac{\mu_s}{l}+3)$. □

Proof of Claim 2

(a) The proof of this Claim is very similar to the proof of Claim 3 in the Proof of Theorem 2 (viz., Appendix A). First, we choose an index j such that for any type of possible non-truncation forgery i, the input to f at the verification of forgery i, namely $x_j'^i + s'^i \times R + j \times R^*$, does collide with any input to f during message signing with low probability. Next, we compute an upper bound for these collisions.

All non-truncation forgeries can be partitioned in a similar manner as that used in the proof of Claim 3 of Theorem 2 (Appendix A). That is, we define extensions of the plaintext of a signed message, which we call the *prefix* case, and the complementary case, which we call *non-prefix* case. The non-prefix case includes two separate subcases, namely when s'^i is different from any message ID p of any message p obtained at signing (i.e., message (x^p, p, w^p)), or when there is a signed message p such that $s'^i = p$. Hence, in the

latter subcase, there must be at least a block position j in the forged message x'^i that is different from the corresponding block of the signed message p. This partition of all possible forgery types shows that a forged message $x'^i = x'^i_1 \quad x'^i_{n'_i}$ which is not a truncation, can be one of the following three complementary types:
(a) $\exists p, 1 \leq p \leq q_s : n'_i > n_p, s'^i = p$ and $\forall k, 1 \leq k \leq n_p : x'^i_k = x^p_k$; i.e., the forged message is an extension of message x^p (the prefix case). The non-prefix case consists of the following two forgery types:
(b1) $s'^i \neq p, \forall p, 1 \leq p \leq q_s$; and
(b2) $\exists p, 1 \leq p \leq q_s : s'^i = p, \exists k, 1 \leq k \leq \min(n'_i, n_p) : x'^i_k \neq x^p_k$; i.e., the forged message is obtained by modifying a queried message starting with some block between the second and last block.

Now we choose index j mentioned above for each type of possible non-truncation forgeries, as follows: for forgeries of type (a), $j = n_p + 1$; for forgeries of type (b1), $j = 1$; and for forgeries of type (b2), j is the smallest index such that $x'^i_j \neq x^p_j, 1 \leq j \leq \min\{n_p, n'_i\}$. In all cases $1 \leq j \leq n'_i$, and hence, the chosen block x'^i_j is well defined.

Event C^i implies that $x'^i_j + s'^i \times R + j \times R^* \in S$. Hence, by union bound

$$Pr[C^i \mid R^s] \leq \sum_{p=1}^{q_s} \sum_{k=1}^{n_p} Pr[x'^i_j + s'^i \times R + j \times R^* = x^p_k + p \times R + k \times R^* \mid R^s].$$

We write the inner sum as a sum of three terms, as follows:

$$\begin{aligned}
& \sum_{k=1}^{n_p} Pr[x'^i_j + s'^i \times R + j \times R^* = x^p_k + p \times R + k \times R^* \mid R^s] \\
= & \sum_{k=1}^{j-1} Pr[x'^i_j + s'^i \times R + j \times R^* = x^p_k + p \times R + k \times R^* \mid R^s] \\
+ & Pr[x'^i_j + s'^i \times R + j \times R^* = x^p_k + p \times R + j \times R^* \mid R^s] \\
+ & \sum_{k=j+1}^{n_p} Pr[x'^i_j + s'^i \times R + j \times R^* = x^p_k + p \times R + k \times R^* \mid R^s],
\end{aligned}$$

By the convention adopted above, the probability terms are zero for undefined collision events. (For example, if $j > n_p$, then $Pr[x'^i_j + s'^i \times R + j \times R^* = x^p_k + p \times R + j \times R^* \mid R^s] = 0$.) For the collision $x'^i_j + s'^i \times R + j \times R^* = x^p_k + p \times R + j \times R^*$, we have $(s'^i \Leftrightarrow p) \times R = x^p_k \Leftrightarrow x'^i_j$. In this expression, if $s'^i = p$, then by the choice of index j we are in case (b2) where $x'^i_j \neq x^p_k$, and the probability of this collision event is zero. If $s'^i \neq p$, then using Fact 2 (Appendix A), we have

$$Pr[x'^i_j + s'^i \times R + j \times R^* = x^p_k + p \times R + j \times R^* \mid R^s] \leq \frac{2^m}{2^l},$$

where $s'^i \Leftrightarrow p = d \times 2^m$ if $s'^i > p$, or $p \Leftrightarrow s'^i = d \times 2^m$ if $p \quad s'^i$. For, the other sums, in a manner similar to the one used in Claim 1, we have:

$$\begin{aligned}
& \sum_{k=1}^{j-1} Pr[x'^i_j + s'^i \times R + j \times R^* = x^p_k + p \times R + k \times R^* \mid R^s] \\
+ & \sum_{k=j+1}^{n_p} Pr[x'^i_j + s'^i \times R + j \times R^* = x^p_k + p \times R + k \times R^* \mid R^s] \\
\leq & \frac{n_p}{2^{l+1}}(\log_2 n_p + 3).
\end{aligned}$$

Hence, by using Fact 4 (Appendix A), we have

$$\begin{aligned}\sum_{p=1}^{q_s}\sum_{k=1}^{n_p} Pr[x_j'^i + s'^i \times R + j \times R^* = x_k^p + p \times R + k \times R^* \mid R^s] &\leq \sum_{p=1}^{q_s} \frac{2^m}{2^l} + \frac{n_p}{2^{l+1}}(\log_2 n_p + 3) \\ &\leq \sum_{p=1}^{q_s} \frac{2^m}{2^l} + \frac{\mu_s}{l2^{l+1}}(\log_2 \frac{\mu_s}{l} + 3).\end{aligned}$$

In the first sum, $\sum_{p=1}^{q_s} \frac{2^m}{2^l}$, we use the fact that $1 \leq p, s'^i \leq q_s, p \neq s'^i$ (as shown in Case (b2) above, the probability is zero when $p = s'^i$), hence

$$\begin{aligned}\sum_{p=1}^{q_s} \frac{2^m}{2^l} &= \sum_{p=1}^{s'^i-1} \frac{2^m}{2^l} + \sum_{p=s'^i+1}^{q_s} \frac{2^m}{2^l} \\ &\leq \frac{s'^i \Leftrightarrow 1}{2^{l+1}}(\log_2(s'^i \Leftrightarrow 1) + 3) + \frac{q_s \Leftrightarrow s'^i \Leftrightarrow 1}{2^{l+1}}(\log_2(q_s \Leftrightarrow s'^i \Leftrightarrow 1) + 3) \\ &\leq \frac{q_s \Leftrightarrow 2}{2^{l+1}}(\log_2(q_s \Leftrightarrow 1) + 3) \leq \frac{q_s}{2^{l+1}}(\log_2 q_s + 3)\end{aligned}$$

Hence,

$$Pr[C^i \mid R^s] \leq \frac{q_s}{2^{l+1}}(\log_2 q_s + 3) + \frac{\mu_s}{l2^{l+1}}(\log_2 \frac{\mu_s}{l} + 3).$$

(b) We find an upper bound for $Pr[\overline{D^i} \mid \overline{C^i}$ and $R^s]$ in a manner very similar to the one used in Claim 3(c) of the Proof of Theorem 3 (viz., Appendix B). Since the message ID does not matter in this case (since all the elements of V_i have the same message ID s'^i, then the bound is identical

$$Pr[\overline{D^i} \mid \overline{C^i} \text{ and } R^s] \leq \frac{n_i'}{2^{l+1}}(\log_2 n_i' + 3).$$

□

Appendix D – Proof [Security of stateful XEBCS-XOR in a Message-Integrity Attack]

Throughout this proof, we use the same notation as in the Proof for Security of the XCBC\$-XOR in a Message-Integrity Attack, Appendix A, and the same facts (i.e., Facts 1 4). Unless mentioned otherwise, we focus on the probability for adversary's success when $f \overset{\mathcal{R}}{\leftarrow} G_S$, and, for simplicity, we will drop the $f \overset{\mathcal{R}}{\leftarrow} G_S$ subscript from the probability equations.

Notation. Let $z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_p$ be the *hidden ciphertext blocks* at the encryption of message p; i.e., for encrypted message $x^p = x_1^p \quad x_{n_p}^p$, we have

$$\begin{aligned} z_k^p &= f(x_k^p + p \times R + k \times R^*), 1 \leq k \leq n_p \\ z_{n_p+1}^p &= f(x_{n_p+1}^p + p \times R). \end{aligned}$$

Let $z_j'^i, 1 \leq i \leq q_v, 1 \leq j \leq n_i' + 1$ be the *hidden ciphertext blocks* at the decryption of forgery i; i.e., for the forgery $y'^i = y_1'^i \quad y_{n_i'}'^i y_{n_i'+1}'^i$ using the message identifier (ID) s'^i ($s'^i \leq q_e$), we have

$$z_j'^i = y_j'^i \Leftrightarrow s'^i \times R \Leftrightarrow j \times R^*, 1 \leq j \leq n_i' + 1.$$

To find an upper bound on the probability of an adversary's success we use the same proof technique as for the XCBC\$-XOR scheme. That is, we (1) define several types of events on which we condition the adversary's success, (2) express the upper bound in terms of the conditional probabilities obtained, and (3) compute upper bounds on these probabilities.

Conditioning Events. To compute an upper bound on the probability of successful forgery, we choose three conditioning events based on collisions in the inputs of f and f^{-1}. We introduce the sets:

$$\begin{aligned} S &: \{z_k^p, 1 \leq p \leq q_e, 1 \leq k \leq n_p + 1\}, \\ V_i &: \{z_j'^i, z_j'^i \notin S, 1 \leq j \leq n_i' + 1\}, \end{aligned}$$

where S is the set of all the hidden ciphertext blocks (outputs of function f at encryption), and V_i is the set of all the inputs of function f^{-1} at decryption of query i that are not in S. Based on sets S and V_i, we introduce the following collision events that arise at the verification of forgery (y'^i, s'^i):

$$C^i \quad : \quad V_i = \emptyset.$$

Event C^i includes all instances when inputs of f^{-1} at forgery decryption collide with outputs of function f at message encryption. Next we define event D^i as follows:

$$\begin{aligned} D^i \quad : \quad & \exists j, 1 \leq j \leq n_i' + 1 : z_j'^i \in V_i \\ & \text{and } z_j'^i \neq z_m'^i, \forall z_m'^i \in V_i, j \neq m, 1 \leq m \leq n_i' + 1. \end{aligned}$$

Event D^i states that there is at least one "new" hidden ciphertext block for forgery i that does not collide with any other "new" hidden ciphertext block for forgery i, where "new" hidden ciphertext blocks refers to hidden ciphertext blocks that are not in the set of hidden ciphertext blocks at encryption, namely S. It is clear that the definition for D^i makes sense only when event C^i is false.

The rationale for introducing events C^i (or, actually, $\overline{C^i}$) and D^i is similar to the one used in the proof of Theorem 2 (Appendix A). That is, we want to find a desirable event which states that there exists a hidden ciphertext block that does not collide with any other output of f at message encryption or with any other input to f^{-1} at the decryption of forgery i. Clearly, if this event is true, then the probability of

verification passing is $1/2^l$ if $f \overset{\mathcal{R}}{\leftarrow} G_S$, where we use the reduction from $f \overset{\mathcal{R}}{\leftarrow} F$ to $f \overset{\mathcal{R}}{\leftarrow} G_S$ as defined in Appendix A (Fact 1). To find this event, however, we must ensure that all other collisions that that may lead to the discovery of R or R^* are also ruled out for this block (viz., Example 3 in Appendix C). For this reason, we introduce event R^e defined below.

$$R^e \quad : \quad z_k^p \neq z_t^s, 1 \leq p, s \leq q_e, 1 \leq k \leq n_p + 1, 1 \leq t \leq n_s + 1, (p,k) \neq (s,t).$$

Event R^e states that the set S is collision-free. Note that event R^e is independent of any forgery i.

Upper bound on the Probability of Successful Forgery. Fact 1 of Appendix A reduces the problem to finding an upper bound for $Pr_{f \overset{\mathcal{R}}{\leftarrow} G_S}[\text{Succ}]$, and

$$Pr_{f \overset{\mathcal{R}}{\leftarrow} F}[\text{Succ}] \leq \epsilon + \frac{\mu_v(\mu_v \Leftrightarrow l)}{l^2 2^{l+1}} + Pr_{f \overset{\mathcal{R}}{\leftarrow} G_S}[\text{Succ}].$$

Unless we state otherwise, assume that $f \overset{\mathcal{R}}{\leftarrow} G_S$ (and drop this subscript from $Pr_{f \overset{\mathcal{R}}{\leftarrow} G_S}[\text{Succ}]$.)

By standard conditioning, we have

$$Pr[\text{Succ}] \leq Pr[\text{Succ} \mid R^e] + Pr[\overline{R^e}].$$

The second term in the sum is bounded as in the following Claim:

Claim 1

$$Pr[\overline{R^e}] \leq \frac{q_e \mu_e}{l 2^{l+1}}(\log_2 q_e + 3) + \frac{\mu_e^2}{l^2 2^{l+1}}(\log_2 \frac{\mu_e}{l} + 3) + \frac{\mu_e(\mu_e \Leftrightarrow l)}{l^2 2^{l+1}}.$$

The proof of Claim 1 is similar to the proof of Claim 1 of Appendix C, and the extra term $\frac{\mu_e(\mu_e - l)}{l^2 2^{l+1}}$ appears because of the distinction between $f \overset{\mathcal{R}}{\leftarrow} P^l$ (since $f \overset{\mathcal{R}}{\leftarrow} G_S$) and $f \overset{\mathcal{R}}{\leftarrow} R^{l,l}$.

To compute an upper bound for the probability of successful forgery, when event R^e is true, we note that the adversary is successful if one of his q_v forgeries is (y'^i, s'^i) is successful, where $y'^i = y_1'^i \quad y_{n_i'+1}'^i$. Hence, by union bound, the probability of adversary's success for all q_v verification queries (when $f \overset{\mathcal{R}}{\leftarrow} G_S$) is:

$$Pr[\text{Succ} \mid R^e] \leq \sum_{i=1}^{q_v} Pr[x_{n_i'+1}'^i = x_1'^i \oplus \quad x_{n_i'}'^i \mid R^e].$$

Hence, we first compute the probability of adversary's success when a single forgery verification is allowed; i.e., we compute $Pr[\text{Succ} \mid R^e]$. For this computation, we partition the space of all possible forgeries into (1) truncation and (2) non-truncation forgeries.

Truncation Forgeries. We call truncation a forgery $y'^i = y_1'^i \quad y_{n_i'+1}'^i$ together with a message identifier s'^i ($s'^i \leq q_e$) such that there exists a ciphertext message $y^p = y_1^p \quad y_{n_p+1}^p$ where $s'^i = p$ and $y_k'^i = y_k^p, \forall k, 1 \leq k \leq n_i' + 1 \quad {}_p + 1$. $<n$

In this case, for any $1 \leq j \leq n_i' + 1$ we have:

$$z_j'^i = z_j^p, \forall j, 1 \leq j \leq n_i' + 1,$$

and thus

$$x_j'^i = f^{-1}(z_j'^i) \Leftrightarrow s'^i \times R \Leftrightarrow j \times R^* = f^{-1}(z_j^p) \Leftrightarrow p \times R \Leftrightarrow j \times R^* = x_j^p,$$

for any $1 \leq j \leq n'_i$, and

$$\begin{aligned} x'^i_{n'_i+1} &= f^{-1}(z'^i_{n'_i+1}) \Leftrightarrow s'^i \times R = f^{-1}(z^p_{n'_i+1}) \Leftrightarrow p \times R \\ &= x^p_{n'_i+1} + p \times R + (n'_i + 1) \times R^* \Leftrightarrow p \times R = x^p_{n'_i+1} + (n'_i + 1) \times R^*. \end{aligned}$$

Hence, the integrity condition

$$x'^i_{n'_i+1} = x'^i_1 \oplus \quad \oplus x'^i_{n'_i}$$

becomes

$$x^p_{n'_i+1} + (n'_i + 1) \times R^* = x^p_1 \oplus \quad \oplus x^p_{n'_i},$$

where $x^p_1, \quad , x^p_{n'_i+1}$ are constants (since $n'_i + 1 \leq n_p$). Hence, by Fact 2, we have for the truncation forgery i

$$Pr[x'^i_{n'_i+1} = x'^i_1 \oplus \quad x'^i_{n'_i} \mid R^e] \leq \frac{2^m}{2^l},$$

where $n'_i + 1 = d \times 2^m$ and d is odd. Thus, since $2^m \leq n'_i + 1$, we have

$$Pr[x'^i_{n'_i+1} = x'^i_1 \oplus \quad x'^i_{n'_i} \mid R^e] \leq \frac{n'_i + 1}{2^l}.$$

Non-Truncation Forgeries. Now, we find an upper bound for $Pr[x'^i_{n'_i+1} = x'^i_1 \oplus \quad \oplus x'^i_{n'_i} \mid R^e]$ for non-truncation forgeries. To compute this upper bound, we define an event such that (1) the probability of successful forgery is $1/2^l$ when this event occurs, and (2) the probability of the complement of this event has a negligible upper bound.

Using the events defined above and by standard conditioning, we obtain:

$$\begin{aligned} Pr[x'^i_{n'_i+1} = x'^i_1 \oplus \quad \oplus x'^i_{n'_i} \mid R^e] &\leq Pr[x'^i_{n'_i+1} = x'^i_1 \oplus \quad \oplus x'^i_{n'_i} \mid \overline{C^i} \text{ and } D^i \text{ and } R^e] + \\ &\quad Pr[C^i \text{ or } \overline{D^i} \mid R^e] \\ &\leq Pr[x'^i_{n'_i+1} = x'^i_1 \oplus \quad \oplus x'^i_{n'_i} \mid \overline{C^i} \text{ and } D^i \text{ and } R^e] + \\ &\quad Pr[C^i \text{ or } \overline{D^i} \mid \overline{C^i} \text{ and } R^e] + Pr[C^i \mid R^e] \\ &= Pr[x'^i_{n'_i+1} = x'^i_1 \oplus \quad \oplus x'^i_{n'_i} \mid \overline{C^i} \text{ and } D^i \text{ and } R^e] + \\ &\quad Pr[\overline{D^i} \mid \overline{C^i} \text{ and } R^e] + Pr[C^i \mid R^e], \end{aligned}$$

since the following events are equivalent:

$$(C^i \text{ or } \overline{D^i} \mid \overline{C^i} \text{ and } R^e) \equiv (\overline{D^i} \mid \overline{C^i} \text{ and } R^e).$$

Event $(\overline{C^i}$ and D^i and $R^e)$ is the desired event mentioned earlier in this proof. If this event happens, then

$$Pr[x'^i_{n'_i+1} = x'^i_1 \oplus \quad \oplus x'^i_{n'_i} \mid \overline{C^i} \text{ and } D^i \text{ and } R^e] = \frac{1}{2^l}.$$

The other probabilities that appear in the expression for the total probability $Pr[x'^i_{n'_i+1} = x'^i_1 \oplus \quad \oplus x'^i_{n'_i} \mid R^e]$ are bounded as in Claim 2, whose proof is similar to that of Claim 2 in Appendix C.

Claim 2
(a)

$$Pr[C^i \mid R^e] \leq \frac{q_e}{2^{l+1}}(\log_2 q_e + 3) + \frac{\mu_e}{l2^{l+1}}(\log_2 \frac{\mu_e}{l} + 3).$$

(b)

$$Pr[\overline{D^i} \mid \overline{C^i} \text{ and } R^e] \leq \frac{n_i'+1}{2^{l+1}}(\log_2(n_i'+1)+3).$$

Based on this claim, for an arbitrary forgery i that is not a truncation, we obtain:

$$Pr[x'^i_{n_i'+1} = x'^i_1 \oplus \quad \oplus x'^i_{n_i'} \mid R^e] \leq \frac{1}{2^l} + \frac{q_e}{2^{l+1}}(\log_2 q_e + 3) + \frac{\mu_e}{l2^{l+1}}(\log_2 \frac{\mu_e}{l} + 3) + \frac{n_i'+1}{2^{l+1}}(\log_2(n_i'+1)+3).$$

For any forgery, the upper bound is the maximum from the upper bounds for truncation and non-truncation forgeries, hence,

$$Pr[x'^i_{n_i'+1} = x'^i_1 \oplus \quad \oplus x'^i_{n_i'} \mid R^e] \leq \frac{1}{2^l} + \frac{q_e}{2^{l+1}}(\log_2 q_e + 3) + \frac{\mu_e}{l2^{l+1}}(\log_2 \frac{\mu_e}{l} + 3) + \frac{n_i'+1}{2^{l+1}}(\log_2(n_i'+1)+3).$$

Hence, for all q_v verification queries, we obtain by union bound,

$$\begin{aligned}
Pr[\text{Succ} \mid R^e] &\leq \sum_{i=1}^{q_v} Pr[x'^i_{n_i'+1} = x'^i_1 \oplus \quad \oplus x'^i_{n_i'} \mid R^e] \\
&\leq \sum_{i=1}^{q_v} \frac{1}{2^l} + \frac{q_e}{2^{l+1}}(\log_2 q_e + 3) + \frac{\mu_e}{l2^{l+1}}(\log_2 \frac{\mu_e}{l} + 3) + \frac{n_i'+1}{2^{l+1}}(\log_2(n_i'+1)+3) \\
&= \frac{q_v}{2^l} + \frac{q_v q_e}{2^{l+1}}(\log_2 q_e + 3) + \frac{q_v \mu_e}{l2^{l+1}}(\log_2 \frac{\mu_e}{l} + 3) + \frac{\mu_v}{l2^{l+1}}(\log_2 \frac{\mu_v}{l} + 3).
\end{aligned}$$

Hence, by Claim 1,

$$\begin{aligned}
Pr[\text{Succ}] \leq\ & \frac{q_v}{2^l} + \frac{q_v q_e}{2^{l+1}}(\log_2 q_e + 3) + \frac{q_v \mu_e}{l2^{l+1}}(\log_2 \frac{\mu_e}{l} + 3) + \frac{\mu_v}{l2^{l+1}}(\log_2 \frac{\mu_v}{l} + 3) + \\
& \frac{q_e \mu_e}{l2^{l+1}}(\log_2 q_e + 3) + \frac{\mu_e^2}{l^2 2^{l+1}}(\log_2 \frac{\mu_e}{l} + 3) + \frac{\mu_e(\mu_e \Leftrightarrow l)}{l^2 2^{l+1}}.
\end{aligned}$$

Finally, when $f \stackrel{\mathcal{R}}{\leftarrow} F$, the probability for adversary's success is bounded as follows:

$$\begin{aligned}
Pr_{f \stackrel{\mathcal{R}}{\leftarrow} F}[\text{Succ}] \leq\ & \epsilon + \frac{\mu_v(\mu_v \Leftrightarrow l)}{l^2 2^{l+1}} + \\
& \frac{q_v}{2^l} + \frac{q_v q_e}{2^{l+1}}(\log_2 q_e + 3) + \frac{q_v \mu_e}{l2^{l+1}}(\log_2 \frac{\mu_e}{l} + 3) + \frac{\mu_v}{l2^{l+1}}(\log_2 \frac{\mu_v}{l} + 3) + \\
& \frac{q_e \mu_e}{l2^{l+1}}(\log_2 q_e + 3) + \frac{\mu_e^2}{l^2 2^{l+1}}(\log_2 \frac{\mu_e}{l} + 3) + \frac{\mu_e(\mu_e \Leftrightarrow l)}{l^2 2^{l+1}} \\
=\ & \epsilon + \frac{\mu_v(\mu_v \Leftrightarrow l)}{l^2 2^{l+1}} + \frac{q_v}{2^l} + \frac{\mu_v}{l2^{l+1}}(\log_2 \frac{\mu_v}{l} + 3) + \\
& q_v + \frac{\mu_e}{l} \quad \frac{q_e}{2^{l+1}}(\log_2 q_e + 3) + \quad q_v + \frac{\mu_e}{l} \quad \frac{\mu_e}{l2^{l+1}}(\log_2 \frac{\mu_e}{l} + 3) + \frac{\mu_e(\mu_e \Leftrightarrow l)}{l^2 2^{l+1}}.
\end{aligned}$$

□