The Security of Cipher Block Chaining

Mihir Bellare[1] and Joe Kilian[2] and Phillip Rogaway[3]

[1] Advanced Networking Laboratory, IBM T.J. Watson Research Center, PO Box 704, Yorktown Heights, NY 10598, USA. e-mail: mihir@watson.ibm.com

[2] NEC Research Institute, 4 Independence Way, Princeton, NJ 08540, USA. e-mail: joe@research.nj.nec.com

[3] Department of Computer Science, University of California at Davis, Davis, CA 95616, USA. e-mail: rogaway@cs.ucdavis.edu

Abstract. The Cipher Block Chaining – Message Authentication Code (CBC MAC) specifies that a message $x = x_1 \cdots x_m$ be authenticated among parties who share a secret key a by tagging x with a prefix of

$$f_a^{(m)}(x) \stackrel{\text{def}}{=} f_a(f_a(\cdots f_a(f_a(x_1) \oplus x_2) \oplus \cdots \oplus x_{m-1}) \oplus x_m) ,$$

where f is some underlying block cipher (eg. $f = \text{DES}$). This method is a pervasively used international and U.S. standard. We provide its first formal justification, showing the following general lemma: that cipher block chaining a pseudorandom function gives a pseudorandom function. Underlying our results is a technical lemma of independent interest, bounding the success probability of a computationally unbounded adversary in distinguishing between a random ml-bit to l-bit function and the CBC MAC of a random l-bit to l-bit function.

1 Introduction

1.1 The problem: Is the CBC MAC secure?

Message authentication lets communicating partners who share a secret key verify that a received message originates with the party who claims to have sent it. This is one of the most important and widely used cryptographic tools. It is most often achieved using a "message authentication code," or MAC. This is a short string $\text{MAC}_a(x)$ computed on the message x to be authenticated and the shared secret key a. The sender transmits $\langle x, \text{MAC}_a(x) \rangle$ and the receiver, who gets $\langle x', \sigma' \rangle$, verifies that $\sigma' = \text{MAC}_a(x')$.

The most common MAC is built using the idea of "cipher block chaining" some underlying block cipher. To discuss this we first need some notation. Given a function f: $\{0,1\}^l \to \{0,1\}^l$ and a number $m \geq 1$ we denote by $f^{(m)}$: $\{0,1\}^{ml} \to \{0,1\}^l$ the function which maps an ml-bit input $x = x_1 \cdots x_m$ ($|x_i| = l$) to the l-bit string

$$f^{(m)}(x_1 \cdots x_m) \ = \ f(f(\cdots f(f(x_1) \oplus x_2) \oplus \cdots \oplus x_{m-1}) \oplus x_m) .$$

Y.G. Desmedt (Ed.): Advances in Cryptology - CRYPTO '94, LNCS 839, pp. 341-358, 1994.

We call $f^{(m)}$ the *(m-fold) cipher block chaining* of f.[4] Now, a block cipher F (with key length k and block size l) specifies a family of permutations f_a: $\{0,1\}^l \rightarrow \{0,1\}^l$, one for each k-bit key a. The CBC MAC constructed from F has an associated parameter $s < l$ which is the number of bits it outputs. The CBC MAC is then defined for any ml-bit string $x = x_1 \ldots x_m$ by

$$\text{CBC-MAC}_a^F(x_1 \ldots x_m) \stackrel{\text{def}}{=} \text{The first } s \text{ bits of } f_a^{(m)}(x_1 \cdots x_m)\,.$$

The CBC MAC is an International Standard [13]. The most popular and widely used special case uses $F = \text{DES}$ (so $k = 56$ and $l = 64$) and $s = 32$, in which case we recover the definition of the corresponding U.S. Standard [1]. These standards are extensively employed in the banking sector and in other commercial sectors. Given this degree of usage and standardization, you might expect that there would be a large body of work aimed at learning if the CBC MAC is secure. Yet this has not really been the case. To the best of our knowledge, it was seen as entirely possible that F could be a perfectly secure block cipher even though CBC-MAC^F might be a completely insecure MAC. There was no reason to be sure that the internal structure of F couldn't "interact badly" with the specifics of cipher block chaining in exactly such a way as to defeat the CBC MAC.

1.2 Our approach

In this paper we will show that CBC MAC construction is secure if the underlying block cipher is secure. To make this statement meaningful we need first to discuss what we mean by security in each case.

What does it mean to assume DES is secure?

To describe the security of a block cipher we adopt the viewpoint introduced by Luby and Rackoff [15, 16] with regard to DES. They suggest that a block cipher should be assumed to be a pseudorandom function (PRF) with respect to "practical" computation. The notion of a PRF is in turn due to Goldreich, Goldwasser and Micali [9]. Roughly said, a function family F is pseudorandom if any reasonable adversary is unable to distinguish the following two types of objects, based on their input/output behavior: a black-box for $f_a(\cdot)$, on a random key a; a black-box for a "truly random" function $f(\cdot)$.

What does it mean for a MAC to be secure?

Our notion of security for a message authentication code adopts the viewpoint of Goldwasser, Micali and Rivest [11] with regard to signature schemes; namely, a secure MAC must resist existential forgery under adaptive message attack. However, what we will show is actually stronger: if F is a pseudorandom function family then $F^{(m)}$, the family of functions $f^{(m)}$ for $f \in F$, is *itself* shown to be

[4] Notice that here and in what follows we require the input to consist of exactly m blocks, not at most m. See Section 1.4 for a discussion of length variability.

a pseudorandom function family. That a PRF automatically makes a secure message authentication code is a well-known observation due to [9, 10]—see Section 6 for details.

Exact security

We wish to obtain results which are meaningful for practice. In particular, in our setting we need to say something about the correct or incorrect use of DES, where there are no asymptotics present. This demands not only that we avoid asymptotics and address security "exactly," but also that we strive for security reductions which are as efficient as possible.

We will only talk about finite families of functions and the resources needed to "learn" things about these finite function families. We will describe the resources necessary to "break" the finite family F given an adversary of specified resources who succeeds in breaking $F^{(m)}$. The parameters of interest are the running time t of the adversary; the number of queries q which she makes to an oracle which is her only point of access to $f(\cdot)$-values for the given $f \in F$; and the adversary's advantage, ϵ, over simple guessing. We emphasize the importance of keeping t and q separate: in practice, oracle queries correspond to observations or interaction with a system whose overall structure often severely limits q (e.g., the system might limit the amount of plaintext encrypted before the key is changed); but t corresponds to off-line computation by the adversary, and so is much less under the good guys control.

Assume that adversary A can (t, q, ϵ)-break $F^{(m)}$. This means she runs in time t, makes q queries of her oracle, and succeeds with advantage ϵ in distinguishing a random member of $F^{(m)}$ from a random function of ml-bits to l-bits. Our results specify (t', q', ϵ') (as functions of t, q, ϵ, m, l) such that there exists an adversary A' (a simple modification of A) that (t', q', ϵ')-breaks F.

Exact security is not new. It is true that most theoretical works only provide asymptotic security guarantees of the form "the success probability of a polynomially bounded adversary is negligible" (everything measured as a function of the security parameter), but the exact security can usually be derived from examination of the proof. However, a lack of concern with the exactness means that in many cases the reductions are very inefficient, and the results are not useful for practice.

Previous works which address exact security explicitly and strive for efficient reductions are [8, 12, 19, 6, 14, 4], the last four on the more practical side.

1.3 Main result

Of course the power of results such as those indicated above depends on what values of t', q', ϵ' one can prove. Our analysis is directed at achieving the best values possible. Our main lemma is stated formally as Lemma 5. Informally, it says the following. Suppose there is an adversary A who (t, q, ϵ)-breaks $F^{(m)}$. Then A can be turned into an adversary A' of comparable size and time complexity to A which, making qm oracle queries, achieves advantage $\epsilon' = \epsilon - 3q^2m^2 \cdot 2^{-l-1}$.

Current knowledge gives us values t', q', ϵ' for which it seems safe to rule out (t', q', ϵ')-breaks on DES. From this we can derive values of t, q, ϵ for which (t, q, ϵ)-breaks of $\mathrm{DES}^{(m)}$ are effectively ruled out. Thus, we reduce the security of $\mathrm{DES}^{(m)}$ to that of DES in a constructive and useful way.

The brunt of the proof addresses the information-theoretic case of the above lemma. Here we consider the problem of distinguishing a random ml-bit to l-bit function from the m-fold CBC of a random l-bit to l-bit function. We prove an absolute bound of $3q^2m^2 \cdot 2^{-l-1}$ on the advantage an adversary can derive. (The bound holds irrespective of the adversary's running time and depends only on m and the number q of queries she makes.) The lemma appears in Section 3.

The proof of this information-theoretic case of the CBC Lemma is not easy. For whatever reasons, it seems quite susceptible to specious arguments and to a general difficulty in moving from intuition to proof.

Section 6 completes the picture by showing that the standard construction of a MAC from a PRF has tight security. In this light, we view our main results as being those discussed above.

1.4 Extensions and corollaries

The CBC Lemma provides an efficient method to produce a PRF to $\leq l$-bits when the input is of fixed length ml. But often the input lengths may vary. We exhibit in Section 5 some simple extensions to the CBC MAC which allow one to correctly authenticate words of arbitrary length. We also demonstrate that a mechanism which is commonly employed —setting $\mathrm{MAC}_f(x_1 \cdots x_m) = f^{(m+1)}(x_1 \cdots x_m\, m)$— does *not* work to generate a secure message authentication code.

Pseudorandom functions are basic tools in cryptography. In addition to shedding light on the security of the CBC MAC our work provides a method of building secure PRFs which can be used in a wide range of applications, in the following way.

Practice readily provides PRFs on fixed input lengths, in the form of block ciphers like DES. On the other hand PRFs are very useful in applications, but one typically needs PRFs on long strings. Our CBC Lemma provides a provably-good way of extending the basic PRFs (which work on short inputs) to PRFs which work on longer inputs. It was based on these facts that PRFs were suggested by [3] as the tools of choice for practical applications, particularly entity authentication and key distribution.

1.5 History and related work

The lack of any theorem linking the security of f to that of $f^{(m)}$ lead previous users of the CBC-MAC to view $f^{(m)}$, and not f, as the basic primitive. Thus for example in Bird et. al. [5], when the authors require a practical message authentication code in order to achieve their higher-level goal of entity authentication they made appropriate assumptions about the CBC MAC.

The cryptanalytic approach to the problem of the security of the CBC MAC is to attack the CBC MAC construction for a particular block cipher F. Refer to [17] for an attempt to directly attack the DES CBC MAC using differential cryptanalysis.

Another approach to studying MACs is rooted in the examination of protocols which use them. Stubblebine and Gligor [20] find flaws in the use of the CBC MAC in some well-known protocols. But as the authors make clear, it is not the CBC MAC itself which is at fault for the indicated protocol failures—it is the manner in which the containing protocols incorrectly embed the CBC MAC. The authors go on to correct some protocols by having them properly use the CBC MAC.

Cipher block chaining is not the only method of constructing a MAC out of a block cipher. Amongst the other methods that have been proposed we note that of [2]. There the authors concern was to provide a construction which, unlike cipher-block chaining, is parallelizable. Their constructions are simple and efficient. The security is analyzed exactly and the bounds achieved are actually somewhat better than what we prove here for cipher block chaining.

1.6 Discussion and open questions

Block ciphers like DES are in fact *permutations*. One open question is whether the permutativity of the block cipher could be exploited to prove a stronger reduction than that in our main lemma. The fact that one typically outputs a number of bits $s < l$ seems relevant and useful in strengthening the bounds that would otherwise be achieved.

Feige and Naor [7] observe that the dependence on q in the bound in Lemma 1 is optimal up to a constant: they can show that there is an adversary who achieves an advantage of $O(mq^2 \cdot 2^{-l})$. An open question is whether our analysis can be tightened to meet this lower bound, or whether the latter can be improved to meet our upper bound of $O(m^2q^2 \cdot 2^{-l})$.

2 Preliminaries

A *finite function family* is a finite set of strings, called keys, each of which names a function according to a fixed and specified manner. To pick a function f at random from a finite function family F means to pick a random key and let f be the corresponding function. Note that two keys can name the same function.

We let $\mathcal{R}_{a \to l}$ denote the set of all functions from $\{0,1\}^a$ to $\{0,1\}^l$. The name of each function $f \in \mathcal{R}_{a \to l}$ is the string which describes its truth table. We let $\mathcal{R}_{*l \to l}$ denote the set of all functions from $\cup_{1 \leq i < 2^l} \{0,1\}^{il}$ to $\{0,1\}^l$. The name of a function $f \in \mathcal{R}_{*l \to l}$ is the string which describes its truth table.

Let A be a Turing machine with access to an oracle $\mathcal{O}$. We say that A is a q-adversary if A makes at most q queries to $\mathcal{O}$. We say that A is a (t,q)-adversary if A runs in at most t steps and makes at most q queries to $\mathcal{O}$.

If F is a finite function family we write $E[A^F] = \Pr[A^f = 1]$ for the probability that A answers 1 when A's oracle is selected to be a random function from F. (If A is probabilistic, the probability is over A's coins as well.) We let

$$advantage_A(F, F') = \frac{E[A^F] - E[A^{F'}]}{2}$$

denote the advantage of A in distinguishing F from F'. Here, following [9], we are considering the following game, or statistical test. Algorithm A is provided as oracle a function g chosen at random from either F or from F', the choice being made at random according to a bit b. The algorithm is trying to predict b. The advantage is $\Pr[A^g = b] - 1/2$, the amount that the probability of A is correct is bounded away from the guessing probability $1/2$.

For $0 \leq \epsilon \leq 1$, we say that A ϵ-distinguishes F and F' if $advantage_A(F, F') \geq \epsilon$. We say that A (q, ϵ)-distinguishes F and F' if A is a q-adversary and A ϵ-distinguishes F and F'. We say that A (t, q, ϵ)-distinguishes F and F' if A is a (t, q)-adversary which ϵ-distinguishes F and F'. Let $d^q(F, F')$ be the supremum of all numbers ϵ, $0 \leq \epsilon \leq 1$, such that there exists a q-adversary A that ϵ-distinguishes F and F'.

Suppose $F \subseteq \mathcal{R}_{a \to b}$. We say that A (q, ϵ)-breaks F if A is a q-adversary who (q, ϵ)-distinguishes F from $\mathcal{R}_{a \to b}$. We say that A (t, q, ϵ)-breaks F if A is a (t, q)-adversary who (t, q, ϵ)-distinguishes F from $\mathcal{R}_{a \to b}$.

Let $m > 0$ be a number and let F be a finite function family whose keys name functions in $\mathcal{R}_{l \to l}$. We define the finite function family $F^{(m)}$ the keys of which are precisely F but the interpretation of $f^{(m)} \in F^{(m)}$ is as a function $f^{(m)} \in \mathcal{R}_{ml \to l}$ defined by

$$f^{(m)}(x_1 \cdots x_m) = f(f(\cdots f(f(x_1) \oplus x_2) \oplus \cdots \oplus x_{m-1}) \oplus x_m) .$$

Let $m > 0$ be a number and let F be a finite function family whose keys name functions in $\mathcal{R}_{l \to l}$. The m-fold CBC-MAC based on F, denoted CBC-MAC^{F,m}, is defined by CBC-MAC$_f^{F,m}(x_1 \ldots x_m) = f^{(m)}(x_1 \ldots x_m)$ for all $f \in F$ and all $x_1, \ldots x_m \in \{0, 1\}^l$.

3 The CBC Lemma: Information-theoretic case

3.1 Statement

The information-theoretic case of the CBC lemma considers an adversary A of unrestricted computational power. She is faced with the following problem. She is given an oracle to a function g chosen in one of the following ways: either g is a random function of ml bits to l bits; or $g = f^{(m)}$ for a random function f of l bits to l bits. The choice between these two possibilities is made according to a hidden coin flip. What is A's advantage in figuring out which type of oracle she has? The answer is specified by a tradeoff which says how A's advantage grows with the number q of queries she makes. The formal statement is made in terms of the distance function defined previously.

Lemma 1. (CBC Lemma: Information-theoretic case.) Let $l \geq 1$ and $m \geq 1$ be integers. Suppose $qm \leq 2^{(l+1)/2}$. Then:

$$d^q \left(\mathcal{R}_{ml \to l},\ \mathcal{R}_{l \to l}^{(m)} \right) \leq \frac{3q^2m^2}{2^{l+1}} .$$

In other words an adversary making $q \leq 2^{(l+1)/2}/m$ queries cannot hope to have an advantage exceeding $3q^2m^2 \cdot 2^{-l-1}$. Thinking of m as being small compared to $q, 2^l$, this means that as long as the total number of queries is roughly $\sqrt{2^l}$, the incremental advantage from the adversary's i-th query is bounded by roughly $i \cdot 2^{-l}$.

3.2 Proof

Fix an adversary A. Since we are not restricting computation time a standard argument shows that we may assume without loss of generality that A is deterministic. The bulk of the proof will be devoted to seeing what happens when A is supplied with a g be chosen at random from $\mathcal{R}_{l \to l}^{(m)}$. We begin with some definitions. The connection of these definitions to the game we are considering will be made later.

Query sequences and labelings. Call the 2^l-ary rooted tree of depth m the *full* tree. A sequence $x_1 \dots x_i$ of l-bit strings $(1 \leq i \leq m)$ names a node at depth i in the natural way. The root is denoted Λ. A sequence of distinct non-root nodes $X_1, \dots, X_n$ is a *query sequence* if for every i there is a $j < i$ such that the parent of X_i is either X_j or Λ. The *query tree* associated to a query sequence $X_1, \dots, X_n$ is the (rooted) subtree of the full tree induced by the nodes $\{\Lambda, X_1, \dots, X_n\}$; it consists of a collection of root emanating paths. Nodes at depth m are called *border* nodes.

A labeling of a query sequence is a map assigning an l-bit string to each node (equivalently, a map assigning an l-bit string to each non-root node of the query tree). A function $f\colon \{0,1\}^l \to \{0,1\}^l$ induces a labeling Z_f of a query sequence $X_1, \dots, X_n$ as follows. Let Λ, x_1, x_1x_2, $\dots$, $x_1x_2 \dots x_i$ be any root emanating path in the query tree. Set

$$Z_f(x_1) = f(x_1) \ ; \ Z_f(x_1 \dots x_j) \ = \ f(Z_f(x_1 \dots x_{j-1}) \oplus x_j)) \quad \text{for } j = 2, \dots, i .$$

A labeling Z of $X_1, \dots, X_n$ induces another labeling Y of the same query sequence, defined as follows. On any root emanating path x_1, x_1x_2, $\dots$, $x_1x_2 \dots x_i$ set:

$$Y(x_1) = x_1 \ ; \ Y(x_1 \dots x_j) \ = \ Z(x_1 \dots x_{j-1}) \oplus x_j \quad \text{for } j = 2, \dots, i .$$

The following is the fundamental relation between Z_f and its induced labeling Y_f:

$$Z_f(X_i) \ = \ f(Y_f(X_i)) \quad \text{for all } i = 1, \dots, n .$$

For this reason, the induced labeling is called the *input* labeling.

A labeling of the query sequence $X_1, \ldots, X_n$ is said to be *collision free* if the n values $Z(X_1), \ldots, Z(X_n)$ are distinct and also the n values $Y(X_1), \ldots, Y(X_n)$ are distinct, where Y is the input labeling induced by Z. A border labeling of a query sequence is a map assigning an l-bit string to each border node in the query tree. A labeling Z is consistent with a border labeling $\hat{Z}$ if the two agree on the border nodes.

A NEW VIEW OF THE GAME. A query $x_1 \ldots x_m$ of the adversary to the g-oracle can be thought of as specifying a root to border path in the full tree. Now imagine a slightly different game in which the adversary has more power. She can sequentially make qm queries, each a node in the full tree, with the restriction that her queries form a query sequence $X_1, \ldots, X_{qm}$ according to the above definition. She receives no answer to queries which are internal nodes of the full tree, but when she queries a border node she receives its Z_f value. It is easy to see that it suffices to prove the lemma for this game.

THE BASIC RANDOM VARIABLES. The query sequence, its Z_f-labeling, and the values returned to the adversary are all random variables over the random choice of $f \in \mathcal{R}_{l \to l}$. We will denote by $\mathsf{X}_1, \ldots, \mathsf{X}_{qm}$ the random variables which are the queries of A. We will denote by Z_n (resp. $\hat{\mathsf{Z}}_n$) the labeling of $\mathsf{X}_1, \ldots, \mathsf{X}_n$ (resp. of the border nodes of the query tree associated to $\mathsf{X}_1, \ldots, \mathsf{X}_n$) specified by Z_f. The input labeling induced by Z_n is denoted Y_n. The *view* of A after her n-th query is the random variable $\mathsf{View}_n = (\mathsf{X}_1, \ldots, \mathsf{X}_n ; \hat{\mathsf{Z}}_n)$. The term labeling usually refers to a value of Z_n; when we want to discuss the induced labeling we talk of the induced or input labeling.

EQUI-PROBABILITY OF COLLISION-FREE LABELINGS. The following lemma fixes the number n of queries that A has made. It then fixes a particular view $(X_1, \ldots, X_n ; \hat{Z})$ of A. It now examines the distribution on labelings from the point of view of A. It says that as far as A can tell, all collision free labelings of $X_1, \ldots, X_n$ consistent with her current view are equally likely.

Lemma 2. Let $1 \le n \le qm$ and let $X_1, \ldots, X_n$ be a query sequence. Let Z_n^1 and Z_n^2 be collision free (output) labelings of $X_1, \ldots, X_n$ which are consistent with a border labeling $\hat{Z}_n$ of $X_1, \ldots, X_n$. Then

$$\Pr\left[\mathsf{Z}_n = Z_n^1 \mid \mathsf{View}_n = (X_1, \ldots, X_n ; \hat{Z}_n)\right]$$
$$= \Pr\left[\mathsf{Z}_n = Z_n^2 \mid \mathsf{View}_n = (X_1, \ldots, X_n ; \hat{Z}_n)\right] .$$

The proof is given in Appendix A.1.

MORE DEFINITIONS. Let $X_1, \ldots, X_n$ be a query sequence. We will discuss labelings $\mathbf{z}$ which assign values only to some specified subset S of this sequence. The input labeling induced by $\mathbf{z}$ assigns values to all nodes of $X_1, \ldots, X_n$ which are at level one and all nodes whose parents are in S. We can discuss collision freeness of such labelings, or their consistency with a border labeling, in the usual way. We denote by Z_n^S the labeling of S given by restricting Z_n to S. Let $\mathsf{ColFree}(Z)$ be true if labeling Z is collision free.

UNPREDICTABILITY OF INTERNAL LABELS. The following lemma fixes the number n of queries that A has made, as well as a particular view $X_1, \ldots, X_n$; $\hat{Z}$ of A. It now makes the assumption that the current labeling Z_n is collision free; think of this fact as being known to A. Given all this, it examines the distribution on labels from the point of view of A. Some labels are known: for example, the Z_n values of border nodes and the Y_n values of nodes at depth one. The lemma says that all other labels are essentially unpredictable. First, it considers a node $x_1 \ldots x_i x_{i+1}$ which is at depth at least two, and says that even given the output labels (i.e. Z_n values) of all nodes except its parent $x_1 \ldots x_i$, the Y_n value of $x_1 \ldots x_i x_{i+1}$ is almost uniformly distributed. Second, it considers a node $x_1 \ldots x_i$ which is not a border node, and says that even given the output labels of all other nodes, the Z_n value of $x_1 \ldots x_i$ is almost uniformly distributed. For technical reasons the lemma requires a bound on the number n of queries that have been made.

Lemma 3. Let $1 \leq n \leq qm-1$ and suppose $n^2/4 + n - 1 \leq 2^l/2$. Let $X_1, \ldots, X_n$ be a query sequence and let $\hat{Z}$ be a labeling of the border nodes of $X_1, \ldots, X_n$. Let

$$\Pr_n[\cdot] = \Pr\left[\cdot \mid \mathsf{View}_n = (X_1, \ldots, X_n ; \hat{Z}) \wedge \mathsf{ColFree}(\mathsf{Z}_n)\right] .$$

Suppose $x_1 \ldots x_i \in \{X_1, \ldots, X_n\}$ is a non-border node and let $S = \{X_1, \ldots, X_n\} - \{x_1 \ldots x_i\}$. Suppose $\mathbf{z}$: $S \rightarrow \{0,1\}^l$ is a collision free labeling of S which is consistent with $\hat{Z}$.

(1) Let $x_1 \ldots x_i x_{i+1} \in S$ be a child of $x_1 \ldots x_i$. Then for any $y^* \in \{0,1\}^l$:

$$\Pr_n\left[\mathsf{Y}_n(x_1 \ldots x_i x_{i+1}) = y^* \mid \mathsf{Z}_n^S = \mathbf{z}\right] \leq 2 \cdot 2^{-l} .$$

(2) For any $z^* \in \{0,1\}^l$:

$$\Pr_n\left[\mathsf{Z}_n(x_1 \ldots x_i) = z^* \mid \mathsf{Z}_n^S = \mathbf{z}\right] \leq 2 \cdot 2^{-l} .$$

The proof is given in Appendix A.2.

BOUNDING THE PROBABILITY OF COLLISIONS. The following lemma fixes the number n of queries that A has made, as well as a particular view $X_1, \ldots, X_n$; $\hat{Z}$ of A. It now makes the assumption that the current labeling Z_n is collision free; think of this fact as being known to A. Given all this, it considers A's adding a new node X_{n+1} to the tree. It says that the labeling is likely to retain its collision freeness; that is, Z_{n+1} is collision free with high probability. The same technical condition on n as in the previous lemma is required.

Note that X_{n+1} is determined by $X_1, \ldots, X_n$; $\hat{Z}$. The value $\mathsf{Z}_{n+1}(X_{n+1})$ has not yet been returned to A, and it makes sense to discuss the distribution of this value given $X_1, \ldots, X_n$; $\hat{Z}$.

Lemma 4. Let $1 \leq n \leq qm-1$ and suppose $n^2/4 + n - 1 \leq 2^l/2$. Let $X_1, \ldots, X_n$ be a query sequence and let $\hat{Z}$ be a labeling of the border nodes of $X_1, \ldots, X_n$. Then

$$\Pr\left[\neg\mathsf{ColFree}(\mathsf{Z}_{n+1}) \mid \mathsf{View}_n = (X_1, \ldots, X_n ; \hat{Z}) \wedge \mathsf{ColFree}(\mathsf{Z}_n)\right] \leq 3n \cdot 2^{-l} .$$

The proof is given in Appendix A.3.

CONCLUDING THE PROOF. We now complete the proof of Lemma 1 given the lemmas above. We need to show that $advantage_A(\mathcal{R}_{ml\to l}, \mathcal{R}_{l\to l}^{(m)}) \leq 3q^2m^22^{-l-1}$. We first consider running A in the above game, with $g = f(m)$ for f chosen at random from $\mathcal{R}_{l\to l}]$. As long as the current labeling Z_n which A has is collision-free, the value of a border node returned to A is a random l bit string distributed independently of anything else. Thus the distribution on A's view is the same as if she were replied to by a function from $\mathcal{R}_{l\to l}$. On the other hand if the labeling Z_n has a collision we pessimistically declare that A has won and stop the game. Thus, if $\Pr_{\mathcal{R}}[\,\cdot\,]$ denotes the probability function when g is drawn randomly from $\mathcal{R}_{ml\to l}$ and $\Pr_{\mathcal{C}}[\,\cdot\,]$ denotes the probability function when g is drawn randomly from $\mathcal{R}_{l\to l}^{(m)}$ then for each $b \in \{0,1\}$ we have

$$\begin{aligned}
&\Pr_{\mathcal{R}}[A^g = b] \\
&\leq \Pr_{\mathcal{C}}[A^g = b \mid \mathsf{ColFree}(\mathsf{Z}_{qm})] + \Pr_{\mathcal{R}}[\neg\mathsf{ColFree}(\mathsf{Z}_{qm})] \\
&\leq \Pr_{\mathcal{C}}[A^g = b \mid \mathsf{ColFree}(\mathsf{Z}_{qm})] \\
&\qquad + \textstyle\sum_{n=1}^{qm-1} \Pr_{\mathcal{R}}[\neg\mathsf{ColFree}(\mathsf{Z}_{n+1}) \mid \mathsf{ColFree}(\mathsf{Z}_n)]\ .
\end{aligned}$$

The first term in the above equals $\Pr_{\mathcal{C}}[A^g = b]$. One can check that our assumption $qm \leq 2^{(l+1)/2}$ implies $n^2/4 + n - 1 \leq 2^l/2$ for all $n = 1, \ldots, qm-1$. Thus we can apply the previous lemmas to argue that the second term above is bounded by

$$\textstyle\sum_{n=1}^{qm-1} 3n \cdot 2^{-l} \leq 3q^2m^2 \cdot 2^{-l-1}\ .$$

Thus $|\Pr_{\mathcal{R}}[A^g = 1] - \Pr_{\mathcal{C}}[A^g = 1]| \leq 3q^2m^2 \cdot 2^{-l}$. Under our definition, adversary A's advantage is given by half of the last bound.

4 The CBC Lemma: The computational case

Let F be a family of functions of l bits to l bits. Think of it as a family of pseudorandom functions. (For concreteness, we could consider $F = \{\mathrm{DES}_a\}_{|a|=56}$ to be the family of functions specified by the DES algorithm; each individual function is specified by a 56 bit key.) Thus it is "hard" to distinguish a random member of F from a random function of l bits to l bits. With f drawn randomly from F, we want to see how $f^{(m)}$ compares to a random function of ml bits to l bits. We'd like to say that $f^{(m)}$ is also pseudorandom. The lemma that follows implies this. But the actual statement is much stronger. It says exactly how the security of $F^{(m)}$ relates to that of F.

Lemma 5. (CBC Lemma: Computational Case). There is an algorithm U and a constant c as follows. Let $l \geq 1$ and $m \geq 1$ be integers. Let $F \subseteq R^{l\to l}$ be a given function family. Suppose $qm \leq 2^{(l+1)/2}$. Let A be an algorithm which (t, q, ϵ)-breaks $F^{(m)}$. Then $U^A(l, m)$ (t', q', ϵ')-breaks F, where

$$t' = t + cqml\ , \quad q' = qm\ , \quad \text{and} \quad \epsilon' = \epsilon - 3q^2m^2 \cdot 2^{-l-1}\ .$$

The constant c is a small number which depends only on the underlying machine model. One should think of t as being much larger than $cqml$ (this is apparent from the definition of the U below) and so the additive $cqml$ term is effectively irrelevant.

Proof. We may assume $\epsilon > \delta \overset{\text{def}}{=} 3q^2m^2 \cdot 2^{-l-1}$ since otherwise there is nothing to prove. Let A be an adversary which (t, q, ϵ)-distinguishes $\mathcal{R}_{ml \to l}$ from $F^{(m)}$. From the triangle inequality at least one of the following must be true:

(1) A (t, q, δ)-distinguishes $\mathcal{R}_{ml \to l}$ from $\mathcal{R}_{l \to l}^{(m)}$; or

(2) A $(t, q, \epsilon - \delta)$-distinguishes $\mathcal{R}_{l \to l}^{(m)}$ from $F^{(m)}$.

However given the assumed bound on qm and the Information Theoretic CBC Lemma (Lemma 1), the first option is ruled out. Thus the second must be true. To complete the proof we now construct an adversary A' which (t', q', ϵ')-distinguishes $\mathcal{R}_{l \to l}$ from F. A' is given access to an oracle for a function f: $\{0,1\}^l \to \{0,1\}^l$. Observe that A' can compute the function $f^{(m)}$ and doing this at a point $x_1 \cdots x_m$ costs A' a total of m queries to its f-oracle and time proportional to ml. Algorithm A''s procedure is to run A and answer its oracle queries according to $f^{(m)}$. Finally A' takes A's prediction as its own. We leave to the reader to check that

$$\mathit{advantage}_{A'}(\mathcal{R}_{l \to l}, F) = \mathit{advantage}_{A}(\mathcal{R}_{l \to l}^{(m)}, F^{(m)}) .$$

This completes the proof. □

5 Length Variability

For simplicity, let us assume throughout this section that strings to be authenticated have length which is a multiple of l bits. This restriction is easy to dispense with by using simple and well-known padding methods: for example, always append a "1" and then append the minimal number of 0's to make the string a multiple of l bits.

THE CBC MAC DOESN'T HANDLE VARIABLE-LENGTH INPUTS. The CBC MAC does not directly give a method to authenticate messages of variable input lengths. In fact, if the length of strings is allowed to vary, it is easy to "break" the basic CBC MAC construction. (This fact is well-known.) As an example, if you request $f_a^{(1)}$ of b, getting back t_b, and then you request $f_a^{(1)}(t_b)$, getting back t_{t_b}, then you have just learned the authentication tag $f_a^{(2)}(b\ 0) = t_{t_x}$ for $b\ 0$—a string for which you have not asked the authentication tag.

APPENDING THE LENGTH DOESN'T WORK. One possible attempt to authenticate messages of varying lengths is to append to each string $x = x_1 \cdots x_m$ the number m, properly encoded as the final l-bit block and then CBC MAC the resulting string $m+1$ blocks. (Of course this imposes a restriction that $m < 2^l$, not likely to be a serious concern.) Let us define $\bar{f}^*(x_1 \cdots x_m) = f^{(m+1)}(x_1 \cdots x_m\ m)$.

We show that $\bar{f}^*$ is not a secure MAC. Take arbitrary l-bit words b, b' and c, $b \neq b'$. It is easy to check that given

(1) $t_b = \bar{f}^*(b)$,
(2) $t_{b'} = \bar{f}^*(b')$, and
(3) $t_{b1c} = \bar{f}^*(b\ 1\ c)$

the adversary has in hand $\bar{f}^*(b'\ 1\ t_b \oplus t_{b'} \oplus c)$ —the authentication tag of a string she has (almost certainly) not asked about before— since this is precisely t_{b1c}.

METHODS WHICH DO WORK. Despite the failure of the method which appends the message length, there are many methods which are almost as simple and which work correctly. We describe three. In each, let F be a block cipher on l bits.

(1) *Input-length key separation.* Set $\dot{f}^*(x)$, where $x = x_1 \cdots x_m$, to be $f_{a_m}^{(m)}(x)$, where $a_m = F_a(m)$. The corresponding finite function family $\dot{F}^*$ is not only a MAC, it can be shown to be computationally close to $\mathcal{R}_{*l \to l}$

(2) *Two-step MAC.* Set $\text{MAC}_a(x)$, where $x = x_1 \cdots x_m$, to be $\langle f_{a'}^{(m)}(x), f_{a''}^{(2)}(m\ f_{a'}^{(m)}(x))\rangle$, where $a' = f_a(0)$ and $a'' = f_a(1)$.

(3) *Length-prepend MAC.* Define $f^*(x_1 \cdots x_m) = f^{(m+1)}(m\ x_1 \cdots x_m)$. The corresponding finite function family F^* is not only a MAC, it can be shown to be computationally close to $\mathcal{R}_{*l \to l}$.

The third of these claims has the most involved proof. We know of no argument which does not involve modifying and verifying that the proof of the CBC Lemma goes through after making this extension.

6 From PRFs to MACs

Recall that justifying the CBC-MAC was the primary motivation of this paper. To formally complete this project we need one more step—to show that pseudorandom functions make good message authentication codes. As we remarked in the introduction the reduction is standard [9, 10]. But we need to see what is the exact security. The following shows that the reduction is almost tight— security hardly degrades at all.

Let G be a finite function family whose keys name functions in $\mathcal{R}_{k \to l}$. Let MAC^G be defined by $\text{MAC}_g^G(y) = g(y)$ for all $g \in G$ and all $y \in \{0,1\}^k$.

Security of MAC^G is discussed via the notion of chosen message attack [11]. An adversary B attacks MAC^G via the following experiment. Pick g at random from G and provide $\text{MAC}_g^G(\cdot)$ to B as an oracle. Suppose B makes q oracle queries and runs for time t, halting with an output (y, σ), where $y \in \{0,1\}^k$ and y is different from any string which B has queried of its oracle. We say that B is successful if $\text{MAC}_g^G(y) = \sigma$. We say that B (t, q, ϵ)-breaks MAC^G if it is successful with probability at least ϵ. The Proposition that follows is the exact security version of the standard reduction of [9, 10].

Proposition 6. There is an algorithm U and a constant c as follows. Let $l \geq 1$ and $k \geq 1$ be integers. Let $G \subseteq R^{k \to l}$ be a given function family. Suppose $q < 2^k$.

Let B be an algorithm which (t,q,ϵ)-breaks MAC^G. Then $U^B(k,l)$ (t',q',ϵ')-breaks G, where

$$t' = t + cq(k+l)\,, \quad q' = q+1\,, \quad \text{and} \quad \epsilon' = (\epsilon - 2^{-l})/2\,.$$

Proof. We provide A such that $advantage_A(G, \mathcal{R}_{k\to l}) \geq \epsilon'$. A has oracle access to g: $\{0,1\}^k \to \{0,1\}^l$. It runs B and answers an oracle query x of B by invoking its own oracle to return $g(x)$. B eventually outputs (y,σ). A makes a last oracle query of y to get $\sigma^* = g(y)$. It outputs 1 if $\sigma^* = \sigma$ and 0 otherwise. It is easy to see that, on the one hand, $E[A^{\mathcal{R}_{k\to l}}] \leq 2^{-l}$ and, on the other hand, $E[A^G]$ is just the success probability of B and hence is at least ϵ. So the advantage is as claimed. The program of A can be easily implemented in the stated complexity by a machine with oracle access to B. □

Let $m > 0$ be a number and let F be a finite function family whose keys name functions in $\mathcal{R}_{l\to l}$. Combining Lemma 5 with the above proposition tells us exactly how secure is the CBC-MAC $\text{CBC-MAC}^{F,m}$ based on F.

Acknowledgments

We thank Uri Feige and Moni Naor for their assistance in the proof of Lemma 1, and also for comments on the paper.

References

1. ANSI X9.9, "American National Standard for Financial Institution Message Authentication (Wholesale)," American Bankers Association, 1981. Revised 1986.
2. M. Bellare, R. Guérin and P. Rogaway, "Fully parallelizable message authentication," Manuscript, April 1994.
3. M. Bellare and P. Rogaway, "Entity authentication and key distribution," *Advances in Cryptology - Crypto 93 Proceedings,* Lecture Notes in Computer Science Vol. 773, Springer-Verlag, D. Stinson, ed., 1994.
4. M. Bellare and P. Rogaway, "Optimal asymmetric encryption," *Advances in Cryptology - Eurocrypt 94 Proceedings,* 1994.
5. R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva and M. Yung, "Systematic design of two-party authentication protocols," *Advances in Cryptology - Crypto 91 Proceedings,* Lecture Notes in Computer Science Vol. 576, Springer-Verlag, J. Feigenbaum, ed., 1991.
6. S. Even, O. Goldreich and S. Micali, "On-line/Off line digital signatures," Manuscript, March 1994. Preliminary version in Crypto 89.
7. U. Feige and M. Naor, Private communication, April 1994.
8. O. Goldreich and L. Levin, "A hard predicate for all one-way functions," *Proceedings of the Twenty First Annual Symposium on the Theory of Computing,* ACM, 1989.
9. O. Goldreich, S. Goldwasser and S. Micali, "How to construct random functions," *Journal of the ACM,* Vol. 33, No. 4, 210–217, (1986).

10. O. Goldreich, S. Goldwasser and S. Micali, "On the cryptographic applications of random functions," *Advances in Cryptology – Crypto 84 Proceedings,* Lecture Notes in Computer Science Vol. 196, Springer-Verlag, B. Blakley, ed., 1985.
11. S. Goldwasser, S. Micali and R. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM Journal of Computing,* 17(2):281–308, April 1988.
12. R. Impagliazzo, L. Levin and M. Luby, "Pseudo-random generation from one-way functions," *Proceedings of the Twenty First Annual Symposium on the Theory of Computing,* ACM, 1989.
13. ISO/IEC 9797, "Data cryptographic techniques – Data integrity mechanism using a cryptographic check function employing a block cipher algorithm," 1989.
14. T. Leighton and S. Micali, "Provably fast and secure digital signature algorithms based on secure hash functions," Manuscript, March 1993.
15. M. Luby and C. Rackoff, "How to construct pseudorandom permutations from pseudorandom functions," *SIAM J. Computation,* Vol. 17, No. 2, April 1988.
16. M. Luby and C. Rackoff, "A study of password security," *Advances in Cryptology – Crypto 87 Proceedings,* Lecture Notes in Computer Science Vol. 293, Springer-Verlag, C. Pomerance, ed., 1987.
17. K. Ohta and M. Matsui, "Differential attack on message authentication codes," *Advances in Cryptology – Crypto 93 Proceedings,* Lecture Notes in Computer Science Vol. 773, Springer-Verlag, D. Stinson, ed., 1994.
18. R. Rivest, "The MD5 message-digest algorithm," IETF Network Working Group, RFC 1321, April 1992.
19. C. Schnorr, "Efficient identification and signatures for smart cards," *Advances in Cryptology – Crypto 89 Proceedings,* Lecture Notes in Computer Science Vol. 435, Springer-Verlag, G. Brassard, ed., 1989.
20. S. Stubblebine and V. Gligor, "On message integrity in cryptographic protocols," Proceedings of the 1992 IEEE Computer Society Symposium on Research in Security and Privacy. May 1992.

A Proof of Lemmas

We present here the proofs of the lemmas in Section 3.2.

A.1 Proof of Lemma 2

The proof is by induction on n. The Lemma holds vacuously for $n = 1$. Assuming the lemma for $1, \ldots, n-1$ we now prove it for n. Let Z^i_{n-1} be the restriction of Z^i_n to $X_1, \ldots, X_{n-1}$ $(i = 1, 2)$. Let $\hat{Z}_{n-1}$ be the restriction of $\hat{Z}_n$ to the border nodes of $X_1, \ldots, X_{n-1}$. Let V_i be the event $\text{View}_i = (X_1, \ldots, X_i\ ;\ \hat{Z}_i)$ and let $\Pr_i[\cdot] = \Pr[\cdot \mid V_i]$, for $i = n-1, n$. Let Y^i_j denote the input labeling induced by Z^i_j for $j = n-1, n$ and $i = 1, 2$. We consider two cases.

Case 1. X_n is not a border node.

For $i = 1, 2$ we have:

$$\begin{aligned}
&\Pr_n\left[\,\mathsf{Z}_n = Z^i_n\,\right] \\
&= \Pr_{n-1}\left[\,\mathsf{Z}_n = Z^i_n\,\right] \\
&= \Pr_{n-1}\left[\,\mathsf{Z}_{n-1} = Z^i_{n-1}\,\right] \cdot \Pr_{n-1}\left[\,\mathsf{Z}_n(X_n) = Z^i_n(X_n) \mid \mathsf{Z}_{n-1} = Z^i_{n-1}\,\right] \\
&= \Pr_{n-1}\left[\,\mathsf{Z}_{n-1} = Z^i_{n-1}\,\right] \cdot 2^{-l}\,.
\end{aligned}$$

The proof is concluded by using the inductive hypothesis. We now justify the above equations. Since X_n is not a border node, it is determined by $X_1, \ldots, X_{n-1}; \hat{Z}_{n-1}$. This means that $\Pr_n[\,\cdot\,]$ equals $\Pr_{n-1}[\,\cdot\,]$ which justifies the first line. The second is just conditioning. Since Z^i_n is collision free, Y^i_n differs from all the points $Y^i_{n-1}(X_1), \ldots, Y^i_{n-1}(X_{n-1})$ on which the underlying randomly chosen f has been evaluated so far. But $\mathsf{Z}_n(X_n) = f(Y^i_n(X_n))$. So the second term in the product in the second line above is indeed 2^{-l}.

Case 2. X_n is a border node.

Both Z^1_n and Z^2_n are by assumption consistent with $\hat{Z}_n$. But since X_n is a border node, the value $\hat{\zeta} \overset{\text{def}}{=} \mathsf{Z}_n(X_n)$ is contained in $\hat{Z}_n$, and $\hat{\zeta} = Z^1_n(X_n) = Z^2_n(X_n)$. Now for $i = 1, 2$ we have:

$$\begin{aligned}
&\Pr_n\left[\,\mathsf{Z}_{n-1} = Z^i_{n-1}\,\right] \\
&= \Pr_{n-1}\left[\,\mathsf{Z}_{n-1} = Z^i_{n-1} \mid \mathsf{Z}_n(X_n) = \hat{\zeta}\,\right] \\
&= \Pr_{n-1}\left[\,\mathsf{Z}_n(X_n) = \hat{\zeta} \mid \mathsf{Z}_{n-1} = Z^i_{n-1}\,\right] \cdot \frac{\Pr_{n-1}\left[\,\mathsf{Z}_{n-1} = Z^i_{n-1}\,\right]}{\Pr_{n-1}\left[\,\mathsf{Z}_n(X_n) = \hat{\zeta}\,\right]} \\
&= 2^{-l} \cdot \frac{\Pr_{n-1}\left[\,\mathsf{Z}_{n-1} = Z^i_{n-1}\,\right]}{\Pr_{n-1}\left[\,\mathsf{Z}_n(X_n) = \hat{\zeta}\,\right]}\,.
\end{aligned}$$

The first equality is because the events V_n and $V_{n-1} \wedge (\mathsf{Z}_n(X_n) = \hat{\zeta})$ are the same. The second line is Bayes rule. That the first term of the product in the second line above is indeed 2^{-l} is argued as in Case 1 based on the fact that Z^i_n is collision free. Now note the denominator in the fraction above is independent of $i \in \{1, 2\}$. Thus, applying the inductive hypothesis, we conclude

$$\Pr_n\left[\,\mathsf{Z}_{n-1} = Z^1_{n-1}\,\right] \;=\; \Pr_n\left[\,\mathsf{Z}_{n-1} = Z^2_{n-1}\,\right]. \tag{1}$$

Now for $i = 1, 2$:

$$\begin{aligned}
\Pr_n\left[\,\mathsf{Z}_n = Z^i_n\,\right] &= \Pr_n\left[\,\mathsf{Z}_{n-1} = Z^i_{n-1}\,\right] \cdot \Pr_n\left[\,\mathsf{Z}_n(X_n) = \hat{\zeta} \mid \mathsf{Z}_{n-1} = Z^i_{n-1}\,\right] \\
&= \Pr_n\left[\,\mathsf{Z}_{n-1} = Z^i_{n-1}\,\right] \cdot 1\,.
\end{aligned}$$

That the second term in the above product is 1 is because V_n contains $\hat{\zeta}$ as the value of $\mathsf{Z}_n(X_n)$. The proof for this case is concluded by applying Equation 1.

A.2 Proof of Lemma 3

Let $x_1 \dots x_i x_{i+1}^u$ $(u = 1, \dots, s)$ be the children of $x_1 \dots x_i$. Let

$$\text{CHILDREN}(x_1 \dots x_i) = \{x_1 \dots x_i x_{i+1}^1, \dots, x_1 \dots x_i x_{i+1}^s\} .$$

Let y: $\{X_1, \dots, X_n\} - \text{CHILDREN}(x_1 \dots x_i) \to \{0,1\}^l$ be the input labeling induced by $\mathbf{z}$. We prove the two claims in turn.

Proof of (1). Let's begin by giving some intuition for the proof. We observe that with z given, if we assign an input label $y \in \{0,1\}^l$ to $x_1 \dots x_i x_{i+1}$ then the value of Z_n at the parent node $x_1 \dots x_i$ is determined; given this, the values of Y_n at the other children of $x_1 \dots x_i$ are also determined. Thus, both Z_n and Y_n are now fully determined for all nodes $X_1, \dots, X_n$. We will show that there is a large set $S(\mathrm{z})$ of these y values for which the determined labeling is collision free. Moreover, all collision free labelings have this form and are equally likely by Lemma 2; thus as far as A can tell, the value at $x_1 \dots x_i x_{i+1}$ is equally likely to be anything from the set $S(\mathrm{z})$. The formal proof follows.

Assume wlog that $x_1 \dots x_i x_{i+1}^1 = x_1 \dots x_i x_{i+1}$. Let $y \in \{0,1\}^l$ be some fixed string. Now define the labeling $Z_{\mathbf{z},y}$: $\{X_1, \dots, X_n\} \to \{0,1\}^l$ by:

$$Z_{\mathbf{z},y}(X_j) = \begin{cases} \mathrm{z}(X_j) & \text{if } X_j \neq x_1 \dots x_i \\ y \oplus x_{i+1}^1 & \text{otherwise.} \end{cases}$$

Let $Y_{\mathbf{z},y}$ denote the input labeling induced by $Z_{\mathbf{z},y}$, and observe that it is given by

$$Y_{\mathbf{z},y}(X_j) = \begin{cases} \mathrm{y}(X_j) & \text{if } X_j \notin \text{CHILDREN}(x_1 \dots x_i) \\ y \oplus x_{i+1}^1 \oplus x_{i+1}^u & \text{if } X_j = x_1 \dots x_i x_{i+1}^u \text{ for some } 1 \le u \le s. \end{cases}$$

Let $S(\mathrm{z})$ be the set of all strings y such that $Z_{\mathbf{z},y}$ is a collision free labeling. We leave to the reader to check that $y \notin S(\mathrm{z})$ if and only if one of the following two conditions is satisfied:

(1) Either $y \oplus x_{i+1}^1 \in \{\, \mathrm{z}(X_j) \;:\; 1 \le j \le n \text{ and } X_j \neq x_1 \dots x_i \,\}$; or

(2) For some $u \in \{1, \dots, s\}$ it is the case that $y \oplus x_{i+1}^1 \oplus x_{i+1}^u \in \{\, \mathrm{y}(X_j) \;:\; 1 \le j \le n \text{ and } X_j \notin \text{CHILDREN}(x_1 \dots x_i) \,\}$.

This implies that $|\{0,1\}^l - S(\mathrm{z})| \le (n-1) + (n-s)s \le n - 1 + n^2/4 \le 2^l/2$. So $|S(\mathrm{z})| \ge 2^l - 2^l/2 \ge 2^l/2$. Now observe that any collision free labeling equals $Z_{\mathbf{z},y}$ for some $\mathbf{z}, y$ as above. Furthermore by Lemma 2 all collision free labelings are equally likely. From this one can prove the desired statement.

Proof of (2). The idea is very similar to the above. This time, observe that with z given, if we assign an output label $z \in \{0,1\}^l$ to $x_1 \dots x_i$ then the values of both Z_n and Y_n are fully determined for all nodes $X_1, \dots, X_n$. We show as before that there is a set $S(\mathrm{z})$ of these z values for which the determined labeling is collision free, and conclude as before using the equi-probability of collision free labelings. The formal proof follows.

Let $z \in \{0,1\}^l$ be some fixed string. Now define the labeling $Z_{\mathbf{z},z}$: $\{X_1,\dots,X_n\} \to \{0,1\}^l$ by:

$$Z_{\mathbf{z},z}(X_j) \;=\; \begin{cases} \mathbf{z}(X_j) \text{ if } X_j \neq x_1\dots x_i \\ z \quad\;\; \text{otherwise.} \end{cases}$$

Let $Y_{\mathbf{z},z}$ denote the input labeling induced by $Z_{\mathbf{z},z}$, and observe that it is given by

$$Y_{\mathbf{z},z}(X_j) \;=\; \begin{cases} \mathbf{y}(X_j) \quad \text{if } X_j \notin \text{CHILDREN}(x_1\dots x_i) \\ z \oplus x_{i+1}^u \text{ if } X_j = x_1\dots x_i x_{i+1}^u \text{ for some } 1 \le u \le s. \end{cases}$$

Let $S(\mathbf{z})$ be the set of all strings z such that $Z_{\mathbf{z},z}$ is a collision free labeling. We leave to the reader to check that $z \notin S(\mathbf{z})$ if and only if one of the following two conditions is satisfied:

(1) Either $z \in \{\, \mathbf{z}(X_j) \;:\; 1 \le j \le n \text{ and } X_j \neq x_1\dots x_i \,\}$; or

(2) For some $u \in \{1,\dots,s\}$ it is the case that $z \oplus x_{i+1}^u \in \{\, \mathbf{y}(X_j) \;:\; 1 \le j \le n \text{ and } X_j \notin \text{CHILDREN}(x_1\dots x_i) \,\}$.

This implies that $|\{0,1\}^l - S(\mathbf{z})| \le (n-1) + (n-s)s \le n - 1 + n^2/4 \le 2^l/2$. So $|S(\mathbf{z})| \ge 2^l/2$. Now observe that any collision free labeling equals $Z_{\mathbf{z},z}$ for some $\mathbf{z}, z$ as above. Furthermore by Lemma 2 all collision free labelings are equally likely. From this one can prove the desired statement.

A.3 Proof of Lemma 4

We'll use the following notation:

$$\Pr_n[\cdot] = \Pr\left[\,\cdot \mid \mathsf{View}_n = (X_1,\dots,X_n\,;\,\hat{Z}) \wedge \mathsf{ColFree}(\mathsf{Z}_n)\right].$$

Case 1. X_{n+1} is at level one.

Let $X_{n+1} = \bar{x}_1$. Note its input label is by definition $\bar{x}_1$. For each $t = 1,\dots,n$ we claim that

$$\Pr_n[\mathsf{Y}_n(X_t) = \bar{x}_1] \;\le\; 2 \cdot 2^{-l}. \tag{2}$$

To see why this is true, consider two cases. First, if X_t is at level one then $\Pr_n[\mathsf{Y}_n(X_t) = \bar{x}_1] = 0$ by definition. On the other hand suppose X_t is at depth at least two. Then $X_t = x_1\dots x_i x_{i+1}$ is the child of some $x_1\dots x_i \in \{X_1,\dots,X_n\}$. Equation 2 now follows by Part 1 of Lemma 3.

Given Equation 2 we can bound the probability of a collision as follows:

$$\begin{aligned} \Pr_n[\neg\mathsf{ColFree}(\mathsf{Z}_{n+1})] \;\le\; & \Pr_n[\bar{x}_1 \in \{\mathsf{Y}_n(X_1),\dots,\mathsf{Y}_n(X_n)\}] + \\ & \Pr_n[\mathsf{Z}_{n+1}(X_{n+1}) \in \{\mathsf{Z}_n(X_1),\dots,\mathsf{Z}_n(X_n)\} \\ & \qquad \mid \bar{x}_1 \notin \{\mathsf{Y}_n(X_1),\dots,\mathsf{Y}_n(X_n)\} \\ \le\; & \frac{2n}{2^l} + \frac{n}{2^l} \le \frac{3n}{2^l}. \end{aligned}$$

Case 2. X_{n+1} is not at level one.

Then $X_{n+1} = x_1 \ldots x_i x_{i+1}$ is the child of some $x_1 \ldots x_i \in \{X_1, \ldots, X_n\}$. Let $S = \{X_1, \ldots, X_n\} - \{x_1 \ldots x_i\}$. We first claim that for any $X_t \in \{X_1, \ldots, X_n\}$:

$$\Pr_n \left[\mathsf{Y}_{n+1}(X_{n+1}) = \mathsf{Y}_n(X_t) \right] \;\leq\; 2 \cdot 2^{-l} \, . \tag{3}$$

To see why this is true, consider two cases. First, if X_t is a sibling of X_{n+1} then

$$\Pr_n \left[\mathsf{Y}_{n+1}(X_{n+1}) = \mathsf{Y}_n(X_t) \right] \;=\; 0$$

by definition. On the other hand suppose X_t is not a sibling of X_{n+1}. Then a collision free labeling $\mathbf{z}$ of S determines $\mathsf{Y}_n(X_t)$. Using this and Part 2 of Lemma 3 we have the following: (The sum here is over all collision free labelings $\mathbf{z}$ of S which are consistent with $\hat{Z}$.)

$$\begin{aligned}
&\Pr_n \left[\mathsf{Y}_{n+1}(X_{n+1}) = \mathsf{Y}_n(X_t) \right] \\
&= \textstyle\sum_{\mathbf{z}} \Pr_n \left[\mathsf{Y}_{n+1}(X_{n+1}) = \mathsf{Y}_n(X_t) \;\middle|\; \mathsf{Z}_n^S = \mathbf{z} \right] \cdot \Pr_n \left[\mathsf{Z}_n^S = \mathbf{z} \right] \\
&= \textstyle\sum_{\mathbf{z}} \Pr_n \left[\mathsf{Z}_n(x_1 \ldots x_i) = \mathsf{Y}_n(X_t) \oplus x_{i+1} \;\middle|\; \mathsf{Z}_n^S = \mathbf{z} \right] \cdot \Pr_n \left[\mathsf{Z}_n^S = \mathbf{z} \right] \\
&\leq \frac{2}{2^l} \cdot \textstyle\sum_{\mathbf{z}} \Pr_n \left[\mathsf{Z}_n^S = \mathbf{z} \right] \;\leq\; \frac{2}{2^l} \, .
\end{aligned}$$

Thus Equation 3 is again established.

Given Equation 3 we can bound the probability of a collision:

$$\begin{aligned}
\Pr_n \left[\neg \mathsf{ColFree}(\mathsf{Z}_{n+1}) \right] \;\leq\; &\Pr_n \left[\mathsf{Y}_{n+1}(X_{n+1}) \in \{ \mathsf{Y}_n(X_1), \ldots, \mathsf{Y}_n(X_n) \} \right] + \\
&\Pr_n [\mathsf{Z}_{n+1}(X_{n+1}) \in \{ \mathsf{Z}_n(X_1), \ldots, \mathsf{Z}_n(X_n) \} \\
&\qquad\qquad | \, \mathsf{Y}_{n+1}(X_{n+1}) \notin \{ \mathsf{Y}_n(X_1), \ldots, \mathsf{Y}_n(X_n) \} \\
&\qquad\quad \leq \frac{2n}{2^l} + \frac{n}{2^l} \leq \frac{3n}{2^l} \, .
\end{aligned}$$

This completes the proof of Lemma 4.