

A Note on the CLRW2 Tweakable Block Cipher Construction

Gordon Procter

Information Security Group,
Royal Holloway, University of London
gordon.procter.2011@rhul.ac.uk

Abstract. In this note, we describe an error in the proof for CLRW2 given by Landecker et al. in their paper at CRYPTO 2012 on the beyond-birthday-bound security for tweakable block ciphers. We are able to resolve the issue, give a new bound for the security of CLRW2, and identify a potential limitation of this proof technique when looking to extend the scheme to provide asymptotic security.

1 Introduction

Tweakable block ciphers were formalised by Liskov, Rivest, and Wagner at CRYPTO 2002 [18]. A tweakable block cipher is a block cipher that admits an additional input (the tweak) to introduce extra variability at the message-block level, in the same way that a nonce or IV introduces variability at the message level. The Hasty Pudding Cipher [23] and Mercy [7] are early examples of ciphers that natively support a tweak; in the case of the Hasty Pudding Cipher, this was called Spice. The tweak may be public and, informally, the security aim is that, for a single key, the permutations indexed by the tweaks are independent; appropriate security notions are defined more formally in Section 2.2.

Liskov et al. [18] describe the syntax and security requirements for tweakable block ciphers and describe two methods for building a tweakable block cipher from a standard block cipher. They also give a method to construct a strong tweakable block cipher¹ from a standard block cipher.

Several tweakable block ciphers have been proposed including Goldenberg et al.'s work on tweaking Luby-Rackoff ciphers [9]; Rogaway's XE and XEX modes [22], which are closely related to the OCB mode of operation [14]; and Threefish [8], which forms part of the hash function Skein [8].

Many block-cipher-based encryption and authentication schemes are secure up to the birthday bound, i.e. provided that fewer than $2^{\frac{n}{2}}$ queries are made, where n is the width of the block cipher (in bits). Beyond this point, one would expect a collision in the input to the block cipher to occur and for this to perhaps leak some information or simplify forgery attempts (as described in, for example, [1,3,4,20]). Several works have studied the security of schemes beyond the birthday bound (for examples, see [10,12]); one related question is how to achieve beyond-birthday-bound security for tweakable block ciphers.

At FSE 2009, Minematsu [21] suggested a method to build a $2n$ -bit tweakable block cipher that provides $\mathcal{O}(2^{\frac{n+m}{2}})$ security from an n -bit block cipher (where m is the size of the tweak). This scheme has a Luby-Rackoff or Feistel structure, but has the disadvantage of only supporting short tweaks and requiring per-invocation block-cipher rekeying which makes changing the tweak computationally expensive.

At CRYPTO 2012 Landecker, Shrimpton, and Terashima [16] continued the study of tweakable block ciphers. Their paper has two main contributions: specifying CLRW2, a tweakable block cipher construction that remains secure beyond the birthday bound (up to approximately

¹ Informally, a strong tweakable block cipher is one that remains secure when the adversary is given also access to a decryption oracle.

$2^{\frac{2n}{3}}$ queries); and giving a proof that TBC-MAC (the analogue of CBC-MAC defined in terms of tweakable block ciphers) is both a PRF and unforgeable. CLRW2 allows arbitrarily long tweaks and does not require excessive rekeying of the block cipher.

The CLRW2 construction was extended by Lampe and Seurin at FSE 2013 [15] who consider longer chains of the LRW2 construction and are able to show (asymptotically in the number of rounds, using a coupling argument) that this provides greater security further beyond the birthday bound than the CLRW2 construction. Their bounds agree with Landecker et al.’s bound in the case of non-adaptive CPA adversaries and they prove a weaker bound for CCA adversaries; they conjecture that the bound proved for non-adaptive CPA adversaries also holds for CCA adversaries.

Contributions

This is an early draft in which we identify a flaw in the proof given by Landecker et al. for CLRW2 [16,17] and are able to resolve it. We describe modifications that correct the proof and give new bounds for the security of CLRW2. We also identify a potential issue which may prevent this proof technique being used to extend these results asymptotically, although this issue does not appear to affect Lampe and Seurin’s results.

Landecker et al. have independently identified and corrected the error in their proof [24]; they correct the proof using a neat coupling argument which results in a tighter bound.

Structure

This note is structured as follows: In Section 2, we introduce the notation, syntax, and security models used in this note; in Section 3, we give a brief summary of Landecker et al.’s scheme and corresponding proof; in Section 4, we describe and demonstrate the error that we have identified in the proof; in Section 5, we give one possible method to fix this error and derive new bounds for the security of CLRW2; and Section 6 contains a description of some issues that may prevent this proof technique being extended asymptotically.

2 Preliminaries

2.1 Notation

For a set \mathcal{X} , we write $x \xleftarrow{\$} \mathcal{X}$ to represent an element x being uniformly sampled from \mathcal{X} . For a bitstring $x \in \{0, 1\}^*$ we use $|x|$ to denote its length. We use $\xi(p)$ to represent a Bernoulli random variable that is 1 with probability p and 0 with probability $1 - p$. For a set $S \subseteq \{0, 1\}^n$ and an element $x \in \{0, 1\}^n$ we define $S \oplus x = \{s \oplus x : s \in S\}$.

We follow the code-based game paradigm of Bellare and Rogaway [5]. We will use X_i to denote plaintext input to a tweakable block cipher and Y_i for ciphertext output. Queries made by an adversary and the value of random variables related to those queries are indexed by a counter i . In games, all boolean flags are initialised to false and arrays are initially undefined at every point.

2.2 Block Cipher Syntax and Security

A block cipher is a family of functions: $E : \mathcal{K} \times \{0, 1\}^n \rightarrow \{0, 1\}^n$, where \mathcal{K} is the keyspace. We require that $E(k, \cdot)$ is a permutation on $\{0, 1\}^n$ for every choice of key k . We will often denote $E(k, \cdot)$ by $E_k(\cdot)$ and the inverse of this permutation by $E_k^{-1}(\cdot)$.

We follow Bellare, Kilian, and Rogaway [2] and say that a block cipher is secure if it is a pseudo-random permutation family (PRP). A pseudo-random permutation family is a set of

permutations $\{E_k | k \in \mathcal{K}\}$ that is indistinguishable from a random permutation family Π also indexed by elements of \mathcal{K} . A strong pseudo-random permutation family (SPRP) is a pseudo-random permutation family that remains indistinguishable from a random permutation family when an adversary is also given access to either a decryption oracle, or to the random permutation's inverse.

Consider an adversary \mathcal{A} and define their SPRP advantage against E as

$$\text{Adv}_E^{\text{sprp}}(\mathcal{A}) = \left| \Pr[1 \leftarrow \mathcal{A}^{(E_k, E_k^{-1})}] - \Pr[1 \leftarrow \mathcal{A}^{(\pi, \pi^{-1})}] \right|$$

where $k \xleftarrow{\$} \mathcal{K}$ and $\pi \xleftarrow{\$} \Pi$. Define $\text{Adv}_E^{\text{sprp}}(q, t)$ to be the maximum advantage that can be realised by an adversary asking no more than q queries and running in time at most t . We say that E is a strong pseudo-random permutation if $\text{Adv}_E^{\text{sprp}}(q, t)$ is sufficiently small.

A tweakable block cipher is a family of functions: $\tilde{E} : \mathcal{K} \times \mathcal{T} \times \{0, 1\}^n \rightarrow \{0, 1\}^n$, where \mathcal{K} is the keyspace and \mathcal{T} is the tweakspace. We require that, for every $k \in \mathcal{K}$ and $T_i \in \mathcal{T}$, $\tilde{E}_k(T_i, \cdot)$ is a permutation on $\{0, 1\}^n$ and we denote the inverse of this permutation by $\tilde{E}_k^{-1}(T_i, \cdot)$. We remark that in the case of CLRW2, a tweakable block cipher key includes two keys for the underlying block cipher and two keys for the universal hash function family, i.e. $k = (k_1, H_1, k_2, H_2)$.

We use the definition of tweakable block cipher security given by Liskov et al. [18,19]. A secure tweakable block cipher is one that is indistinguishable from a tweaked random permutation family $\tilde{\Pi}$, where $\tilde{\Pi}$ is a set of random permutations on $\{0, 1\}^n$ indexed by elements of $\mathcal{K} \times \mathcal{T}$. A strong tweakable block cipher is one that is indistinguishable from a tweaked random permutation family when an adversary is given either access to encryption and decryption oracles for the tweakable block cipher or access to a family of tweaked random permutations and their inverses. In the definition of both tweakable block ciphers and strong tweakable block ciphers, the adversary is able to choose both the input and the tweak for each query.

Consider an adversary \mathcal{A} and define their tweakable-SPRP advantage against \tilde{E} as

$$\text{Adv}_{\tilde{E}}^{\widetilde{\text{sprp}}}(\mathcal{A}) = \left| \Pr[1 \leftarrow \mathcal{A}^{(\tilde{E}_k, \tilde{E}_k^{-1})}] - \Pr[1 \leftarrow \mathcal{A}^{(\tilde{\pi}, \tilde{\pi}^{-1})}] \right|$$

where $k \xleftarrow{\$} \mathcal{K}$ and $\tilde{\pi} = \{\pi(k, T_i, \cdot) \in \tilde{\Pi} : T_i \in \mathcal{T}\}$. Define $\text{Adv}_{\tilde{E}}^{\widetilde{\text{sprp}}}(q, t)$ to be the maximum advantage that can be realised by an adversary making no more than q queries and running in time at most t . We say that \tilde{E} is a strong tweakable block cipher if $\text{Adv}_{\tilde{E}}^{\widetilde{\text{sprp}}}(q, t)$ is sufficiently small.

2.3 Universal Hash Functions

A family of hash functions will be denoted $\mathcal{H} = \{h_H : \{0, 1\}^* \rightarrow \{0, 1\}^n \mid H \in \mathcal{K}_{\mathcal{H}}\}$ with each hash function h_H indexed by a key $H \in \mathcal{K}_{\mathcal{H}}$. For simplicity and clarity, we will abbreviate h_{H_j} to h_j .

A family of hash functions is said to be ϵ -almost XOR universal [13] if for every $M, M' \in \{0, 1\}^*$ with $M \neq M'$ and for every $c \in \{0, 1\}^n$, $\Pr_{H \in \mathcal{K}_{\mathcal{H}}} [h_H(M) \oplus h_H(M') = c] < \epsilon$. Throughout this paper ϵ -almost XOR universal will be abbreviated to ϵ -AXU.

3 Description of the CLRW2 Tweakable Block Cipher Construction

The scheme proposed by Landecker et al. [16] combines an ϵ -AXU hash function and a block cipher. The ciphertext Y_i is computed from plaintext X_i using key $k = (k_1, H_1, k_2, H_2)$ and tweak T_i as follows:

$$Y_i = E_{k_2} (E_{k_1} (X_i \oplus h_1(T_i)) \oplus h_1(T_i) \oplus h_2(T_i)) \oplus h_2(T_i)$$

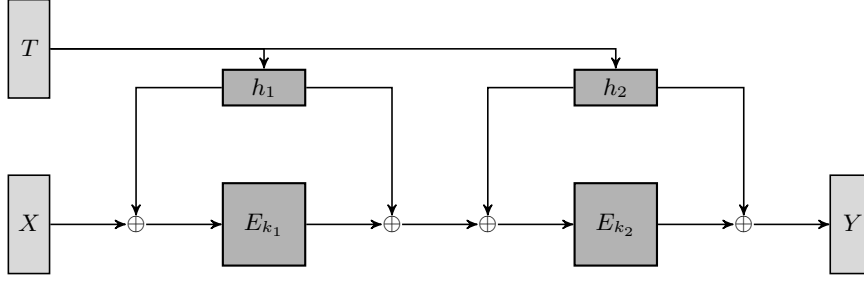


Fig. 1. The CLRW2 tweakable block cipher construction.

This construction is illustrated in Figure 1. The intuition behind the security of CLRW2 is that an adversary can only obtain a ‘birthday-bound-style’ advantage by causing a collision at both inputs to the block cipher. The reduction given by Landecker et al. (which relies on the SPRP security of the block cipher to show the strong tweakable block cipher security of CLRW2) proceeds via a series of games. The aim of these games is to show that the output of CLRW2 is close to that of an ideal tweakable block cipher, in the absence of a query that causes a collision at both block cipher inputs and that the likelihood of such a query is sufficiently low.

3.1 Additional Notation

We largely follow the notation used by Landecker et al. [16,17] to avoid introducing confusion.

Throughout this paper, we will lazy-sample random permutations instead of defining the permutation up front. When referring to the domain and range of a permutation $\pi : \{0, 1\}^n \rightarrow \{0, 1\}^n$, we will use $\text{Dom}_{\text{full}}(\pi)$ and $\text{Rng}_{\text{full}}(\pi)$ to denote the set $\{0, 1\}^n$ in order to make clear the context that this set relates to. When lazy-sampling a permutation $\pi : \{0, 1\}^n \rightarrow \{0, 1\}^n$ we will use the sets $\text{Dom}_{\text{lazy}}(\pi) \subseteq \text{Dom}_{\text{full}}(\pi) = \{0, 1\}^n$ and $\text{Rng}_{\text{lazy}}(\pi) \subseteq \text{Rng}_{\text{full}}(\pi) = \{0, 1\}^n$ to keep track of which values have been defined in the domain and range (respectively) of π . We will often drop the subscript for the sets $\text{Dom}_{\text{lazy}}(\pi)$ and $\text{Rng}_{\text{lazy}}(\pi)$ provided that the meaning is clear. The ‘lazy’ sets have an implicit query index because they are only defined relative to the previous queries and random choices within them.

We use $\mathcal{Y}_i = \{0, 1\}^n \setminus \{Y_j : j < i \text{ and } T_j = T_i\}$ and $\overline{\mathcal{Y}}_i = \{0, 1\}^n \setminus \mathcal{Y}_i$. That is, if $Y_i \in \overline{\mathcal{Y}}_i$ then Y_i is not a possible output from an ideal tweakable block cipher given the output from previous queries (recalling that $E(k, T_i, \cdot)$ is a permutation). Hence, \mathcal{Y}_i is the set of possible output values from an ideal tweakable block cipher.

We define the sets S_j as they are defined in Landecker et al.’s paper and present them graphically in Figure 2. These sets partition $\{0, 1\}^n$ for a particular input (X_i, T_i) according to the whether the corresponding inputs and outputs of each permutation have already been defined. As above, these sets have an implicit query index.

$$\begin{aligned}
S_1 &= \{Y_i \in \mathcal{Y}_i : Y_i \oplus h_2(T_i) \notin \text{Rng}(\pi_2)\} \\
S_2 &= \{Y_i \in \mathcal{Y}_i : Y_i \oplus h_2(T_i) \in \text{Rng}(\pi_2) \wedge \pi_2^{-1}(Y_i \oplus h_2(T_i)) \oplus h_1(T_i) \oplus h_2(T_i) \notin \text{Rng}(\pi_1)\} \\
S_3 &= \{Y_i \in \mathcal{Y}_i : Y_i \oplus h_2(T_i) \in \text{Rng}(\pi_2) \wedge \pi_2^{-1}(Y_i \oplus h_2(T_i)) \oplus h_1(T_i) \oplus h_2(T_i) \in \text{Rng}(\pi_1)\} \\
S_4 &= \{Y_i \in \overline{\mathcal{Y}}_i\}
\end{aligned}$$

That is, S_1 is the set of output values that correspond to undefined outputs from π_2 and S_2 is the set of output values corresponding to defined outputs from π_2 , but for which the output of π_1 is undefined. The set S_3 contains output values for which the all inputs and outputs to the block cipher are defined. Because we only consider these sets when $L_i \notin \text{Rng}(\pi_1)$, these

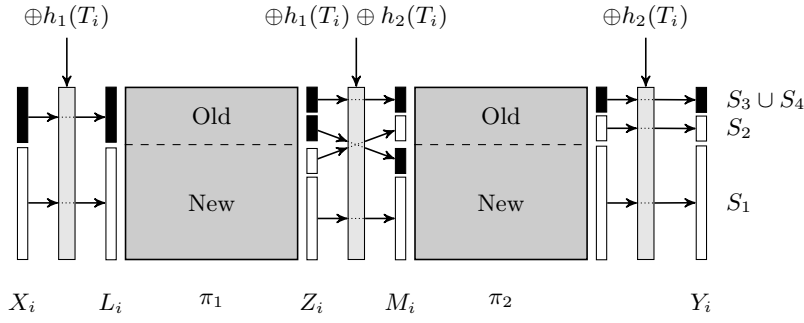


Fig. 2. An illustration of the definition for sets S_i . The domain and range of each permutation are divided into two sets: the input/output pairs that are ‘Old’ (in $\text{Dom}(\pi_j)$ and $\text{Rng}(\pi_j)$), shown above the dashed line; and the points that are ‘New’ (in $\overline{\text{Dom}(\pi_j)}$ and $\overline{\text{Rng}(\pi_j)}$), shown below the dashed line. In the case that $L_i \notin \text{Dom}(\pi_1)$, any value corresponding to a black box in the above diagram is impossible.

output values are inconsistent with the responses to previous queries. Finally, $S_4 = \overline{\mathcal{Y}_i}$ is the set of values that are not possible for either an ideal tweakable block cipher or the CLRW2 construction; responding to a query with an element of this set would violate the requirement for each $\tilde{E}(k, T_i, \cdot)$ to be a permutation.

At a first glance, S_3 appears to be the difference between CLRW2 and an ideal tweakable block cipher: elements in S_4 are not possible in either case; elements in S_1 and S_2 are possible in both cases; while elements in S_3 are not excluded from the output of an ideal tweakable block cipher but are impossible if CLRW2 is used. This informal summary does not give the full detail – if the situation were this simple, the original proof would be correct.

We use $p_{TBC}(Y_i)$ to denote the probability that Y_i is the output of the ideal tweakable block cipher and $p_{G3}(Y_i)$ for the probability that Y_i is the output of the intermediate cipher defined in Game 3 of the proof; these probabilities are both conditioned on all previous queries and responses.

3.2 Overview of Proof

We give a brief overview of Landecker et al.’s proof below but refer to the original papers [16,17] for the full details.

Game 0 defines the scheme when it is instantiated with a particular block cipher and is given in Figure 3, to introduce the random variables corresponding to inputs and outputs to the random permutations in later games.

Game 0

- 1: **procedure** $\tilde{E}(T, X)$
- 2: $i \leftarrow i + 1$; $X_i \leftarrow X$; $T_i \leftarrow T$
- 3: $L_i \leftarrow X_i \oplus h_1(T_i)$
- 4: $Z_i \leftarrow E_{k_1}(L_i)$
- 5: $M_i \leftarrow Z_i \oplus h_1(T_i) \oplus h_2(T_i)$
- 6: $Y_i \leftarrow E_{k_2}(M_i) \oplus h_2(T_i)$
- return** Y_i

Fig. 3. Game 0 defines the scheme when it is instantiated with a specific block cipher. It is included here to introduce the random variables corresponding to inputs and outputs to the random permutations in later games.

Games 1 to 3 consider the case in which there is a collision at the input to the first block cipher. The conclusion of this analysis is that in the absence of a collision at both block ciphers

on the same query, the output of CLRW2 is indistinguishable from an ideal tweakable block cipher and that the probability of two block cipher collisions being caused by the same query is sufficiently small that the scheme remains secure up to approximately $2^{\frac{2n}{3}}$ queries.

In Game 1, the block cipher is replaced with a random permutation, which is lazy-sampled. The security of this scheme relies on the assumption that the block cipher used to instantiate the scheme is a strong pseudorandom permutation and the distinguishing advantage between Game 0 and Game 1 is bounded above by $2\text{Adv}_E^{\text{SPP}}(\mathcal{A})$.

Between Games 1 and 3, the only changes to the definition of these games are in section of code that is executed when the image of L_i under π_1 has already been defined. The differences between these games are as follows. In Game 1, the existing definition of $\pi_2(M_i)$ is used. In Game 2, Y_i is sampled from \mathcal{Y}_i (the set of possible values) and this definition is checked to ensure that it does not contradict any existing definition of $\pi_2(M_i)$ or $\pi_2^{-1}(Y_i \oplus h_2(T_i))$. If the sampling of Y_i from \mathcal{Y}_i does cause a contradiction, one of bad_1 and bad_2 is set to `true` and Y_i is redefined. Game 3 is identical to Game 2, except that in Y_i is never redefined in Game 4, even if a `bad` flag is set to `true`. The difference between Game 1 and Game 2 is purely syntactic; the distribution of all random variables in the two games are identical. The distributions of all random variables in Games 2 and 3 are identical unless either bad_1 or bad_2 is set to `true`.

Games 4 to 6 address the case in which there is no first round collision; it is in this case that we have identified the problem with the proof. In these games, no changes are made to the code that is executed when $\pi_1(L_i)$ is already defined.

Between Game 3 and Game 4, the order in which Z_i and Y_i are sampled is changed, however the joint distribution is not changed. Between Games 4 and 5, the method used to sample Z_i and Y_i is changed, with the aim that the final joint distribution is unchanged. We will see that this aim is not achieved and that this is where the error in the proof occurs.

Game 5 and Game 6 are identical unless it is necessary to redefine Y_i . If it is necessary to redefine Y_i in Game 5, then bad_3 is set to `true` and so these two games are identical until `bad`, in a similar way to Game 2 and Game 3. Game 6 and Game 7 are functionally identical and Game 7 realises an ideal tweakable block cipher.

Game 8 is essentially identical to Game 7, but gives the adversary control over the Y_i values. The motivation for this step is that this makes it no harder for an adversary to trigger any of the `bad` events, but makes it easier to reason about the probability of a `bad` event occurring.

4 The Error in the Proof

4.1 Games 4 and 5 are not Identical

In the proof of security given for CLRW2 [16,17], Landecker et al. assert that the output distributions of Games 4 and 5 are identical. However, this is only the case if $p_{G3}(Y_i) - p_{TBC}(Y_i)$ is non-negative for $Y_i \in S_1 \cup S_2$. This is because the resampling step in Game 5 (at Lines 618-9) only produces an identical distribution to that of Game 4 if $p_{G3}(Y_i) \geq p_{TBC}(Y_i)$ for $Y_i \in S_1 \cup S_2$.

It is possible for a series of queries to result in $p_{G3}(Y_i) - p_{TBC}(Y_i)$ being negative for all $Y_i \in S_1$. This results in a contradiction to the claim that the output distributions of Games 4 and 5 are always identical.

In Section 5 we describe one method of modifying Games 5-8 to fix this issue with the proof. We redefine $\Delta_i = \sum_{Y_i \in S_i} p_{G3}(Y_i) - p_{TBC}(Y_i)$ (without the modulus signs from Landecker et al.'s definition) and note that $|\Delta_i| = \sum_{Y_i \in S_i} |p_{G3}(Y_i) - p_{TBC}(Y_i)|$ because for a given set S_j either $p_{G3}(Y_i) - p_{TBC}(Y_i) \geq 0$ for every Y_i in S_j or $p_{G3}(Y_i) - p_{TBC}(Y_i) < 0$ for every Y_i in S_j . Using our definition, $|\Delta_j|$ corresponds precisely with Landecker et al.'s definition of Δ_j .

We briefly describe the differences between the sampling methods employed when L_i is new in the relevant games. The resulting probability distributions are illustrated in Figures 7 to 9.

In Game 3, Z_i is chosen uniformly at random from $\overline{\text{Dom}(\pi_1)}$ and Y_i is defined to be consistent with this choice of Z_i . In Game 4, an appropriately weighted coin is tossed and Y_i is chosen from either S_1 or S_2 so that the distributions of Z_i and Y_i are identical to the distribution in Game 3. In Game 5, Y_i is chosen uniformly from \mathcal{Y}_i and if Y_i is in S_3 then it is resampled, from either S_1 or S_2 .

When Y_i is resampled in Game 5, the probability that it is chosen in S_2 is $\frac{|\Delta_2|}{|\Delta_1|+|\Delta_2|}$. This probability is used because if $|\Delta_1| + |\Delta_2| = |\Delta_3|$ then the distribution of Y_i does not change between Games 4 and 5 [17, p. 9]. However, if $p_{G3}(Y_i) - p_{TBC}(Y_i) < 0$ then $|\Delta_2| = |\Delta_1| + |\Delta_3|$, so $|\Delta_1| + |\Delta_2| > |\Delta_3|$ and the distribution of Y_i does change.

In fact, the difference between the distributions in Games 4 and 5 is exaggerated by the method used by Landecker et al. to resample from $S_1 \cup S_2$ if $Y_i \in S_3$, as illustrated in Figure 11. In the case that $p_{G3}(Y_i) - p_{TBC}(Y_i) < 0$ for $Y_i \in S_1$, the desired difference between the distributions in Game 4 and Game 5 for $Y_i \in S_1$ is $-|\Delta_1|$, i.e.

$$\sum_{Y_i \in S_1} p_{G3}(Y_i) = \sum_{Y_i \in S_1} p_{TBC}(Y_i) - |\Delta_1|$$

but using the sampling method described, this increases this to $|\Delta_1| \cdot \frac{|\Delta_3|}{|\Delta_1|+|\Delta_2|}$, so that

$$\sum_{Y_i \in S_1} p_{G3}(Y_i) = \sum_{Y_i \in S_1} p_{TBC}(Y_i) + |\Delta_1| \cdot \frac{|\Delta_3|}{|\Delta_1| + |\Delta_2|}.$$

Similarly, the difference for $Y_i \in S_2$ is decreased from $|\Delta_2|$ to $|\Delta_2| \cdot \frac{|\Delta_3|}{|\Delta_1|+|\Delta_2|}$.

4.2 How to make $p_{G3}(Y_i) - p_{TBC}(Y_i) < 0$

The proof given by Landecker et al. [16,17] is correct provided that $p_{G3}(Y_i) - p_{TBC}(Y_i) \geq 0$ when $Y_i \in S_1 \cup S_2$. We will call the situation in which $p_{G3}(Y_i) - p_{TBC}(Y_i) < 0$ for some $Y_i \in S_i \cup S_2$ an *inversion*. To demonstrate an inversion, first recall that:

$$\begin{aligned} p_{TBC}(Y_i) &= \frac{1}{2^n - |S_4|} \text{ for } Y_i \notin S_4, \\ p_{G3}(Y_i) &= \frac{N - |S_2|}{N|S_1|} \text{ for } Y_i \in S_1, \text{ and} \\ p_{G3}(Y_i) &= \frac{1}{N} \text{ for } Y_i \in S_2, \end{aligned}$$

where $N = |\overline{\text{Dom}(\pi_1)}|$.

$Y_i \in S_1$ We consider the possibility of an inversion occurring for $Y_i \in S_1$ and show that an adversary can force an inversion to happen for Y_i in S_1 with high probability. In this case, an inversion occurs when

$$\frac{1}{2^n - |S_4|} = p_{TBC}(Y_i) > p_{G3}(Y_i) = \frac{N - |S_2|}{N|S_1|}$$

Suppose that the adversary asks a number of queries so that $|\text{Dom}(\pi_1)| = a$ and $|\text{Dom}(\pi_2)| = b$, with no restrictions on how X_i and T_i are chosen. Then, one way that an inversion can occur with high probability is if: the adversary uses a new tweak for the next query; L_i is new; and for every $Z_i \in \text{Rng}(\pi_1)$, $Z_i \oplus h_1(T_i) \oplus h_2(T_i) \in \overline{\text{Dom}(\pi_2)}$.

In this case:

$$N = 2^n - a, \quad |S_1| = 2^n - b, \quad |S_2| = b, \quad \text{and } |S_3| = |S_4| = 0.$$

Starting with the observation that $ab > 0$, the following statements are equivalent:

$$\begin{aligned} ab &> 0 \\ 2^{2n} - (a+b)2^n + ab &> 2^{2n} - (a+b)2^n \\ (2^n - a)(2^n - b) &> 2^n(2^n - a - b) \\ \frac{1}{2^n} &> \frac{2^n - a - b}{(2^n - a)(2^n - b)} \\ \frac{1}{2^n - |S_4|} &> \frac{N - |S_2|}{N|S_1|} \\ p_{TBC}(Y_i) &> p_{G3}(Y_i) \end{aligned}$$

which is the condition for an inversion.

This situation can occur and indeed it is easy for an adversary to force this event to happen. If $T_1 \neq T_2$ then an inversion occurs on the second query (where $a = b = 1$) with probability $1 - \epsilon \approx \frac{2^n - 1}{2^n}$, which is the probability that $h_j(T_1) \neq h_j(T_2)$ when $T_1 \neq T_2$ (this probability is conditioned on the event $L_i \in \overline{\text{Dom}(\pi_1)}$ having occurred).

$Y_i \in S_2$ We also consider the possibility of an inversion occurring for $Y_i \in S_2$. For an inversion to occur we need

$$\frac{1}{2^n - |S_4|} = p_{TBC}(Y_i) > p_{G3}(Y_i) = \frac{1}{N}$$

Now note that

$$2^n - |S_4| \geq |S_1| + |S_2| \geq N$$

and so

$$\frac{1}{2^n - |S_4|} \leq \frac{1}{|S_1| + |S_2|} \leq \frac{1}{N}$$

Therefore, for $Y_i \in S_2$, there is no situation in which $p_{TBC}(Y_i) \geq p_{G3}(Y_i)$.

A Minor Issue with Landecker et al.'s Graph We also make the minor remark that the graph given by Landecker et al. [16, Fig.3] shows $p_{G3}(Y_1) \geq p_{G3}(Y_2)$, where $Y_1 \in S_1$ and $Y_2 \in S_2$, when in fact this situation can never occur as $N - |S_2|$ is never any larger than $|S_1|$ and

$$N - |S_2| \leq |S_1| \Leftrightarrow \frac{N - |S_2|}{|S_1|} \leq 1 \Leftrightarrow \frac{N - |S_2|}{N|S_1|} \leq \frac{1}{N} \Leftrightarrow p_{G3}(Y_1) \leq p_{G3}(Y_2).$$

5 How to Fix Landecker et al.'s Proof

The proof given by Landecker et al. can be fixed by modifying Games 5 to 8. The strategy we have adopted is to not change the distribution of Y_i between Games 4 and 5, then to bound the distance between the distributions that result from an ideal tweakable block cipher and from Game 5. This is the same strategy followed by Landecker et al. but we are careful to ensure that the distribution of Y_i does not change between Games 4 and 5 when $p_{G3}(Y_i) - p_{TBC}(Y_i) < 0$. This requires us to reduce the probability of Y_i being sampled from S_1 when $\Delta_1 < 0$ in Game 5; we do this naïvely by tossing an appropriately weighted coin to decide whether to resample

Y_i from S_2 . We add conditional branches to differentiate between the cases $\Delta_1 \geq 0$ and $\Delta_1 < 0$; this is a simple approach, but appears to work well and we lose only a small factor in the bound.

In Appendix A, we give the revised games (Figures 4, 5, and 6) and graphically represent the probability distributions realised by each of these games (Figures 7, 8, 9, 10, 11, and 12). We have only specified the encryption algorithm for each of game; it is straightforward to derive the corresponding decryption algorithms.

The distributions of all random variables in Games 4 and 5' are identical. Games 5' and 6' are identical unless either bad_3 or bad_4 gets set to true. The distributions of random variables in Games 6' and 7' are identical, with Game 7' simplifying some of the program flow. Game 8' gives the adversary control over Y_i values, so the bad flags can be set at least as easily as they can in Game 7'. We note that Game 8 and Game 8' are identical except for the addition of lines 14 to 17, so a large majority of the original analysis still applies.

Landecker et al. have independently identified an alternative method to correct the error in their proof [24], using a coupling argument which results in a tighter bound.

5.1 A Bound on Δ_1

To bound the advantage an adversary gains when we change from Game 5' to Game 6', we need to bound $|\Delta_1| \frac{(2^n - |S_4|)}{|S_1|}$ in the case that $\Delta_1 < 0$.

By noting that

$$N \geq 2^n - q, \quad |S_1| \geq 2^n - q, \quad |S_2| \leq q, \quad |S_3| \geq 0, \quad \text{and} \quad |S_4| \geq 0$$

we can bound $|\Delta_1|$ and $|\Delta_1| \cdot \frac{(2^n - |S_4|)}{|S_1|}$:

$$|\Delta_1| \leq \frac{q^2}{2^n(2^n - q)} \quad \text{and} \quad |\Delta_1| \cdot \frac{(2^n - |S_4|)}{|S_1|} \leq \frac{q^2}{(2^n - q)^2}$$

More details are given in Appendix B. This bound is tight, in the sense that it is possible for an adversary to ask a series of q queries and for $|\Delta_1| \cdot \frac{(2^n - |S_4|)}{|S_1|}$ to be as large as described by this bound. For this to occur we require that $|S_4| = |S_3| = 0$, $|S_2| = q$, and $N = 2^n - q$.

We also need to compare this to $\beta_1 + \Pr[Q]$ as calculated by Landecker et al., who show that $\beta_1 + \Pr[Q] \leq \frac{2q^3\hat{\epsilon}^2}{1 - q^3\hat{\epsilon}^2}$.

Now, note that $\frac{1}{2^n - q} \leq \frac{1}{2^n - 2q} \leq \hat{\epsilon}$ and that $1 - q^3\hat{\epsilon}^2 \leq 1$. So

$$\frac{q^2}{(2^n - q)^2} \leq q^2\hat{\epsilon}^2 \leq \frac{q^3\hat{\epsilon}^2}{1 - q^3\hat{\epsilon}^2}$$

and we can see that

$$\Pr[\mathcal{A}^{G8'} : \text{bad}_4] \leq \frac{q^2}{(2^n - q)^2} \leq q^3\hat{\epsilon}^2 \leq \frac{q^3\hat{\epsilon}^2}{1 - q^3\hat{\epsilon}^2}.$$

5.2 Game Hopping Probabilities

Recalling Landecker et al.'s observation [16,17] that

$$2\Pr[\mathcal{A}^{G8'} : \text{bad}_3] + \Pr[\mathcal{A}^{G8'} : \text{bad}_1 \vee \text{bad}_2] \leq \frac{6q^3\hat{\epsilon}^2}{1 - q^3\hat{\epsilon}^2},$$

we are able to compute the adversary's advantage, using the Fundamental Lemma of Game Playing [5]:

$$\begin{aligned}
\Pr[\mathcal{A}^{G^1} \rightarrow 1] &\leq \Pr[\mathcal{A}^{G^4} \rightarrow 1] + \Pr[\mathcal{A}^{G^4} : \text{bad}_1 \vee \text{bad}_2] \\
&\leq \Pr[\mathcal{A}^{G^{5'}} \rightarrow 1] + \Pr[\mathcal{A}^{G^{5'}} : \text{bad}_1 \vee \text{bad}_2] \\
&\leq \Pr[\mathcal{A}^{G^{6'}} \rightarrow 1 \wedge \neg(\text{bad}_3 \vee \text{bad}_4)] + \Pr[\mathcal{A}^{G^{6'}} \rightarrow 1 \wedge (\text{bad}_3 \vee \text{bad}_4)] \\
&\quad + \Pr[\mathcal{A}^{G^{6'}} : (\text{bad}_1 \vee \text{bad}_2) \wedge (\text{bad}_3 \vee \text{bad}_4)] \\
&\quad + \Pr[\mathcal{A}^{G^{6'}} : (\text{bad}_1 \vee \text{bad}_2) \wedge \neg(\text{bad}_3 \vee \text{bad}_4)] \\
&\leq \Pr[\mathcal{A}^{G^{6'}} \rightarrow 1] + 2\Pr[\mathcal{A}^{G^{6'}} : \text{bad}_3 \vee \text{bad}_4] + \Pr[\mathcal{A}^{G^{6'}} : \text{bad}_1 \vee \text{bad}_2] \\
&\leq \Pr[\mathcal{A}^{(\tilde{\pi}, \tilde{\pi}^{-1})} \rightarrow 1] + 2\Pr[\mathcal{A}^{G^{8'}} : \text{bad}_3] + 2\Pr[\mathcal{A}^{G^{8'}} : \text{bad}_4] \\
&\quad + \Pr[\mathcal{A}^{G^{8'}} : \text{bad}_1 \vee \text{bad}_2] \\
&\leq \Pr[\mathcal{A}^{(\tilde{\pi}, \tilde{\pi}^{-1})} \rightarrow 1] + \frac{6q^3\epsilon^2}{1 - q^3\epsilon^2} + 2\frac{q^2}{(2^n - q)^2} \\
&\leq \Pr[\mathcal{A}^{(\tilde{\pi}, \tilde{\pi}^{-1})} \rightarrow 1] + \frac{8q^3\epsilon^2}{1 - q^3\epsilon^2}
\end{aligned}$$

So, there is an adversary \mathcal{B} against the SPRP security of E , such that:

$$\text{Adv}_{\tilde{E}}^{\widetilde{\text{sprp}}}(\mathcal{A}) \leq \text{Adv}_E^{\text{sprp}}(\mathcal{B}) + \frac{8q^3\epsilon^2}{1 - q^3\epsilon^2}.$$

This is in contrast to the original result, which concludes that there is an adversary \mathcal{B} against the SPRP security of E , such that:

$$\text{Adv}_{\tilde{E}}^{\widetilde{\text{sprp}}}(\mathcal{A}) \leq \text{Adv}_E^{\text{sprp}}(\mathcal{B}) + \frac{6q^3\epsilon^2}{1 - q^3\epsilon^2}.$$

6 A Limitation of this Proof Technique

A natural extension of the work of Landecker et al. [16] is to consider longer chains of the LRW2 construction, as Lampe and Seurin have done at FSE 2013 [15]. The naïve approach to proving results in this case, which we emphasise is not the approach taken by Lampe and Seurin, would be to mimic Landecker et al.'s proof but to increase the number of sets S_i to describe where the last non-colliding input to a permutation occurs. This technique does not seem to succeed in the presence of the errors described in this note. A second remark about this approach is that it fundamentally depends on the ability to sample from a set $\text{Rng}_{\text{lazy}}(\pi_i) \cap (\text{Dom}_{\text{lazy}}(\pi_{i+1}) \oplus h_i(T_i) \oplus h_{i+1}(T_i))$. If this set is ever empty, it will be possible for an adversary to make a query that cannot be answered using the method described and a different proof method will be required. If this set is ever empty, it will be possible for an adversary to make a query that cannot be answered using the method described and a different proof method will be required.

We bound the number of queries that may be asked before the sampling method described above fails by $q < 2^{n-1}$. We emphasise that this does not constitute an error in Landecker et al.'s proof and does not appear to be an issue in Lampe and Seurin's work [15], it simply prevents Landecker et al.'s technique being naïvely extended asymptotically.

This bound is obtained as follows: Note that for every j , $|\text{Dom}_{\text{lazy}}(\pi_j)| = |\text{Rng}_{\text{lazy}}(\pi_j)|$ after every query and, for every tweak, $h_j(T_i) \oplus h_{j+1}(T_i)$ defines a perfect matching $\{0, 1\}^n \rightarrow \{0, 1\}^n$

(representing $\text{Rng}_{\text{full}}(\pi_j)$ and $\text{Dom}_{\text{full}}(\pi_{j+1})$). Suppose that we remove (up to) two edges from every matching when answering each query: one edge that matches the output of π_j and one edge matching the input to π_{j+1} . Then it is possible to respond to any later query, as long as every matching has at least one edge remaining. If there is a matching with no remaining edges, then there is a tweak for which $\overline{\text{Rng}_{\text{lazy}}(\pi_j) \cap (\text{Dom}_{\text{lazy}}(\pi_{j+1}) \oplus h_j(T_i) \oplus h_{j+1}(T_i))}$ is empty and an input value X_i such that $X_i \oplus h_1(T_i) \notin \text{Dom}_{\text{lazy}}(\pi_1)$. If there is a matching with no remaining edges, then there is a tweak for which $\overline{\text{Rng}_{\text{lazy}}(\pi_j) \cap (\text{Dom}_{\text{lazy}}(\pi_{j+1}) \oplus h_j(T_i) \oplus h_{j+1}(T_i))}$ is empty and an input value X_i such that $X_i \oplus h_1(T_i) \notin \text{Dom}_{\text{lazy}}(\pi_1)$.

We can guarantee that $\overline{\text{Rng}_{\text{lazy}}(\pi_j) \cap (\text{Dom}_{\text{lazy}}(\pi_{j+1}) \oplus h_j(T_i) \oplus h_{j+1}(T_i))}$ is not empty for every i and j , provided that $q < 2^{n-1}$. While it may remain possible to sample from this set beyond this bound, it is not guaranteed and depends on both the adversary's queries and the random choices in the lazy sampling of the functions.

This does not cause a problem until $n - 1$ block ciphers are chained together because up to that point security is only provided for $q < 2^{n-1}$. However if $n - 1$ block ciphers are chained together using the CLRW2 construction, then this issue can occur and it may not be possible to respond to queries using this method.

7 Discussion and Conclusions

We have shown that a minor error exists in the proof given for the tweakable block cipher CLRW2 [16,17]. Fortunately it is possible for this error to be corrected and the scheme still provides a similar level of security.

We also identify a potential limitation to extending this result asymptotically, by bounding the number of queries that can be made before it may be impossible to use Landecker et al.'s sampling method. We emphasise that this does not constitute an error in Landecker et al.'s proof and does not appear to be an issue in Lampe and Seurin's work; it simply prevents Landecker et al.'s technique being naïvely extended asymptotically.

Recently, flaws have been found in the proofs for two high-profile ciphers: GCM [11] and XCB [6]. In order for the community to have faith in the proofs given for schemes, it is important that security proofs are correct and any errors are removed. The factor lost by correcting the bound in this case is significantly smaller than for these errors, however it remains important that any errors in a security reduction are removed.

References

1. M. Bellare, A. Desai, E. Jorjani, and P. Rogaway. A concrete security treatment of symmetric encryption. In *Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on*, pages 394–403, Oct 1997.
2. Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of cipher block chaining. In Yvo G. Desmedt, editor, *Advances in Cryptology CRYPTO 94*, volume 839 of *Lecture Notes in Computer Science*, pages 341–358. Springer Berlin Heidelberg, 1994.
3. Mihir Bellare, Ted Krovetz, and Phillip Rogaway. Luby-Rackoff backwards: Increasing security by making block ciphers non-invertible. In Kaisa Nyberg, editor, *Advances in Cryptology EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 266–280. Springer Berlin Heidelberg, 1998.
4. Mihir Bellare, Krzysztof Pietrzak, and Phillip Rogaway. Improved Security Analyses for CBC MACs. In Victor Shoup, editor, *Advances in Cryptology CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 527–545. Springer Berlin Heidelberg, 2005.
5. Mihir Bellare and Phillip Rogaway. The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426. Springer Berlin Heidelberg, 2006.
6. Debrup Chakraborty, Vicente Hernandez-Jimenez, and Palash Sarkar. Another Look at XCB. *Cryptology ePrint Archive*, Report 2013/823, 2013.

7. Paul Crowley. Mercy: A Fast Large Block Cipher for Disk Sector Encryption. In Gerhard Goos, Juris Hartmanis, Jan Leeuwen, and Bruce Schneier, editors, *Fast Software Encryption*, volume 1978 of *Lecture Notes in Computer Science*, pages 49–63. Springer Berlin Heidelberg, 2001.
8. Niels Ferguson, Stefan Lucks, Bruce Schneier, Doug Whiting, Mihir Bellare, Tadayoshi Kohno, Jon Callas, and Jesse Walker. The Skein Hash Function Family. <http://www.skein-hash.info/sites/default/files/skein1.3.pdf>, 2010.
9. David Goldenberg, Susan Hohenberger, Moses Liskov, Elizabeth Crump Schwartz, and Hakan Seyalioglu. On Tweaking Luby-Rackoff Blockciphers. In Kaoru Kurosawa, editor, *Advances in Cryptology ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 342–356. Springer Berlin Heidelberg, 2007.
10. Tetsu Iwata. New Blockcipher Modes of Operation with Beyond the Birthday Bound Security. In Matthew Robshaw, editor, *Fast Software Encryption*, volume 4047 of *Lecture Notes in Computer Science*, pages 310–327. Springer Berlin Heidelberg, 2006.
11. Tetsu Iwata, Keisuke Ohashi, and Kazuhiko Minematsu. Breaking and Repairing GCM Security Proofs. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 31–49. Springer Berlin Heidelberg, 2012.
12. A. Joux. On the security of blockwise secure modes of operation beyond the birthday bound. *Information Theory, IEEE Transactions on*, 56(3):1239–1246, March 2010.
13. Hugo Krawczyk. LFSR-based Hashing and Authentication. In Yvo G. Desmedt, editor, *Advances in Cryptology CRYPTO 4*, volume 839 of *Lecture Notes in Computer Science*, pages 129–139. Springer Berlin Heidelberg, 1994.
14. Ted Krovetz and Phillip Rogaway. The Software Performance of Authenticated-Encryption Modes. In Antoine Joux, editor, *Fast Software Encryption*, volume 6733 of *Lecture Notes in Computer Science*, pages 306–327. Springer Berlin Heidelberg, 2011.
15. Rodolphe Lampe and Yannick Seurin. Tweakable Blockciphers with Asymptotically Optimal Security. *Fast Software Encryption 2013* (to appear).
16. Will Landecker, Thomas Shrimpton, and R. Seth Terashima. Tweakable Blockciphers with Beyond Birthday-Bound Security. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology, CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 14–30. Springer Berlin Heidelberg, 2012.
17. Will Landecker, Thomas Shrimpton, and R. Seth Terashima. Tweakable Blockciphers with Beyond Birthday-Bound Security. *Cryptology ePrint Archive*, Report 2012/450, 2012. Version 20120808:065419.
18. Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable Block Ciphers. In Moti Yung, editor, *Advances in Cryptology, CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 31–46. Springer Berlin Heidelberg, 2002.
19. Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable Block Ciphers. *Journal of Cryptology*, 24(3):588–613, 2011.
20. David McGrew. Impossible plaintext cryptanalysis and probable-plaintext collision attacks of 64-bit block cipher modes. *Cryptology ePrint Archive*, Report 2013/623, 2013.
21. Kazuhiko Minematsu. Beyond-Birthday-Bound Security Based on Tweakable Block Cipher. In Orr Dunkelman, editor, *Fast Software Encryption*, volume 5665 of *Lecture Notes in Computer Science*, pages 308–326. Springer Berlin Heidelberg, 2009.
22. Phillip Rogaway. Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. In Pil Joong Lee, editor, *Advances in Cryptology - ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 16–31. Springer Berlin Heidelberg, 2004.
23. Rich Schroepel. Hasty Pudding Cipher Specification. <http://web.archive.org/web/20070206162154/http://www.cs.arizona.edu/people/rcs/hpc/hpc-spec>, 1999.
24. Thomas Shrimpton and R. Seth Terashima. personal communication, Jan 2014.

Appendix A More Details on the Fix for the Proof

Game $\boxed{5'}$, $6'$

- 1: **procedure** $\tilde{E}(T, X)$
- 2: $i \leftarrow i + 1; X_i \leftarrow X; T_i \leftarrow T$
- 3: $L_i \leftarrow X_i \oplus h_1(T_i)$
- 4: **if** $L_i \in \text{Dom}(\pi_1)$ **then**
- 5: $M_i \leftarrow \pi_1(L_i) \oplus h_1(T_i) \oplus h_2(T_i)$
- 6: $Y_i \stackrel{\$}{\leftarrow} \mathcal{Y}_i$
- 7: **if** $M_i \in \text{Dom}(\pi_2)$ **then**
- 8: $\text{bad}_1 \leftarrow \text{true}$
- 9: **else if** $Y_i \oplus h_2(T_i) \in \text{Rng}(\pi_2)$ **then**
- 10: $\text{bad}_2 \leftarrow \text{true}$
- 11: $\pi_2(M_i) \leftarrow Y_i \oplus h_2(T_i)$
- 12: **else**
- 13: $Y_i \stackrel{\$}{\leftarrow} \mathcal{Y}_i$
- 14: **if** $\Delta_1 \geq 0$ **then**
- 15: **if** $Y_i \in S_1$ **then**
- 16: $V_i \leftarrow 0$
- 17: **else if** $Y_i \in S_2$ **then**
- 18: $V_i \leftarrow 1$
- 19: **else if** $Y_i \in S_3$ **then**
- 20: $\text{bad}_3 \leftarrow \text{true}$
- 21: $V_i \stackrel{\$}{\leftarrow} \xi\left(\frac{|\Delta_2|}{|\Delta_1| + |\Delta_2|}\right)$
- 22: $Z_i \leftarrow \pi_2^{-1}(Y_i \oplus h_2(T_i)) \oplus h_1(T_i) \oplus h_2(T_i)$
- 23: **if** $V_i = 1$ **then**
- 24: $Y_i \stackrel{\$}{\leftarrow} S_2$
- 25: **else if** $V_i = 0$ **then**
- 26: $Y_i \stackrel{\$}{\leftarrow} S_1$
- 27: **else if** $\Delta_1 < 0$ **then**
- 28: **if** $Y_i \in S_3$ **then**
- 29: $\text{bad}_3 \leftarrow \text{true}$
- 30: $Y_i \stackrel{\$}{\leftarrow} S_2$
- 31: **if** $Y_i \in S_1$ **then**
- 32: $V_i \leftarrow 0$
- 33: $U_i \stackrel{\$}{\leftarrow} \xi\left(\frac{|\Delta_1|(2^n - |S_4|)}{|S_1|}\right)$
- 34: **if** $U_i = 1$ **then**
- 35: $\text{bad}_4 \leftarrow \text{true}$
- 36: $Y_i \stackrel{\$}{\leftarrow} S_2$
- 37: **if** $Y_i \in S_2$ **then**
- 38: $V_i \leftarrow 1$
- 39: **if** $Y_i \in S_2$ **then**
- 40: $Z_i \leftarrow \pi_2^{-1}(Y_i \oplus h_2(T_i)) \oplus h_2(T_i) \oplus h_1(T_i)$
- 41: **else if** $Y_i \in S_1$ **then**
- 42: $Z_i \stackrel{\$}{\leftarrow} \text{Rng}(\pi_1) \setminus (\text{Dom}(\pi_2) \oplus h_2(T_i) \oplus h_1(T_i))$
- 43: $\pi_2(Z_i \oplus h_1(T_i) \oplus h_2(T_i)) \leftarrow Y_i \oplus h_2(T_i)$
- 44: $\pi_1(L_i) \leftarrow Z_i$
- 45: $M_i \leftarrow \pi_1(L_i) \oplus h_1(T_i) \oplus h_2(T_i)$
- return** Y_i

Fig. 4. Between Game 4 and Game $5'$, the order in which V_i and Y_i are sampled is reversed. Game $5'$ is identical to Game 5 and Game $6'$ is identical to Game 6, except for the addition of lines 27 to 38. Game $6'$ is identical to Game $5'$ until one of bad_3 or bad_4 is set to true.

Game 7'

```

1: procedure  $\tilde{E}(T, X)$ 
2:    $i \leftarrow i + 1; X_i \leftarrow X; T_i \leftarrow T$ 
3:    $Y_i \stackrel{\$}{\leftarrow} \mathcal{Y}_i$ 
4:    $L_i \leftarrow X_i \oplus h_1(T_i)$ 
5:   if  $L_i \in \text{Dom}(\pi_1)$  then
6:      $M_i \leftarrow \pi_1(L_i) \oplus h_1(T_i) \oplus h_2(T_i)$ 
7:     if  $M_i \in \text{Dom}(\pi_2)$  then
8:        $\text{bad}_1 \leftarrow \text{true}$ 
9:     else if  $Y_i \oplus h_2(T_i) \in \text{Rng}(\pi_2)$  then
10:       $\text{bad}_2 \leftarrow \text{true}$ 
11:       $\pi_2(M_i) \leftarrow Y_i \oplus h_2(T_i)$ 
12:   else
13:     if  $Y_i \in S_1$  then
14:        $V_i \leftarrow 0$ 
15:       if  $\Delta_1 < 0$  then
16:          $U_i \stackrel{\$}{\leftarrow} \xi\left(\frac{|\Delta_1|(2^n - |S_4|)}{|S_1|}\right)$ 
17:         if  $U_i = 1$  then
18:            $\text{bad}_4 \leftarrow \text{true}$ 
19:       else if  $Y_i \in S_2$  then
20:          $V_i \leftarrow 1$ 
21:       else if  $Y_i \in S_3$  then
22:          $\text{bad}_3 \leftarrow \text{true}$ 
23:          $Z_i \leftarrow \pi_2^{-1}(Y_i \oplus h_2(T_i)) \oplus h_1(T_i) \oplus h_2(T_i)$ 
24:       if  $V_i = 1$  then
25:          $Z_i \leftarrow \pi_2^{-1}(Y_i \oplus h_2(T_i)) \oplus h_2(T_i) \oplus h_1(T_i)$ 
26:       else if  $V_i = 0$  then
27:          $Z_i \stackrel{\$}{\leftarrow} \overline{\text{Rng}(\pi_1)} \setminus (\text{Dom}(\pi_2) \oplus h_2(T_i) \oplus h_1(T_i))$ 
28:          $\pi_2(Z_i \oplus h_1(T_i) \oplus h_2(T_i)) \leftarrow Y_i \oplus h_2(T_i)$ 
29:          $\pi_1(L_i) \leftarrow Z_i$ 
30:          $M_i \leftarrow \pi_1(L_i) \oplus h_1(T_i) \oplus h_2(T_i)$ 
return  $Y_i$ 

```

Fig. 5. The distributions of random variables in Games 6' and 7' are identical, with Game 7' simplifying some of the program flow. Game 7' is identical to Game 7, except for the addition of lines 15 to 18.

Game 8'

```

1: procedure  $\tilde{E}(T, X, Y)$ 
2:    $i \leftarrow i + 1$ ;  $X_i \leftarrow X$ ;  $T_i \leftarrow T$ 
3:    $Y_i \leftarrow Y$ 
4:    $L_i \leftarrow X_i \oplus h_1(T_i)$ 
5:   if  $L_i \in \text{Dom}(\pi_1)$  then
6:      $M_i \leftarrow \pi_1(L_i) \oplus h_1(T_i) \oplus h_2(T_i)$ 
7:     if  $M_i \in \text{Dom}(\pi_2)$  then
8:        $\text{bad}_1 \leftarrow \text{true}$ 
9:     else if  $Y_i \oplus h_2(T_i) \in \text{Rng}(\pi_2)$  then
10:       $\text{bad}_2 \leftarrow \text{true}$ 
11:       $\pi_2(M_i) \leftarrow Y_i \oplus h_2(T_i)$ 
12:   else
13:     if  $Y_i \in S_1$  then
14:       if  $\Delta_1 < 0$  then
15:          $U_i \xleftarrow{\$} \xi\left(\frac{|\Delta_1|(2^n - |S_4|)}{|S_1|}\right)$ 
16:         if  $U_i = 1$  then
17:            $\text{bad}_4 \leftarrow \text{true}$ 
18:            $Z_i \xleftarrow{\$} \overline{\text{Rng}(\pi_1)} \setminus (\text{Dom}(\pi_2) \oplus h_2(T_i) \oplus h_1(T_i))$ 
19:            $\pi_2(Z_i \oplus h_1(T_i) \oplus h_2(T_i)) \leftarrow Y_i \oplus h_2(T_i)$ 
20:         else if  $Y_i \in S_2$  then
21:            $Z_i \leftarrow \pi_2^{-1}(Y_i \oplus h_2(T_i)) \oplus h_1(T_i) \oplus h_2(T_i)$ 
22:         else if  $Y_i \in S_3$  then
23:            $\text{bad}_3 \leftarrow \text{true}$ 
24:            $Z_i \leftarrow \pi_2^{-1}(Y_i \oplus h_2(T_i)) \oplus h_1(T_i) \oplus h_2(T_i)$ 
25:          $\pi_1(L_i) \leftarrow Z_i$ 
26:          $M_i \leftarrow \pi_1(L_i) \oplus h_1(T_i) \oplus h_2(T_i)$ 
return  $Y_i$ 

```

Fig. 6. Game 8' gives the adversary control over Y_i values, so the 'bad flags' can be set at least as easily as they can in Game 7'. Game 8' is identical to Game 8, except for the addition of lines 14 to 17.

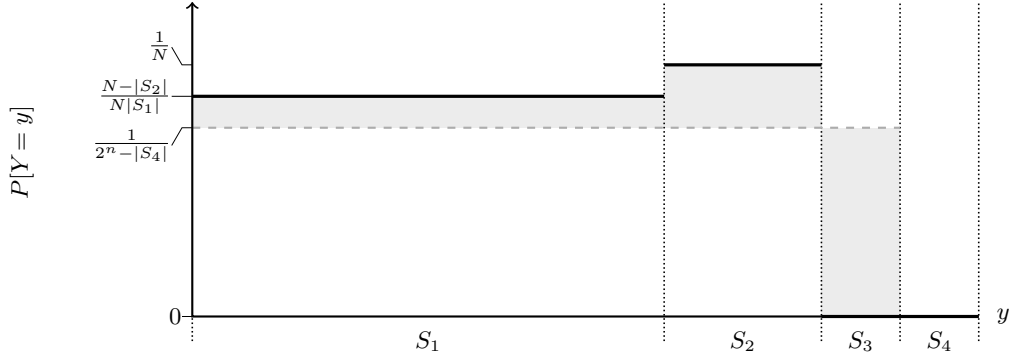


Fig. 7. Game 4, when $\Delta_1 \geq 0$. The output of CLRW2 is denoted by solid, black lines. The dashed line indicates the behaviour of an ideal tweakable block cipher. The shaded areas correspond to the Δ_i s. We do not need to redefine Game 4 when $\Delta_1 \geq 0$.

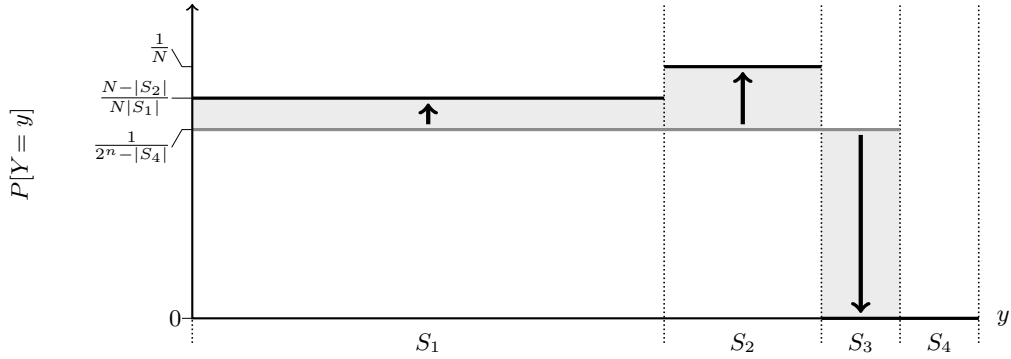


Fig. 8. Game 5, when $\Delta_1 \geq 0$. The solid, grey line denotes the distribution from which the output is initially sampled; this is the output distribution according to an ideal tweakable block cipher. The output is resampled according to the bold arrows. The solid, black lines represent the final distribution, which is identical to that of Game 4. Game 5 is identical to Game 5' when $\Delta_1 \geq 0$.

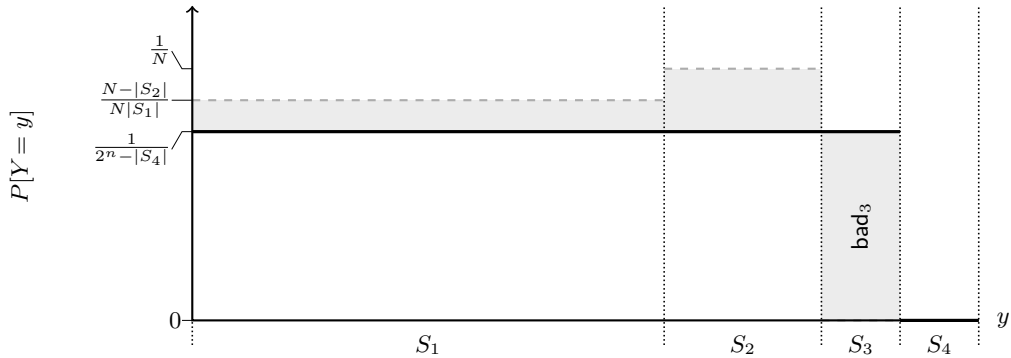


Fig. 9. Game 6, when $\Delta_1 \geq 0$. The solid, black line denotes the distribution of CLRW2 in Game 6, which coincides with the distribution of outputs from an ideal tweakable block cipher. bad_3 is set to true if the output is sampled from the labelled region. The dashed line indicates the behaviour of an CLRW2 in previous games; Game 6 is identical to Game 5 unless bad_3 is set to true. Game 6 is identical to Game 6' when $\Delta_1 \geq 0$.

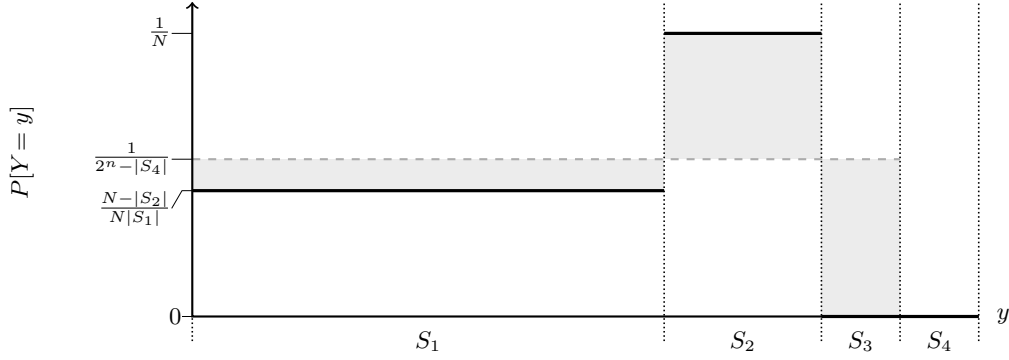


Fig. 10. Game 4, when $\Delta_1 < 0$. The output of CLRW2 is denoted by solid, black lines. The dashed line indicates the behaviour of an ideal tweakable block cipher. We do not need to redefine Game 4 when $\Delta_1 < 0$.

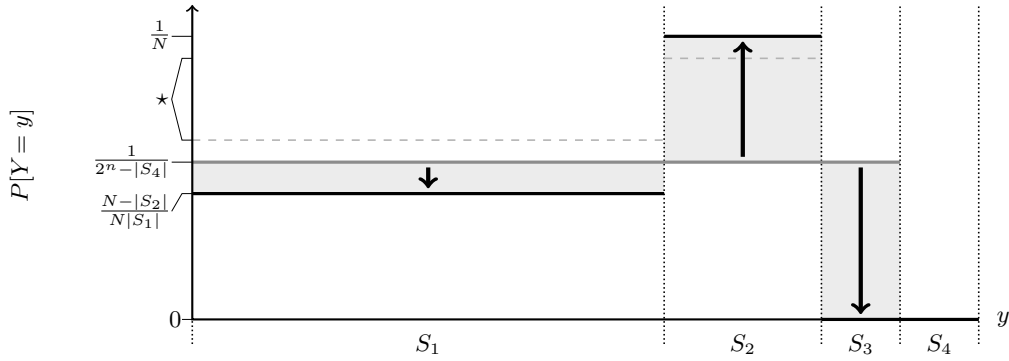


Fig. 11. Game 5', when $\Delta_1 < 0$. The solid, grey line denotes the distribution from which the output is initially sampled; this is the output distribution according to an ideal tweakable block cipher. The output is resampled according to the bold arrows. The solid, black lines represents the final distribution, which is identical to that of Game 4. The dashed lines labelled by \star indicate the incorrect probabilities realised in Game 5 from Landecker et al.'s paper if $\Delta_1 < 0$.

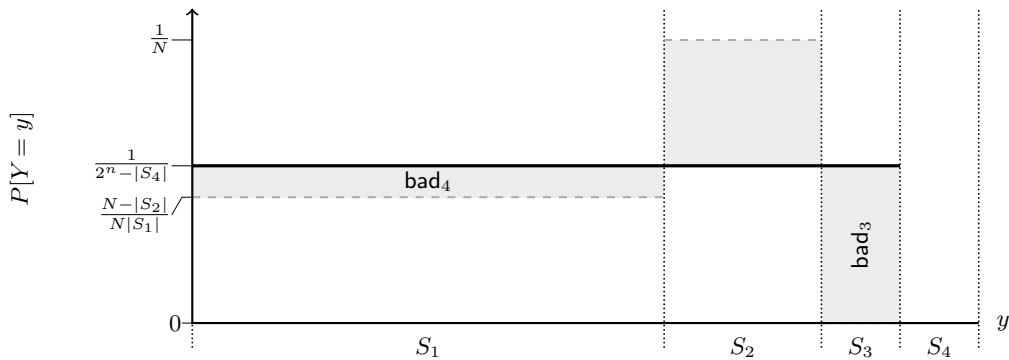


Fig. 12. Game 6', when $\Delta_1 < 0$. The solid, black line denotes the distribution of CLRW2 in Game 6, which coincides with the distribution of outputs from an ideal tweakable block cipher. `bad3` and `bad4` are set to true if the output is sampled from the respectively labelled regions. The dashed line indicates the behaviour of an CLRW2 in previous games; Game 6' is identical to Game 5' unless `bad3` or `bad4` is set to true.

Appendix B Bounding Δ_1

As described in Section 5.1, we can bound $|\Delta_1|$ in the case that $\Delta_1 < 0$ as follows:

$$\begin{aligned}
 |\Delta_1| &= \frac{|S_1|}{2^n - |S_4|} - \frac{N - |S_2|}{N} \\
 &= \frac{|S_1|}{2^n - |S_4|} - 1 + \frac{|S_2|}{N} \\
 &= \frac{|S_1| - (2^n - |S_4|)}{2^n - |S_4|} + \frac{|S_2|}{N} \\
 &= \frac{-|S_2| - |S_3|}{2^n - |S_4|} + \frac{|S_2|}{N} \\
 &\leq -\frac{|S_2| + |S_3|}{2^n} + \frac{|S_2|}{N} \\
 &= |S_2| \left(\frac{1}{N} - \frac{1}{2^n} \right) - \frac{|S_3|}{2^n} \\
 &\leq |S_2| \left(\frac{1}{N} - \frac{1}{2^n} \right) \\
 &\leq q \left(\frac{1}{N} - \frac{1}{2^n} \right) \\
 &\leq q \left(\frac{1}{2^n - q} - \frac{1}{2^n} \right) \\
 &= \frac{q^2}{2^n(2^n - q)}
 \end{aligned}$$

So:

$$\begin{aligned}
 |\Delta_1| \cdot \frac{(2^n - |S_4|)}{|S_1|} &\leq \frac{q^2}{2^n(2^n - q)} \cdot \frac{(2^n - |S_4|)}{|S_1|} \\
 &\leq \frac{q^2}{2^n(2^n - q)} \cdot \frac{2^n}{(2^n - q)} \\
 &= \frac{q^2}{(2^n - q)^2}
 \end{aligned}$$