Domain Extension for MACs Beyond the Birthday Barrier

Yevgeniy Dodis[*] John Steinberger[†]

February 4, 2011

Abstract

Given an n-bit to n-bit MAC (e.g., a fixed key blockcipher) with MAC security ε against q queries, we design a variable-length MAC achieving MAC security $O(\varepsilon q \text{poly}(n))$ against queries of total length qn. In particular, our construction is the first to break the "birthday barrier" for MAC domain extension from noncompressing primitives, since our security bound is meaningful even for $q = 2^n/\text{poly}(n)$ (assuming ε is the best possible $O(1/2^n)$). In contrast, the previous best construction for MAC domain extension for n-bit to n-bit primitives, due to Dodis and Steinberger [13], achieved MAC security of $O(\varepsilon q^2 (\log q)^2)$, which means that q cannot cross the "birthday bound" of $2^{n/2}$.

[*] Computer Science Dept. NYU. Email: `dodis@cs.nyu.edu`.

[†] Institute for Theoretical Computer Science, Tsinghua University, China. Email: `jpsteinb@gmail.com`.

1 Introduction

Most primitives in symmetric-key cryptography are built from block ciphers, such as AES. In this paper, we will concentrate on the question of designing variable-input-length (VIL) message authentication codes (MACs) from block ciphers. This question is very well studied, as we survey below. However, with few exceptions, most existing constructions and their analyses make the following two assumptions: (a) **Pseudorandomness**: the block cipher is modeled as a pseudorandom permutation (PRP); and (b) **Secrecy of Intermediate Results**: the attacker only learns the input/output behavior of the corresponding VIL-MAC, but does not learn any of the intermediate results. As observed by Dodis et al. [11–13], each of these assumptions might either be unnecessarily strong, or simply too unrealistic in the following two scenarios.

DOMAIN EXTENSION OF MACS. This is our main question. Since the desired MAC primitive only needs to be *unpredictable*, it would be highly desirable to only assume that the block cipher is unpredictable as well, as opposed to pseudorandom. Indeed, it seems that assuming the block cipher is unpredictable is a *much weaker* assumption than assuming full pseudorandomness: to break the latter, all one needs to do is to predict one bit of "random-looking" information about the block cipher with probability just a little over $1/2$, while the former requires one to fully compute the value of the block cipher on a new point with non-trivial probability. For example, in the non-uniform model, any block cipher (in fact, even non-trivial pseudorandom generator) with an n-bit key can be very efficiently distinguished from random with advantage $2^{-n/2}$ [10, 13]. To the best of our knowledge, no such lower bounds are known for breaking unpredictability, meaning that close to 2^{-n} MAC security might be possible for such a block cipher. To put it differently, while we hope that existing block ciphers are actually PRPs, it seems quite reasonable to assume that their MAC security could be *noticeably better* than their PRP security. Thus, if we can design a VIL-MAC whose security is tightly related to the *unpredictability* of the block cipher, this VIL-MAC might be more secure than the MAC whose analysis assumes the *pseudorandomness* of the cipher.

Of course, one might hope that existing block-cipher based VIL-MACs, such as CBC-MAC [5, 27] and HMAC [3, 6] (whose compression function, under the hood, also uses a block cipher), are already secure when the block cipher is unpredictable. Unfortunately, as detailed in Dodis et al. [11–13] (see especially [13]), this is not the case: with few exceptions mentioned shortly, standard constructions are *completely insecure* when instantiated with unpredictable block ciphers, — often despite having simple proofs of security when one models the block cipher as a PRP.

RESILIENCE TO SIDE-CHANNELS. Even if the block cipher is a very good PRP, in practice many cryptographic implementations fall prey to various forms of side-channel attacks [7, 8, 14, 16–18, 29], where the physical realization of a cryptographic primitive can leak additional information, such as the computation-time, power-consumption, radiation/noise/heat emission etc. Thus, hardware people are paying special attention to securing block ciphers, such as AES, against such side-channel attacks. Although this might be a daunting task, it appears reasonable that specialized hardware implementations of AES might be pretty resistent to common forms of side-channel attacks. On the other hand, when the block cipher is used in some more complicated application, such as the design of a VIL-MAC, it might be hard or impractical to design a specialized "leakage-resilient" implementation for each such application, instead of doing so for a single, fixed-length building block (such as AES). Motivated by these considerations, Dodis et al. [11–13] proposed the model where the internals of the block cipher implementation are assumed secure, as usual, but all the external input/output behavior of the block cipher could potentially leak to the attacker (say, via side-channel attack).

To give this model a name while simultaneously making it more general, we say that a construction of a (deterministic) MAC P using some lower level keyed primitive(s) f is *transparent* (w.r.t. f), if (a) the key for P only consists of one of more keys for f; (b) when making a query M to P, the attacker not only gets $P(M)$, but also gets all the input/output pairs for every call to f made during the evaluation of $P(M)$. Since P is deterministic and all keys reside "inside" f, this indeed provides the attacker with the entire *transcript* of $P(M)$, short of what is happening during the calls to f. Coming back to our setting, we are interested in building a *transparent* VIL-MAC out of a block cipher. As we will see, this question is highly non-trivial even if the block cipher is assumed pseudorandom, let alone unpredictable. Indeed, as observed by [11–13], most existing VIL-MACs, including

CBC-MAC [5, 27] and HMAC [3, 6], are *completely insecure* when the intermediate results are leaked, even when instantiated with a PRP.

OUR MAIN RESULT. Motivated by these applications, we ask the same question as Dodis et al. [11–13], which simultaneously addresses both of the above concerns.

Question 1 *Can one build a* transparent *VIL-MAC P out of a block cipher f which is only assumed* unpredictable*?*

As already mentioned, most standard VIL-MACs built from block ciphers fail to address either MAC-preservation or transparency (even with a PRP). So we turn to the known secure approaches. As it turns out, all of them followed the principle of An and Bellare [2] of first constructing a compressing *Weakly Collision Resistant* (WCR)[1] hash function F from m to n bits, for some fixed $m > n$, then iterating this fixed-length WCR F using some variant of the Merkle-Damgård transform, and finally composing the output with a freshly keyed block cipher. As argued by Preneel and van Oorschot [28], any construction of this kind can achieve at most *birthday security*. Translated to the MAC-preservation setting, even if our original MAC f cannot be forged with probability ε using q queries, the resulting VIL-MAC P cannot have security greater than $O(\varepsilon q^2)$, meaning that q cannot cross $2^{n/2}$, even is ε is assumed to be (the best possible) $1/2^n$.

Interestingly, even achieving birthday security turned out to be challenging when the block cipher is only assumed unpredictable. The first secure construction of Dodis and Puniya [12], based on the Feistel network, only achieved security $O(\varepsilon q^6)$. The bound was then improved to $O(\varepsilon q^4)$ by Dodis, Pietrzak and Puniya [11] using the "enhanced CBC" construction. Finally, Dodis and Steinberger [13] showed (nearly) birthday security $\tilde{O}(\varepsilon q^2)$ using a new analysis of the Shrimpton-Stam [30] compression function. All these constructions were also transparent.

We ask the question if it is possible to build (hopefully, transparent) VIL-MACs from block ciphers with *beyond birthday security*. Most ambitiously, if f cannot be forged with probability ε using q queries, we would like to build a VIL-MAC P with security close to $O(\varepsilon q)$, meaning our security is meaningful even for values of q approaching 2^n, provided ε is assumed to be (the best possible) $1/2^n$. As our main result, we answer this question in the affirmative. Informally (see Section 4 for more details),

Theorem 1 *There exist fixed polynomials $a(n)$ and $b(n)$ and a construction P of a transparent VIL-MAC from an n-bit block cipher f, such that the rate[2] of P is $a(n)$ and the MAC security ε' of P against q' queries of total length qn is at most $O(b(n)q\varepsilon)$, where ε is the MAC-security of f against q queries. In particular, this bound is meaningful for q (and q') approaching 2^n.*

OTHER RELATED WORK. As we mentioned, the question of achieving beyond-birthday security for building VIL-MACs from unpredictable block ciphers was open prior to our work. In fact, the only domain extension results for MACs with beyond birthday security we obtained just recently by Yasuda [32] and Lee and Steinberger [19]. However, both results started with a shrinking MAC from strictly more than $2n$ to n bits. As we will see below, building such shrinking MACs (with beyond birthday security) from unpredictable block ciphers is highly non-trivial, and will be one of the key challenges we resolve on route to proving our main result. (However, we note that our result does not[3] simply reduce to building a $2n$ to n bit MAC from an n-bit to n-bit MAC.)

Another related area is that of for building VIL *pseudorandom functions* (PRFs) with beyond birthday security from PRPs, or more generally, fixed-length PRFs. In particular, several such constructions were found by [1, 4, 21, 24–26]. However, it is easy to see that none of them work either for the MAC domain extension, or even for building VIL-MACs (let alone PRFs) when the intermediate computation results are leaked. For example, the corollary of our main result, giving a *transparent* VIL-MAC from a (q, ε_{prp})-secure PRP with security $\varepsilon_{prp} + \tilde{O}(q/2^n)$, appears to be new.

Perhaps the closest work to ours is a paper of Maurer and Tessaro [23], who showed how to build a variable-length random oracle from an n-to-n bit random oracle. Their construction, analyzed in the indifferentiability

[1] WCR security states that it is infeasible to find collisions in F given oracle access to F.

[2] Defined as the average number of calls to the block cipher f per n-bit input block.

[3] We cannot just build beyond birthday $(2+\epsilon)n$ to n bit MAC and then compose it with the beyond birthday VIL-MAC construction of [19, 32], as each construction would lose a factor of q in exact security, resulting in already known "birthday" security $O(\varepsilon q^2)$.

framework of [9, 22], has fixed polynomial rate (just like our construction) and security $2^{(1-\delta)n}$, for any $\delta > 0$. However, although our construction borrows some ideas from that of [23], the two settings appear incomparable. On the one hand, the Maurer-Tessaro paper has to build an "indifferentiability simulator" for their setting (which required "input extraction" not required in our setting). However, they assumed a truly random function, and could use various probability calculations in deriving their result. In our setting, the block cipher is only unpredictable, and we have to make an explicit reduction to unforgeability, which makes matters substantially more delicate.

1.1 Outline of Our Construction

Our construction is quite involved, although we abstract it into several self-contained layers. As a side benefit, some of these layers (see below) are of potentially independent interest, and could be used for other purposes.

STEP 1: REDUCING TO $3n$-TO-$2n$ WCR AND $2n$-TO-n MAC. First, we notice that the above mentioned birthday limitation [28] of the An-Bellare approach no longer holds provided we build a WCR hash function F from m to $2n$ bits (for some $m > 2n$, say $m = 3n$). Namely, "birthday on $2n$ bits" might still give good enough security 2^n. However, even if we succeed in doing so with beyond birthday security (which will be one of our key results), we now also have to build a "final" MAC G from $2n$ to n bits. Thus, using known techniques but with different parameters (see Lemma 1 and Figure 1), our problem reduces to building beyond birthday WCR F from $3n$ to $2n$ bits and a beyond birthday MAC G from $2n$ to n bits.

STEP 2: REDUCING TO COVER-FREE FUNCTIONS. It so turns out that both of these tasks—i.e. the construction of the WCR function F and the construction of the MAC G—can be achieved from a more powerful (keyed) primitive which we introduce, called a *cover-free* function. Informally, a keyed function g from $\{0,1\}^m$ (recall, we will have $m = 3n$) to $(\{0,1\}^n)^t$ (for some parameter t), where $g(s) = (z_1(s), \ldots, z_t(s)) \in (\{0,1\}^n)^t$, is called *cover-free* (CF) if, given oracle access to g, it is infeasible to produce a sequence of (distinct) queries $s_1, s_2, \ldots, s_q \in \{0,1\}^m$ such that, for some $1 \leq j \leq q$, $z_\ell(s_j) \in \{z_\ell(s_1), \ldots, z_\ell(s_{j-1})\}$ for all $\ell \in [t]$. In other words, for each new query s_j one of the coordinates of $g(s_j)$ must be "uncovered" by previous coordinates of that index. The case $t = 1$ corresponds to the standard m to n bit WCR security,[4] however better (and in particular beyond-birthday) cover-free security can be achieved with larger values of t.

First, as depicted on the left side of Figure 2, we can compose CF g with t independently keyed block ciphers $f_1, \ldots f_t$, by setting $G(s) = f_1(z_1) \oplus \ldots \oplus f_t(z_t)$, where $g(s) = (z_1, \ldots, z_t)$. We show that the resulting G is easily seen to be a secure MAC from m bits to n bits. Moreover, the MAC security of G is tightly related to the CF security of g and the MAC security of f (see Lemma 2). Intuitively, a new forgery for G will give a new forgery for at least one of the f_ℓ's, by the CF security of g. Since $m = 3n > 2n$, this already gives us the needed $2n$ to n bit MAC.

More interestingly, as depicted on the right side of Figure 2, we show how to compose a CF function g with $2t$ independently keyed block ciphers $f_1, \ldots f_t, f'_1, \ldots, f'_t$ (in a variant of the "double-pipe" mode of [20]) to get a WCR function F from m bits to $2n$ bits. Moreover, the WCR security of F will be "roughly" $O(\varepsilon' + q\varepsilon)$, where ε' is the CF security of g and ε is the MAC security of f (see Lemma 3). Thus, as long as we can build CF g with security ε' close to $O(q\varepsilon)$, the WCR security of F will also be such. The proof of this result critically uses the "multi-collision to forgery" technique of Dodis and Steinberger [13].

Summarizing the discussion above, our task of building a VIL-MAC P thus reduces to building a CF function g with security $\varepsilon' \approx O(q\varepsilon)$ where ε is the MAC security of the underlying n-bit to n-bit primitive f. We also wish to build the CF function g with t as small as possible (which is relevant since the efficiency of P, including the size of key, is proportional to t). See Lemma 4.

STEP 3: BUILDING CF FUNCTIONS. This is, by far, the most involved part of our construction. The inspiration for this construction came from the afore-mentioned paper of Maurer and Tessaro [23], who showed how to build a VIL random oracle from an n-to-n bit random oracle. As we mentioned already, the setting of [23] is incomparable to our setting, especially since we cannot assume that our block cipher is (pseudo)random. However, our actual construction of CF functions is quite similar to the corresponding "cover-free" layer of the construction of [23],

[4] By analogy with collision-resistance, we could have called such families "weakly cover-free", but since we do not envision their use in the "unkeyed" setting, we stick with the current name.

although we had to make some changes (actually, *simplifications*) to the construction of [23], and our analyses are completely different. Our CF construction has three layers which we informally call combinatorial, cryptographic and algebraic. An impatient reader can look at Figure 3 for a concrete example (with $t = 3$ and other notation explained below).

STEP 3A: USING INPUT-RESTRICTING FAMILIES. This purely combinatorial step is precisely the same as in [23], and is also the most expensive step of our construction. We will use an *unkeyed* function E from $\{0,1\}^m$ to $(\{0,1\}^n)^r$ (here r is a parameter) called an *input-restricting function family* (IRFF; see Definition 1). Intuitively, IRFF has the property that after any q queries $s_1 \dots s_q$ to E, the number Q of new inputs s for which the r-tuple $E(s)$ is covered by the union of $E(s_1), \dots, E(s_q)$ is "not much larger" than q, and this should be true even when q is almost 2^n. Recall, our final goal is to ensure that it is hard to produce *any* such new input s. While IRFFs do not (and *cannot*!)[5] quite get us there, they ensure that there are not that many choices for the attacker of which new inputs to "cover" by old inputs.

We discuss the known constructions of IRFFs in Section 4, but mention that the constructions of IRFFs are completely combinatorial, and closely related to constructions of certain types of highly unbalanced bipartite expander graphs. While well-studied, these types of expander graphs are not yet completely understood, and in particular the "extreme" setting of parameters relevant to our case has not been the object of much attention. Therefore, although the existence of IRFFs with "good parameters" is known (and lead to the asymptotic bound claimed in Theorem 1), the concrete constructions are pretty inefficient. Nevertheless, as these parameters and efficiency are improved by future research in computational complexity, so will our final construction.

STEPS 3B-C: ADDING CONFUSION AND MIXING. Recall, IRFFs convert our input s into an r-tuple $(x_1 \dots, x_r)$. To get the final t-tuple $(z_1, \dots, z_t)$ for our CF function g, we can imagine repeating the following two-step precedure (steps 3b and 3c) t times, each time with a freshly keyed block cipher $\mathbf{F}$ (so the total number of block cipher keys for g will be t). First, we pass all r values $x_1, \dots, x_r$ through the block cipher $\mathbf{F}$ ("confusion step"), getting the values $y_1 \dots, y_r$. This is the cryptographic "confusion" layer. Then we algebraically "mix" all $2r$ values $(x_1 \dots x_r, y_1 \dots y_r)$ through a fixed, degree-r multivariate polynomial p (see Equation 3). This gives us one of the t outputs values $z_1 \dots z_t$.

The intuition for these last two steps is hard to explain (and, indeed, our analysis is quite involved). At a high level, the confusion step (evaluating $\mathbf{F}(x_1) \dots \mathbf{F}(x_r)$) is certainly needed to make a reduction to unforgeability, while the mixing step uses the fact that low-degree polynomials have few roots, so a "non-trivial" collision on the output of p will enable one to guess one of the values y_ℓ we are trying to forge. Of course, the difficulty is to make a successful guess for when and where the non-trivial collision to p will happen, with probability roughly $1/Q$, where Q is the guarantee given by IRFF (so Q is close to q). It turns out, there is a trivial strategy to make such a guess with "birthday" probability $1/Q^2 \approx 1/q^2$, even when $t = 1$. Of course, such probability is too low, and this is why we repeat steps 3b-c t times, for an appropriately chosen parameter t. We then show that the required guessing strategy can be reduced to the analysis of two "balls-and-bins" games. The relevance of such games to the domain extension of MACs was first introduced by Dodis and Steinberger [13]. Unfortunately, the two "balls-and-bins" games we have to analyze are significantly more complicated than the game of [13]. Nevertheless, as our most involved technical step, we show that both games can be won with probability roughly $1/(Q \cdot Q^{1/t})$. Thus, by choosing $t > \log Q$ (say, $t = n$), we get the desired bound $O(1/Q) \approx O(1/q)$.

EFFICIENCY. Our final VIL-MAC construction uses $5t$ keys for f, where we recall that the minimal value of $t \approx \log q \leq n$. Theoretically, we can reduce key material down to a single key for f, by "keying" f via fixed, reserved input bits. Namely, to simulate (at most) $5n$ keys this way we need to reserve $\lceil \log_2(5n) \rceil$ bits of input (and "truncate" the same number of bits in the output), effectively reducing the block length of the construction from n down to $n' = n - \lceil \log_2(5n) \rceil$. Due to the output truncation, we now also need to guess the missing $\lceil \log_2(5n) \rceil$ output bits not returned by our forger, incurring an (acceptable) additional $O(n)$ degradation of the security bound.

Our final VIL-MAC also achieves rate roughly proportional to $O(rt) = O(rn)$. Achieving a low value of r (coming from the combinatorial IRFF part) is more problematic (see Section 4), although existentially one can also

[5] Because they do not have a key and do not rely on any computational assumptions.

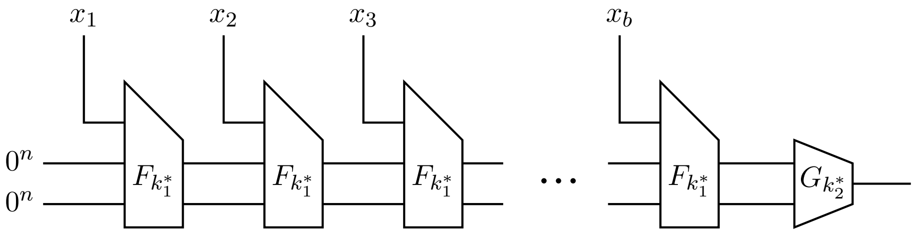

Figure 1: A high-level view of our construction $\mathrm{MD}[F, G]$. The input x is padded in a suffix-free manner into n-bit blocks $x_1, \ldots, x_b$. All wires shown carry n-bit values. $F_{k_1^*} : \{0,1\}^{3n} \to \{0,1\}^{2n}$ and $G_{k_2^*} : \{0,1\}^{2n} \to \{0,1\}^n$ are compression functions keyed by independent keys k_1^*, k_2^*.

make $r = O(n)$. So the best rate we can hope to achieve using our approach is $O(n^2)$. Therefore, we primarily view our result as an important *feasibility* result, much like the result of Maurer and Tessaro [23]. Nevertheless, our feasibility opens the door for future, potentially more efficient constructions.

2 Preliminaries

A *keyed function family* is a map $f : \{0,1\}^\kappa \times Dom(f) \to \{0,1\}^v$ where $Dom(f) \subseteq \{0,1\}^*$. The strings in $\{0,1\}^\kappa$ are the *keys* of f and we write $f_k(x)$ for $f(k, x)$ for $k \in \{0,1\}^\kappa$ and $x \in Dom(f)$.

For MACs we consider the following game, where A is a halting adversary with oracle access to f_k:

```
Game Forge($A, f$):
    $k \leftarrow \{0,1\}^\kappa$; $(x, y) \leftarrow A^{f_k}$
    If $x \in Dom(f)$, $f_k(x) = y$ and $x$ was not a query of $A$ then $A$
      wins, otherwise $A$ looses.
```

We define the insecurity of f as a MAC to be

$$\textbf{InSec}_f^{\mathrm{mac}}(T, q, \mu) := \max_A \Pr[A \text{ wins } \mathrm{Forge}(A, f)]$$

where the maximum is taken over all adversaries A making at most q queries of total combined length at most μ (after padding, if any) and of "running time" at most T. The "running time" is defined to be the total running time of the experiment, including the time necessary to compute the answers to A's queries. Moreover we "bill" the final verification query $f_k(x)$ (and its length) to A (so that A must in fact make $q - 1$ queries if $x \in Dom(f)$; seen another way, we ask A to verify its own forgery, if it attempts one). When f has fixed input length (i.e. $Dom(f) = \{0,1\}^m$ for some $m \in \mathbb{N}$) then μ is a function of q and it is convenient to elide the last argument, writing $\textbf{InSec}_f^{\mathrm{mac}}(t, q)$ instead of $\textbf{InSec}_f^{\mathrm{mac}}(t, q, \mu)$.

The *weak collision resistance* or "wcr" security of a function family f is measured as the maximum advantage of an adversary in finding a collision for a randomly keyed member of f when given oracle access to this member. We write

$$\textbf{InSec}_f^{\mathrm{wcr}}(T, q)$$

for the maximum such advantage over all adversaries A making at most q queries of running time at most T. (Here we do not measure the total query length, as we will only measure the wcr security of fixed input length constructions.) We skip a formal pseudocode-based definition of this standard notion, but mention that the adversary must make the queries verifying its collision, not merely output a colliding pair.

Given a block length n and a message x, we let $\mathrm{Pad}_n(x)$ be a suffix-free encoding of x of length a multiple of n bits (for example, the standard Merkle-Damgård padding of x, which appends the length of x as the last block[6]). Furthermore, given two keyed compression functions $F : \{0,1\}^{\kappa_1} \times \{0,1\}^{3n} \to \{0,1\}^{2n}$, $G : \{0,1\}^{\kappa_2} \times$

[6] This limits the message length to at most 2^n blocks, but this is not a serious limitation for practical values of n.

$\{0,1\}^{2n} \to \{0,1\}^n$ we define a keyed function $\mathrm{MD}[F,G] : \{0,1\}^{\kappa_1+\kappa_2} \times \{0,1\}^* \to \{0,1\}^n$ by

$$\mathrm{MD}[F,G]_{k_1^*,k_2^*}(x) = G_{k_2^*}(F_{k_1^*}(x_b \| F_{k_1^*}(x_{b-1} \cdots F_{k_1^*}(x_1 \| 0^{2n}) \cdots)))$$

where $\mathrm{Pad}_n(x) = x_1 x_2 \cdots x_b$ and each x_i has n bits, for all $k_1^* \in \{0,1\}^{\kappa_1^*}, k_2^* \in \{0,1\}^{\kappa_2}$ (see Fig. 1).

The proof of the following (standard) lemma is given in Appendix B:

Lemma 1 *Let* $F : \{0,1\}^{\kappa_1} \times \{0,1\}^{3n} \to \{0,1\}^{2n}$, $G : \{0,1\}^{\kappa_2} \times \{0,1\}^{2n} \to \{0,1\}^n$*, and consider* $\mathrm{MD}[F,G]$ *as a function of key space* $\{0,1\}^{\kappa_1+\kappa_2}$*. Then, for* $q = \mu/n$*,*

$$\textbf{\textit{InSec}}^{\mathrm{mac}}_{\mathrm{MD}[F,G]}(T, \tilde{q}, \mu) \le \textbf{\textit{InSec}}^{\mathrm{wcr}}_{F}(T, q) + \textbf{\textit{InSec}}^{\mathrm{mac}}_{G}(T, q)$$

Informally speaking, Lemma 1 reduces our task to building, from an n-bit to n-bit primitive f, compression functions F and G such that F has beyond-birthday wcr security and G has beyond-birthday mac security, where these securities must be based only the mac security of f (i.e., breaking the wcr security of F must imply breaking the mac security of f, and breaking the mac security of G must likewise imply breaking the mac security of f).

To the latter end we introduce in this paper the notion of a *cover-free* keyed function family $g : \{0,1\}^\kappa \times \{0,1\}^m \to (\{0,1\}^n)^t$. Here t is a parameter of the definition and we write the output of $g_k(x)$ as $(z_1^k(x), \ldots, z_t^k(x)) \in (\{0,1\}^n)^t$ where $z_i^k(x) \in \{0,1\}^n$ for each i; later we will simply write $(z_1(x), \ldots, z_t(x))$ when the dependence on a key k is understood. In the cover-free game, an adversary adaptively queries g_k on distinct points $s_1, s_2, \ldots \in \{0,1\}^m$, and wins if for some j each block of output of $g_k(s_j)$ is "covered" by a previous output, in the sense that $z_\ell^k(s_j) \in \{z_\ell^k(s_i) : i < j\}$, $1 \le \ell \le t$. The following game formalizes this:

```
Game Cover($A, g$):
    $k \leftarrow \{0,1\}^\kappa$;
    If $A^{g_k}$ makes distinct queries $s_1, \ldots, s_q \in \{0,1\}^m$ to $g_k$ such that
        $z_\ell^k(s_j) \in \{z_\ell^k(s_i) : i < j\}$, $1 \le \ell \le t$, for some $j \le q$,
    Then $A$ wins; Otherwise, $A$ looses.
```

We define the cover-free (CF) insecurity of g as

$$\mathbf{InSec}_g^{\mathrm{cover}}(T, q) := \max_A \Pr[A \text{ wins } \mathrm{Cover}(A, g)]$$

where the maximum is taken over all adversaries A making at most q queries and of running time at most T, with the same conventions as above on the running time. We (informally) say that a function family is *cover-free* to mean it has small cover-free insecurity.

Given a (cover-free) function family $g : \{0,1\}^\kappa \times \{0,1\}^m \to (\{0,1\}^n)^t$ where the ℓ-th block of g_k is given by the function $z_\ell^k : \{0,1\}^m \to \{0,1\}^n$ and a function family $f : \{0,1\}^{\kappa'} \times \{0,1\}^n \to \{0,1\}^n$ we define the composed function family $f \circ g : \{0,1\}^{\kappa + t\kappa'} \times \{0,1\}^m \to \{0,1\}^n$ by

$$(f \circ g)_{kk_1 \cdots k_t}(s) = \bigoplus_{\ell=1}^{t} f_{k_\ell}(z_\ell^k(s))$$

where $k \in \{0,1\}^\kappa$ and $k_1, \ldots, k_t \in \{0,1\}^{\kappa'}$, and $kk_1 \cdots k_t$ is a shorthand for the concatenation of k, $k_1, \ldots, k_t$. See Figure 2. We also define a *parallel composition* $f \mathbin{\bar{\circ}} g : \{0,1\}^{\kappa + 2t\kappa'} \times \{0,1\}^m \to \{0,1\}^{2n}$ of f and g, defined by

$$(f \mathbin{\bar{\circ}} g)_{kk_1 \cdots k_t k_1' \cdots k_t'}(s) = (f \circ g)_{kk_1 \cdots k_t}(s) \| (f \circ g)_{kk_1' \cdots k_t'}(s).$$

In other words, $f \mathbin{\bar{\circ}} g$ is simply the concatenation of two functions $f \circ g$ instantiated with the same g-key but independent f-keys.

Recall that our construction $\mathrm{MD}[F,G]$ takes as parameters keyed compression functions $F : \{0,1\}^{\kappa_1} \times \{0,1\}^{3n} \to \{0,1\}^{2n}$ and $G : \{0,1\}^{\kappa_2} \times \{0,1\}^{2n} \to \{0,1\}^n$. Given a cover-free function family $g : \{0,1\}^\kappa \times$

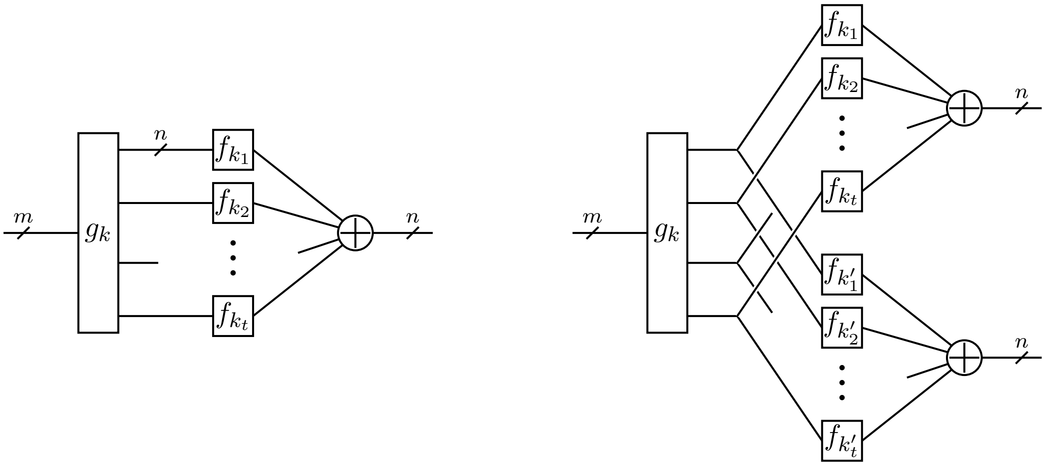

Figure 2: On the left, the composition $(f \circ g)_{kk_1\ldots k_t} : \{0,1\}^m \to \{0,1\}^n$. On the right, the parallel composition $(f \mathbin{\overline{\circ}} g)_{kk_1\cdots k_tk'_1\cdots k'_t} : \{0,1\}^m \to \{0,1\}^{2n}$.

$\{0,1\}^{3n} \to (\{0,1\}^n)^t$ and a function family $f : \{0,1\}^{\kappa'} \times \{0,1\}^n \to \{0,1\}^n$, we will set $\kappa_1 = \kappa + 2t\kappa'$, $\kappa_2 = \kappa + t\kappa$, and define

$$F_{k_1^*}(s) = (f \mathbin{\overline{\circ}} g)_{k_1^*}(s) \tag{1}$$

$$G_{k_2^*}(r) = (f \circ g)_{k_2^*}(0^n \| r) \tag{2}$$

for all $s \in \{0,1\}^{3n}$, $r \in \{0,1\}^{2n}$, $k_1^* \in \{0,1\}^{\kappa_1}$, $k_2^* \in \{0,1\}^{\kappa_2}$. The specification of our construction is thus now reduced to defining the cover-free function family g. We note that the n-bit to n-bit function family f is a parameter of the scheme (not constructed from any lower-level primitive) whereas g must be instantiated from f, and its cover-free security reduced to the mac security of f; see the next section for details on the construction of g.

Recall that, by Lemma 1, we are interested in bounding $\mathbf{InSec}_F^{\mathrm{wcr}}(T,q)$ and $\mathbf{InSec}_G^{\mathrm{mac}}(T,q)$ in terms of $\mathbf{InSec}_f^{\mathrm{mac}}(T,q)$. Towards this goal, we give two lemmas that upper bound $\mathbf{InSec}_{f \mathbin{\overline{\circ}} g}^{\mathrm{wcr}}(T,q)$ and $\mathbf{InSec}_{f\circ g}^{\mathrm{mac}}(T,q)$ as a function of $\mathbf{InSec}_g^{\mathrm{cover}}(T,q)$ and $\mathbf{InSec}_f^{\mathrm{mac}}(T,q)$. The proofs of both lemmas are given in Appendix B.

Lemma 2 *Let* $g : \{0,1\}^\kappa \times \{0,1\}^m \to (\{0,1\}^n)^t$, $f : \{0,1\}^{\kappa'} \times \{0,1\}^n \to \{0,1\}^n$. *Then*

$$\boldsymbol{InSec}_{f\circ g}^{\mathrm{mac}}(T,q) \le \boldsymbol{InSec}_g^{\mathrm{cover}}(T,q) + t \cdot \boldsymbol{InSec}_f^{\mathrm{mac}}(T,q).$$

Lemma 3 *Let* $g : \{0,1\}^\kappa \times \{0,1\}^m \to (\{0,1\}^n)^t$, $f : \{0,1\}^{\kappa'} \times \{0,1\}^n \to \{0,1\}^n$. *Then*

$$\boldsymbol{InSec}_{f \mathbin{\overline{\circ}} g}^{\mathrm{wcr}}(T,q) \le \boldsymbol{InSec}_g^{\mathrm{cover}}(T,q) + 2tq\log q \cdot \boldsymbol{InSec}_f^{\mathrm{mac}}(T + \tilde{O}(q), q).$$

(We note that, unlike Lemmas 1 and 2, the proof of Lemma 3 is not a triviality. In particular, it requires a "multi-collision to forgery" (MTF) bin-filling game of the type used in [13].) Combining Lemmas 1, 2 and 3 we directly obtain:

Lemma 4 *Let* $g : \{0,1\}^\kappa \times \{0,1\}^{3n} \to (\{0,1\}^n)^t$, $f : \{0,1\}^{\kappa'} \times \{0,1\}^n \to \{0,1\}^n$ *and let* F, G *be as in* (1), (2). *Then, if* $q = \mu/n$,

$$\boldsymbol{InSec}_{\mathrm{MD}[F,G]}^{\mathrm{mac}}(T,\tilde{q},\mu) \le 2 \cdot \boldsymbol{InSec}_g^{\mathrm{cover}}(T,q) + (2tq\log q + t) \cdot \boldsymbol{InSec}_f^{\mathrm{mac}}(T + \tilde{O}(q), q)$$

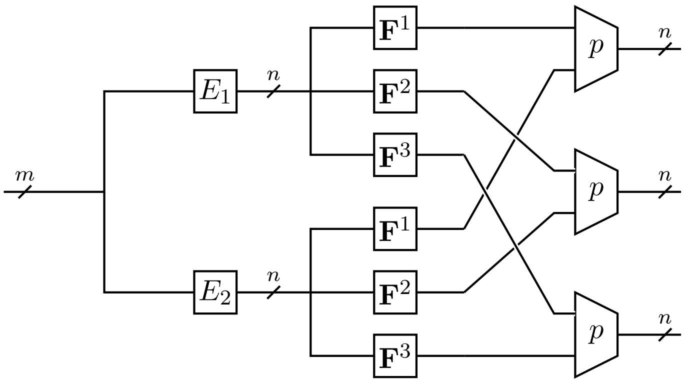

Figure 3: Illustration of the cover-free function $\mathbf{Z}_{m,n}^{\mathcal{E},r,t} : \{0,1\}^m \to (\{0,1\}^n)^t$ for parameters $r = 2$, $t = 3$. Additional wires not shown on the diagram carry the input of each $\mathbf{F}^i$ to the i-th copy of p.

Lemma 4 reduces our problem to constructing the cover-free function family g from the function family f such that $\mathbf{InSec}_g^{\mathrm{cover}}(T, q)$ can be bounded in terms of $\mathbf{InSec}_f^{\mathrm{mac}}(T, q)$. This is the topic of the next section, and the paper's main technical achievement.

When a keyed function is built from a smaller primitive, where the function's key consists of a finite set of keys for the smaller primitive (which is potentially called several times with different keys), the notions of MAC, WCR and cover-free securities naturally extend to a *transparent* model, where the adversary receives a full transcript of the function's computation at each query, up to calls to the primitive (namely, calls to the lower-level primitive appear as oracle calls in the transcript, so as not to reveal the primitive's keys). In fact, *all* results and proofs of this paper can be (easily) interpreted and are valid in this stronger "transparent" model. However, to keep the presentation simple, we will not further remind this from here on.

3 Building Cover-Free Function Families from MACs

This section contains our main result, the construction of a cover-free function family based on n-bit to n-bit primitives, that achieves beyond-birthday security assuming only good MAC security from the primitives. We note in passing that an *unkeyed* function $g : \{0,1\}^m \to (\{0,1\}^n)^t$ cannot be cover-free against information-theoretic adversaries unless $t2^n \geq 2^m$ or unless t is as large as the desired query security, which gives values of t that are too large to be practical for most settings.

Our construction uses the notion of an *input-restricting function family* (IRFF), introduced by Maurer and Tessaro [23]. The following definition is slightly modified for our purposes.

Definition 1 *Let $K = K(n) \leq 2^n$ and let $m > n$. A (m, n, r, δ, K)-*IRFF *is a set $\mathcal{E}$ of functions $E_1, \ldots, E_r : \{0,1\}^m \to \{0,1\}^n$ such that* (i) *$r \geq 2$ and $E_h(s) \neq E_{h'}(s)$ for all $s \in \{0,1\}^m$ and all $h \neq h'$,* (ii) *for all $s \neq s' \in \{0,1\}^m$ there exists $h \in \{1, \ldots, r\}$ such that $E_h(s) \neq E_h(s')$, and* (iii) *for any subset $\mathcal{U} \subseteq \{0,1\}^n$ such that $|\mathcal{U}| \leq rK$ we have*

$$\left|\{s \in \{0,1\}^m : E_h(s) \in \mathcal{U} \text{ for all } h = 1 \ldots r\}\right| \leq \delta|\mathcal{U}|.$$

The constructions of input-restricting function families are discussed in Section 4.

Our cover-free function family is also adapted from [23]. The construction takes as parameters $m \geq n$ as well as integers $r, t \geq 1$ and a (m, n, r, δ, K)-IRFF $\mathcal{E} = \{E_1, \ldots, E_r\}$. Let $\mathbf{F}^1, \ldots, \mathbf{F}^t$ be n-bit to n-bit primitives (later to be instantiated as members of function family $f : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$, possibly fixed-key blockciphers). The construction also uses a (concrete, unkeyed) function $p : \{0,1\}^{2rn} \to \{0,1\}^n$ described below. Let $\mathbf{Z}_{m,n}^{\mathcal{E},r,t} : \{0,1\}^m \to (\{0,1\}^n)^t$ be defined by

$$\mathbf{Z}_{m,n}^{\mathcal{E},r,t}(s) = (z_1(s), \ldots, z_t(s))$$

where

$$z_\ell(s) = p(E_1(s), \ldots, E_r(s), \mathbf{F}^\ell(E_1(s)), \ldots, \mathbf{F}^\ell(E_r(s)))$$

for $1 \leq \ell \leq t$ (see Figure 3). From $\mathbf{Z}^{\mathcal{E},r,t}_{m,n}$ we obtain a keyed function family of signature $\{0,1\}^{t\kappa} \times \{0,1\}^m \to (\{0,1\}^n)^t$ by instantiating each $\mathbf{F}^\ell$ with a member of a function family $f : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$; however, we opt for the unkeyed notation (in which $\mathbf{F}^1, \ldots, \mathbf{F}^t$ are implicitly keyed) when possible to reduce notational overhead.

As for the function p, it is the polynomial

$$p(x_1, \ldots, x_r, y_1, \ldots, y_r) = \sum_{j=1}^{r} \sum_{i=1}^{r} x_i y_j^i \tag{3}$$

where $x_1, \ldots, y_r$ are n-bit strings treated as elements of the field $\mathbb{F}_{2^n}$. The only properties of p that matter are the two following:

I. **Invertibility.** For any $1 \leq j \leq r$ and any values $x_1, \ldots, x_r, y_1, \ldots, y_{j-1}, y_{j+1}, \ldots, y_r$, $z \in \mathbb{F}_{2^n}$ such that $x_1, \ldots, x_r$ are not all zero, there are few values y_j such that $p(x_1, \ldots, x_r, y_1, \ldots, y_r) = z$, and these values y_j are efficiently enumerable.

II. **Collision Invertibility.** For any $1 \leq j, j' \leq r$ and any values $x_1, \ldots, x_r$, $y_1, \ldots, y_{j-1}$, $y_{j+1}, \ldots, y_r$, $x'_1, \ldots, x'_r$, $y'_1, \ldots, y'_{j'-1}, y'_{j'+1}, \ldots, y'_r \in \mathbb{F}_{2^n}$ such that $(x_1, \ldots, x_r) \neq (x'_1, \ldots, x'_r)$ there are few values $y_j = y'_{j'}$ such that

$$p(x_1, \ldots, x_r, y_1, \ldots, y_r) = p(x'_1, \ldots, x'_r, y'_1, \ldots, y'_r),$$

and these values are efficiently enumerable.

Both properties are easily verifiable from the fact that $p(x_1, \ldots, x_r, y_1, \ldots, y_r)$ is a polynomial of y_j of the type $c + x_1 y_j + \cdots + x_r y_j^r$, where c does not depend on y_j. Maurer and Tessaro use a different construction instead of p which does not obviously satisfy either property above, that requires enlarging the set of functions $\{\mathbf{F}^\ell\}$ to a set $\{\mathbf{F}^{\ell,v}\}$ where v ranges from 1 to $\lceil m/n + 1 \rceil$.

To state our main theorem, let $\mathsf{InvTime}(\mathcal{E}, q)$ be the amount of time required to list the values $\{s \in \{0,1\}^m : E_{h_0}(s) = v$ and $E_h(s) \in \mathcal{U}$ for $h \neq h_0\}$ for any given $h_0 \in [r]$, $v \in \{0,1\}^n$ and set $\mathcal{U} \subseteq \{0,1\}^n$ such that $|\mathcal{U}| \leq rq$. We have:

Theorem 2 *Let $\mathcal{E}$ be a (m, n, r, δ, K)-IRFF, let $f : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ be a function family, and consider $\mathbf{Z}^{\mathcal{E},r,t}_{m,n}$ as a keyed function family of key space $\{0,1\}^{\kappa t}$ by setting $\mathbf{F}^\ell = f_{k_\ell}$ for any $k_1 \cdots k_t \in \{0,1\}^{\kappa t}$. Then*

$$\textbf{\textit{InSec}}^{\text{cover}}_{\mathbf{Z}^{\mathcal{E},r,t}_{m,n}}(T, q) \leq 6rQt^3Q^{1/t} \cdot \textbf{\textit{InSec}}^{\text{mac}}_{f}(T_{mac}, q) \tag{4}$$

for any $q \leq K$, where $Q = qr\delta$ and

$$T_{mac} = T + \tilde{O}(Qt) + qr\mathit{InvTime}(\mathcal{E}, q) + \mathit{RootTime}_r(n)$$

where $\mathit{RootTime}_r(n)$ is the time required to find all the roots of a polynomial of degree r in a field of size $\mathbb{F}_{2^n}$. In particular, when $t = n$ and $q \leq 2^n/(r\delta)$, we have

$$\textbf{\textit{InSec}}^{\text{cover}}_{\mathbf{Z}^{\mathcal{E},r,t}_{m,n}}(T, q) \leq (12r^2\delta n^3) \cdot q \cdot \textbf{\textit{InSec}}^{\text{mac}}_{f}(T_{mac}, q)$$

Proof. Let A' be an adversary for the game $\text{Cover}(\cdot, \mathbf{Z}^{\mathcal{E},r,t}_{m,n})$ that runs in time T and that has success probability $\varepsilon_{A'}$. It suffices to design an adversary B for the game $\text{Forge}(\cdot, f)$ with advantage at least

$$\varepsilon_{A'}(6rQt^3Q^{1/t})^{-1}$$

and that runs in time T_{mac}.

B has access to a random member f_{k_0} of f. B chooses t random keys $k_1, \ldots, k_t \in \{0,1\}^\kappa$, and selects a random index $\ell_0 \in [t]$. Then B simulates A' with oracle $\mathbf{Z}^{\mathcal{E},r,t}_{m,n}$, instantianting the function $\mathbf{F}^\ell$ with f_{k_ℓ} if $\ell \neq \ell_0$ and instantianting $\mathbf{F}^{\ell_0}$ with f_{k_0}, using its oracle. Moreover B proceeds to "forget" the value of ℓ_0, treats each of the functions $\mathbf{F}^\ell$ as an oracle, and tries to forge any one of them (predicting their output on an unqueried input), making only one such forgery attempt during the game. Since B has chance $1/t$ of forging $\mathbf{F}^{\ell_0}$ if it does make a correct forgery, it suffices for B to make such a forgetful forgery with chance at least

$$\varepsilon_{A'}(6rQt^2Q^{1/t})^{-1}$$

in order for it to forge f_{k_0} with chance at least $\varepsilon_{A'}(6rQt^3Q^{1/t})^{-1}$.

It is easier to consider a modified version of A', which we call simply A, that directly issues $\mathbf{F}$-queries rather than $\mathbf{Z}^{\mathcal{E},r,t}_{m,n}$-queries; more precisely, A issues a sequence of queries $x_1, \ldots, x_{q'}$ where $q' \leq qr$ and each $x_j \in \{0,1\}^n$; B answers the query x_j with the tuple $(\mathbf{F}^1(x_j), \ldots, \mathbf{F}^t(x_j))$. We can assume A never makes the same query twice. We let $\mathcal{Q}_i = \{x_j : j \leq i\}$ and let $\mathcal{S}_i = \{s \in \{0,1\}^m : E_h(s) \in \mathcal{Q}_i \text{ for } 1 \leq h \leq r\}$ for $0 \leq i \leq q'$ (with $\mathcal{Q}_0 = \mathcal{S}_0 = \emptyset$). Note that

$$|\mathcal{S}_i| \leq |\mathcal{S}_{q'}| \leq |\mathcal{Q}_{q'}|\delta \leq qr\delta = Q$$

by the input-restricting property of $\mathcal{E}$. We also let $\Delta\mathcal{S}_i = \mathcal{S}_i \backslash \mathcal{S}_{i-1}$ for $1 \leq i \leq q'$ and put $z_\ell(\mathcal{C}) = \{z_\ell(s) : s \in \mathcal{C}\}$ for any $\mathcal{C} \subseteq \{0,1\}^m$ (which B can compute after it answers A's i-th query as long as $\mathcal{C} \subseteq \mathcal{S}_i$). We say A "wins the generous cover-free game" at the i-th query if there exists an $s \in \mathcal{S}_i$ such that $z_\ell(s) \in z_\ell(\mathcal{S}_i \backslash \{s\})$ for $1 \leq \ell \leq t$. Clearly, there exists an A of same running time as A' whose advantage ε_A in the generous game is at least as great as $\varepsilon_{A'}$, since A can simply simulate A' and ask the various $\mathbf{F}$-queries needed to compute the answers to A''s queries; by definition, A wins if A' wins Cover$(A, \mathbf{Z}^{\mathcal{E},r,t}_{m,n})$. (It is easy to check that if A' makes (distinct) queries $z_1, \ldots z_j \in \{0,1\}^m$ such that $z_\ell(s_j) \in \{z_\ell(s_i) : i < j\}$, then A wins the generous cover-free game by the time it has finished asking the queries necessary to compute the answer to the query s_j of A'.) Thus it is sufficient to have B forge one of the $\mathbf{F}$-functions with probability at least $\varepsilon_A(6rQt^2Q^{1/t})^{-1}$. We now view B as simply answering A's $\mathbf{F}$-queries (as opposed to computing answers to $\mathbf{Z}^{\mathcal{E},r,t}_{m,n}$-queries) though in reality B is running the whole computation, including the simulation of A' by A.

We view each value $s \in \mathcal{S}_i$ as a "bin" with t "slots"; the ℓ-th slot of bin s "receives a ball" or "becomes full" at query $j \geq i$ if $s \in S_j$ (namely, if the bin already exists at that point), if $z_\ell(s) \in z_\ell(\mathcal{S}_j \backslash \{s\})$, and if either $s \notin \mathcal{S}_{j-1}$ or $z_\ell(s) \notin z_\ell(\mathcal{S}_{j-1} \backslash \{s\})$. Once a bin receives a ball in a slot, the slot remains full. A slot cannot receive more than one ball, and bins are never removed; we note that no bins exist at the start, and that $|\Delta\mathcal{S}_i|$ bins are added at the i-th query. Under these definitions, A wins the "generous" cover-free game precisely if some bin becomes full (i.e., all its slots become full). It is helpful to picture A and B as playing an adversarial game in which A wins if it fills a bin without B forging one of the functions $\mathbf{F}^1, \ldots, \mathbf{F}^t$, and where B wins otherwise (in fact, we may even picture A as choosing the answers to its queries, while B observes and tries to guess an answer before it is revealed).

We say that ball ℓ of a bin $s \in \Delta\mathcal{S}_i$ is "early" if $z_\ell(s) \in z_\ell(\mathcal{S}_i \backslash \{s\})$ and "late" otherwise; thus a ball is early if and only if it is added to a bin at the same A-query which creates the bin. B plays one of two different forging strategies with equal probability. The first strategy is designed to prevent too many early balls from appearing in bins while the second strategy is designed to prevent A from filling a bin (the second strategy only functions well if not too many early balls appear in bins, whence the necessity of the first strategy). We name the two strategies "early prevention" and "late prevention"; despite these names, we emphasize the two strategies are not played sequentially; instead, B flips a coin at the start to decide which strategy to use.

We start by describing B's early prevention strategy. Let $Q = qr\delta$; as noted above, $Q \geq |\mathcal{S}_{q'}|$, so Q is an upper bound for the total number of bins created during the game. The goal of B's early prevention strategy is to prevent A from creating, for every $1 \leq k \leq t$, $Q^{1-k/t}$ or more bins that each have k or more early balls in them. In other words, we only require this strategy to work (i.e. forge a function $\mathbf{F}^\ell$ with "good enough" probability) if there is some $1 \leq k \leq t$ such that $Q^{1-k/t}$ or more bins are created with k or more early balls in them.

We model the early prevention strategy via a slightly simplified balls-in-bins game described below. To connect this balls-in-bins game with the "real" game played by B and A, it is helpful to first review the process via which bins are created and early balls are added to them. Consider a query x_i made by A. Then

$$\Delta\mathcal{S}_i = \{s \in \{0,1\}^m : E_{h_0}(s) = x_i \text{ for some } h_0 \in [r] \text{ and } E_h(s) \in \mathcal{Q}_{i-1} \text{ for } h \neq h_0\}$$

and the elements of $\Delta\mathcal{S}_i$ are the new bins created by this query. Each bin $s \in \Delta\mathcal{S}_i$ has t slots and the "value" $z_\ell(s)$ of the ℓ-th slot of s is revealed when B makes the query $\mathbf{F}^\ell(x_i)$; after the value $z_\ell(s)$ is revealed, an early ball is added to the ℓ-th slot of s according to whether there exists an $s' \in \mathcal{S}_i \backslash \{s\}$ such that $z_\ell(s) = z_\ell(s')$ or not (notice that $z_\ell(s')$ is known at this point for all $s' \in \mathcal{S}_i$). Thus, the process of filling the newly created bins with early balls consists in t "phases" (the queries $\mathbf{F}^1(x_i), \ldots, \mathbf{F}^t(x_i)$, which are made sequentially by B), where the ℓ-th phase simultaneously reveals the values of the ℓ-th slots of all the new bins, and whether these slots receive early balls or not. The following balls-in-bins game thus abstracts the process of creation of new bins and early balls:

'EARLY PREVENTION' BALLS-AND-BINS GAME. This game is played between two adversaries $\overline{A}$ and $\overline{B}$. Parameters are integers $t, q', Q \geq 1$. Rules are as follows:

- The game proceeds in q' rounds. At round i, $\overline{A}$ announces some number $v_i \geq 0$ of bins such that $\sum_{j \leq i} v_j \leq Q$.
- At the beginning of each round the v_i bins are empty. Each bin has t slots. Each round consists of t phases. At the ℓ-th phase, $\overline{A}$ reveals which of the v_i bins have their ℓ-th slot "filled" by a "ball".
- Before each phase of each round, $\overline{B}$ is allowed to secretly predict a bin that will receive a ball at that phase; $\overline{B}$ wins if it makes a correct guess, but it is only allowed to make one guess during the entire game.
- Let $b_{k,i}$ be the number of bins that receive k or more balls at round i, and let $b_k = \sum_i b_{k,i}$ where the sum is taken over all the rounds. Then $\overline{A}$ is required to fill bins such that $b_k \geq Q^{1-k/t}$ for at least one value of k, $1 \leq k \leq t$.

In Proposition 1 of Appendix A we exhibit a strategy for $\overline{B}$ that gives it at least $(t^2 Q^{1/t})^{-1}$ chance of winning the above game, regardless of $\overline{A}$'s strategy. Thus, if $Q^{1-k/t}$ or more bins each receive k or more early balls for some $1 \leq k \leq t$, and if B uses this strategy, B has chance $(t^2 Q^{1/t})^{-1}$ of correctly predicting, before the answer to some query $\mathbf{F}^\ell(x_i)$ is given, that the value returned by this query will result in slot ℓ of some (specific) bin $s \in \Delta\mathcal{S}_i$ receiving an early ball. To guess $\mathbf{F}^\ell(x_i)$, B further chooses a random $s' \in \mathcal{S}_i \backslash \{s\}$, and solves $z_\ell(s) = z_\ell(s')$ in order to guess $\mathbf{F}^\ell(x_i)$ (since s receives an early ball in slot ℓ precisely when there exists an $s' \in \mathcal{S}_i \backslash \{s\}$ such that $z_\ell(s) = z_\ell(s')$). To see that $z_\ell(s) = z_\ell(s')$ is really "solvable" two different cases must be considered, according to whether $s' \in \Delta\mathcal{S}_i$ or not. If $s' \notin \Delta\mathcal{S}_i$ then s' was created by an earlier A-query and the value of its slots are known, in particular the value $z_\ell(s')$ of its ℓ-th slot is known. Let $\overline{x}_h = E_h(s)$ for $1 \leq h \leq r$, let $h_0 \in [r]$ be the unique index such that $\overline{x}_{h_0} = x_i$ and let $\overline{y}_h = \mathbf{F}^\ell(\overline{x}_h)$ for $1 \leq h \leq r$. Then all the values $\overline{x}_1, \ldots, \overline{x}_r$, $\overline{y}_1, \ldots, \overline{y}_r$ are known to B except for the value $\overline{y}_{h_0}$, which it needs to guess using the equation

$$p(\overline{x}_1, \ldots, \overline{x}_r, \overline{y}_1, \ldots, \overline{y}_r) = z_\ell(s'). \tag{5}$$

By condition (i) of Definition 1 $(\overline{x}_1, \ldots, \overline{x}_r) \neq (0, \ldots, 0)$ so, by the 'Invertibility' property of p, there are few values $\overline{y}_{h_0}$ that solve (5). More precisely, since $p(\overline{x}_1, \ldots, \overline{y}_r)$ is a nonzero polynomial of degree at most r in $\overline{y}_{h_0}$, B has to choose from the at most r roots of $p(\overline{x}_1, \ldots, \overline{y}_r) - z_\ell(s')$, where $z_\ell(s')$ is just a constant. In the second case, $s' \in \Delta\mathcal{S}_i$ and $z_\ell(s')$ is not known (like $z_\ell(s)$, it is about to be revealed). Let $\overline{x}'_h = E_h(s')$, let $h'_0 \in [r]$ be the unique index such that $\overline{x}'_{h'_0} = x_i$ and let $\overline{y}'_h = \mathbf{F}^\ell(\overline{x}'_h)$ for $1 \leq h \leq r$. Then all the values $\overline{x}'_1, \ldots, \overline{x}'_r$, $\overline{y}'_1, \ldots, \overline{y}'_r$ are known to B except $\overline{y}'_{h'_0}$, and B needs to solve

$$p(\overline{x}_1, \ldots, \overline{x}_r, \overline{y}_1, \ldots, \overline{y}_r) = p(\overline{x}'_1, \ldots, \overline{x}'_r, \overline{y}'_1, \ldots, \overline{y}'_r) \tag{6}$$

(this is $z_\ell(s) = z_\ell(s')$) for $\overline{y}_{h_0}$, $\overline{y}'_{h_0}$ (or at least for $\overline{y}_{h_0}$). But $\overline{y}_{h_0} = \overline{y}'_{h_0}$; since $\overline{x}_{h_0} = \overline{x}'_{h'_0} = x_i$; also, by the injectivity of $\mathcal{E}$, $(\overline{x}_1, \ldots, \overline{x}_r) \neq (\overline{x}'_1, \ldots, \overline{x}'_r)$, so it follows by the 'Collision Invertibility' property of p that there are few values $\overline{y}_{h_0} = \overline{y}'_{h'_0}$ solving (6); in fact these are the at most r different roots of $p(\overline{x}_1, \ldots, \overline{y}_r) - p(\overline{x}'_1, \ldots, \overline{y}'_r)$, considered as a polynomial in $\overline{y}_{h_0} = \overline{y}'_{h'_0}$. The term $\mathsf{RootTime}_r(n)$ in Theorem 2 accounts for B's root-finding costs, which are incurred only once in the computation.

Naturally, B's further guessing of s' and of the correct root $\overline{y}_{h_0}$ erodes its probability of making a correct forgery even it has correctly guessed an early ball is about to be added to a bin slot, but it is easy to bound this erosion: B has chance at least $1/|\mathcal{S}_i| \geq 1/Q$ of correctly guessing s' and chance at least $1/r$ of correctly guessing the root. Thus, if $Q^{1-k/t}$ or more bins each receive k or more early balls for some $1 \leq k \leq t$ and if B is using its 'early prevention' strategy (which we have just finished describing), then B has chance at least

$$\frac{1}{rQt^2Q^{1/t}}$$

of forging. As B uses this strategy with probability $\frac{1}{2}$, we can therefore assume that fewer than $Q^{1-k/t}$ bins receive k early balls for every $1 \leq k \leq t$, or else B already reaches the requisite probability of success of $\varepsilon_A(6rQt^2Q^{1/t})^{-1}$.

We now discuss B's 'late prevention' strategy. Here B attempts to prevent A from filling a bin with t balls by guessing the arrival of late balls. We note that, if a query $\mathbf{F}^\ell(x_i)$ results in some late ball being placed in the ℓ-th slot of bin s, then $s \notin \Delta\mathcal{S}_i$ (by definition of 'late') and so the values $z_1(s), \ldots, z_t(s)$ are already known prior to the answer of the query $\mathbf{F}^\ell(x_i)$. Moreover the fact that the query $\mathbf{F}^\ell(x_i)$ results in a late ball appearing in bin s means there is some $s' \in \Delta\mathcal{S}_i$ such that (i) $E_{h_0}(s') = x_i$ for some $h_0 \in [r]$, (ii) the queries $\mathbf{F}^\ell(E_h(s'))$ have already been made[7] for $h \neq h_0$, and (iii) $z_\ell(s) = z_\ell(s')$ (the value $z_\ell(s')$ will become known when $\mathbf{F}^\ell(x_i)$ is answered). Let $\overline{x}'_1 = E_1(s'), \ldots, \overline{x}'_r = E_r(s')$ (so $\overline{x}'_{h_0} = x_i$) and $\overline{y}'_1 = \mathbf{F}^\ell(\overline{x}'_1), \ldots, \overline{y}'_r = \mathbf{F}^\ell(\overline{x}'_r)$, all of which are known to B except $\overline{y}'_{h_0}$. Then, if B has correctly guessed a late ball is going to appear in the ℓ-th slot of bin s *and* has correctly guessed the value of $s' \in \Delta\mathcal{S}_i$, it can predict $\mathbf{F}^\ell(x_i)$ by solving

$$p(\overline{x}'_1, \ldots, \overline{x}'_r, \overline{y}'_1, \ldots, \overline{y}'_r) = z_\ell(s) \tag{7}$$

for $\overline{y}'_{h_0}$, for which there are at most r solutions. (This is the second (and last) place we require the 'Invertibility' property of p.) Given these observations, the following balls-and-bins game clearly models B's 'late prevention' task, up to but not including guessing the root of (7):

'LATE PREVENTION' BALLS-AND-BINS GAME. This game is played between two adversaries $\overline{A}$ and $\overline{B}$. Parameters are integers $t, q', Q \geq 1$. Rules are as follows:

- The game involves "bins" with t slots each, where each slot can contain either contain a ball or not. At the beginning of the game, there are no bins. Bins are added to the game as described below, and never removed.
- The game proceeds in q' rounds, each of which consists of t "phases".
- At the beginning of round i, $\overline{A}$ announces some number $v_i \geq 0$ such that $\sum_{j \leq i} v_j \leq Q$. If $v_i = 0$, the round is skipped.
- At phase ℓ of round i, $1 \leq \ell \leq t$, $\overline{A}$ chooses a subset (possibly empty) of the currently existing bins that do not yet have a ball in their ℓ-th slot, and places balls in all of their ℓ-th slots, simultaneously. Moreover, $\overline{A}$ labels each ball placed with a number from 1 to v_i. (Multiple balls with the same label are allowed, and not all labels are required to appear.)
- At the end of round i, $\overline{A}$ introduces v_i new bins to the game, each possibly already containing some balls. Throughout the game, the total number of new bins introduced with k or more balls already in them must be less than $Q^{1-k/t}$ for all $1 \leq k \leq t$.
- Before each phase of each round, $\overline{B}$ is allowed to secretly predict a bin that will receive a ball at that phase and a label for that ball; $\overline{B}$ wins if it guesses both correctly. It is only allowed to make one guess during the game.

[7] This means A has made the queries $E_h(s')$ for $h \neq h_0$ so that, in fact, all queries $\mathbf{F}^{\ell'}(E_h(s'))$ for $1 \leq \ell' \leq t$ and $h \neq h_0$ have already been made (not just $\ell' = \ell$).

- $\overline{A}$ must fill some bin with t balls by the end of the game.

We note that the new bins introduced at the end of round i correspond to the elements of $\Delta\mathcal{S}_i$ and that v_i corresponds to $|\Delta\mathcal{S}_i|$. The “label” for a ball placed in a bin s at phase ℓ corresponds to an element $s' \in \Delta\mathcal{S}_i$ such that $z_\ell(s) = z_\ell(s')$, discussed above. (In the ‘real game’ between B and A several such elements s' may exist, so that more accurate modelization would allow $\overline{A}$ to choose a nonempty list of labels rather than a single label for each ball; however, seeking to minimize the guessing advantage of $\overline{B}$, $\overline{A}$ would automatically make each of these lists a singleton anyway.)

In Proposition 2 of Appendix A we exhibit a strategy for $\overline{B}$ in the ‘late prevention’ game that succeeds with probability $(3Qt^2Q^{1/t})^{-1}$ regardless of $\overline{A}$’s strategy. The ‘late prevention’ strategy of B consists simply of coupling the $\overline{B}$-strategy of Proposition 2 with a guessing of the root of (7). Thus, as long as fewer than $Q^{1-k/t}$ bins receive k or more early balls for $1 \le k \le t$, as long as A fills some bins with t balls and as long as B uses its late prevention strategy, B has chance at least

$$\frac{1}{3rQt^2Q^{1/t}}$$

of forging. Since B uses the ‘late prevention’ strategy with probability $\frac{1}{2}$, this concludes the proof.

□

4 Implications

Replacing g in Lemma 4 by our cover-free function $\mathbf{Z}^{\mathcal{E},r,t}_{m,n}$ and using Theorem 2 with $m = 3n$, we obtain:

Theorem 3 *Let $\mathcal{E}$ be a $(3n, n, r, \delta, K)$-IRFF, let $f : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$, and consider $\mathbf{Z}^{\mathcal{E},r,t}_{3n,n}$ as a keyed function family of key space $\{0,1\}^{\kappa t}$ like in Theorem 2. Define F, G by* (1), (2) *with $g = \mathbf{Z}^{\mathcal{E},r,t}_{3n,n}$. Then*

$$\begin{aligned} \textbf{\textit{InSec}}^{\mathrm{mac}}_{\mathrm{MD}[F,G]}(T, \tilde{q}, \mu) \quad &\le \quad 12rQt^3Q^{1/t} \cdot \textbf{\textit{InSec}}^{\mathrm{mac}}_{f}(T_{mac}, q) \\ &\quad + (2tq\log q + t) \cdot \textbf{\textit{InSec}}^{\mathrm{mac}}_{f}(T + \tilde{O}(q), q) \end{aligned} \tag{8}$$

where $q = \mu/n$ and $Q = qr\delta$ as long as $q \le K$, and where

$$T_{mac} = T + \tilde{O}(Qt) + qr\mathsf{InvTime}(\mathcal{E}, q) + \mathsf{RootTime}_r(n).$$

In particular, when $t = n$ and $Q \le 2^n$ (i.e. $q \le 2^n/r\delta$) and $q \le K$ we have

$$\begin{aligned} \textbf{\textit{InSec}}^{\mathrm{mac}}_{\mathrm{MD}[F,G]}(T, \tilde{q}, \mu) \quad &\le \quad 24r^2\delta n^3 q \cdot \textbf{\textit{InSec}}^{\mathrm{mac}}_{f}(T_{mac}, q) \\ &\quad + (2nq\log q + n) \cdot \textbf{\textit{InSec}}^{\mathrm{mac}}_{f}(T + \tilde{O}(q), q) \end{aligned} \tag{9}$$

By default we shall apply the second part of Theorem 3, choosing $t = n$. In order to interpret (9) we need to know what values of r, δ and K are achievable via IRFFs and to know $\mathsf{InvTime}(\mathcal{E}, q)$ for those IRFFs, as this term dominates T_{mac}.

The question of instantiating the IRFF $\mathcal{E}$ was already studied by Maurer and Tessaro [23], who reduced it to the construction of certain types of highly unbalanced bipartite expander graphs. While well-studied, these types of expander graphs are not yet completely understood, and in particular the setting of parameters relevant to our case has not been the object of much attention. Here we mention bounds achieved by two explicit constructions as well as those achieved by a non-explicit, probabilistic construction. In all cases we set $m = 3n$. We note that $\mathsf{InvTime}(\mathcal{E}, q)$ can always be upper bounded by q^3 by appending three functions to the IRFF that read off each block of input via the identity. Moreover, we can easily enforce condition (i) of Definition 1 as long as $r \le 2^n$. Since the family sizes r in question are anyway polynomial in n, we assume these tweaks without further mention.

Existential construction. A probabilistic construction [23] achieves a $(3n, n, r, \delta, K)$-IRFF $\mathcal{E}$ with $r = O(n)$, $\delta \approx 1$ and $K = \Omega(\frac{2^n}{n})$. In this case $Q = qr\delta = O(nq)$. Then the right-hand side of (9) becomes

$$O(n^5 q) \cdot \textbf{InSec}_f^{\mathrm{mac}}(T_{\mathrm{mac}}, q).$$

Assuming $\textbf{InSec}_f^{\mathrm{mac}}(T_{\mathrm{mac}}, q) \approx 1/2^n$, $\mathrm{MD}[F, G]$ achieves query security up to $q = \Omega(2^n/n^5)$. However, this construction is inexplicit.

Expanders of [31]. Expanders of Ta-Shma, Umans and Zuckerman yield an explicit $(3n, n, r, \delta, K)$-IRFF $\mathcal{E}$ with $r = \mathrm{poly}(n)$, $\delta = \mathrm{poly}(n)$ and $K = \Omega(\frac{2^n}{\mathrm{poly}(n)})$. In this case $Q = q\mathrm{poly}(n)$. The right-hand side of (9) becomes

$$O(\mathrm{poly}(n)q) \cdot \textbf{InSec}_f^{\mathrm{mac}}(T_{\mathrm{mac}}, q).$$

Assuming $\textbf{InSec}_f^{\mathrm{mac}}(T_{\mathrm{mac}}, q) \approx 1/2^n$ we can then achieve query security up to $q = \Omega(2^n/\mathrm{poly}(n))$. (We note this construction is strictly better from all standpoints than the one presented by Maurer and Tessaro [23].)

Expanders of [15]. Expanders of Guruswami, Umans and Venkatesan yield an explicit $(3n, n, r, \delta, K)$-IRFF $\mathcal{E}$ with $r = n^{O(\frac{1}{\varepsilon})}$, $\delta = \mathrm{poly}(n)$ and $K = 2^{n(1-\varepsilon)}$ for any $\varepsilon \in (0, 1)$. In this case $Q = q\mathrm{poly}(n)n^{O(\frac{1}{\varepsilon})}$. We can set $t = \log(Q) = \log q + O(\frac{1}{\varepsilon} \log n)$. For constant ε the right-hand side of (9) again becomes

$$O(\mathrm{poly}(n)q) \cdot \textbf{InSec}_f^{\mathrm{mac}}(T_{\mathrm{mac}}, q).$$

Assuming $\textbf{InSec}_f^{\mathrm{mac}}(T_{\mathrm{mac}}, q) \approx 1/2^n$ the insecurity thus remains negligible as long as $q \leq K = 2^{n(1-\varepsilon)}$. The advantage of this construction is that it affords efficient inversion time of $O(q\,\mathrm{poly}(n))$ (as opposed to $O(q^3)$ for the previous two constructions).

Interpretation. The assumption $\textbf{InSec}_f^{\mathrm{mac}}(T_{\mathrm{mac}}, q) \approx 1/2^n$ is only realistic as long as T_{mac} does not allow to do an exhaustive search over the key space of f; assuming the latter has size $2^\kappa \geq 2^n$, this implies that our upper bounds are only meaningful if $T_{\mathrm{mac}} \approx \mathsf{InvTime}(\mathcal{E}, q) \ll 2^\kappa$ (since T_{mac} is dominated by $\mathsf{InvTime}(\mathcal{E}, q)$). The first two constructions, which are only known to have $\mathsf{InvTime}(\mathcal{E}, q) = O(q^3)$, therefore only give a meaningful bound for $q \ll 2^{\kappa/3}$. Thus, with the current understanding of $\mathsf{InvTime}(\mathcal{E}, q)$, they might become beyond birthday only if $\kappa > 3n/2$ (and approach $q \approx 2^n$ only if $\kappa > 3n$). However, the last construction, having $\mathsf{InvTime}(\mathcal{E}, q) = O(q\,\mathrm{poly}(n))$, yields beyond-birthday security even if $\kappa = n$, which is the case of AES-128. Once again, though, we stress that the current limitations of our approach are due only to the limitations in the current constructions of expander graphs, and are not related to any "cryptographic" difficulties. Needless to say, future advances in the constructions of expander graphs will not only improve our parameters, but will likely have other applications in many areas of theoretical computer science.

Heuristic Instantiation. In practice, we expect a variety of heuristic instantiations to potentially approach the IRFF parameters of the non-explicit construction, most important of which is the setting of $r = O(n)$, which directly affects the rate and the efficiency. Here is one such construction based on block ciphers. We simply implement each $E_i : \{0,1\}^{3n} \rightarrow \{0,1\}^n$ as the XOR of three (independently keyed) fixed key block ciphers, i.e. $E_i(x\|y\|z) = f_{k_{i,1}}(x) \oplus f_{k_{i,2}}(y) \oplus f_{k_{i,3}}(z)$. We note that in this case the $3r$ keys $k_{1,1}, \ldots, k_{r,3}$ do not constitute key material, but rather fixed constants of the construction. We conjecture that, for a good enough block cipher, this construction might achieve the parameters $r = O(n)$, $\delta = \mathrm{poly}(n)$ (or possibly even $\delta = O(1)$) and $K = 2^n/\mathrm{poly}(n)$.

ACKNOWLEDGMENTS. Yevgeniy Dodis was supported by by NSF grants 1017471, 0831299, 0716690 and the Google Faculty Award. John Steinberger was supported by the National Natural Science Foundation of China Grant 60553001, by the National Basic Research Program of China Grant 2007CB807900, 2007CB807901 and by NSF grant CNS 0904380.

References

[1] W. Aiello and R. Venkatesan. *Foiling birthday attacks in length-doubling transformations — Benes: a non-reversible alternative to Feistel*, Eurocrypt 1996, pages 307–320.

[2] Jee Hea An, Mihir Bellare, *Constructing VIL-MACs from FIL-MACs: Message Authentication under Weakened Assumptions*, CRYPTO 1999, pages 252–269.

[3] Mihir Bellare, *New Proofs for NMAC and HMAC: Security without Collision-Resistance*, CRYPTO 2006, pages 602–619.

[4] Mihir Bellare, Oded Goldreich and Hugo Krawczyk, *Stateless Evaluation of Pseudorandom Functions: Security beyond the Birthday Barrier*. CRYPTO 1999, pages 270–287.

[5] Mihir Bellare, Joe Kilian, Phillip Rogaway, *The Security of Cipher Block Chaining*, CRYPTO 1994, pages 341–358.

[6] Mihir Bellare, Ran Canetti, Hugo Krawczyk, *Keying Hash Functions for Message Authentication*, CRYPTO 1996, pages 1–15.

[7] D. Boneh, R. A. DeMillo, and R. J. Lipton. On the importance of checking cryptographic protocols for faults (extended abstract). In *EUROCRYPT*, pages 37–51, 1997.

[8] E. Biham and A. Shamir. Differential fault analysis of secret key cryptosystems. In B. S. K. Jr., editor, *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, pages 513–525. Springer, 1997.

[9] J.-S. Coron, Y. Dodis, C. Malinaud and P. Puniya, *Merkle-Damgård Revisited: How to Construct a Hash Function*, Advances in Cryptology, Crypto 2005, pages 430–448.

[10] Anindya De, Luca Trevisan and Madhur Tulsiani. *Time Space Tradeoffs for Attacks against One-Way Functions and PRGs.* CRYPTO 2010, pp. 649–665.

[11] Yevgeniy Dodis, Krzysztof Pietrzak, Prashant Puniya, *A New Mode of Operation for Block Ciphers and Length-Preserving MACs*, EUROCRYPT 2008, pages 198–219.

[12] Yevgeniy Dodis, Prashant Puniya, *Feistel Networks Made Public, and Applications*, EUROCRYPT 2007, pages 534–554.

[13] Yevgeniy Dodis, John Steinberger, *Message Authentication Codes from Unpredictable Block Ciphers.* CRYPTO 2009, LNCS 5677, pp. 267–285. Full version available at `http://people.csail.mit.edu/dodis/ps/tight-mac.pdf`.

[14] K. Gandolfi, C. Mourtel, and F. Olivier. Electromagnetic analysis: Concrete results. In Çetin Kaya Koç, D. Naccache, and C. Paar, editors, *CHES*, volume 2162 of *Lecture Notes in Computer Science*, pages 251–261. Springer, 2001.

[15] Venkatesan Guruswami, Christopher Umans and Salil P. Vadhan. *Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes*. J. ACM 56(4), (2009).

[16] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten. Lest we remember: cold-boot attacks on encryption keys. *Commun. ACM*, 52(5):91–98, 2009.

[17] P. C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In N. Koblitz, editor, *CRYPTO*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.

[18] P. C. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In *CRYPTO*, pages 388–397, 1999.

[19] Jooyoung Lee and John Steinberger, *Multi-property preservation using polynomial-based modes of operation*, Eurocrypt 2010.

[20] S. Lucks. A failure-friendly design principle for hash functions. Asiacrypt 2005, LNCS 3788, pp. 474–494, Springer-Verlag, 2005.

[21] Ueli Maurer and Krzysztof Pietrzak. *The security of Many-Round Luby-Rackoff Pseudo-Random Permutations*, Eurocrypt 2003, pages 544–561.

[22] U. Maurer, R. Renner and R. Holenstein. Indifferentiability, impossibility results on reductions, and apllications to the random oracle methodology. TCC 2004, LNCS 2951, pp. 21–39, Springer-Verlag, 2008.

[23] Ueli Maurer and Stefano Tessaro, *Domain Extension of Public Random Functions: Beyond the Birthday Barrier,* Advances in Cryptology - CRYPTO 2007, Lecture Notes in Computer Science, Springer-Verlag, vol. 4622, pp. 187-204, Aug 2007.

[24] Jacques Patarin. *Luby-Rackoff: 7 rounds are enough for* $2^{n(1-\varepsilon)}$ *security*. Crypto 2003, pages 513–529.

[25] Jacques Patarin. *Security of Random Feistel Schemes with 5 or More Rounds*. Crypto 2004, pages 106–122.

[26] Jacques Patarin and André Montreuil. *Benes and Butterfly Schemes Revisited*. ISISC 2005.

[27] Erez Petrank, Charles Rackoff, *CBC MAC for Real-Time Data Sources*, J. Cryptology 13(3):315–338, 2000.

[28] Bart Preneel and Paul C. van Oorschot, *MD-x MAC and building fast MACs from hash functions*, CRYPTO 1995, pages 1–14.

[29] J.-J. Quisquater and D. Samyde. Electromagnetic analysis (ema): Measures and counter-measures for smart cards. In I. Attali and T. P. Jensen, editors, *E-smart*, volume 2140 of *Lecture Notes in Computer Science*, pages 200–210. Springer, 2001.

[30] Thomas Shrimpton and Martijn Stam, *Building a Collision-Resistant Compression Function from Non-Compressing Primitives*, ICALP 2008, pages 643–654. Also available at Cryptology ePrint Archive: Report 2007/409.

[31] Amnon Ta-Shma, Christopher Umans and David Zuckerman. *Lossless Condensers, Unbalanced Expanders, And Extractors.* Combinatorica 27(2), pages 213–240 (2007).

[32] K. Yasuda. A double-piped mode of operation for MACs, PRFs and PROs: Security beyond the birthday barrier. Eurocrypt 2009, LNCS 5479, pp. 242–259, Springer-Verlag, 2009.

A Early and Late Balls-in-Bins Games

Proposition 1 *There exists an adversary* $\overline{B}$ *for the 'Early Prevention' balls-and-bins game of Section 3 whose advantage is at least* $(t^2Q^{1/t})^{-1}$ *for any adversary* $\overline{A}$*. Moreover* $\overline{B}$ *runs in time* $\tilde{O}(Qt)$*.*

Proof. $\overline{B}$'s strategy is as follows:

1. $\overline{B}$ chooses a random index m between 0 and $t-1$, a random index v between 1 and $\lfloor Q^{1-m/t} \rfloor$ and a random index u between 1 and t.

2. $\overline{B}$ counts and catalogs the bins that receive m balls or more, updating its list before each new phase. When $\overline{B}$ finds its list has v members or more, $\overline{B}$ identifies the v-th member of the list (in order it was added to the list) and, if the last phase of the current round was less than u, $\overline{B}$ predicts that a ball will be added to this bin at the u-th phase of the current round.

3. In all other cases (i.e. if its list never contains v members or if the last phase number is greater than or equal to u) $\overline{B}$ does not guess.

Let c_j be the number of balls placed by $\overline{A}$ into bins that already have j or more balls in them at the moment of placement, $0 \leq j \leq t-1$, and let b_j be the number of bins that receive j or more balls, $0 \leq j \leq t$. Note $b_j \leq c_{j-1}$ for $1 \leq j \leq t$.

Assuming by contradiction that $\overline{B}$'s chance of success is less than $(t^2 Q^{1/t})^{-1}$ we will show by induction on j that $b_j < Q^{1-j/t}$ for $1 \leq j \leq t$, which will contradict $\overline{A}$'s obligation to make $b_j \geq Q^{1-j/t}$ for all least one value of j, $1 \leq j \leq t$.

Let $0 \leq m_0 < t$ and assume that $b_j \leq Q^{1-j/t}$ for $j \leq m_0$. This holds for $m_0 = j = 0$ since the total number of bins b_0 does not exceed Q. We will show $b_{m_0+1} < Q^{1-(m_0+1)/t}$, thus proving by induction that $b_j < Q^{1-j/t}$ for $1 \leq j \leq t$.

Say $\overline{B}$ selects $m = m_0$ (which happens with probability $1/t$). Then for each ball that $\overline{A}$ places into a bin with m or more balls already in it the index of that ball's bin in $\overline{B}$'s catalog of bins receiving m or more balls must be at most $b_m \leq \lfloor Q^{1-m/t} \rfloor$, so $\overline{B}$ has chance $1/t\lfloor Q^{1-m/t} \rfloor$ of correctly guessing the index v and the slot u for that ball; moreover, if $\overline{B}$ correctly guesses v and u, the phase at which $\overline{B}$ adds the v-th bin to its list will be less than u, since otherwise $\overline{A}$ could not longer add a ball at slot u to that bin. Thus each ball $\overline{A}$ places into a bin with m or more balls gives $\overline{B}$ chance $1/t\lfloor Q^{1-m/t} \rfloor$ of winning. Since these probabilities refer to disjoint events ($\overline{B}$ can never guess two different balls in the same game), $\overline{B}$'s chance of winning if it selects $m = m_0$ is therefore at least $c_{m_0}/tQ^{1-m_0/t}$. Since $\overline{B}$ selects $m = m_0$ with probability $1/t$ and since we are assuming $\overline{B}$'s chance of success is less than $(t^2 Q^{1/t})^{-1}$, we get

$$\frac{c_{m_0}}{t^2 Q^{1-m_0/t}} < \frac{1}{t^2 Q^{1/t}}$$

namely

$$c_{m_0} < Q^{1-(m_0+1)/t}$$

which implies $b_{m_0+1} < Q^{1-(m_0+1)/t}$ since $b_{m_0+1} \leq c_{m_0}$, as desired.

Finally the running of $\overline{B}$ is dominated by the task of maintaining a list of bins having received m or more balls. Since $\overline{A}$ places at most Qt balls and there are at most Q bins, this takes time $\tilde{O}(Qt)$. □

Proposition 2 *There exists an adversary $\overline{B}$ for the 'Late Prevention' balls-and-bins game of Section 3 whose advantage is at least $(3Qt^2Q^{1/t})^{-1}$ for any adversary $\overline{A}$. Moreover $\overline{B}$ runs in time $\tilde{O}(Qt)$.*

Proof. $\overline{B}$'s strategy is as follows:

1. $\overline{B}$ chooses a random index v between 1 and Q, a random index u between 1 and t, and a random index m between 0 and $t-1$.

2. $\overline{B}$ waits until a round i such that $\sum_{j \leq i} v_i \geq v$, and lets $v' = v - \sum_{j \leq i-1} v_j$. Then $1 \leq v' \leq v_i$.

3. $\overline{B}$ waits until phase u of round i, then chooses a bin uniformly at random from all bins having already at least m balls in them, and predicts that a ball of label v' is about to be added to this bin.

4. In all other cases (e.g. if there is no i such that $\sum_{j \leq i} v_i \geq v$, or no bin with m balls in them already at round u) $\overline{B}$ does not guess.

We note that $\overline{B}$ ignores the fact that balls are assigned to particular slots, and may in fact "stupidly" guess a bin at phase u for which the u-th slot is already full.

To analyze this strategy, let c_j be the number of balls placed by $\overline{A}$ into bins that already have j or more balls in them at the moment of placement, $0 \leq j \leq t-1$, and let b_j be the number of bins introduced with j or more balls already in them, $0 \leq j \leq t$. We have $b_t = 0$ since $b_t < Q^{1-t/t} = 1$ and $c_{t-1} \geq 1$ since $\overline{A}$ fills some bin with t balls. Let d_j be the number of bins at end of the game with j or more balls in them. Note that $d_0 \leq Q$ and that $d_j \leq c_{j-1} - c_j + b_j \leq c_{j-1} + b_j$ for $j \geq 1$.

Fix a value of m. For every ball placed by $\overline{A}$ into a bin already containing m balls, $\overline{B}$ has chance $1/Qt$ of guessing the correct round, label and phase for which that ball is thrown, and then has chance at least $1/d_m$ of guessing the correct bin. Thus $\overline{B}$ has chance at least c_m/Qtd_m of winning, since there are c_m such balls and since successful guesses for these balls constitute disjoint events. Since $\overline{B}$ selects each value of m with chance at least $1/t$, $\overline{B}$'s chance of winning is thus at least

$$\frac{1}{Qt^2}\sum_{m=0}^{t-1}\frac{c_m}{d_m}. \tag{10}$$

We claim the sum in (10) is at least $\beta := Q^{-1/t}/3$. Indeed, otherwise each ratio c_m/d_m less than β, so

$$c_0 < \beta d_0 \leq \beta Q$$

and

$$c_m < \beta d_m \leq \beta(c_{m-1} + b_m) \leq \beta(c_{m-1} + Q^{1-m/t})$$

for $m \geq 1$. Unfolding these inequalities, we get

$$\begin{aligned}
c_{t-1} &< \beta(\beta(\beta(\ldots\beta(\beta Q + Q^{1-1/t})\ldots) + Q^{1-(t-2)/t}) + Q^{1-(t-1)/t}) \\
&= \beta^t Q + \sum_{j=1}^{t-1}\beta^j Q^{1-(t-j)/t} \\
&\leq \frac{1}{3} + \sum_{j=1}^{\infty}(\beta Q^{1/t})^j \\
&\leq \frac{1}{3} + \frac{1}{2}
\end{aligned}$$

a contradiction since $c_{t-1} \geq 1$. Thus $\overline{B}$'s chance of winning is at least

$$\frac{1}{3Qt^2Q^{1/t}}$$

as claimed. The running time of $\overline{B}$ is dominated by the task of maintaining a list of bins with m or more balls in them. As $\overline{A}$ may throw at most Qt balls it is easy to see such a list can be maintained in time $\tilde{O}(Qt)$. □

B Proofs of lemmas 1, 2 and 3

Proof of Lemma 1. Let A be a mac-adversary for $\mathrm{MD}[F,G]$ whose queries have total length at most μ and that achieves advantage ε_A. Define a wcr adversary B for F as follows: given a randomly keyed member $F_{k_1^*}$ of F as oracle, B chooses a random key $k_2^{*'} \in \{0,1\}^{\kappa_2}$ and simulates A on oracle $\mathrm{MD}[F,G]_{k_1^*,k_2^{*'}}$, using its own oracle to compute $F_{k_1^*}$; then B wins if some $F_{k_1^*}$-collision occurs during this computation. Let ε_B be the advantage of B. Also define a mac adversary C for G as follows: given a randomly keyed member $G_{k_2^*}$ of G as oracle, C chooses a random key $k_1^{*'} \in \{0,1\}^{\kappa_1}$ and simulates A on oracle $\mathrm{MD}[F,G]_{k_1^{*'},k_2^*}$, using its own oracle to compute $G_{k_2^*}$; when

A announces a forgery, C "copies" the induced forgery $G_{k_2^*}$, which succeeds as long as the input to $G_{k_2^*}$ is fresh. Let ε_C be the advantage of C. Then we have $\varepsilon_A \leq \varepsilon_B + \varepsilon_C$ since if A succeeds a forgery, either the input to G for its forgery is fresh or else some collision for F has occurred among the computation of its queries (this uses the suffix-freeness of $\mathrm{Pad}_n(\cdot)$). Since $\varepsilon_B \leq \textbf{InSec}_F^{\mathrm{wcr}}$ and $\varepsilon_C \leq \textbf{InSec}_G^{\mathrm{mac}}$, we are done. □

Proof of Lemma 2. Let $z_\ell : \{0,1\}^m \to \{0,1\}^n$ be the function giving the ℓ-th block of g's output for some implicit selection of a key $k \in \{0,1\}^\kappa$.

Let A be a q-query MAC adversary for $f \circ g$. We can assume A makes only distinct queries. Let Forge the the event that A wins Forge$(A, f \circ g)$ and let Cover be the event that in A's sequence of queries $s_1, \ldots, s_q \in \{0,1\}^m$ there is a query s_j such that $z_\ell(s_j) \in \{z_\ell(s_i) : i < j\}$, $1 \leq \ell \leq t$; also let $\varepsilon_{\mathrm{mac}}$ be the probability of Forge and $\varepsilon_{\mathrm{cover}}$ be the probability of Cover. Then obviously

$$\varepsilon_{\mathrm{cover}} \leq \textbf{InSec}_g^{\mathrm{cover}}(T, q)$$

since an andversary A' for the game Cover(A, g) can simply simulate A by choosing random keys $k_1, \ldots, k_t$ for f in order to answer A's queries to $f \circ g$. Since A is arbitrary, and by definition of $\textbf{InSec}_{f \circ g}^{\mathrm{mac}}(T, q)$, it thus suffices to show

$$\varepsilon_{\mathrm{mac}} \leq \varepsilon_{\mathrm{cover}} + t \cdot \textbf{InSec}_f^{\mathrm{mac}}(T, q). \tag{11}$$

For this we construct an adversary B for the game Forge$(\cdot, f)$ running in time T and making at most q queries, of advantage at least $(\varepsilon_{\mathrm{mac}} - \varepsilon_{\mathrm{cover}})/t$. By definition of the game Forge(B, f), B has oracle access to some member f_{k_0} of f for some unknown key $k_0 \in \kappa'$. B samples t random keys $k_1, \ldots, k_t \in \{0,1\}^{\kappa'}$ and a random key $k \in \{0,1\}^\kappa$, chooses a random index $\ell_o \in [t]$, and simulates A on the oracle $(f \circ g)_{k\overline{k_1}\cdots\overline{k_t}}$ where $\overline{k_i} = k_i$ if $i \neq \ell_0$ and $\overline{k_i} = k_0$ otherwise (in the latter case, using its own oracle). When A announces its forgery $(s, y) \in \{0,1\}^m \times \{0,1\}^n$, B computes $f_k(s) = (z_1(s), \ldots, z_\ell(s))$. If $z_{\ell_0}(s) \notin \{z_{\ell_0}(s') : s'$ was previously queried by $A\}$ then B can win by outputting the forgery $(z_{\ell_0}(s), w)$ where

$$w = y \oplus \bigoplus_{\ell \neq \ell_0} g_{k_\ell}(z_\ell(s))$$

as long as A is correct in its forgery. Since ℓ_0 is in the set

$$\{\ell \in [t] : z_\ell(s) \notin \{z_\ell(s') : s' \text{ was previously queried by } A\}\}$$

with probability at least $1/t$ as long as this set is nonempty and since this set is nonempty as long as the event Cover doesn't occur, B thus wins with probability at least $\frac{1}{t}(\varepsilon_{\mathrm{mac}} - \varepsilon_{\mathrm{cover}})$. □

The proof of Lemma 3 requires a "multicollision to forgery" (MTF) bin-filling game of the type used in [13, 19]. The bin-filling game used for the proof of Lemma 3 is, in fact, slightly simpler than any of the games presented in [13, 19], and represents in some sense the basic core of an MTF game. For this reason, and also to distinguish it from two other MTF games presented in Section 3, we call it the "Plain" game.

'PLAIN' BALLS-AND-BINS GAME. This game is played between two adversaries $\overline{A}$ and $\overline{B}$. Parameters are integers $t, q \geq 1$. The rules are as follows:

- There is a set of "bins" (possibly infinite), which are empty at the start of the game. The game proceeds in q rounds.

- At round i, $\overline{A}$ places a ball in one of the bins. $\overline{A}$ must fill some bin with more than t balls by the end of the game.

- Before any round, $\overline{B}$ can secretly guess where $\overline{A}$ will place its next ball. $\overline{B}$ is only allowed one guess in the game. $\overline{B}$ wins if and only if it makes a correct guess.

Proposition 3 *There is a strategy for the player $\overline{B}$ in the 'Plain' balls-in-bins game that gives $\overline{B}$ chance of success at least $(qq^{1/t})^{-1}$ against any $\overline{A}$. Moreover $\overline{B}$ runs in time $\tilde{O}(q)$.*

Proof. $\overline{B}$ chooses a random index m between 1 and t and random index i between 1 and q. At the i-th round of the game, $\overline{B}$ guesses uniformly among all bins already containing m balls, or gives up if no such bins exist. For computations showing this strategy achieves chance of success at least $(qq^{1/t})^{-1}$ see [13] or [19]. For $\overline{B}$'s running time, we note that its most costly task is maintaining a list of bins with m or more balls in them. This can be done in time $\tilde{O}(q)$. □

Note: When $t = \log(q)$, a closer analysis can show $\overline{B}$'s chance of success is $q^{-1}/(1+o(1))$ with the above strategy.

For the following corollary (as in the rest of the paper) logs are base 2.

Corollary 1 *If $t \geq \log q$, there is a strategy giving $\overline{B}$ chance of success at least $1/2q$ in the Plain balls-and-bins game, where $\overline{B}$ runs in time $T + \tilde{O}(q)$.*

Proof of Lemma 3. Let $z_\ell : \{0,1\}^m \to \{0,1\}^n$ be the function giving the ℓ-th block of g's output for some implicit selection of a key $k \in \{0,1\}^\kappa$.

Let A be a q-query wcr adversary for $f \,\overline{\circ}\, g$ with chance of success $\varepsilon_{\text{coll}}$ and of running time T. Let Cover be the event that A makes (distinct) queries $s_1, \ldots, s_q$ such that there is some $j \leq q$ for which $z_\ell(s_j) \in \{z_\ell(s_i) : i < j\}$, $1 \leq \ell \leq t$. Let $\varepsilon_{\text{cover}}$ be the probability of Cover. Then it suffices to show there is some q-query adversary B of running time $T + \tilde{O}(q)$ whose chance of winning Forge(B, f) is at least

$$(\varepsilon_{\text{coll}} - \varepsilon_{\text{cover}})/2tq \log q. \tag{12}$$

B has oracle access to some random member f_{k_0} of f. To begin, B chooses a random $k \in \{0,1\}^\kappa$ and $2t$ random values $k_1, \ldots, k_t, k'_1, \ldots, k'_t \in \{0,1\}^{\kappa'}$. Then B plays one of the two following strategies, each with probability $\frac{1}{2}$:

Strategy 1: B selects a random index $\ell_0 \in [t]$ and simulates A on oracle $(f \overline{\circ} g)_{k\overline{k}_1 \cdots \overline{k}_t k'_1 \cdots k'_t}$ where $\overline{k}_\ell = k_\ell$ if $\ell \neq \ell_0$ and $\overline{k}_\ell = k_0$ otherwise; namely B computes queries to $f_{k_{\ell_0}}$ by calling its own oracle f_{k_0}, and computes all other f-queries using its own keys. B then plays a 'Plain' balls-in-bins game with A in which bins are n-bit values (one bin for each string in $\{0,1\}^n$). Each query made by A constitutes a ball, whereby the query $s \in \{0,1\}^m$ is placed in bin number $(f \circ g)_{k\overline{k}_1 \cdots \overline{k}_t}(s)$ (this this the first output half of $(f \overline{\circ} g)_{k\overline{k}_1 \cdots \overline{k}_t k'_1 \cdots k'_t}(s)$). When B guesses that a ball (query) s is about to be placed in bin y, it stops and outputs the forgery $(z_{\ell_0}(s), w)$ where

$$w = y \oplus \bigoplus_{\ell \neq \ell_0} f_{k_\ell}(z_\ell(s)).$$

Strategy 2: B selects a random index $\ell_0 \in [t]$ and simulates A on oracle $(f \overline{\circ} g)_{kk_1 \cdots k_t \overline{k}'_1 \cdots \overline{k}'_t}$ where $\overline{k'}_\ell = k'_\ell$ if $\ell \neq \ell_0$ and $\overline{k'}_\ell = k_0$ otherwise. B chooses a random index j between 1 and q, and when A makes its j-th query s_j B evaluates only the first half of output $u := (f \circ g)_{kk_1 \cdots k_t}(s_j) \in \{0,1\}^n$. B then finds the values $I = \{s_i : i < j, (f \circ g)_{kk_1 \cdots k_t}(s_i) = u\}$, and chooses a random member $s_0 \in I$, or gives up if none exists. Then B outputs the forgery $(z_{\ell_0}(s_j), w)$ where

$$w = (f \circ g)_{k\overline{k}'_1 \cdots \overline{k}'_t}(s_0) \oplus \bigoplus_{\ell \neq \ell_0} f_{\overline{k}'_\ell}(z_\ell(s_j)).$$

To analyze's B chance of succeeding, let $S = \{s_1, s_2, \ldots\}$ be the values queried by A and let $\mathsf{FullBin}$ be the event that A obtains an r-multicollision on the first output halves of its queries for some $r > \log(q)$, namely that there exists some $S' \subseteq S$, $|S'| > \log q$, such that $(f \circ g)_{k\overline{k}_1 \cdots \overline{k}_t}(s') = (f \circ g)_{k\overline{k}_1 \cdots \overline{k}_t}(s'')$ (resp. $(f \circ g)_{kk_1 \cdots k_t}(s') =$

$(f \circ g)_{kk_1 \cdots k_t}(s''))$ for all $s', s'' \in S$ under Strategy 1 (resp. Strategy 2). (For the convenience of this definition, we may assume B completes the simulation of A after it outputs its forgery.) Then $\Pr[\mathsf{FullBin}]$ is the same under Strategy 1 or Strategy 2 since the key k_0 has the same distribution as the keys $k_1, \ldots, k_\ell, k'_1, \ldots, k'_\ell$, and may therefore be indistinguishably substituted for any of these.

If $\mathsf{FullBin}$ occurs and B uses Strategy 1, then B has chance at least $1/2q$ of winning the balls-in-bins game (by Corollary 1). If it wins the balls-in-bins game, B's forgery will be correct if in addition ℓ_0 is in the set

$$\{\ell \in [t] : z_\ell(s) \text{ has not yet been queried to } f_{\overline{k}_\ell}\}$$

(where s is the ball (query) for which B makes its guess), which happens with probability at least $1/t$ as long as this set is nonempty, which itself happens as long as the event Cover does not occur. Since B uses Strategy 1 with probability $\frac{1}{2}$, we thus get that B has chance at least

$$\frac{1}{2}(\Pr[\mathsf{FullBin} \wedge \neg\mathsf{Cover}]/qt)$$

of winning from its first strategy.

If $\neg\mathsf{FullBin}$ occurs and B uses Strategy 2 *and* A obtains a collision, B has chance at least $1/q$ of guessing the index j of the second colliding input of the collision correctly and, if this guess is correct, chance at least $1/\log(q)$ of correctly selecting the first colliding input from the set I, because $\neg\mathsf{FullBin}$ means $|I| \leq \log(q)$. If all this occurs, B's forgery will be correct if in addition ℓ_0 is in the set

$$\{\ell \in [t] : z_\ell(s_j) \text{ has not yet been queried to } f_{\overline{k}'_\ell}\}$$

which happens with probability at least $1/t$ as long as this set is nonempty, which itself happens as long as Cover does not occur. Thus from its second strategy B has chance of succeeding at least

$$\frac{1}{2}(\Pr[\neg\mathsf{FullBin} \wedge \neg\mathsf{Cover} \wedge \mathsf{Collision}]/qt\log q)$$

where $\mathsf{Collision}$ denotes the event that A wins (obtains a collision). Thus B's chance of winning is at least

$$\begin{aligned}
& \frac{1}{2qt\log q}(\Pr[\mathsf{FullBin} \wedge \neg\mathsf{Cover}] + \Pr[\neg\mathsf{FullBin} \wedge \neg\mathsf{Cover} \wedge \mathsf{Collision}]) \\
\geq \quad & \frac{1}{2qt\log q}(\Pr[\mathsf{FullBin} \wedge \neg\mathsf{Cover} \wedge \mathsf{Collision}] + \Pr[\neg\mathsf{FullBin} \wedge \neg\mathsf{Cover} \wedge \mathsf{Collision}]) \\
\geq \quad & \frac{1}{2qt\log q}(\Pr[\neg\mathsf{Cover} \wedge \mathsf{Collision}] + \Pr[\neg\mathsf{Cover} \wedge \mathsf{Collision}]) \\
\geq \quad & \frac{1}{2qt\log q}(\varepsilon_{\mathrm{coll}} - \varepsilon_{\mathrm{cover}})
\end{aligned}$$

verifying (12), as desired. Finally, concerning B's running time, we see that its overhead in Strategy 1 is bounded in Proposition 1, whereas its overhead in Strategy 2 consists in the one-time task of enumerating previous queries having u as their first half of output, which takes time $O(q)$. □