An extended abstract of this paper appears in *Advances in Cryptology – CRYPTO '99*, Lecture Notes in Computer Science Vol. 1666, M. Wiener ed., Springer-Verlag, 1999. This is the full version.

Stateless evaluation of pseudorandom functions: Security beyond the birthday barrier

Mihir Bellare[*] Oded Goldreich[†] Hugo Krawczyk[‡]

June 1999

Abstract

Many cryptographic solutions based on pseudorandom functions (for common problems like encryption, message-authentication or challenge-response protocols) have the following feature: There is a stateful (counter based) version of the scheme that has high security, but if, to avoid the use of state, we substitute a random value for the counter, the security of the scheme drops below the birthday bound. In some situations the use of counters or other forms of state is impractical or unsafe. Can we get security beyond the birthday bound without using counters?

This paper presents a paradigm for strengthening pseudorandom function usages to this end, the idea of which is roughly to use the XOR of the values of a pseudorandom function on a small number of distinct random points in place of its value on a single point. We establish two general security properties of our construction, "pseudorandomness" and "integrity", with security beyond the birthday bound. These can be applied to derive encryption schemes, and MAC schemes (based on universal hash functions), that have security well beyond the birthday bound, without the use of state and at moderate computational cost.

Keywords: Pseudorandom functions, MACs, concrete security, birthday attacks.

[*] Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. E-Mail: mihir@cs.ucsd.edu. URL: http://www-cse.ucsd.edu/users/mihir. Supported in part by NSF CAREER Award CCR-9624439 and a 1996 Packard Foundation Fellowship in Science and Engineering.

[†] Department of Computer Science, Weizmann Institute of Science, Rehovot, Israel. E-Mail: oded@wisdom.weizmann.ac.il

[‡] Department of Electrical Engineering, Technion, Haifa 32000, Israel, and IBM T.J. Watson Research Center, New York, USA. Email: hugo@ee.technion.ac.il. Supported by the Fund for the Promotion of Research at the Technion.

Contents

1 Introduction

Pseudorandom functions [8] are an essential tool in many cryptographic solutions. They can be used to generate a pseudorandom pad for symmetric encryption, to mask a universal hash function for producing a secure message-authentication (MAC), to implement secure challenge-response mechanisms, and so on. In practice, one might use, in the role of pseudorandom functions, various concrete primitives, such as block ciphers or keyed hash functions under the assumption that they do possess the pseudorandomness properties in question.

The danger of repetition. In usages of pseudorandom functions such as those mentioned above, the same pseudorandom function will be applied to many values in the function's domain. In many such cases, security can be compromised if one applies the pseudorandom function twice to the same point. Consider as an example the following method of encryption. Two parties share a key which specifies a function $f\colon \{0,1\}^n \to \{0,1\}^m$ from some fixed pseudorandom function family. In order to encrypt a message M of length m, the sender computes f on an element $v \in \{0,1\}^n$ and then sends the pair $(v, M \oplus f(v))$. Clearly, the security of such a scheme depends on never re-using the same value v for encrypting different messages. The same problem arises in other applications of pseudorandom functions, including MACs and challenge-response protocols.

1.1 Counters versus coins

Using counters. A natural way to avoid repetition is for the sender to use (as the points on which to evaluate the function) an increasing counter, or other form of varying, non-repeating state, which is updated with each application of the function. This does very well in terms of avoiding repetition, but can have various drawbacks depending on the setting and application.

Maintaining a counter, or other state information, might in some settings be impractical or unsafe. This can happen, for example, whenever maintaining a synchronized state across different applications of the function is unsafe or impossible. Such is the case of a function that is used across different sessions (or invocations) of a protocol, or used (possibly simultaneously) by different users or components of a system. Additional examples include the use of smart-cards, or authentication tokens, that store the key to a pseudorandom function in persistent memory but are not equipped with non-volatile writeable memory to store the varying value of a counter. Even in cases where such a varying state can be stored, security is susceptible to system failures that may reset the value of that counter.

Also some applications require more for security than mere non-repetitiveness of the value to which the pseudorandom function is applied; e.g., the value might be a challenge which should be unpredictable, and a counter value is of course highly predictable. In this case too, the use of counters is not possible at all.

Using coins. Another possibility is to use *random values* as those on which to evaluate the function. This can avoid the need to store varying information, and also yield unpredictability, thereby avoiding the drawbacks of counters. However, randomness might do less well at the task we first highlighted, namely avoiding repetition. This is due to the "birthday" phenomenon, which means that if the domain of the function has size $N = 2^n$, and we apply the function to a sequence of q points selected at random from the domain, we have probability about q^2/N of seeing a repetition in the selected points. In the encryption example discussed above, this represents a significant decrease in the number of messages that can be safely encrypted: only $\sqrt{N}$ if we use random values for the point v, but up to N (depending on the security of the pseudorandom function family) if we use counters.

	Construction	**Insecurity**		**No. f-appls.**
		Upper bound	Lower bound	
1.	CBC-2	$\frac{12q^2}{N}$ [6]	$\Omega(\frac{q^2}{N})$ [15]	2
2.	Feistel-t ($t = 3, 4$)	$\frac{q^2}{N}$ [11]	$\Omega(\frac{q^2}{N})$	t
3.	[12]	$O(\frac{q^2}{N})$ [12]	$\Omega(\frac{q^2}{N})$	2
4.	Benes [1]	$O\left(\frac{q}{N}\right)$ [1]	$\Omega\left(\frac{q}{N}\right)$	8
5.	Ω_t ($t \geq 1$) [13]	$\frac{q^{t+1}}{(t+1)N^t}$ [13]	?	$2t$
6.	Feistel-6	$O(\frac{q^4}{N^3} + \frac{q^2}{N^2})$ [14]	?	6

Figure 1: Input-length doubling transformations: Constructing g: $\{0,1\}^{2n} \to \{0,1\}^n$ given f: $\{0,1\}^n \to \{0,1\}^n$. The insecurity is the maximum adversarial success in q queries. Both upper bounds and lower bounds (attacks) on the insecurity are shown. Here $N = 2^n$. "No. f-apps" is the number of applications of f used in one computation of g, and is the main cost. "Feistel-t" means t rounds, and "CBC-2" means CBC on two blocks. Constructions **2,3,4,6** yield maps of $2n$ bits to $2n$ bits; in our context it is implicit that the outputs are truncated. Question marks mean we don't know. See the text for (even) more discussion.

Thus the birthday bound for query collisions may become the security bottleneck of the whole application. This is particularly evident when using 64-bit input pseudorandom functions, such as those based on DES. In this case a number $q = 2^{32}$ of queries nullifies the quantified security; even $q = 2^{25}$ leaves us with an insecurity (ie. chance that the scheme may be broken) of $q^2/N = 2^{-14}$, which is fairly high. With 128-bit blocks (such as in the AES proposals) the probability of repeated queries leaves less security than usually intended: in this case $q = 2^{32}$ provides 2^{-64} insecurity, less than the usually conjectured "128-bit security" for these ciphers.

The above discussion raises the natural question of to what extent the use of varying state (e.g. counters) is *essential* for avoiding the quadratic degradation in the security of the function. In other words, can we combine the advantages of coins and counters: get security beyond the birthday bound, yet avoid the need to maintain state?

Using input-length doubling transformations. One approach is to change the pseudorandom function and use instead one with a larger domain. For example, instead of f: $\{0,1\}^n \to \{0,1\}^m$, we use a pseudorandom function g: $\{0,1\}^{2n} \to \{0,1\}^m$. This however can be impractical, or may not increase security in the desired way, as we now discuss.

Since total redesign of the function is typically not desirable, one would usually try to build g in a generic way from f. Figure 1 summarizes the main known designs. (It sets $m = n$ for simplicity.) For example, one can use the popular CBC-MAC construction. Another alternative is to use one of many known transformations of pseudorandom functions on n bits to pseudorandom permutations (or functions) on $2n$ bits, and simply drop all but the last m bits of the output. (Constructions **2,3,4,6** of the table fall in this class, while construction **5** is directly of $2n$ bits to n bits.) Figure 1 indicates the best known analyses upper bounding the insecurity, the best known attacks lower bounding the insecurity, and the cost measured in terms of the number of applications of f needed to make one computation of g. As the table indicates, the most efficient known constructions are still vulnerable to attacks that in q queries achieve success related to q^2/N where $N = 2^n$ is

	Property	**Insecurity**		**No. f-appls.**
		Upper bound	Lower bound	
1.	Pseudorandomness	$O(t!) \cdot \frac{q^2}{N^t}$	$\Omega(t!) \cdot \frac{q^2}{N^t}$	t
2.	Integrity	$(t \lg N)^{O(t)} \cdot \frac{q^3}{N^t}$	$\Omega(t^t) \cdot \frac{q^3}{N^t}$	t

Figure 2: The two security properties of the t-fold parity construction for $t \geq 1$: Parameters are as in Figure 1. This is true for $q < N/(2e^2 t)$, and t is odd in **2**. Bounds shown are approximate.

the domain size of the *original* function. (In particular **1,2,3**). The last three constructions have better bounds on the insecurity, but as the table shows, their computational cost (the number of f-applications) is relatively high. In particular, as we will see (Figure 2), it is higher than the cost of our methods discussed below.

1.2 The parity method and results in brief

Construction. In this paper we propose and investigate a simple mechanism to go beyond the birthday barrier without using counters or state information. We call it the *"parity method"*. Instead of computing the function at a single random point, compute it at several random (but distinct) points (typically two or three points will suffice) and take the parity of the results (namely, XOR these values). For instance, in the above encryption example, if the sender wants to encrypt plaintext M, he will choose two different random values r_1, r_2 from the domain of the function, and send to the other party as the ciphertext the triple $(r_1, r_2, M \oplus f(r_1) \oplus f(r_2))$. Similar methods will be used for other applications such as challenge-response, message authentication, or key derivation. As a result our methods offer a sateless alternative to achieve the high security of stateful schemes at a moderate computational cost but with increased use of random bits.

Security. We are interested in proving general security properties of the parity method that can later be applied to prove the security of specific encryption schemes (such as the one discussed above) or MAC schemes (such as we will discuss below). Accordingly, we begin by considering the probabilistic function that embodies the parity construct, namely

$$F(r_1, \ldots, r_t) \; = \; \bigoplus_{i=1}^{t} f(r_i) \tag{1}$$

where the r_i's are uniformly chosen different n-bit numbers. The first security property we consider is pseudorandomness, or "distinguishability distance" from true randomness, of the (randomized) function F. This corresponds to passive attacks. The second security property we call "integrity", and it corresponds to certain kinds of active attacks. (In the coming sections we will discuss these properties in more depth, and see how they apply to encryption and MAC respectively.) In either case we are interested in how the security of this randomized function degrades after q queries relative to the security of the original pseudorandom function f. Our analyses reduce this question to a purely information-theoretic setting, and show that the parity method amplifies security at quite a high rate, enabling one to move well beyond the birthday barrier. Our results are displayed in Figure 2 and discussed below.

Pseudorandomness amplification and encryption. An adversary sees q vectors $(r_1, \ldots, r_t)$ and the output of the parity function on them. We define a certain "bad" event and show that subject to its not happening, the outputs look uniform. Exploiting and extending a connection of [4], the bad event is that a certain matrix associated to the vectors is not of full rank. Lemma 3.3 bounds this probability roughly by:

$$O(t!) \cdot \frac{q^2}{N^t} \quad \text{for} \quad q \leq \frac{N}{e^2 t} \ , \tag{2}$$

where $N = 2^n$ is the size of the domain of the function.[1] (The bound on q is necessary: we note in Section 3.2 why the parity construct is not pseudorandom when $q > N$.) Remarkably, the bound Equation (2) shows that if f is chosen as a truly random function then the effect of the parity construct of Equation (1) on limiting the degradation of security due to repeated queries is, for $q < O(N/t)$ and small t, close to the effect of applying a random function on single inputs of length tn. Indeed, in the latter case the distance from randomness is, using the birthday argument, of the order of $\frac{q^2}{N^t}$. That is, we approximate the effect of a t-fold increase in the queries size without necessitating any change to the underlying function f. We note that the bound is tight.

The encryption scheme discussed above, a special case of the CTR scheme in [2], was shown by the latter to have insecurity (under a chosen-plaintext attack of $q < N$ messages) at most ϵ, the maximum possible attainable advantage in breaking the underlying pseudorandom function in q queries and time related to that allowed the encryption attacker. The insecurity of the randomized (stateless) version is only bounded by $\epsilon + q^2/N$ due to birthday attacks. In Section 3.3 we consider the (counter-less) encryption scheme in which to encrypt plaintext M, we choose t distinct random values $r_1, \ldots, r_t$ and set the ciphertext to $(r_1, \ldots, r_t, F(r_1, \ldots, r_t) \oplus M)$. Theorem 3.7 bounds its insecurity by the term of Equation (2) modulo an additive term corresponding to the insecurity of F under tq queries. Considering the case $t = 2$ discussed above, for $q = O(\sqrt{N})$, the new scheme has security which is close to the counter-version of the basic CTR scheme, whereas the coin-version of the basic scheme is totally insecure at $q = \sqrt{N}$. Furthermore the security gets even better with larger t.

Our improvements are more than in merely going beyond the birthday barrier. The insecurity of the parity construct grows much more slowly with q than the insecurity of constructs from Figure 1 when $q > \sqrt{N}$. This is true already with $t = 2$, and with $t = 3$ the gap is quite large. For more information see the plots in Figure 3.

Integrity amplification and message authentication. In the Carter-Wegman paradigm [17], the MAC of message M is $(C, h(M) \oplus f(C))$, where C is a counter value, f is a pseudorandom function (PRF), and h is a δ-AXU hash function [10]. When trying to make this stateless by substituting a random string for C, security drops to the birthday bound. The same situation arises in the XOR MAC schemes of [4]. A counter based variant of their scheme has high security, but the stateless version substitutes a random value for the counter and security drops to the birthday bound. The modified (stateless) Carter-Wegman MAC scheme we propose is that the MAC of message M be $(r_1, \ldots, r_t, h(M) \oplus F(r_1, \ldots, r_t))$ where $r_1, \ldots, r_t \in \{0,1\}^n$ are random but distinct points, and f, h are as before. Here t is a parameter, and the higher we set it, the more security we get, though each increment to t costs one extra application of the PRF.

The pseudorandomness of the parity construct does not by itself guarantee security of the above due to the fact that an adversary in a MAC setting is allowed an active attack, and can attempt a forgery in which the values $r_1, \ldots, r_t$ are of its own choice. We propose another property of the

[1] We are simplifying a little, but the deviation from the more complex bound of Equation (5) is insignificant for reasonable values of t. The constant in the big-oh in $O(t!)$ can be taken to be one in all practical situations, except for the case $t = 2$ when it is larger, about 30. (Practical means N is large and t is small.) See Corollary 3.4 and Lemma 3.3 for more information.

parity construct we call "integrity". We again reduce the analysis to the question of whether the matrix associated to the points on which the parity function is evaluated has a certain property, which we call "vulnerability" and is defined in Section 4. Improvement over the birthday bound occurs only at $t \geq 3$. Specifically, for odd t, Lemma 4.2 bounds the probability of vulnerability by

$$d'(t, \lg N) \cdot \frac{q^3}{N^t} \quad \text{for} \quad q \leq \frac{N}{2e^2 t}\ , \tag{3}$$

where $N = 2^n$ and $d'(t, \lg N)$ is a polynomial in $\lg N$ for each fixed t, whose value is specified by Equation (17). (Curiously enough, the bound for even $t \geq 4$ is typically inferior to the bound for $t-1$. Specifically, for even t our bound is $d'(t, \lg N) \cdot \frac{q^2}{N^{t/2}}$, which is tight.) Note that this expression is inferior to the one obtained in Equation (2). Still, it suffices for our applications. We apply this to get Theorem 4.4, an analysis of the security of the MAC scheme discussed above.

1.3 Discussion and related work

One should note that getting security beyond the birthday bound (both in the case where one uses counters, and in our setting where one does not) requires that we use a pseudorandom function family which itself has security beyond the birthday bound. This precludes the direct use of block ciphers; since they are permutations, their security does not go beyond the birthday bound. The question of designing pseudorandom functions (with security beyond the birthday bound) out of pseudorandom permutations (which model block ciphers) was first considered by Bellare, Krovetz and Rogaway [7] and later by Hall, Wagner, Kelsey and Schneier [9] and Bellare and Impagliazzo [5]. These works provide several constructions that one might use. (The works of [7, 9] were also motivated by the desire to get beyond the birthday bound for encryption, but were using a counter-based encryption scheme: their applications are not stateless.)

Shoup [16] considers various ways of providing better security tradeoffs when using pseudorandom functions or permutations as masks in universal-hash function based MACs. He gets the security to decrease slower as a function of the number of queries, but does not get security beyond the birthday bound without the use of state.

1.4 Organization

In Section 2 we recall definitions of pseudorandom functions, encryption schemes and MAC schemes and their security. Section 3 considers the pseudorandomness of parity and its application to encryption, while Section 4 considers the integrity properties of parity and their application to message authentication. A wider perspective is provided in the Appendix A where we consider an arbitrary randomized process which is being applied iteratively on the same random-pad (or random function).

2 Definitions

Primitives discussed in this paper include pseudorandom function families [8], symmetric encryption schemes, and MACs. Security of all these will be treated in a concrete framework along the lines of works like [6, 2]. Since this approach is by now used in many places, we will briefly summarize the concepts and terms we need.

The definitional paradigm we employ is to associate to any scheme an *insecurity function* which, given some set of parameters defining resource limitations, returns the maximum possible success probability of an adversary limited to the given resources. The definition of "success" various with the goal of the primitive, as do the resources considered.

Throughout the paper we assume some fixed RAM model of computation and measure computation time (of a given algorithm on certain inputs) by the number of steps in this model. This enables us to consider the computational complexity of tasks defined on finite domains. When we refer below to "running time" we mean the time in this sense, plus the size of the description of the algorithm (namely, the code).

Pseudorandom function families. [Notion of [8], concretized as per [6]]. To a family F of functions (in which each function maps $\{0,1\}^n$ to $\{0,1\}^m$) we associate an *insecurity function* $\mathbf{InSec}^{\rm prf}(F,\cdot,\cdot)$ defined as follows: For integers q, T the quantity $\mathbf{InSec}^{\rm prf}(F,q,T)$ is the maximum possible "advantage" that an adversary can obtain in distinguishing between the cases where its given oracle is a random member of F or a truly random function of $\{0,1\}^n$ to $\{0,1\}^m$, when the adversary is restricted to q oracle queries and running time T.

More precisely let R be the family of all functions each mapping $\{0,1\}^n$ to $\{0,1\}^m$, and F a subset of R. Goldreich, Goldwasser and Micali [8] define the advantage $\mathsf{Adv}^{\rm prf}(D)$ of a (distinguisher) adversary D in breaking F as

$$\mathsf{Adv}^{\rm prf}(D) \;=\; \left| \Pr_{f \overset{R}{\leftarrow} F}\left[D^f = 1\right] - \Pr_{f \overset{R}{\leftarrow} R}\left[D^f = 1\right] \right| \,.$$

The insecurity of F as a pseudorandom function family is the function $\mathbf{InSec}^{\rm prf}(F,\cdot,\cdot)$, where $\mathbf{InSec}^{\rm prf}(F,q,T)$ is the maximum value of $\mathsf{Adv}^{\rm prf}(D)$, taken over all adversaries D that make up to q queries of their function oracle f and run in time at most T [6].

Symmetric encryption schemes. [Following [2]]. To a symmetric encryption scheme ENC (consisting of a probabilistic encryption algorithm and deterministic decryption algorithm) we associate an *insecurity function* $\mathbf{InSec}^{\rm enc}(\mathsf{ENC},\cdot,\cdot)$ defined as follows: For integers μ, T the quantity $\mathbf{InSec}^{\rm enc}(\mathsf{ENC},\mu,T)$ is the maximum possible probability that an adversary can "break" the encryption scheme under a chosen-plaintext attack in which a total of μ plaintext bits are encrypted and the running time of the adversary is restricted to T. ("Break" here means in the sense of real-or-random security as detailed in [2].)

MACs. [Following [4]]. Unlike what is traditionally called a "MAC," ours generate tags (called macs) probabilistically. Verification thus cannot be done by tag re-computation. Accordingly, the description of a specific MAC scheme MAC includes two procedures, a mac generation procedure we denote $\mathsf{MAC.gen}$, and a mac verification procedure we denote $\mathsf{MAC.vf}$. The first (which is randomized) takes K, M and returns a mac; the second (which is deterministic) takes K, M, σ' and returns 0 or 1. Naturally, macs generated by the generation procedure are accepted by the verification procedure. Following [4], the success $\mathsf{Succ}^{\rm mac}(A)$ of an adversary A attacking a specific scheme MAC is the probability (over the choice of key K and the coins of A) that the following experiment returns 1–

```
Choose key $K$ at random and let $(M,\sigma) \leftarrow A^{\mathsf{MAC.gen}(K,\cdot)\,,\,\mathsf{MAC.vf}(K,\cdot,\cdot)}$
If $\mathsf{MAC.vf}(K,M,\sigma) = 1$ and $M$ was never queried of $\mathsf{MAC.gen}(K,\cdot)$
    then return 1 else return 0
```

The insecurity of MAC as a MAC is the function $\mathbf{InSec}^{\rm mac}(\mathsf{MAC},\cdot,\cdot,\cdot)$, where $\mathbf{InSec}^{\rm mac}(\mathsf{MAC},q_a, q_v,T)$ is the maximum of $\mathsf{Succ}^{\rm mac}(A)$ taken over over all adversaries A that make up to q_a mac generation queries and q_v mac verification queries and run for time at most T. We adopt the convention that the adversary makes a verification query on the attempted forgery it outputs, so that q_v is always at least 1.

Conventions. In any insecurity function, we might drop the time argument T, and it is to be understood then that the time allowed the adversary is not restricted, meaning we are in an information theoretic setting. Indeed, this will be the important case in analyses.

3 Pseudorandomness of parity and application to encryption

We need a bit of terminology. A sequence $R = (r_1, \ldots, r_t)$ of n-bit strings is called *non-colliding* if the t strings $r_1, \ldots, r_t$ are all distinct. We let $D(n, t)$ denote the set of all non-colliding t-sequences of n-bit strings. We let $R(n, m)$ denote the set of all functions of $\{0,1\}^n$ to $\{0,1\}^m$.

3.1 Distributions and matrix connection

Parity distribution. Consider the following game. A random function f from $R(n, m)$ is chosen and fixed. Then q non-colliding sequences, $R_i = (r_{i,1}, \ldots, r_{i,t})$ for $i = 1, \ldots, q$, are chosen randomly and independently. An adversary is provided these sequences together with the q corresponding output values of the parity function, namely $b_i = f(r_{i,1}) \oplus \cdots \oplus f(r_{i,t})$ for $i = 1, \ldots, q$. In applications, it is typical that as long as $b_1, \ldots, b_q$ look like random independent m-bit strings (given the other information), the adversary will not be able to derive any "advantage" in "breaking" the security of the application, whatever that may be. This will be seen more clearly and specifically later, but for the moment we wish only to give some clue as to the motivation for what we now look at. Namely, the experiment which produces the output just described, which we call $\mathrm{Par}(n, m, q, t)$. We wish to "compare" this to the output of the experiment which picks $R_1, \ldots, R_q$ the same way, and $b_1, \ldots, b_q$ randomly. The experiments are described below.

```
Experiment Par$(n, m, q, t)$
    $f \stackrel{R}{\leftarrow} R(n, m)$
    For $i = 1, \ldots, q$ do
        $R_i = (r_{i,1}, \ldots, r_{i,t}) \stackrel{R}{\leftarrow} D(n, t)$
        $b_i \leftarrow \bigoplus_{j=1}^{t} f(r_{i,j})$
    End For
    Output $(R_1, b_1, \ldots, R_q, b_q)$
```

```
Experiment Rnd$(n, m, q)$
    For $i = 1, \ldots, q$ do
        $R_i = (r_{i,1}, \ldots, r_{i,t}) \stackrel{R}{\leftarrow} D(n, t)$
        $b_i \stackrel{R}{\leftarrow} \{0,1\}^m$
    End For
    Output $(R_1, b_1, \ldots, R_q, b_q)$
```

A natural comparison measure is the statistical distance between the output distributions of these experiments, defined as

$$\begin{aligned} &\textsc{StatDist}\,[\mathrm{Par}(n, m, q, t), \mathrm{Rnd}(n, m, q)] \\ &\quad = \max_J \left\{ \left| \Pr\left[J(\omega) = 1 \,:\, \omega \stackrel{R}{\leftarrow} \mathrm{Par}(n, m, q, t) \right] - \Pr\left[J(\omega) = 1 \,:\, \omega \stackrel{R}{\leftarrow} \mathrm{Rnd}(n, m, q) \right] \right| \right\} , \end{aligned}$$

the maximum being over all (computationally unlimited) "judging" algorithms J that return either 0 or 1 on any input. We would like to upper bound this. In fact we will need a stronger claim. We will define a certain "bad" event, and upper bound its probability. We will also assert that conditioned on the bad event not occurring, the outputs of the two experiments are identically distributed. (The bad event will depend only on the choices of $R_1, \ldots, R_q$ hence is defined and has the same probability under both experiments.) In other words, when the bad event does not occur, the outputs $b_1, \ldots, b_q$ of the parity experiment are random and uniform. As Proposition 3.2 indicates it follows that the statistical distance between the output distributions of the two experiments is bounded by the probability of the bad event, but applications will in fact exploit the stronger assertion.

Matrix to pseudorandomness connection. The definition of the bad event is based on an association of a matrix to the parity distribution. This connection is taken from [4], where it is used to analyze a MAC construction based on the XOR operation. We adapt it for our purposes. Then the bulk of our analysis focuses on this matrix. Let us now describe the matrix and explain more precisely the connection to the pseudorandomness of parity.

To any non-colliding sequence $R = (r_1, \ldots, r_t)$ of n-bit strings is associated its characteristic vector of length $N = 2^n$, denoted $\mathrm{ChVec}(R)$. Namely, if we consider the values r_i as representing integer numbers between 0 and $N-1$ then the characteristic vector of $r_1, \ldots, r_t$ will have a value of 1 in the positions corresponding to these t numbers and 0 elsewhere. If $R_1, \ldots, R_q$ are non-colliding sequences we denote by $\mathrm{MTX}_{N,q}(R_1, \ldots, R_q)$ the q by N matrix (of zeros and ones) whose i-th row is $\mathrm{ChVec}(R_i)$ for $i = 1, \ldots, q$. We are interested in the rank of our matrix when it is viewed as a random variable over the choices of $R_1, \ldots, R_q$ from $D(n,t)$. Specifically, we want the matrix to have full rank, meaning rank equal to the number of rows q. We consider the probability that this does not happen:

$$\mathsf{NFRProb}(N,q,t) \quad = \quad \Pr\left[\mathrm{MTX}_{N,q}(R_1, \ldots, R_q) \text{ is not of rank } q \;:\; R_1, \ldots, R_q \stackrel{R}{\leftarrow} D(n,t)\right] .$$

Now, let $b_i = f(r_{i,1}) \oplus \cdots \oplus f(r_{i,t})$ for $i = 1, \ldots, q$. View the values $b_1, \ldots, b_q$ as arranged in a column vector consisting of q strings, each m-bits long. Then notice that this vector is given by the following matrix vector product, where as before we identify $\{0,1\}^n$ with $\{0, 1, \ldots, N-1\}$ for simplicity:

$$\mathrm{MTX}_{N,q}(R_1, \ldots, R_q) \cdot \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(N-1) \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_q \end{bmatrix} . \qquad (4)$$

Namely $b_1 = f(r_{1,1}) \oplus \cdots \oplus f(r_{1,t}) = \sum_j f(j)$, the sum being taken over all values j for which the j-th coordinate of $\mathrm{ChVec}(R_1)$ is 1, and so on.

The following lemma says that as long as the matrix has full rank, the entries of the output vector are uniformly and independently distributed over $\{0,1\}^m$. That is, they look like the outputs of a random function with range $\{0,1\}^m$ being evaluated at q distinct points. It is an adaption of a lemma of [4] to our setting, and is informally stated. We will not prove this here; the reader is referred to [4].

Lemma 3.1 Conditioned on the event that $\mathrm{MTX}_{N,q}(R_1, \ldots, R_q)$ is of rank q, the outputs of experiment $\mathrm{Par}(n,m,q,t)$ and experiment $\mathrm{Rnd}(q,t)$ are identically distributed.

The implication in terms of the usage of the parity construct is that as long as the matrix maintains full rank, seeing the outputs of the parity construct yields no information at all to an adversary. It is just like seeing values of a random function on distinct points. Accordingly, adversarial success will only happen when the matrix is *not* of full rank. For this reason, our efforts are concentrated on upper bounding $\mathsf{NFRProb}(N,q,t)$. Before we do that, however, let us state some relations between the probability we are considering and the statistical distance discussed earlier. Although we don't use these facts directly, they are useful in understanding the results.

Proposition 3.2 Let $N = 2^n$. Then

$$[1 - 2^{-m}] \cdot \mathsf{NFRProb}(N,q,t) \quad \leq \quad \textsc{StatDist}\left[\mathrm{Par}(n,m,q,t), \mathrm{Rnd}(n,m,q)\right] \quad \leq \quad \mathsf{NFRProb}(N,q,t) .$$

Proof: The upper bound follows from Lemma 3.1. For the lower bound we present a judge J achieving advantage equal to the claimed bound. The judge J gets input $\omega = (R_1, b_1, \ldots, R_q, b_q)$ and wants to tell from which of the two distributions ω was drawn. If $\mathrm{MTX}_{N,q}(R_1, \ldots, R_q)$ has rank q then J can obtain no advantage, and hence outputs a random bit. Else it fixes some i and some non-empty $S \subseteq \{1, \ldots, q\} - \{i\}$ such that $R_i = \oplus_{j \in S} R_j$. If $b_i = \oplus_{j \in S} b_j$ then it guesses that ω was from $\mathrm{Par}(n, m, q, t)$ (meaning it outputs 1) else it guesses ω was from $\mathrm{Rnd}(n, m, q)$ (meaning it outputs 0) The advantage of this judge is the claimed lower bound. ▌

In other words, the statistical distance is fully captured by the probability that the matrix is not of full rank: to within a constant factor, it is both an upper and a lower bound.

3.2 Main lemma: Bound on $\mathsf{NFRProb}(N, q, t)$

The heart of our analysis reduces by the above to upper bounding the probability that the matrix $\mathrm{MTX}_{N,q}(R_1, \ldots, R_q)$ is not of full rank when $R_1, \ldots, R_q$ are randomly and independently chosen non-colliding vectors. The bound is given in terms of $N = 2^n$, t and q in the following lemma. Here e is the base of the natural logarithm.

Lemma 3.3 Let t be such that $1 \leq t \leq \sqrt{N/(e \lg N)}$. Then for any $q < N/(e^2 t)$ we have

$$\mathsf{NFRProb}(N, q, t) \;\leq\; d_1(t) \cdot \frac{q^2}{N^t} \;+\; \begin{cases} d_2(t, \lg N) \cdot \dfrac{q^3}{N^{3t/2}} & \text{if } t \text{ is even} \\[2ex] d_2(t, \lg N) \cdot \dfrac{q^4}{N^{2t}} & \text{if } t \text{ is odd ,} \end{cases} \tag{5}$$

where

$$d_1(t) \;=\; 0.76 \cdot t! \tag{6}$$

and

$$d_2(t, n) \;=\; \begin{cases} 3e^{3+3t/2} 2^{-3} t^{-3+3t} n^{-3+3t/2} & \text{if } t \text{ is even} \\ e^{4+2t} 2^{-4} t^{-4+4t} n^{-4+2t} & \text{if } t \text{ is odd.} \end{cases} \tag{7}$$

Discussion of the bounds. Let us now interpret the bounds of Lemma 3.3. First, the upper bound on t is a technicality insignificant in practice, and safely ignored. (For example if $N = 2^{64}$ it says roughly that $t \leq 2^{29}$, and we are interested in values like $t = 2, 3, 4$.) The bound on q indicates that we are not expecting security for q above N; in fact q must be $O(N)$. This is necessary, as explained below. The main thing is Equation (5) which says that $\mathsf{NFRProb}(N, q, t)$ is roughly bounded by q^2/N^t. This is modulo a small constant factor, and also an additive term. The additive term has a factor of $q^s/N^{st/2}$ with $s \geq 3$, which is small enough to make the whole additive term negligible, even given the somewhat large seeming coefficient $d_2(t, \lg N)$. Accordingly it is safe to view the above bound as essentially $d_1(t) \cdot q^2/N^t$ for common values of N, t, and in later parts of this paper we will make this simplifying assumption.

Examples. It is worth looking at some special cases to see how these bounds behave. Typical pseudorandom functions candidates (eg. block ciphers) have $n \geq 64$ so let us make this assumption. That means $N \geq 2^{64}$. We are interested in values of t as small as possible. The following indicates that there is little reason to go beyond $t = 3$, and even $t = 2$ is very good.

Corollary 3.4 Let $n \geq 64$ and let $N = 2^n$. Assume $q < N/23$. Then

$$\mathsf{NFRProb}(N, q, t) \quad \leq \quad 57 \cdot \frac{q^2}{N^2} \quad \text{if } t = 2, \tag{8}$$

and

$$\mathsf{NFRProb}(N, q, t) \quad \leq \quad 5 \cdot \frac{q^2}{N^3} \quad \text{if } t = 3. \tag{9}$$

Proof: Apply Lemma 3.3. We take $q < N/(3e^2)$ to cover the cases $t = 2$ and $t = 3$. For $t = 2$ we have $d_2(t, n) = 3e^6$. Thus the bound is

$$\begin{aligned}
\mathsf{NFRProb}(N, q, t) &\leq d_1(2) \cdot \frac{q^2}{N^2} + 3e^6 \cdot \frac{q^3}{N^3} \\
&\leq \left[d_1(2) + 3e^6 \cdot \frac{q}{N} \right] \cdot \frac{q^2}{N^2} \\
&\leq \left[1.52 + 3e^6 \cdot \frac{1}{3e^2} \right] \cdot \frac{q^2}{N^2} \,.
\end{aligned}$$

Simplifying yields Equation (8). Note we did not need the lower bound on n here since $d_2(2, n)$ does not depend on n.

For $t = 3$ the bound becomes

$$\begin{aligned}
\mathsf{NFRProb}(N, q, t) &\leq d_1(3) \cdot \frac{q^2}{N^3} + e^{10} 2^{-4} 3^8 n^2 \cdot \frac{q^4}{N^6} \\
&\leq \left[d_1(3) + e^{10} 2^{-4} 3^8 n^2 \cdot \frac{q^2}{N^3} \right] \cdot \frac{q^2}{N^3} \\
&\leq \left[4.56 + e^{10} 2^{-4} 3^8 n^2 \cdot \frac{1}{3^2 e^4 N} \right] \cdot \frac{q^2}{N^3} \\
&\leq \left[4.56 + e^6 2^{-4} 3^6 \cdot \frac{\lg^2(N)}{N} \right] \cdot \frac{q^2}{N^3} \\
&\leq \left[4.56 + e^6 2^{-4} 3^6 \cdot \frac{64^2}{2^{64}} \right] \cdot \frac{q^2}{N^3} \\
&\leq \left[4.56 + 2^{-37} \right] \cdot \frac{q^2}{N^3} \,.
\end{aligned}$$

This yields Equation (9). ▌

As these calculations indicate, the influence of the second term in the bound effectively vanishes for $t \geq 3$ as long as N is reasonably large. That is why, as indicated above, there is little loss in thinking of the bound as $O(t!) \cdot q^2/N^t$ in general.

Figure 3 plots for comparison's sake the functions of Corollary 3.4 and also some functions from Figure 1. The conclusion is that the parity construct with $t = 2$ does as well as the best previous known construct although at lower cost, while with $t = 3$ the security is higher than any previous known construct, yet the cost stays lower.

Tightness of the bound. The upper bound of Lemma 3.3 can be proven to be approximately tight by considering the event in which two rows in $\mathrm{MTX}_{N,q}(R_1, \ldots, R_q)$ are identical. This is an instance of the usual birthday paradox: We are selecting q rows from a universe of $\binom{N}{t}$ possible

Note: Use ghostview to see the graphs in color.

Here we have set $N = 2^{64}$ and $x = \lg(q)$ and plotted the following functions:

$$\begin{aligned}
\mathrm{bd}(x) &= \frac{2^{2x}}{N} &&= \frac{q^2}{N} \\
\mathrm{l}(x) &= \frac{2^{x}}{N} &&= \frac{q}{N} \\
\mathrm{p2}(x) &= 57 \cdot \frac{2^{2x}}{N^2} &&= 57 \cdot \frac{q^2}{N^2} \\
\mathrm{p3}(x) &= 5 \cdot \frac{2^{2x}}{N^3} &&= 5 \cdot \frac{q^2}{N^3}\ .
\end{aligned}$$

The first function is the birthday bound, refelcting the insecurity of the first few constructions of Figure 1. We refer to $\mathrm{l}(x)$ as the linear bound: it is that of the of Aiello and Venkatesan's Benes construction [1], also listed in Figure 1. The third is the bound for the pseudorandomness of the parity construct when $t = 2$ as given by Corollary 3.4, and the last is the case $t = 3$ as given by the same Corollary. The first graph shows the birthday behavior: function $\mathrm{bd}(x)$ goes to 1 as q hits $2^{32} = \sqrt{N}$, while the others stay small much longer. A closer look at the "tail" is provided by the second graph. It indicates that insecurity function of the parity construct with $t = 2$ is lower than the linear function upto about $q = 2^{58}$, then gets higher. The insecurity function of the parity construct with $t = 3$ however is essentially zero; at these scales it simply does not lift off the x-axis in the above pictures. In summary the parity construction with $t = 2$ yields essentially the same security as the best previous construction but at lower cost, while that with $t = 3$ yields better security and still at lower cost.

Figure 3: Comparitive plots of insecurity functions.

rows. Then a standard birthday calculation (we take the specific estimates used here from [4]) says that for $2 \leq q \leq \sqrt{\binom{N}{t}}$ the probability of collisions is at least

$$0.16 \cdot \frac{q^2}{\binom{N}{t}} \;\geq\; 0.16 \cdot \frac{q^2}{N^t/t!} \;\geq\; 0.16 \cdot t! \cdot \frac{q^2}{N^t} \; .$$

Comparing with the first term in the bound of Lemma 3.3 we see that the bounds are tight to within a constant that is independent of N, t, q.

Necessity of bound on q. Lemma 3.3 holds only for $q < O(N)$. It turns out the bound on q is necessary up to the constant factor: if $q > N$ then the matrix $\mathrm{MTX}_{N,q}(R_1, \ldots, R_q)$ obviously does not have rank q, since q (the number of rows) is more than N (the number of columns). That is, $\mathsf{NFRProb}(N, q, t) = 1$ when $q > N$. Note that this reflects the fact that the parity construct is not pseudorandom when $q > N$: applying Proposition 3.2 with $q > N$ we have

$$\textsc{StatDist}\,[\mathrm{Par}(n, m, q, t), \mathrm{Rnd}(n, m, q)] \;\geq\; \frac{1}{2} \; ,$$

meaning the statistical distance between the distributions output by experiments $\mathrm{Par}(n, m, q, t)$ and $\mathrm{Rnd}(n, m, q)$ is large when $q > N$. So indeed, no security can be expected from the parity construct when $q > N$.

Pseudorandomness of the parity construct. Putting together Proposition 3.2 and Lemma 3.3 tells us that except with probability about $d_1(t) \cdot q^2/2^{nt}$, the outputs of experiments $\mathrm{Par}(n, m, q, t)$ and $\mathrm{Rnd}(q, t)$ are identically distributed. More precisely:

Corollary 3.5 Let $n \geq 1$ and $N = 2^n$. Let t be such that $1 \leq t \leq \sqrt{2N/(e \lg N)}$. Let $d_1(t)$ and $d_2(t, n)$ be as in Lemma 3.3, and let $q < N/(e^2 t)$. Then

$$\textsc{StatDist}\,[\mathrm{Par}(n, m, q, t), \mathrm{Rnd}(q, t)] \;\leq\; d_1(t) \cdot \frac{q^2}{N^t} \;+\; \begin{cases} d_2(t, \lg N) \cdot \dfrac{q^3}{N^{3t/2}} & \text{if } t \text{ is even} \\[2ex] d_2(t, \lg N) \cdot \dfrac{q^4}{N^{2t}} & \text{if } t \text{ is odd .} \end{cases}$$

Proof: Follows from Proposition 3.2 and Lemma 3.3. ▌

Proof of Lemma 3.3. Finally we prove the main lemma.

Proof of Lemma 3.3: The case of $t = 1$ corresponds to the well-known birthday bound (i.e., we are interested in the probability that two rows have their single 1-entry in the same column). The proof thus focuses on (and assumes) $t \geq 2$. In the following, it is understood that the probabilities are over the choices of $R_1, \ldots, R_q$ uniformly and independently from $D(n, t)$.

$$\begin{aligned} \mathsf{NFRProb}(N, q, t) &= \sum_{i=2}^{q-1} \Pr[\,\mathrm{MTX}_{N,q}(R_1, \ldots, R_q) \text{ has rank } i\,] \\ &\leq \sum_{i=2}^{q-1} \sum_{1 \leq j_1 < \cdots < j_i \leq q} \Pr[\,\text{Rows } j_1, \ldots, j_i \text{ of } \mathrm{MTX}_{N,q}(R_1, \ldots, R_q) \text{ sum to zero}\,] \; . \end{aligned}$$

Let $p(N, i, t)$ denote the probability that a i-by-N matrix over Z_2, in which each row is a random N-string with exactly t ones, has row-sum zero. Since the probability above does not depend on which rows we consider we have

$$\mathsf{NFRProb}(N,q,t) \;\leq\; \sum_{i=2}^{q-1} \binom{q}{i} \cdot p(N,i,t)\ .$$

Notice that if t is odd then three rows of the matrix cannot sum to zero. So set $s=3$ if t is even and $s=4$ if t is odd. Then our bound becomes

$$\mathsf{NFRProb}(N,q,t) \;\leq\; \binom{q}{2} \cdot p(N,2,t) \;+\; \sum_{i=s}^{q-1} \binom{q}{i} \cdot p(N,i,t)\ . \tag{10}$$

Claim: For any $2 \leq i \leq q-1$ we have

$$p(N,i,t) \leq \begin{cases} \dfrac{2d_1(t)}{N^t} & \text{if } i=2 \\ \left(\dfrac{eti}{2N}\right)^{ti/2} & \text{if } i \geq 3\ . \end{cases}$$

Proof of Claim: Let R denote a matrix selected according to the above distribution. If $i=2$ then $p(N,2,t)$ is just the probability of a collision when two balls are thrown into $\binom{N}{t}$ buckets. This is

$$\frac{1}{\binom{N}{t}} \;=\; \frac{t!(N-t)!}{N!} \;=\; \frac{t!}{N(N-1)\cdots(N-t+1)} \;\leq\; \frac{t!}{(N-t+1)^t}\ .$$

By assumption $t \leq \sqrt{N/(e\lg N)}$ so we can lower bound the denominator by

$$\left(N-\sqrt{N}\right)^t \;=\; N^t \cdot \left(1-\frac{1}{\sqrt{N}}\right)^t \;\geq\; N^t \cdot \left(1-\frac{t}{\sqrt{N}}\right) \;\geq\; N^t \cdot \left(1-\frac{1}{(e\lg N)^{1/2}}\right)\ .$$

The lowest value of N meeting the conditions in the lemma statement is $N=9$ and hence the above is at most $0.659 \cdot N^t$. Putting all this together we get

$$p(N,2,t) \;\leq\; \frac{1.517 \cdot t!}{N^t} \;\leq\; 2d_1(t) \cdot N^{-t}$$

as desired.

Now consider $i \geq 3$. Each column in R having some 1-entry, must have at least 2 such entries. Thus, the probability that the rows of R sum to zero is upper bounded by the probability that R has 1-entries in at most $it/2$ columns. We can view the choice of a row as that of picking at random a subset of exactly t columns in which to place ones. Thus

$$p(N,i,t) \;\leq\; \binom{N}{ti/2} \cdot \left[\frac{\binom{ti/2}{t}}{\binom{N}{t}}\right]^i \;=\; \binom{N}{ti/2} \cdot \left[\frac{\prod_{j=0}^{t-1} \frac{ti}{2}-j}{\prod_{j=0}^{t-1} N-j}\right]^i\ .$$

Now use the fact that $a \leq b$ implies $(a-1)/(b-1) \leq a/b$. This can be applied since $ti/2 \leq N/2$, the latter being true because $i \leq q \leq N/(2e^2t)$. This bounds the above by

$$\binom{N}{ti/2} \cdot (ti/2N)^{ti} \;\leq\; \left(\frac{Ne}{ti/2}\right)^{ti/2} \cdot (ti/2N)^{ti}\ .$$

Simplifying the last term yields the claim. □

From Equation (10) and the Claim we get

$$\begin{aligned} \mathsf{NFRProb}(N,q,t) &\leq \binom{q}{2}\cdot p(N,2,t) + \sum_{i=s}^{q-1}\left(\frac{qe}{i}\right)^{i}\cdot\left(\frac{eti}{2N}\right)^{ti/2} \\ &= \binom{q}{2}\cdot\frac{2d_1(t)}{N^t} + \sum_{i=s}^{q-1}\left[eq\cdot\left(\frac{et}{2N}\right)^{t/2}\cdot i^{\frac{t}{2}}-1\right]^{i} . \qquad (11) \end{aligned}$$

The first term of Equation (11) is at most

$$\frac{q^2}{2}\cdot\frac{2d_1(t)}{N^t} = d_1(t)\cdot\frac{q^2}{N^t} . \qquad (12)$$

This yields the first term in the bound claimed in the lemma statement. Now we consider the sum

$$S = \sum_{i=s}^{q-1}\left[eq\cdot\left(\frac{et}{2N}\right)^{t/2}\cdot i^{\frac{t}{2}}-1\right]^{i}$$

and show it is bounded by the second term in the lemma statement.

Let α be a value to be determined. Then

$$\begin{aligned} S &= \sum_{i=s}^{\alpha\lg N}\left[eq\cdot\left(\frac{et}{2N}\right)^{t/2}\cdot i^{\frac{t}{2}-1}\right]^{i} + \sum_{i=1+\alpha\lg N}^{q}\left[eq\cdot\left(\frac{et}{2N}\right)^{t/2}\cdot i^{\frac{t}{2}-1}\right]^{i} \\ &\leq \sum_{i=s}^{\alpha\lg N}\left[eq\cdot\left(\frac{et}{2N}\right)^{t/2}\cdot(\alpha\lg N)^{\frac{t}{2}-1}\right]^{i} + \sum_{i=1+\alpha\lg N}^{q}\left[eq\cdot\left(\frac{et}{2N}\right)^{t/2}\cdot q^{\frac{t}{2}-1}\right]^{i} \\ &\leq \sum_{i=s}^{\alpha\lg N}\left[eq\cdot\left(\frac{et}{2N}\right)^{t/2}\cdot(\alpha\lg N)^{\frac{t}{2}}-1\right]^{i} + \sum_{i=1+\alpha\lg N}^{q}\left[e\cdot\left(\frac{etq}{2N}\right)^{t/2}\right]^{i} . \qquad (13) \end{aligned}$$

We will impose upper bounds on q that guarantee

$$A \stackrel{\text{def}}{=} eq\cdot\left(\frac{et}{2N}\right)^{t/2}\cdot(\alpha\lg N)^{\frac{t}{2}}-1 \leq \frac{1}{2} \text{ and } B \stackrel{\text{def}}{=} e\cdot\left(\frac{etq}{2N}\right)^{t/2} \leq \frac{1}{2} . \qquad (14)$$

In that case, each of the sums of Equation (13) is bounded by twice its first term, so we can bound the sum itself by

$$\begin{aligned} & 2\cdot\left[eq\cdot\left(\frac{et}{2N}\right)^{t/2}\cdot(\alpha\lg N)^{\frac{t}{2}}-1\right]^{s} + \left[e\cdot\left(\frac{etq}{2N}\right)^{t/2}\right]^{\alpha\lg N} \\ \leq & \left[2e^{\frac{st}{2}+s}(t/2)^{st/2}(\alpha\lg N)^{\frac{st}{2}-s}\right]\cdot\frac{q^s}{N^{st/2}} + 2^{-\alpha\lg N} . \end{aligned}$$

Now set $\alpha = 2t$. The second term is $N^{-\alpha} = N^{-2t}$ and hence we get

$$S \leq \left[3e^{\frac{st}{2}+s}t^{st-s}2^{-s}(\lg N)^{\frac{st}{2}-s}\right]\cdot\frac{q^s}{N^{st/2}} .$$

To complete the proof, put this together with Equation (12), plug in the appropriate value of $s = 3$ if t is even and $s = 4$ if t is odd, and simplify. This yields the bound in the lemma statement.

It remains to see what conditions on q, t are imposed by Equation (14). Recalling that $\alpha = t$, some calculations show that the conditions imposed by $A \leq 1/2$ and $B \leq 1/2$ are, respectively,

$$q \leq \frac{t\lg N}{e}\left(\frac{N}{et^2\lg N}\right)^{t/2} \quad \text{and} \quad q \leq \frac{N}{e^2 t} .$$

```
$\mathsf{ENCRX}_t[F]$: encryption procedure
Input: Key $f$, plaintext $M$
    Pick distinct, random points
        $r_1, \ldots, r_t \in \{0,1\}^n$
    Let $mk = f(r_1) \oplus f(r_2) \oplus \cdots \oplus f(r_t)$
    Let $mdM = mk \oplus M$
    Return $(r_1, \ldots, r_t, mdM)$
```

```
$\mathsf{ENCRX}_t[F]$: decryption procedure
Input: Key $f$, ciphertext $(r_1, \ldots, r_t, mdM)$
    Let $mk = f(r_1) \oplus f(r_2) \oplus \cdots \oplus f(r_t)$
    Let $M = mdM \oplus mk$
    Return $M$
```

Figure 4: $\mathsf{ENCRX}_t[F]$: Our encryption scheme: Here $M \in \{0,1\}^m$ is the plaintext and $f \in F$ is the key.

As long as $N \geq et^2 \lg N$, some more calculation shows that

$$\frac{N}{e^2 t} \;\leq\; \frac{t \lg N}{e} \left(\frac{N}{et^2 \lg N} \right)^{t/2} \quad .$$

To ensure $N \geq et^2 \lg N$ we have made the requirement $t \leq \sqrt{N/(e \lg N)}$. Now if $q \leq N/e^2 t$ then we are ensured $A, B \leq 1/2$. The proof is complete. ▌

3.3 Application to encryption

See Section 2 for definitions of encryption and PRF related terms used below. Let F be a family of functions with domain $\{0,1\}^n$ and range $\{0,1\}^m$. (In "practice" this will be a pseudorandom function family, but the important case in the analysis is when F is the set R of all functions with the given domain and range.) For simplicity we look at the problem of encrypting a message of m-bits. (The methods can be adapted to encrypt messages of longer and varying lengths.)

CTR mode encryption. A standard mode to encrypt an m-bit message M is to pick a value $r \in \{0,1\}^n$ and set the ciphertext to $(r, f(r) \oplus M)$. Here $f \in F$ is the (secret) key under which encryption and decryption are performed. The counter version sets r to a counter value that is incremented with each message encrypted. Denoting it by $\mathsf{StandardENC\text{-}Ctr}$, the insecurity is shown in [2] be be bounded as indicated below. For any number $q < N$ of m-bit messages queried in a chosen-plaintext attack, setting $N = 2^n$–

$$\mathbf{InSec}^{\mathrm{enc}}(\mathsf{StandardENC\text{-}Ctr}, qm, T) \;\leq\; 2 \cdot \mathbf{InSec}^{\mathrm{prf}}(F, q, T') + 2^{-m} \; . \tag{15}$$

Here $T' = T + O(q(n+m))$. When a stateless scheme is desired, the standard paradigm would pick r at random. A chosen-plaintext attack of q messages results in a collision in r values with probability $\Theta(q^2/N)$, and when this happens the encryption scheme is broken, in the sense that partial information about the plaintext is leaked. We wish to apply the parity construct to get better security, comparable or superior to that of the counter version.

Our scheme. The idea is that instead of picking one point r, the encryptor picks t distinct random points $r_1, \ldots, r_t$, and sets the ciphertext of M to $(r_1, \ldots, r_t, f(r_1) \oplus \cdots \oplus f(r_t) \oplus M)$, the setting being the same as above.

More precisely, we associate to F an encryption scheme $\mathsf{ENCRX}_t[F]$, parameterized by the integer $t \geq 1$. It consists of two algorithms, one to encrypt and the other to decrypt. These algorithms are described in Figure 5. The encryption algorithm takes as input a key f and a message $M \in \{0,1\}^m$, while the decryption algorithm takes the same key and a ciphertext. Here f is a random member of F. It is understood that f is accessible as an oracle. (When F is pseudorandom, a seed explicitly supplied to the algorithms names a particular function in the family and thus enables computation of the oracle. But the view of f as an oracle better suits the analysis.)

Connection to matrix rank. In the information theoretic case, the insecurity of our scheme can be upper bounded in terms of the probability that the matrix associated to the execution is not of full rank.

Lemma 3.6 Let R be the family of all functions with domain $\{0,1\}^n$ and range $\{0,1\}^m$, and let $N = 2^n$. Let $t \geq 1$ and let $\mathsf{ENCRX}_t[R]$ be the associated encryption scheme as defined above. Let $q \geq 0$. Then

$$\mathbf{InSec}^{\mathrm{enc}}(\mathsf{ENCRX}_t[R], qm) \leq \mathsf{NFRProb}(N, q, t) .$$

Proof Sketch: This can be proved by combining Lemma 3.1 with the analysis of [2]. ▌

Security of our scheme. We now turn to the security of a concrete instantiation of our scheme under some given pseudorandom function family F. The insecurity of our encryption scheme will be bounded in terms of the insecurity of F (as a prf family), and the bound on the not-full-rank-probability of the matrix we have computed above. For simplicity we view the bound of Lemma 3.3 as $O(t!) \cdot q^2/N^t$ which as we have seen is essentially true.

Theorem 3.7 Let F be a family of (pseudorandom) functions with domain $\{0,1\}^n$ and range $\{0,1\}^m$, and let $N = 2^n$. Let $t \geq 1$ and let $\mathsf{ENCRX}_t[F]$ be the associated encryption scheme as defined above. Assume $1 \leq q \leq N/(e^2 t)$. Then

$$\mathbf{InSec}^{\mathrm{enc}}(\mathsf{ENCRX}_t[F], qm, T) \ \leq \ d(t) \cdot \frac{q^2}{N^t} \ + \ 2 \cdot \mathbf{InSec}^{\mathrm{prf}}(F, tq, T') ,$$

where $T' = T + O(tq(n+m))$ and $d(t) = O(t!)$.

Proof Sketch: A standard "simulation" based argument for pseudorandom function based constructions shows that

$$\mathbf{InSec}^{\mathrm{enc}}(\mathsf{ENCRX}_t[F], qm, T) \ \leq \ \mathbf{InSec}^{\mathrm{enc}}(\mathsf{ENCRX}_t[R], qm) + 2 \cdot \mathbf{InSec}^{\mathrm{prf}}(F, tq, T') .$$

Now bound the first term by combining Lemma 3.6 and Lemma 3.3. ▌

4 Integrity of parity and application to MACs

4.1 Motivational discussion and matrix connection

When the parity construct is used in an application such as MAC where the adversary is active, further properties are required to ensure security. It turns out we need to consider the following. An adversary A sees an output $(R_1, b_1, \ldots, R_q, b_q)$ of experiment $\mathrm{Par}(n, m, q, t)$. Now A tries to create a non-colliding sequence $R_{q+1} = (r_{q+1,1}, \ldots, r_{q+1,t})$ and a value b_{q+1} such that $R_{q+1} \notin \{R_1, \ldots, R_q\}$ and $b_{q+1} = f(r_{q+1,1}) \oplus \cdots \oplus f(r_{q+1,t})$. Notice that this is easy for A to do if there is some subset S of the rows of $\mathrm{MTX}_{N,q}(R_1, \ldots, R_q)$ which sums up to a N-vector v of exactly t ones, because then A can define R_{q+1} via $v = \mathrm{ChVec}(R_{q+1})$ and then set b_{q+1} to $\oplus_i b_i$, the XOR being over all i such that $\mathrm{ChVec}(R_i)$ is a row in S. We will see that in fact this is the only condition under which A can do it. Thus we want to make sure no subset of rows S has this property. This will imply that if A creates some non-colliding sequence $R_{q+1} \notin \{R_1, \ldots, R_q\}$, then A's chance of predicting

$f(r_{q+1,1})\oplus\cdots\oplus f(r_{q+1,t})$ correctly is at most 2^{-m}. Based on this it will be possible to prove the security of our MAC scheme.

The problem can be formulated by extending the experiments $\mathrm{Par}(n,m,q,t)$ and $\mathrm{Rnd}(n,m,q)$ to consider an adversary as discussed above. However since we went through that approach before, we will not do it again. Rather we will skip to the essential step and lemma based on which we can directly prove the security of the applications. This lemma is again about the probability that $\mathrm{MTX}_{N,q}(R_1,\ldots,R_q)$ has certain properties.

We need to consider the probability that one may augment the given matrix $\mathrm{MTX}_{N,q}(R_1,\ldots,R_q)$ by a row with t 1-entries, different from all current rows, so as to result in a matrix of rank at most q. Actually, we will ask for a little more, to simplify the analysis.

We say a subset S of its rows sums is *bad* if it sums up to a N-vector v such that $v \notin S$ but v contains exactly t 1-entries. We say that $\mathrm{MTX}_{N,q}(R_1,\ldots,R_q)$ is *t-vulnerable* if one of the following is true: (1) It has two identical rows, or (2) some subset of its rows is bad. We let

$$\mathsf{VulProb}(N,q,t) \;=\; \Pr\left[\mathrm{MTX}_{N,q}(R_1,\ldots,R_q) \text{ is } t\text{-vulnerable} \;:\; R_1,\ldots,R_q \overset{R}{\leftarrow} D(n,t)\right].$$

The following lemma considers an arbitrary adversary that given an output of experiment $\mathrm{Par}(n,m,q,t)$ attempts to create a new R_{q+1} and the corresponding f value. It says that A has no better strategy than to guess, as long as the matrix is not t-vulnerable.

Lemma 4.1 Fix any adversary A that on any input $(R_1,b_1,\ldots,R_q,b_q) \in D(n,t)\times\{0,1\}^m\times\cdots\times D(n,t)\times\{0,1\}^m$ outputs some $R_{q+1}=(r_{q+1,1},\ldots,r_{q+1,t}) \in D(n,t)-\{R_1,\ldots,R_q\}$ and a string $b_{q+1}\in\{0,1\}^m$. In experiment $\mathrm{Par}(n,m,q,t)$, conditioned on the event that $\mathrm{MTX}_{N,q}(R_1,\ldots,R_q)$ is not t-vulnerable, the probability that $b_{q+1}=f(r_{q+1,1})\oplus\cdots\oplus f(r_{q+1,t})$ is at most 2^{-m}.

Motivated by this we proceed to bound $\mathsf{VulProb}(N,q,t)$ (the proof of next lemma is omitted – see [3]).

4.2 Main lemma: Bound on $\mathsf{VulProb}(N,q,t)$

Notice that a bad subset of the rows must have cardinality greater than 1. Also notice that if $\mathrm{MTX}_{N,q}(R_1,\ldots,R_q)$ is not t-vulnerable then it has full rank. (The first condition above means that no two rows sum to zero. If a subset T of three or more rows sums to zero, removing one row from T leaves a bad subset of rows, which the second condition disallows.)

Lemma 4.2 Let t be such that $1\le t\le\sqrt{N/(2e\lg N)}$, then for any $q<N/(2e^2t)$ we have

$$\mathsf{VulProb}(N,q,t) \;\le\; \begin{cases} d'(t,\lg N)\cdot\dfrac{q^2}{N^{t/2}} & \text{if } t \text{ is even} \\[2ex] d'(t,\lg N)\cdot\dfrac{q^3}{N^t} & \text{if } t \text{ is odd}\,, \end{cases} \tag{16}$$

where

$$d'(t,n) \;=\; \begin{cases} e^{2+3t/2}2^{3t/2}3^{-t/2}t^{-2+3t/2}n^{t-2} & \text{if } t \text{ is even} \\ e^{3+2t}2^{-3}t^{-3+5t/2}n^{t-2} & \text{if } t \text{ is odd.} \end{cases} \tag{17}$$

Notice the difference in the bounds for odd versus even t. This phenomenon is explained later.We will focus on odd t. In comparison with Lemma 3.3 the main term in the bound, namely q^3/N^t, has an extra factor of q. Other than that things are pretty similar. To get an idea of the relative values of the various terms, consider $N = 2^{64}$ and $t = 3$. Then the lemma says that for $q \leq N/46$ we have $\mathsf{VulProb}(N, q, 3) \leq 2^{24} \cdot q^3/N^3$.

Proof of Lemma 4.2: Consider separately the two conditions, namely that (1) the matrix has two identical rows, or (2) the matrix has a bad subset of rows. The probability of the first is easily bounded by

$$\begin{aligned} \frac{q^2}{2 \cdot \binom{N}{t}} &= q^2 \cdot \frac{t!}{2N(N-1)\cdots(N-t+1)} \\ &\leq q^2 \cdot \frac{t^{t-1}}{2(N-t+1)^t} \\ &\leq 2^{t-1}t^{t-1} \cdot \frac{q^2}{N^t} \ , \end{aligned} \tag{18}$$

the last bound being due to the fact that $q < N/2$ by assumption.

We proceed to bound the probability that there is some bad subset. Notice that if t is odd then a subset of two rows cannot sum to a N-string of exactly t ones, so the smallest possible bad subset has size 3. To capture this difference between odd and even values of t we let $s = 2$ if t is even and $s = 3$ if t is odd. Then we can bound the probability that $\mathrm{MTX}_{N,q}(R_1, \ldots, R_q)$ has a bad subset of rows by

$$\begin{aligned} &\sum_{i=s}^{q} \sum_{1 \leq j_1 < \cdots < j_i \leq q} \Pr[\,\text{The subset } \{j_1, \ldots, j_i\} \text{ of the rows of } \mathrm{MTX}_{N,q}(R_1, \ldots, R_q) \text{ is bad}\,] \\ \leq\ &\sum_{i=s}^{q} \binom{q}{i} \cdot p'(N, i, t) \ , \end{aligned} \tag{19}$$

where $p'(N, i, t)$ the probability that a i-by-N matrix over Z_2, in which each row is a random N-string with exactly t ones, has row-sum which is a N-string containing exactly t ones.

Claim: For any $s \leq i \leq q$ we have

$$p'(N, i, t) \ < \ \left[\frac{2eN}{t(s+1)}\right]^{t/2} \cdot \left(\frac{eti}{N}\right)^{ti/2} \ .$$

Proof of Claim: Let R denote a matrix selected according to the above distribution. For the row-sum of R to be a N-string having exactly t ones it must be that all 1-entries in R must lie in a subset of at most $t + \frac{ti-t}{2} = \frac{ti+t}{2}$ columns. So

$$p'(N, i, t) \ \leq \ \binom{N}{(ti+t)/2} \cdot \left[\frac{\binom{(ti+t)/2}{t}}{\binom{N}{t}}\right]^i \ .$$

We proceed to bound this as in the proof of the corresponding Claim in the proof of Lemma 3.3. We get

$$\begin{aligned} p'(N, i, t) \ &\leq \ \left[\frac{Ne}{(ti+t)/2}\right]^{(ti+t)/2} \cdot \left(\frac{ti+t}{2N}\right)^{ti} \\ &\leq \ \left[\frac{2eN}{ti+t}\right]^{t/2} \cdot \left[\frac{Ne}{(ti+t)/2}\right]^{ti/2} \cdot \left(\frac{ti+t}{2N}\right)^{ti} \\ &\leq \ \left[\frac{2eN}{ts+t}\right]^{t/2} \cdot \left[\frac{e(ti+t)}{2N}\right]^{ti/2} \ . \end{aligned}$$

The claim follows since $i + 1 < 2i$. □

From Equation (19) and the Claim we can bound the probability that $\mathrm{MTX}_{N,q}(R_1, \ldots, R_q)$ contains some bad subset of rows by

$$\left[\frac{2eN}{t(s+1)}\right]^{t/2} \cdot \sum_{i=s}^{q} \left[\frac{qe}{i} \cdot \left(\frac{eti}{N}\right)^{t/2}\right]^{i} \;=\; \left[\frac{2eN}{t(s+1)}\right]^{t/2} \cdot \sum_{i=s}^{q} \left[eq \cdot \left(\frac{et}{N}\right)^{t/2} \cdot i^{\frac{t}{2}-1}\right]^{i} . \qquad (20)$$

The sum is the same as the one in the proof of Lemma 3.3 except that it starts at $i = s$ rather than $i = 2$, and the denominator contains N rather than $2N$. We can thus bound it the same way. Briefly, in breaking up the sum into two parts we choose this time $\alpha = 2t$. Set

$$A \overset{\mathrm{def}}{=} eq \cdot \left(\frac{et}{N}\right)^{t/2} \cdot (2t \lg N)^{\frac{t}{2}} - 1 \qquad \text{and} \qquad B \overset{\mathrm{def}}{=} e \cdot \left(\frac{etq}{N}\right)^{t/2} . \qquad (21)$$

We will ensure $A, B \leq 1/2$. Then the quantity of Equation (20) is bounded by

$$\left[\frac{2eN}{t(s+1)}\right]^{t/2} \cdot \sum_{i=s}^{2t \lg N} A^i \;+\; \left[\frac{2eN}{t(s+1)}\right]^{t/2} \cdot \sum_{i=1+2t\lg N}^{q} B^i \;\leq\; \left[\frac{2eN}{t(s+1)}\right]^{t/2} \cdot \left[2A^s + B^{2t \lg N}\right] .$$

We have $B^{2t \lg N} \leq N^{-2t}$ because $B \leq 1/2$. Plugging in the value of A and simplifying gives a bound of

$$\begin{aligned} & \left[\frac{2eN}{t(s+1)}\right]^{t/2} \cdot 2 \cdot \left[eq \cdot \left(\frac{et}{N}\right)^{t/2} \cdot (2t \lg N)^{\frac{t}{2}} - 1\right]^{s} \;+\; \left[\frac{2e}{t(s+1)}\right]^{t/2} \cdot N^{-3t/2} \\ \leq\;\; & 2 \cdot \left[2^{\frac{t}{2}+\frac{st}{2}-s} e^{\frac{t}{2}+\frac{st}{2}+s} t^{st-\frac{t}{2}-s} (s+1)^{-\frac{t}{2}} (\lg N)^{\frac{st}{2}-s}\right] \cdot \frac{q^s}{N^{t(s-1)/2}} \;+\; N^{-3t/2} \end{aligned}$$

We need to add in the term of Equation (18), and recall that $s = 2$ if t is even and $s = 3$ if t is odd. The bound in the lemma statement then follows.

To complete the proof we need to determine the conditions imposed on q, t by the requirements $A, B \leq 1/2$. The calculations are similar to those in the proof of Lemma 3.3. Briefly under the condition $N \geq 2et^2 \lg N$ it suffices that $q \leq N/(2e^2t)$. These conditions are imposed by the upper bounds on t and q, respectively, in the lemma statement. ▌

Tightness of the above bound. Suppose that $q < N$ (which is required and assumed anyhow). Consider, first, an even t. Then the probability that a q-by-N matrix is t-vulnerable is lower bounded by $\Omega(q^2)$ times the probability that two t-vectors add-up to another t-vector. The probability for this event is computed by first selecting and fixing the first vector, and next computing probability that the second vector agrees with it on exactly $t/2$ 1-entries. The latter probability is $\Theta((t/N)^{t/2})$.

For odd t, we consider the event that three distinct t-vectors add up to a different t-vector. Fix any random non-overlapping choice for the first two t-vectors, and consider the probability that the third resides fully in these $2t$ columns (but does not equal any of the first two vectors). The latter probability is $\Theta((2t/N)^t)$. Considering all $\binom{q}{3}$ choices of the rows, the claim follows.

Is it odd that odd $t \geq 3$ is better than even $t + 1$? Considering small t's and ignoring logarithmic factors, for odd $t \geq 3$ the upper bound is q^3/N^t which is typically much smaller than the (tight!) upper bound $q^2/N^{(t+1)/2}$ provided for the even t+1. We note that a similar phenomenon occurs for large t's: Consider an even $t = N/2$. Then the probability that two random rows with t 1-entries sum-up to a row with t 1-entries is $\Theta(1/\sqrt{N})$. On the other hand, for odd t this event

```
$\mathsf{MACRX}_t[F]$: mac generation
INPUT: Key $\langle h, f\rangle$, message $M$
    Pick distinct, random points
        $r_1, \ldots, r_t \in \{0,1\}^n$
    Let $mk = f(r_1) \oplus f(r_2) \oplus \cdots \oplus f(r_t)$
    Let $mhM = mk \oplus h(M)$
    Return $(r_1, \ldots, r_t, mhM)$
```

```
$\mathsf{MACRX}_t[F]$: mac verification
INPUT: Key $\langle h, f\rangle$, message $M$, candidate mac $\sigma$
    Check that $\sigma$ has form $(r_1, \ldots, r_t, \mu)$ for
        $t$ distinct strings $r_1, \ldots, r_t \in \{0,1\}^n$
        and some $\mu \in \{0,1\}^m$
    Let $mk = f(r_1) \oplus f(r_2) \oplus \cdots \oplus f(r_t)$
    Let $mhM = mk \oplus h(M)$
    If $mhM = \mu$ then return 1 else return 0
```

Figure 5: $\mathsf{MACRX}_t[F]$: Our message authentication scheme: Here $M \in D$ is the text to be authenticated and $\langle h, f\rangle \in H \times F$ is the key.

never happens. In general, the discrepancy is due to the fact that for even t, one should consider the contribution of pairs of rows; whereas for odd t only larger subsets are relevant.

4.3 Application to message authentication

Given. Let D be some domain consisting of messages we want to authenticate. (For example D could be $\{0,1\}^*$, or all strings of length up to some maximum length.) We fix a family H of ϵ-AXU hash functions in which each function $h \in H$ maps from D to $\{0,1\}^n$. We also let F be a family of functions with domain $\{0,1\}^n$ and range $\{0,1\}^m$. (In "practice" this will be a pseudorandom function family, but the important case in the analysis is when F is the set R of all functions with the given domain and range.)

Universal hash based MACs. The standard paradigm is that to authenticate message $M \in D$, pick a value $r \in \{0,1\}^n$ and set the mac to $(r, f(r) \oplus h(M))$. Here $\langle h, f\rangle$ is the (secret) key under which macs are created and verified, where $h \in H$ and $f \in F$. The counter version sets r to a counter value that is incremented with each message authenticated. Denoting it by StandardMAC-Ctr,

$$\mathbf{InSec}^{\mathrm{mac}}(\mathsf{StandardMAC\text{-}Ctr}, q_a, q_v, T) \quad \leq \quad q_v \epsilon + \mathbf{InSec}^{\mathrm{prf}}(F, q_a + q_v, T') \ .$$

where $q_a < N$, $q_v \geq 1$, $N = 2^n$ and $T' = T + O((q_a + q_v)(n+m))$. When a stateless scheme is desired, the standard paradigm would pick r at random. A chosen-message attack of q messages results in a collision in r values with probability $\Theta(q^2/N)$, and when this happens forgery is possible. We wish to apply the parity construct to get better security, comparable or superior to that of the counter version.

Our scheme. The idea is that instead of picking one point r, the generator of the mac picks t distinct random points $r_1, \ldots, r_t$, and sets the mac of M to $(r_1, \ldots, r_t, f(r_1) \oplus \cdots \oplus f(r_t) \oplus h(M))$, the setting being the same as above.

More precisely, with H fixed we associate to F a message authentication scheme $\mathsf{MACRX}_t[F]$, parameterized by the integer $t \geq 1$. It consists of two algorithms, one to generate macs, and the other to verify candidate macs. (The distinction is necessary since the mac generation algorithm is probabilistic.) These algorithms are described in Figure 5. The mac generation algorithm takes as input a key $\langle h, f\rangle$ and a message $M \in D$, while the verification algorithm takes the same key, a message, and a candidate mac for it. Here h is a random hash function from H while f is a random member of F. It is understood that f is accessible as an oracle. (When F is pseudorandom, a seed explicitly supplied to the algorithms names a particular function in the family and thus enables computation of the oracle. But the view of f as an oracle better suits the analysis.)

We stress one aspect of the verification procedure, namely to check that the candidate tag really contains t points (not more or less) and that these are distinct. Without this check, forgery is possible.

Connection to matrix vulnerability. In the information theoretic case, the insecurity of our scheme can be upper bounded in terms of the quality of H as an AXU family (namely ϵ), the vulnerability of the matrix associated to the number of authentication queries involved, and a term corresponding to guessing a correct mac in the number of verification queries involved.

Lemma 4.3 Let H be a family of ϵ-AXU hash functions with range $\{0,1\}^n$. Let R be the family of all functions with domain $\{0,1\}^n$ and range $\{0,1\}^m$. Let $t \geq 1$ and let $\mathsf{MACRX}_t[R]$ be the associated MAC as defined above. Let $q_a, q_v \geq 1$. Then

$$\mathbf{InSec}^{\mathrm{mac}}(\mathsf{MACRX}_t[R], q_a, q_v) \leq q_v\epsilon + \mathsf{VulProb}(2^n, q_a, t) \ .$$

Proof Sketch: This uses the same ideas as the standard connection between universal hashing and MACs so we only indicate briefly the source of the various terms in the bound. Consider an adversary A making q_a authentication queries and q_v verification queries. First assume for simplicity the verification queries are all made after all the authentication queries are complete. The authentication queries give rise to a matrix $\mathrm{MTX}_{N,q}(R_1, \ldots, R_q)$. If this matrix is t-vulnerable, we give up, accounting for this term in the bound. So assume not. As observed above, non-t-vulnerable implies full rank so by Lemma 3.1 the adversary is getting no information about h via the authentication queries. By Lemma 4.1 it is also getting no information about the XOR of f on the entries in some new non-colliding t-sequence. Thus its chance of forgery is limited by $q_v\epsilon$. Finally, one must deal with the verification queries. If no information has been released, guessing is the only possible strategy. ▌

Security of our scheme. We now turn to the security of a concrete instantiation of our scheme under some given pseudorandom function family F. The insecurity of our message authentication scheme will be bounded in terms of the insecurity of F (as a prf family), the quality of the AXU family H, the bound on matrix vulnerability we have computed above, and the guessing term mentioned just above. We focus on the case of odd t because our bounds are better here.

Theorem 4.4 Let H be a family of ϵ-AXU hash functions with domain D and range $\{0,1\}^n$. Let F be a family of (pseudorandom) functions with domain $\{0,1\}^n$ and range $\{0,1\}^m$. Let $N = 2^n$ and assume t is an odd integer satisfying $1 \leq t \leq \sqrt{N/(2e\lg N)}$. Let $\mathsf{MACRX}_t[F]$ be the associated MAC as defined above. Assume $1 \leq q_a \leq N/(2e^2t)$ and $q_v \geq 1$. Then

$$\begin{aligned}&\mathbf{InSec}^{\mathrm{mac}}(\mathsf{MACRX}_t[F], q_a, q_v, T) \leq \\ &\qquad q_v\epsilon + d'(t,n)\cdot\frac{q_a^3}{N^t} + \mathbf{InSec}^{\mathrm{prf}}(F, t(q_a+q_v), T') \ ,\end{aligned}$$

where $T' = T + O(t(q_a+q_v)(n+m))$ and $d'(t,n)$ is as in Equation (17).

Proof Sketch: A standard "simulation" based argument for pseudorandom function based constructions shows that

$$\mathbf{InSec}^{\mathrm{mac}}(\mathsf{MACRX}_t[F], q_a, q_v, T) \ \leq \ \mathbf{InSec}^{\mathrm{mac}}(\mathsf{MACRX}_t[R], q_a, q_v) + \mathbf{InSec}^{\mathrm{prf}}(F, t(q_a+q_v), T') \ .$$

Now bound the first term by combining Lemma 4.3 and Lemma 4.2. ▌

Thus, $\mathsf{MACRX}_3[F]$ offers better security than $\mathsf{MACRX}_1[F]$, and for $q_a < 2^{2n/3}$ its security is comparable to the counter-version as given in Equation (22). $\mathsf{MACRX}_5[F]$ is comparable in security to the counter-version.

Stronger bounds can be obtained if we make stronger assumptions about the family from which h is selected. For example, if we may assume that an adversary cannot compute $\bigoplus_{M \in S} h(M)$, for a set S of messages of its choice then it suffices to consider passive attacks on F (as done in the previous sections), and the bound given in Equation (2) applies. Thus, under such assumptions $\mathsf{MACRX}_3[F]$ is comparable in security to the counter-version.

On the insufficiency of the full rank condition in the MAC setting. We show that for ensuring the security against forgery of the MAC scheme, presented in Section 4.3 and in Theorem 4.4, it is not enough to require that the matrix $\mathrm{MTX}_{N,q}(R_1, \ldots, R_q)$ be full rank. The additional requirement of not being t-vulnerable is necessary too. That is, we show an attack against the MAC scheme that succeeds if the attacker can obtain a linear combination of the rows that result in a vector over Z_2^N with exactly t non-zero entries. For this we assume that the universal hash functions in use are additive over Z_2 (this is a common property of many universal hash families [17, 10]). The attack proceeds as follows. The attacker asks to see the MAC value of $q < 2^n/(2e^2t)$ random messages $M_1, \ldots, M_q$, say each uniformly chosen in $\{0,1\}^{2n}$. That is, for each such message M_i, the attacker sees the value $\sigma_i = h(M_i) \oplus \bigoplus_{l=1}^{t} r_{i,l}$ together with the t random values $r_{i,1}, \ldots, r_{i,t}$ (note that the later values represent the positions with non-zero entries in the i-th row of the matrix). Now suppose that the XOR of rows $i_1, ..., i_k$ in M result in a vector (different from all these rows) with exactly t ones in positions $r_1, \ldots, r_t$. The attacker computes a message $M = \bigoplus_{j=1}^{k} M_{i_j}$, and outputs as its forgery the tag value $(r_1, \ldots, r_t, \bigoplus_{j=1}^{k} \sigma_{i_j})$. It is easy to verify that

$$\bigoplus_{j=1}^{k} \sigma_{i_j} = \bigoplus_{j=1}^{k} h(M_{i_j}) \oplus \bigoplus_{j=1}^{k} \bigoplus_{l=1}^{t} r_{i_j,l} = h\Big(\bigoplus_{j=1}^{k} M_{i_j}\Big) \oplus \bigoplus_{l=1}^{t} r_l = h(M) \oplus \bigoplus_{l=1}^{t} r_l \, .$$

Since we assumed the M_i were random messages (in $\{0,1\}^{2n}$) then with very high probability M is different than all previously queried messages (as the M_i's are linearly independent), and the forgery is successful.

Acknowledgments

We thank R. Venkatesan and B. Yee for helpful discussions on the subject of the uses and drawbacks of counters in computer systems.

References

[1] W. Aiello, and R. Venkatesan. Foiling birthday attacks in length-doubling transformations. *Advances in Cryptology – EUROCRYPT '96*, Lecture Notes in Computer Science Vol. 1070, U. Maurer ed., Springer-Verlag, 1996.

[2] M. Bellare, A. Desai, E. Jokipii and P. Rogaway. A concrete security treatment of symmetric encryption: Analysis of the DES modes of operation. *Proceedings of the 38th Symposium on Foundations of Computer Science*, IEEE, 1997.

[3] M. Bellare, O. Goldreich and h. Krawczyk. Stateless evaluation of pseudorandom functions: Security beyond the birthday barrier. Extended abstract of this paper, *Advances in Cryptology – CRYPTO '99*, Lecture Notes in Computer Science Vol. 1666, M. Wiener ed., Springer-Verlag, 1999.

[4] M. Bellare, R. Guérin and P. Rogaway. XOR MACs: New Methods for Message Authentication using Finite Pseudorandom Functions. Full version available via `http://www-cse.ucsd.edu/users/`

mihir. Preliminary version in *Advances in Cryptology – CRYPTO '95*, Lecture Notes in Computer Science Vol. 963, D. Coppersmith ed., Springer-Verlag, 1995.

[5] M. Bellare and R. Impagliazzo. A tool for obtaining tighter security analyses of pseudorandom function based constructions, with applications to PRP→PRF conversion. Manuscript, February 1999.

[6] M. Bellare, J. Kilian and P. Rogaway. The Security of Cipher Block Chaining. *Advances in Cryptology – CRYPTO '94*, Lecture Notes in Computer Science Vol. 839, Y. Desmedt ed., Springer-Verlag, 1994.

[7] M. Bellare, T. Krovetz and P. Rogaway. Luby-Rackoff backwards: Increasing security by making block ciphers non-invertible. *Advances in Cryptology – EUROCRYPT '97*, Lecture Notes in Computer Science Vol. 1233, W. Fumy ed., Springer-Verlag, 1997.

[8] O. Goldreich, S. Goldwasser and S. Micali. How to construct random functions. *Journal of the ACM*, Vol. 33, No. 4, 1986, pp. 210–217.

[9] C. Hall, D. Wagner, J. Kelsey and B. Schneier. Building PRFs from PRPs. *Advances in Cryptology – CRYPTO '98*, Lecture Notes in Computer Science Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.

[10] H. Krawczyk. LFSR-based Hashing and Authentication. *Advances in Cryptology – CRYPTO '94*, Lecture Notes in Computer Science Vol. 839, Y. Desmedt ed., Springer-Verlag, 1994.

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

[12] M. Naor and O. Reingold. On the construction of pseudorandom permutations: Luby-Rackoff revisited. *J. of Cryptology* Vol. 12, No. 1, 1999, pp. 29–66.

[13] J. Patarin. Improved security bounds for pseudorandom permutations. *Proceedings of the 5th Annual Conference on Computer and Communications Security*, ACM, 1997.

[14] J. Patarin. About Feistel schemes with six (or more) rounds. *Proceedings of the 5th Fast Software Encryption Workshop*, Lecture Notes in Computer Science Vol. 1372, Springer-Verlag, 1998.

[15] B. Preneel and P. van Oorschott. MDx-MAC and building fast MACs from hash functions. *Advances in Cryptology – CRYPTO '95*, Lecture Notes in Computer Science Vol. 963, D. Coppersmith ed., Springer-Verlag, 1995.

[16] V. Shoup. On Fast and Provably Secure Message Authentication Based on Universal Hashing. *Advances in Cryptology – CRYPTO '96*, Lecture Notes in Computer Science Vol. 1109, N. Koblitz ed., Springer-Verlag, 1996.

[17] M. Wegman and L. Carter. New hash functions and their use in authentication and set equality. *J. of Computer and System Sciences,* vol. 22, 1981, pp. 265-279.

A Appendix: An abstract generalization

Underlying our applications is a basic information theoretic question: What is the best possible one-time pad based encryption scheme that is stateless? That is, suppose that two parties share a one-time pad which is a random string *pad* of length $N = 2^n$. (Such a pad corresponds to a random function $f : \{0,1\}^n \mapsto \{0,1\}$ considred in the main text.) Let us say the sender wants to encrypt a sequence of messages, each one bit long. The encryption must be stateless. With loss of little generality, we may consider schemes in which some function g: $\{0,1\}^N \to \{0,1\}$ is randomly selected and the ciphertext is obtained by "masking" (i.e., XORing) a plaintext bit with $g(pad)$. That is, the sender sends $(\langle g \rangle, g(pad) \oplus m)$, where $\langle g \rangle$ is a description of g, as the encryption of message bit m. Each time, the function g is selected according to some fixed probability distribution over $\text{Maps}(\{0,1\}^N, \{0,1\})$, the set of all functions mapping $\{0,1\}^N$ to $\{0,1\}$. No history dependence is allowed: each time the sender picks a new function according to the prescribed process. This captures what seems a most general possible notion of a stateless scheme.

The question is what methods of generating g lead to the least loss in information about the message sequence as a function of the number of messages encrypted. To formalize this, we need not talk about the message; we just consider the process of generating the function g.

Definition A.1 A stateless encryption scheme is specified by a *stateless function specifier*. This is a probabilistic algorithm FnSp that takes input N and outputs a function $g \in \mathrm{Maps}(\{0,1\}^N, \{0,1\})$.

Notice that FnSp is probabilistic. We write $\mathsf{FnSp}(N; R)$ to be its output on input N and coins R. We now consider two distributions:

```
Distribution $\mathsf{FnSp}_{N,q}$
    Pick a random $N$-bit string $pad$
    For $i = 1, \ldots, q$ do
        Pick $R_i$ at random
        Let $g_i \overset{R}{\leftarrow} \mathsf{FnSp}(N, R_i)$
        Let $b_i \leftarrow g_i(pad)$
    End For
Output $(\langle g_1 \rangle, b_1), \ldots, (\langle g_q \rangle, b_q)$
```

```
Distribution $\mathsf{Rand}_{N,q}$
    For $i = 1, \ldots, q$ do
        Pick $R_i$ at random
        Let $g_i \overset{R}{\leftarrow} \mathsf{FnSp}(N, R_i)$
        Pick a bit $b_i$ at random
    End For
Output $(\langle g_1 \rangle, b_1), \ldots, (\langle g_q \rangle, b_q)$
```

In each distribution, we output a sequence of pairs. The first component of each pair is the output function of the function specifier. The second component is either the application of this function to the pad, or an independently selected random bit. We are interested in the statistical distance between these distributions.

Definition A.2 The $\mathsf{discrepency}$ of a function specifier, FnSp, is defined as the statistical difference between the above distributions, as a function of N and q; that is,

$$\mathbf{Dist}(\mathsf{FnSp}, N, q) \;=\; \textsc{StatDist}\left[\mathsf{FnSp}_{N,q}, \mathsf{Rand}_{N,q}\right]$$

We are interested in the growth rate of this function, as a function of q, for fixed N. Furthermore, we want to construct efficient function specifiers for which this value is as low as possible.

A.1 Lower bounds

To guide our study we consider some lower bounds. We start by observing that the discrepancy cannot be zero as soon as we output more than one bit. This is in contrast to the stateful case, where the distance may remain zero upto N bits.

Proposition A.3 $\mathbf{Dist}(\mathsf{FnSp}, N, q) > 0$, for any function specifier FnSp and any $q \geq 2$.

Proof: There is a non-zero probability that the random strings R_1, R_2 chosen in the first two tries are equal. In this case, in the first experiment, we get back the same bit both times. In the random experiment, we get back independent random bits both times. ▌

On the other hand, it is clear that one cannot out-perform the stateful schemes. That is, Shannon's bounds continue to hold here.

Proposition A.4 $\mathbf{Dist}(\mathsf{FnSp}, N, N+i) \geq 1 - 2^{-i}$, for any function specifier FnSp and any $i \geq 1$.

Proof: Fixing any sequence $g_1, ..., g_{N+i}$, we consider the residual distribution of the bits $b_1, ..., b_{N+i}$ in $\mathsf{FnSp}_{N,q}$ and $\mathsf{Rand}_{N,q}$, denoted X and Y, respectively. Clearly, Y is uniform over $\{0,1\}^{N+i}$, whereas X has a support of size at most 2^N (as it is obtained by performing a fixed mapping on the uniform distribution over $\{0,1\}^N$). Thus, the statistical difference between X and Y is $1 - \frac{\text{support}(X)}{2^{N+i}} \geq 1 - 2^{-i}$. ▌

A.2 Upper bounds

We are interested in a particular class of function specifiers, namely those where the output function g is the XOR of some subset of bits in its argument. Let $\text{Lin}_N \subseteq \text{Maps}(\{0,1\}^N, \{0,1\})$ be the set of all these maps; such a generic map, denoted χ_S, is specified by $S \subseteq [N] \stackrel{\text{def}}{=} \{1, ..., N\}$, and is defined by $\chi_S(pad) = \oplus_{i \in S} pad_i$, where $pad = pad_1 \cdots pad_N$. (Lin_N is also called the set of parity functions.)

Definition A.5 An function specifier is called a *parity function specifier* if it always outputs functions g in the set Lin_N.

We can analyze parity function specifiers using the connection to matrix rank as in [4]. We associate to any such function specifier FnSp a random variable $\text{Mtx}(\mathsf{FnSp}, N, q)$ (defined below). This random variable takes as value a q-by-N matrix of zeros and ones. To define it, think of χ_S as an N-element vector with a 1 in position j iff $j \in S$.

```
Random Variable Mtx(FnSp, $N$, $q$)
    Initialize $M$ to an empty matrix
    For $i = 1, \ldots, q$ do
        Pick $R_i$ at random and let $\chi_{S_i} \stackrel{R}{\leftarrow} \mathsf{FnSp}(N; R_i)$
        Make $\chi_{S_i}$ the $i$-th row of matrix $M$
    End For
Output $M$
```

This is a random variable over the choices of $R_1, \ldots, R_q$ made.

Proposition A.6 [Following [4]] For any parity function specifier FnSp it is the case that**Dist**(FnSp, N, q) is bounded above by the probability that$\text{Mtx}(\mathsf{FnSp}, N, q)$ is not of full rank.

Consider the following simple parity function specifier, called the *full parity function specifier.* It simply selects a subset $S \subseteq \{0,1\}^N$ at random:

```
Full Parity Function Specifier FuParSp($N$)
    Select $S \subseteq \{0,1\}^N$ at random
Output $\chi_S$
```

It is useful to visualize the set S being chosen item by item. That is, for every $j = 1, ..., N$, put j in S with probability one-half. Note that the Full Parity Function Specifier is very bad in terms of integrity (i.e., for the application to MACs): Given two invocations of it, $\chi_{S_1} \leftarrow \mathsf{FuParSp}(N)$ and $\chi_{S_2} \leftarrow \mathsf{FuParSp}(N)$, and the values $\chi_{S_1}(pad)$ and $\chi_{S_2}(pad)$, an active adversary may set S to be the symmetric difference of S_1 and S_2, and predict the value $\chi_S(pad)$ $(= \chi_{S_1}(pad) \oplus \chi_{S_2}(pad))$. However, the Full Parity Function Specifier fares very well with respect to a passive attack (i.e., the pseudorandomness property equivalent to discrepancy):

Proposition A.7 For any N and $q \geq 1$ the discrepancy of the full parity function specifier is upper bounded as follows:

$$\mathbf{Dist}(\mathsf{FuParSp}, N, q) \; < \; \frac{2^q}{2^N} \; .$$

Proof: By the above proposition, the statistical distance between $\mathsf{FnSp}_{N,q}$ and $\mathsf{Rand}_{N,q}$ is bounded above by the probability that the associated matrix is not of full rank. This probability is easily computed: Having chosen $i-1$ boolean vectors, what is the probability that an i-th, randomly chosen one, is a linear combination of one of the previous ones? There are 2^{i-1} linear combinations of the given $i-1$ vectors, and 2^N choices for the i-th one, so this probability is $2^{i-1}/2^N$. This is true for each choice for $i = 1, \ldots, q$, so we get

$$\mathbf{Dist}(\mathsf{FuParSp}, N, q) \; \leq \; \sum_{i=1}^{q} \frac{2^{i-1}}{2^N} \; < \; \frac{2^q}{2^N} \; ,$$

as desired. ▮

As explained in the introduction, the full parity function specifier is useless in applications (such as ours) where we need to operate in poly(n)-time, where $n = \lg N$. Recall that in practice the N-bit random pad, *pad*, will be defined by a succinct (pseudorandom) function $f : \{0,1\}^n \to \{0,1\}$. This leads to our main results which refer to partity function specifiers for which the subset of XORed bits is small.

Definition A.8 An function specifier is called a *t-parity function specifier* if it always outputs functions $\chi_S \in \mathrm{Lin}_N$ so that $|S| = t$. The *t-uniform parity function specifier* selects uniformly a t-subset, S, and outputs χ_S.

Our main result is

Theorem A.9 For any $t \geq 1$, the t-uniform parity function specifier has discrepancy at most $d_1(t) \cdot \frac{q^2}{N^t}$.

This result follows immediately by combining Proposition A.6 and Lemma 3.3.