Stateless evaluation of pseudorandom functions: Security beyond the birthday barrier

Mihir Bellare[1], Oded Goldreich[2], and Hugo Krawczyk[3]

[1] 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.

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

[3] Department of Electrical Engineering, Technion, Haifa 32000, Israel, and IBM T.J. Watson Research Center, New York, USA. Email: hugo@ee.technion.ac.il.

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.

1 Introduction

Pseudorandom functions [7] 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.

Michael Wiener (Ed.): CRYPTO'99, LNCS 1666, pp. 270-287, 1999.

Consider as an example the following method of encryption. Two parties share a key which specifies a function f: $\{0,1\}^n \to \{0,1\}^m$ 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.

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 smartcards, 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.

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. Even with 128-bit blocks (such as in the AES proposals)

	Construction	**Insecurity**		**No. f-appls.**
		Upper bound	Lower bound	
1.	CBC-2	$\frac{12q^2}{N}$ [5]	$\Omega(\frac{q^2}{N})$ [14]	2
2.	Feistel-t ($t = 3, 4$)	$\frac{q^2}{N}$ [10]	$\Omega(\frac{q^2}{N})$	t
3.	[11]	$O(\frac{q^2}{N})$ [11]	$\Omega(\frac{q^2}{N})$	2
4.	Benes [1]	$O\left(\frac{q}{N}\right)$ [1]	$\Omega\left(\frac{q}{N}\right)$	8
5.	Ω_t ($t \geq 1$) [12]	$\frac{q^{t+1}}{(t+1)N^t}$ [12]	?	$2t$
6.	Feistel-6	$O(\frac{q^4}{N^3} + \frac{q^2}{N^2})$ [13]	?	6

Fig. 1. Input-length doubling transformations: Constructing g: $\{0,1\}^{2n} \rightarrow \{0,1\}^n$ given f: $\{0,1\}^n \rightarrow \{0,1\}^n$. 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.

the probability of repeated queries leaves less security than usually intended. In this case $q = 2^{32}$ provides 2^{-64} security which is much weaker than the usually conjectured "128-bit security" for these ciphers.

Beating the birthday bound. 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: $\{0,1\}^n \rightarrow \{0,1\}^m$, we use a pseudorandom function g: $\{0,1\}^{2n} \rightarrow \{0,1\}^m$. 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 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.

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

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

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

	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

Fig. 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 $q < N/O(t)$, and t is odd in **2.**. Bounds shown are approximate.

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 2 bounds this probability roughly by:

$$d_1(t) \cdot \frac{q^2}{N^t} \qquad \text{for } q \leq \tfrac{N}{e^2 t} \text{ and } d_1(t) = 0.76 \cdot t!, \tag{2}$$

where $N = 2^n$ is the size of the domain of the function. (The bound on q is necessary: given the q sequences of $(r_1, ..., r_t)$'s, the randomness in the process of Equation (1) is only due to f itself which has N bits of randomness.) 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 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 1 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.

Integrity amplification and message authentication. In the Carter-Wegman paradigm [16], 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 [9]. 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 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 bounds the probability of vulnerability by

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

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 (13). (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 2, an analysis of the security of the MAC scheme discussed above.

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 [6] and later by Hall, Wagner, Kelsey and Schneier [8]. These two works provide several constructions that one might use. The works of [6, 8] 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 [15] 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.

2 Definitions

Primitives discussed in this paper include pseudorandom function families [7], symmetric encryption schemes, and MACs. Security of all these will be treated in a concrete framework along the lines of works like [5, 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. The following will suffice either for an experienced reader or for one wanting to understand our results at a first, high level. More precise definitions can be found in [3].

PSEUDORANDOM FUNCTION FAMILIES. [Notion of [7], concretized as per [5]]. To a family F of functions (in which each function maps $\{0,1\}^n$ to $\{0,1\}^m$) we associate an *insecurity function* $\mathbf{InSec}^{\mathrm{prf}}(F,\cdot,\cdot)$ defined as follows: For integers q, T the quantity $\mathbf{InSec}^{\mathrm{prf}}(F,q,T)$ 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.

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* $\mathbf{InSec}^{\mathrm{enc}}(\mathsf{ENC},\cdot,\cdot)$ defined as follows: For integers μ, T the quantity $\mathbf{InSec}^{\mathrm{enc}}(\mathsf{ENC},\mu,T)$ 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 [2] .)

MACS. [Following [4]]. To a message authentication scheme MAC (consisting of a probabilistic mac generation algorithm and deterministic mac verification algorithm[1]) we associate an *insecurity function* $\mathbf{InSec}^{\mathrm{mac}}(\mathsf{MAC},\cdot,\cdot,\cdot)$ defined as follows: For integers q_a, q_v, T the quantity $\mathbf{InSec}^{\mathrm{mac}}(\mathsf{MAC},q_a,q_v,T)$ is the maximum possible probability that an adversary can forge a mac of a new message under an attack in which it obtains valid macs of q_a texts of its choice, verifies up to q_v candidate message/mac pairs of its choice, and runs in time at most T.

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

[1] Traditional MACs are deterministic, so verification can be done by mac recomputation. Our mac generation process is probabilistic, so a separate verification procedure must be prescribed.

restricted, meaning we are in an information theoretic setting. Indeed, this will be the important case in analyses.

3 Pseudorandomness of Parity

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 $\{0,1\}^m$.

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

```
Experiment $\mathrm{Rnd}(n,q)$
    For $i = 1, \ldots, q$ do
        $R_i = (r_{i,1}, \ldots, r_{i,t}) \overset{R}{\leftarrow} D(n,t)$
        $b_i \overset{R}{\leftarrow} \{0,1\}^m$
    End do
    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, and we would like to upper bound it. 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. It follows in particular 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 $D(n,t)$. This is captured by the following quantity:

$$\begin{aligned} &\mathsf{NFRProb}(N,q,t) \\ &= \Pr\left[\mathrm{MTX}_{N,q}(R_1,\ldots,R_q) \text{ is not of full rank } : \ R_1,\ldots,R_q \overset{R}{\leftarrow} D(n,t)\right]. \end{aligned}$$

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:

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

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.

Lemma 1. *Conditioned on the event that* $\mathrm{MTX}_{N,q}(R_1,\ldots,R_q)$ *is of full rank, the outputs of experiment* $\mathrm{Par}(n,q,t)$ *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 $\mathsf{NFRProb}(N,q,t)$.

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

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

where $d_1(t) \ = \ 0.76 \cdot t!$ *and*

$$d_2(t,n) \ = \ \begin{cases} 3e^{3+3t/2} 2^{-3} t^{-3+3t} n^{-3+3t/2} & \text{if } t \text{ is even} \\ e^{4+2t} 2^{-4} t^{-4+4t} n^{-4+2t} & \text{if } t \text{ is odd.} \end{cases} \tag{6}$$

Discussion of the bounds. Let us now interpret the bounds a bit. 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 $t \le 2^{29}$, and we are interested in values like $t = 2, 3, 4, 5$.) 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 noted in the introduction, for entropy reasons alone. 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 $s \ge 3$, 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 $d_1(t) \cdot q^2/N^t$.

Example 1. Take for example $N = 2^{64}$ and $t = 3$. Then $d_1(t) \le 4.6$ and $d_2(3, 64) < 2^{31}$ so

$$\mathsf{NFRProb}(N,q,3) \ \le\ 4.6 \cdot \frac{q^2}{N^3} + 2^{31} \cdot \frac{q^4}{N^6} \ \le\ 4.6 \cdot \frac{q^2}{2^{64*3}} + \frac{q^4}{2^{64*6-31}} \ \le\ 5 \cdot \frac{q^2}{N^3} \ .$$

as long as $q \le N/23$. Thus, we are off from q^2/N^3 by only the small factor of 5. Note in particular the bound is essentially equal to $d_1(t) \cdot q^2/N^t$.

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

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

Comparing with the first term in the bound of Lemma 2 we see that the bounds are tight to within a constant that is independent of N, t, q.

Proof of Lemma 2: 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

$$\mathsf{NFRProb}(N, q, t) \;\leq\; \binom{q}{2} \cdot p(N, 2, t) \;+\; \sum_{i=s}^{q-1} \binom{q}{i} \cdot p(N, i, t) \,. \qquad (7)$$

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

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

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 numerator 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 (7) and the Claim we get

$$\begin{aligned} \mathsf{NFRProb}(N,q,t) &\leq \binom{q}{2} \cdot p(N,2,t) \;+\; \sum_{i=s}^{q-1} \left(\frac{qe}{i}\right)^i \cdot \left(\frac{eti}{2N}\right)^{ti/2} \\ &= \binom{q}{2} \cdot \frac{2d_1(t)}{N^t} \;+\; \sum_{i=s}^{q-1} \left[eq \cdot \left(\frac{et}{2N}\right)^{t/2} \cdot i^{\frac{t}{2}} - 1\right]^i \; . \qquad (8) \end{aligned}$$

The first term of Equation (8) is at most $q^2/2 \cdot 2d_1(t)/N^t = d_1(t) \cdot q^2/N^t$. 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 some calculations show that

$$S \leq \sum_{i=s}^{\alpha \lg N} \left[eq \cdot \left(\frac{et}{2N}\right)^{t/2} \cdot (\alpha \lg N)^{\frac{t}{2}} - 1\right]^i \;+\; \sum_{i=1+\alpha \lg N}^{q} \left[e \cdot \left(\frac{etq}{2N}\right)^{t/2}\right]^i \qquad (9)$$

We will impose upper bounds on q that guarantee

$$A \stackrel{\text{def}}{=} eq \cdot \left(\frac{et}{2N}\right)^{t/2} \cdot (\alpha \lg N)^{\frac{t}{2}} - 1 \;\leq\; \frac{1}{2} \text{ and } B \stackrel{\text{def}}{=} e \cdot \left(\frac{etq}{2N}\right)^{t/2} \;\leq\; \frac{1}{2} \, . \qquad (10)$$

In that case, each of the sums of Equation (9) is bounded by twice its first term, so we can bound the sum itself by

$$2 \cdot \left[eq \cdot \left(\frac{et}{2N}\right)^{t/2} \cdot (\alpha \lg N)^{\frac{t}{2}} - 1\right]^s \;+\; \left[e \cdot \left(\frac{etq}{2N}\right)^{t/2}\right]^{\alpha \lg N}$$

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

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 the above, 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 (10). Recalling that $\alpha = t$, some calculations show that the conditions imposed by $A \leq 1/2$ and $B \leq 1/2$ are, respectively,

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

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

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

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. ▮

CTR MODE ENCRYPTION. Let F be a family of functions with domain $\{0,1\}^n$ and range $\{0,1\}^m$. In this section we look at the problem of encrypting a message of m-bits. (In [3] we discuss how to encrypt messages of longer and varying lengths.)

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 StandardENC-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 $N = 2^n$–

$$\mathbf{InSec}^{\mathrm{enc}}(\mathsf{StandardENC\text{-}Ctr}, qm, T) \; \leq \; 2 \cdot \mathbf{InSec}^{\mathrm{prf}}(F, q, T') + 2^{-m} \; . \qquad (11)$$

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

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

```
ENCRXt[F]: decryption procedure
Input: Key $f$, ciphertext $(r_1, \dots, r_t, mdM)$
    Let $mk = f(r_1) \oplus f(r_2) \oplus \cdots \oplus f(r_t)$
    Let $M = mdM \oplus mk$
    Return $M$
```

Fig. 3. $\mathsf{ENCRX}_t[F]$: Our encryption scheme: Here $M \in \{0,1\}^m$ is the plaintext and $f \in F$ is the key.

the other to decrypt. These algorithms are described in Figure 4. 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.)

The security of our scheme can be analyzed via a connection to matrix rank and Lemma 2, as detailed in [4], to yield the following.

Theorem 1. *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*

$$\mathbf{InSec}^{\mathrm{enc}}(\mathsf{ENCRX}_t[F], qm, T) \;\leq\; d_1(t) \cdot \frac{q^2}{N^t} \;+\; 2 \cdot \mathbf{InSec}^{\mathrm{prf}}(F, tq, T') \;,$$

where $T' = T + O(tq(n+m))$ and $d_1(t)$ is as in Equation (2).

4 Integrity of Parity and Application to MACs

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 $(R_1, b_1, \dots, R_q, b_q)$ of experiment $\mathrm{Par}(n,q,t)$. Now A tries to create a non-colliding sequence $R_{q+1} = (r_{q+1,1}, \dots, r_{q+1,t})$ and a value b_{q+1} such that $R_{q+1} \notin \{R_1, \dots, R_q\}$ 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 $\mathrm{MTX}_{N,q}(R_1, \dots, R_q)$ 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 $R_{q+1} \notin \{R_1, \dots, R_q\}$, 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 $\mathrm{Par}(n, q, t)$ and $\mathrm{Rnd}(n, q)$ 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

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

The following lemma considers an arbitrary adversary that given an output of experiment $\mathrm{Par}(n, q, t)$ 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 3. *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* $\mathrm{Par}(n, q, t)$*, 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]).

Lemma 4. *Let t be such that $1 \le t \le \sqrt{N/(2e \lg N)}$, then for any $q < N/(2e^2 t)$ we have*

$$\mathsf{VulProb}(N, q, t) \;\le\; \begin{cases} d'(t, \lg N) \cdot \dfrac{q^2}{N^{t/2}} & \text{if } t \text{ is even} \\[2ex] d'(t, \lg N) \cdot \dfrac{q^3}{N^t} & \text{if } t \text{ is odd}, \end{cases} \tag{12}$$

where

$$d'(t, n) \;=\; \begin{cases} e^{2+3t/2} 2^{3t/2} 3^{-t/2} t^{-2+3t/2} n^{t-2} & \text{if } t \text{ is even} \\ e^{3+2t} 2^{-3} t^{-3+5t/2} n^{t-2} & \text{if } t \text{ is odd.} \end{cases} \tag{13}$$

Notice the difference in the bounds for odd versus even t. We will focus on odd t. In comparison with Lemma 2 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 $\mathsf{VulProb}(N, q, 3) \leq 2^{24} \cdot q^3/N^3$.

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.

Universal hash based MACs. We now discuss the application to message authentication. 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 $\{0,1\}^m$.

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,

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

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 4. 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

```
$\mathsf{MACRX}_t[F]$: 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 $\langle h, f\rangle$, $M, \sigma$
    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
```

Fig. 4. $\mathsf{MACRX}_t[F]$: 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.

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.

The security of our scheme can be analyzed via a connection to matrix vulnerability and Lemma 4, as detailed in [4], to yield the following.

Theorem 2. *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 $1 \le t \le \sqrt{N/(2e \lg N)}$. Let $\mathsf{MACRX}_t[F]$ be the associated MAC as defined above. Assume $1 \le q_a \le N/(2e^2 t)$ and $q_v \ge 1$. Then*

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

where $T' = T + O(t(q_a + q_v)(n + m))$ and $d'(t,n)$ is as in Equation (13).

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 (14). $\mathsf{MACRX}_5[F]$ is comparable in security to the counter-version.

Acknowledgments

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

Mihir Bellare was supported in part by NSF CAREER Award CCR-9624439 and a 1996 Packard Foundation Fellowship in Science and Engineering. Hugo Krawczyk was supported by the Fund for the Promotion of Research at the Technion.

References

1. W. Aiello, and R. Venkatesan. Foiling birthday attacks in length-doubling transformations. *Advances in Cryptology – Eurocrypt 96 Proceedings*, 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. Beyond the birthday barrier, without counters. Full version of this paper, available via `http://www-cse.ucsd.edu/users/mihir`.
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 Proceedings*, Lecture Notes in Computer Science Vol. 963, D. Coppersmith ed., Springer-Verlag, 1995.
5. M. Bellare, J. Kilian and P. Rogaway. The Security of Cipher Block Chaining. *Advances in Cryptology – Crypto 94 Proceedings*, Lecture Notes in Computer Science Vol. 839, Y. Desmedt ed., Springer-Verlag, 1994.
6. M. Bellare, T. Krovetz and P. Rogaway. Luby-Rackoff backwards: Increasing security by making block ciphers non-invertible. *Advances in Cryptology – Eurocrypt 97 Proceedings*, Lecture Notes in Computer Science Vol. 1233, W. Fumy ed., Springer-Verlag, 1997.
7. O. Goldreich, S. Goldwasser and S. Micali. How to construct random functions. *Journal of the ACM,* Vol. 33, No. 4, 1986, pp. 210–217.
8. C. Hall, D. Wagner, J. Kelsey and B. Schneier. Building PRFs from PRPs. *Advances in Cryptology – Crypto 98 Proceedings*, Lecture Notes in Computer Science Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.
9. H. Krawczyk. LFSR-based Hashing and Authentication. *Advances in Cryptology – Crypto 94 Proceedings*, Lecture Notes in Computer Science Vol. 839, Y. Desmedt ed., Springer-Verlag, 1994.
10. M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Computing,* Vol. 17, No. 2, April 1988.
11. 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.
12. J. Patarin. Improved security bounds for pseudorandom permutations. *Proceedings of the Fourth Annual Conference on Computer and Communications Security*, ACM, 1997.
13. 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.
14. B. Preneel and P. van Oorschott. MDx-MAC and building fast MACs from hash functions. *Advances in Cryptology – Crypto 95 Proceedings*, Lecture Notes in Computer Science Vol. 963, D. Coppersmith ed., Springer-Verlag, 1995.
15. V. Shoup. On Fast and Provably Secure Message Authentication Based on Universal Hashing. *Advances in Cryptology – Crypto 96 Proceedings*, Lecture Notes in Computer Science Vol. 1109, N. Koblitz ed., Springer-Verlag, 1996.
16. 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.