A One-Pass Mode of Operation for Deterministic Message Authentication— Security beyond the Birthday Barrier

Kan Yasuda

NTT Information Sharing Platform Laboratories, NTT Corporation
3-9-11 Midoricho Musashino-shi, Tokyo 180-8585 Japan
yasuda.kan@lab.ntt.co.jp

Abstract. We present a novel mode of operation which iterates a compression function $f : \{0,1\}^{n+b} \to \{0,1\}^n$ meeting a condition $b \geq 2n$. Our construction can be viewed as a way of domain extension, applicable to a fixed-input-length PRF (pseudo-random function) $f_k : \{0,1\}^b \to \{0,1\}^n$ meeting the condition $b \geq 2n$, which yields an arbitrary-input-length PRF $F_k : \{0,1\}^* \to \{0,1\}^n$. Our construction accomplishes both high security (beyond the birthday barrier) and high efficiency (one-pass), with engineering considerations of being stateless, deterministic and single-keyed.

Keywords: pseudo-random function, domain extension, birthday barrier, compression function, mode of operation, message authentication code, tweak, checksum, quasi-random function.

1 Introduction

Birthday Barrier. A message authentication code (MAC) is often constructed of a compression function (*e.g.*, HMAC [1]) via a mode of operation or a block cipher (*e.g.*, CBC-MAC [2]). The security of HMAC and CBC-MAC is based on the fact that they are pseudo-random functions (PRFs), assuming that the underlying primitives (*i.e.*, the compression function and the block cipher) are PRFs. Unfortunately, HMAC and CBC-MAC are inherently vulnerable to birthday attacks due to their naively-chained internal structure [3, 4]. That is, using an n-bit-output compression function or block cipher, HMAC or CBC-MAC gets forged after about $2^{n/2}$ (which is much smaller than the desired 2^n) queries. This generic principle is known as the *birthday barrier*.

For modern compression functions and block ciphers the above attacks require, for example, 2^{128} and 2^{64} queries, which are unlikely to be a practical threat in most scenarios. It is rather a theoretical challenge to construct a mode with security beyond the birthday barrier at minimal costs over existing modes of operation.

Two Already-Known Ways of Breaking the Barrier. It seems that there exist roughly two approaches of breaking the barrier, and hence constructing MACs whose security is beyond the birthday bound. One is to allow use of either nonce elements or random salts. The other is to allow use of multiple passes. Yet, neither of these two approaches is satisfactory, as explained below.

Nonce elements are often used in encryption (*e.g.*, stream ciphers, the counter mode of block ciphers, *etc.*), but their presence is sometimes unwelcome in practical MAC applications; if a nonce value is used in a MAC scheme, then the value needs to be communicated, synchronized and maintained among all parties generating tags and/or verifying message-tag pairs. If instead a random salt is used, then these constraints become somewhat relaxed, but it still leaves problematic properties: the tag size gets enlarged, and the parties creating tags are required to possess a random-number generator.

The use of multiple passes offers construction without counters nor coins but results in inefficiency. Although usually parallelizable owing to their multi-pass structure, these schemes require more numbers of invocations to the underlying primitive, and the performance advantage due to the parallelism depends on each implementation and is generally limited.

Our Contributions. In this paper we devise a novel approach of breaking the birthday barrier. Namely, we utilize some techniques from the area of tweakable block ciphers and combine them with "checksum construction." The combination enables us to provide a one-pass mode of operation that overcomes the birthday limit without relying on the use of counters or coins.

Our starting primitive (*i.e.*, building block) is a compression function $f : \{0,1\}^{n+b} \to \{0,1\}^n$. We require that $b \geq 2n$. We emphasize that this requirement is essential in our construction; we utilize this condition in two (completely) different places.[1] Then using this primitive f, we construct a PRF $F_k : \{0,1\}^* \to \{0,1\}^n$ that satisfies the following seven properties:

1. The security of F is beyond the birthday barrier,
2. F is one-pass, that is, to process a message $M \in \{0,1\}^*$ only requires $|M|/b$ plus a small constant number of invocations to f,
3. Workings outside f consist of only simple machine operations,
4. F is stateless, avoiding use of nonce values or counters,
5. F is deterministic, avoiding use of random salts,
6. F is single-keyed, invoking f only with a fixed key $k \in \{0,1\}^n$ via $f_k(m) \stackrel{\text{def}}{=} f(k\|m)$ for a message block $m \in \{0,1\}^b$, and
7. The security of F is based on the sole assumption that f_k is a PRF.

It appears that no prior mode of operation, iterating either a compression function $f : \{0,1\}^{n+b} \to \{0,1\}^n$ or a block cipher $f_k : \{0,1\}^n \to \{0,1\}^n$, accomplishes the above features concurrently.

[1] We remark that the condition $b \geq 2n$ is not severe limitation in practice. In fact, off-the-shelf compression functions, such as $\mathsf{sha1} : \{0,1\}^{160+512} \to \{0,1\}^{160}$ and $\mathsf{sha256} : \{0,1\}^{256+512} \to \{0,1\}^{256}$, satisfy this requirement.

Organization of the Paper. Section 2 goes through previous work in this field. We then review necessary notions from the area of tweakable block ciphers in Sect. 3. We introduce our mode of operation in Sect. 4. The security proofs of our mode are given in Sect. 5. A couple of techniques to improve the performance of our mode are discussed in Sect. 6. We mention some open problems regarding the domain extension of PRFs in Sect. 7, prior to concluding the paper in Sect. 8.

2 Previous Work

In this section we briefly look over previous constructs that break the birthday barrier, including the ones that take the two approaches mentioned in Sect. 1. Other known results, which have some relevance to the techniques used in the paper, are also cited in Sect. 3, 7 and 8.

Stateful or Randomized Construction. XOR MAC [5] is a parallelizable MAC that is based on a compression function. RMAC [6] is a serial MAC that is based on a block cipher. Both of these MACs guarantee security beyond the birthday barrier, yet XOR MAC is nonce-based and RMAC is a randomized algorithm.

Multiple-Pass-Based Construction. The idea of using two (or more) passes of data processing dates back to the design of RIPEMD and its application to Two-Track MAC [7]. A similar approach appears in the context of keyless hash functions as "Double-Pipe" hash [8]. These constructs effectively preclude birthday attacks, but the problem is that they are twice or more slower than their "single-pass" versions (even though they are somewhat parallelizable). The L-Lane scheme [9] performs better than a naively-doubled construction, but it is still less efficient as compared to a truly-single-pass construction.

Universal-Hash-Based Construction. A similar situation applies to MACs based on universal hash functions. UMAC [10] and MACRX [11] achieve security beyond the birthday barrier, but UMAC is nonce-based and MACRX is randomized. Once these MACs are made deterministic (in the obvious way), the security of such MACs gets degraded behind the birthday barrier immediately.

"Wide-Pipe" and Others. If we used a "wide-pipe" compression function $f : \{0,1\}^{2n+b} \to \{0,1\}^{2n}$ or a "wide" universal hashing with a collision probability $\varepsilon \approx 2^{-2n}$ (in a deterministic MAC), then we could certainly preclude the birthday attacks (in a provably secure way). However, such a method does not solve our problem at hand in nature; such a function deserves $2n$-bit security, not n-bit, or not to mention the fact that schemes based on wide functions would become inefficient.

Yet another approach is to construct a PRF $f'_{k'} : \{0,1\}^{2n} \to \{0,1\}^{2n}$ from a PRF $f_k : \{0,1\}^n \to \{0,1\}^n$ in a birthday-resistant way. Examples include

Benes [12], Ω_t [13] and Feistel-6 [14]. These constructs however require too many (4 or more) invocations to f, and consequently schemes based on such an f' would be inefficient.

Lastly, we mention the Sum construction [15] which gives a way to construct a PRF from PRPs. The security of the resulting PRF is shown to be beyond the birthday limit, but the construction requires at least two invocations to f when instantiated with a single PRP f.

3 Preliminaries

In this section we first review the notion of pseudo-random functions (PRFs) and that of quasi-random functions (QRFs). We then give an overview of the theory of tweakable PRFs. Notice that such a theory is usually based on the framework of block ciphers, but we carefully restate the theory in the language of compression functions (rather than block ciphers). Some parts of the theory are directly translated into the new setting, while other parts need to receive local treatment in the context of compression functions.

Pseudo-random Functions (PRFs). Let $\{f_k : X \to Y\}$ be a family of functions with keys $k \in K$. Informally, we say that f is *pseudo-random* if f_k with a key $k \xleftarrow{\$} K$ randomly chosen is indistinguishable from a truly random function $\varphi : X \to Y$ (*i.e.*, $\varphi \xleftarrow{\$} \mathrm{Func}(X, Y)$ where $\mathrm{Func}(X, Y)$ denotes the set of all functions from X to Y), by computationally-bounded adversaries.

To be more precise, let A denote an adversary trying to distinguish between f and φ. That is, A is given access to either the "real" oracle f or the "random" oracle φ. The f-oracle picks a random key $k \xleftarrow{\$} K$ at the beginning of each experiment and, upon a query $x \in X$ made by A, returns the value $y = f_k(x)$ to A. On the other hand, the φ-oracle picks a random function $\varphi \xleftarrow{\$} \mathrm{Func}(X, Y)$ at the beginning of each experiment and returns the value $y = \varphi(x)$ upon a query $x \in X$. Then the *advantage* of adversary A is defined by

$$\mathrm{Adv}_f^{\mathrm{prf}}(A) \stackrel{\mathrm{def}}{=} \Pr\bigl[A^f \Rightarrow 1\bigr] - \Pr\bigl[A^\varphi \Rightarrow 1\bigr],$$

where by the notation $A^{\mathcal{O}} \Rightarrow 1$ we denote the event that A, given access to oracle $\mathcal{O}$, outputs value 1.

In order for the advantage function to be well-defined, the resources of adversary A need to be bounded. We define

$$\mathrm{Adv}_f^{\mathrm{prf}}(t, q, \ell) \stackrel{\mathrm{def}}{=} \max_A \mathrm{Adv}_f^{\mathrm{prf}}(A),$$

where max runs over all adversaries, whose time complexity is at most t, making at most q oracle queries, each query being at most length ℓ (in some appropriate units). In order to measure the time complexity t, we fix some model of computation. The time complexity includes the maximum time for adversary A to execute each overlying experiment, including the time consumed by oracles, plus the code size of A. If f accepts only fixed-length inputs, then the quantity ℓ is simply omitted from the notation.

Quasi-random Functions (QRFs). The notion of QRFs is an information-theoretic version of that of PRFs [16]. A QRF $\boldsymbol{\psi}$ is a family of functions, indexed not by a key but by smaller random function(s). An adversary A attacking $\boldsymbol{\psi}$ is computationally unbounded. The advantage function is defined similarly:

$$\mathrm{Adv}_{\boldsymbol{\psi}}^{\mathbf{qrf}}(A) \stackrel{\text{def}}{=} \Pr\big[A^{\boldsymbol{\psi}} \Rightarrow 1\big] - \Pr\big[A^{\varphi} \Rightarrow 1\big],$$

and we also define

$$\mathrm{Adv}_{\boldsymbol{\psi}}^{\mathbf{qrf}}(q,\ell) \stackrel{\text{def}}{=} \max_{A} \mathrm{Adv}_{\boldsymbol{\psi}}^{\mathbf{qrf}}(A),$$

where again, ℓ may be omitted from the notation if irrelevant.

Tweaking Pseudo-random Functions. Here we recast the theory of tweakable block ciphers in the context of compression functions. In fact, developing the theory in the style of compression function is easier, because block ciphers are permutations, whilst compression functions are functions, which in particular means that we do not need to exercise the PRP $\leftrightarrow$ PRF Switching Lemma. Also, we utilize the condition $b \geq 2n$ here, which is something impossible with block ciphers where there exists an innate relation $b = n$.

The purpose of tweaking a PRF f_k is to construct many functions $f_1, f_2, \ldots$ from f which are indistinguishable from a collection of (truly) random functions $\varphi_1, \varphi_2, \ldots$. In order to do this, we begin with defining an initial value Δ_0 of masks to be the leftmost b bits of

$$f_k(1) \,\|\, f_k(2) \,\|\, \cdots \,\|\, f_k(\lceil b/n \rceil),$$

where integers $1, 2, \ldots, \lceil b/n \rceil$ are represented as b-bit strings by some canonical encoding. We then modify this value Δ_0 sequentially, by "incrementing" as

$$\Delta_1\,,\ \Delta_2\,,\ \ldots\,,\ \Delta_\ell,$$

up until about $\ell \approx 2^n$. It is essential here that the values $\Delta_1, \Delta_2, \ldots, \Delta_\ell$ are all distinct. In addition, we also need a "special" set of offsets

$$\bar{\Delta}_{L,1}\,,\ \bar{\Delta}_{L,2}\,,\ \bar{\Delta}_{L,3},$$

for each $L \in \{1, 2, \ldots, \ell\}$. All of these values need to be distinct among themselves and from the above list of ℓ-many values.

In our construction a message $M \in \{0,1\}^*$ needs to be padded so that the length becomes a multiple of b bits, before being processed. This would cause the length to increase by b bits in case $|M|$ is already a multiple of b. If one wants to avoid the extra block of computation when $|M|$ happens to be exactly equal to a multiple of b bits, then one needs another special set of offsets

$$\bar{\bar{\Delta}}_{L,1}\,,\ \bar{\bar{\Delta}}_{L,2}\,,\ \bar{\bar{\Delta}}_{L,3},$$

for performance optimization (saving one block of computation). For the sake of simplicity, we do not make use of such masks $\bar{\bar{\Delta}}_1, \bar{\bar{\Delta}}_2, \bar{\bar{\Delta}}_3$ and do contend ourselves with the trivial padding $M\|10^*$. Our construction always requires three blocks of extra computation in any event, so the effectiveness of such optimization is limited. All the proofs carry over with such optimization but only become more complicated.

Incrementing Masks. It remains to describe the ways of "incrementing" the masks. There are several known methods [17–20], and some of them can be transformed into the context of compression functions. In the following we modify the method in [20] so that it becomes compatible with our construction.

The basic framework of [20] is to let $\Delta_i \stackrel{\text{def}}{=} \alpha^i \cdot \Delta_0$, where the multiplication is done in the finite field $\mathbb{F}_{2^b}$, and $\alpha \in \mathbb{F}_{2^b}^\times$ is a non-zero element whose multiplicative order is large enough (say $\geq 2^n$). The functions f_i are created via $f_i(m) \stackrel{\text{def}}{=} f_k(m \oplus \Delta_i)$. The special offsets are created via $\bar{\Delta}_{L,j} \stackrel{\text{def}}{=} \alpha^L \cdot \beta^j \cdot \Delta_0$, where $\beta \in \mathbb{F}_{2^b}^\times$ is an element such that $\alpha^L\beta^j$ can be guaranteed to be distinct from α^is. A preferred choice of α, β is usually $\alpha = 2$ and $\beta = 3$.

The finite field needs to be represented by an irreducible polynomial $g(x) \in \mathbb{F}_2[x]$ of degree b, with $\alpha = x(=$ "2") being a generator of $\mathbb{F}_{2^b}^\times$ (so that its multiplicative order is 2^b-1). Then we compute $\log_x(x+1)$ in this field and verify that it is huge, which enables us to choose $\beta = x + 1(=$ "3"). Computing such discrete logarithms for block ciphers has been feasible owing to small parameters such as $b = 64$ and $b = 128$ [20].

Yet, now we are dealing with a compression function with a parameter such as $b = 512$, which most likely stops us from computing such discrete logarithms. So instead we choose an irreducible polynomial $g(x)$ so that $\alpha = 2$ generates only a subgroup of $\mathbb{F}_{2^b}^\times$ but its order being large enough ($\geq 2^n$). Then we merely need to verify that $\beta = 3$ generates the subgroup "missed" by $\alpha = 2$.

For example, consider the case $b = 512$ and $n = 128$. We are then working in the multiplicative group $\mathbb{F}_{2^{512}}^\times$ of the field with 2^{512} elements, and the order of the group $2^{512}-1$ can be factored as $2^{512}-1 = (2^1+1)(2^2+1)\cdots(2^{128}+1)(2^{256}+1)$. In particular, the term $2^{128}+1$ can be further factored as [21]:

$$2^{128} + 1 = 59649589127497217 \times 5704689200685129054721.$$

It can be directly verified that these two prime factors appear nowhere else in the factorization of $2^{512} - 1$.

Now we choose $x^{512} + x^{12} + x^7 + x^2 + 1 \in \mathbb{F}_2[x]$ as an irreducible polynomial to represent the field $\mathbb{F}_{2^{512}}$ and verify that $x^{(2^{512}-1)/59649589127497217} \neq 1$ and $x^{(2^{512}-1)/5704689200685129054721} \neq 1$ in this field, which ensures that the multiplicative order of the element x is at least $2^{128}+1$. On the contrary, notice that $x^{(2^{512}-1)/17} = 1$, where $17 = 2^4 + 1$ appears only once in the factorization of $2^{512} - 1$, from which we deduce that the element x does not "generate" the subgroup of order 17 in the multiplicative group $\mathbb{F}_{2^{512}}^\times$. On the other hand, observe that $(x+1)^{(2^{512}-1)/17} \neq 1$, which implies that the group generated by $x+1$ <u>does</u> contain the subgroup of order 17.

After the above verification we are able to set

$$\Delta_i \stackrel{\text{def}}{=} x^i \Delta_0 \text{ and } \bar{\Delta}_{L,j} \stackrel{\text{def}}{=} x^L(x+1)^j \Delta_0$$

for $i, L \in \{1, 2, \ldots, 2^{128}\}$ and $j \in \{1, 2, 3\}$. These masks are all distinct because of the following three reasons: (1) We have $\Delta_i \neq \Delta_{i'}$ if $i \neq i'$, owing to the high order of the element x; (2) We have $\Delta_i \neq \bar{\Delta}_{L,j}$ for any i, L, j in the above ranges, because $x^L(x+1)^j$ generates a group that contains the subgroup of order 17 while x^i does not; (3) We have $\bar{\Delta}_{L,j} \neq \bar{\Delta}_{L',j'}$ as long as $(L, j) \neq (L', j')$, for if the equality $x^L(x+1)^j\Delta_0 = x^{L'}(x+1)^{j'}\Delta_0$ holds in the field with i, L, j being in the above ranges, then by looking at the subgroup of order 17 we see that $j = j'$, which immediately implies that $L = L'$.

Lemma 1. *If f is a PRF and the masks $\Delta_1, \Delta_2, \ldots, \Delta_\ell \in \{0,1\}^b$ are all distinct, created via $\Delta_i \stackrel{\text{def}}{=} \gamma_i \cdot \Delta_0 \in \mathbb{F}_{2^b}$ with γ_i being some (public) function of i independent of the value Δ_0, then the functions $f_1, f_2, \ldots, f_\ell$ defined by $f_i(m) \stackrel{\text{def}}{=} f_k(m \oplus \Delta_i)$ are indistinguishable from random functions $\varphi_1, \varphi_2, \ldots, \varphi_\ell$, by an adversary having time complexity at most t and making at most $q \geq \lceil b/n \rceil$ queries to each f_i (or φ_i), except for the probability at most*

$$\mathrm{Adv}_f^{\mathrm{prf}}(t, q') + \frac{q^2}{2^{2n-1}},$$

where $q' = (\ell + 1)q$.

Proof. The proof is done via hybrid argument. Consider an intermediate oracle $\boldsymbol{\rho}$ which chooses a random function $\rho : \{0,1\}^b \to \{0,1\}^n$ at the beginning of each experiment and upon a query m to f_i returns $\boldsymbol{\rho_i}(m) \stackrel{\text{def}}{=} \rho(m \oplus \Delta_i)$ instead. Here, Δ_0 is computed as the leftmost b bits of

$$\rho(1) \parallel \rho(2) \parallel \cdots \parallel \rho(\lceil b/n \rceil),$$

and the masks $\Delta_1, \Delta_2, \ldots, \Delta_\ell$ are generated accordingly, which are all distinct as long as $\Delta_0 \neq 0^b$.

Now let A be an adversary trying to distinguish between $f_1, f_2, \ldots, f_\ell$ and $\varphi_1, \varphi_2, \ldots, \varphi_\ell$. Assume that A has time complexity at most t and makes at most q queries to each f_i (or φ_i). It is straightforward to see that the probability that A distinguish between $f_1, f_2, \ldots, f_\ell$ and $\boldsymbol{\rho_1}, \boldsymbol{\rho_2}, \ldots, \boldsymbol{\rho_\ell}$ is at most

$$\mathrm{Adv}_f^{\mathrm{prf}}(t, q'),$$

where $q' \stackrel{\text{def}}{=} (\ell + 1)q \geq \ell q + \lceil b/n \rceil$.

We next show that $\boldsymbol{\rho}$ is quasi-random. Observe that functions $\boldsymbol{\rho_1}, \boldsymbol{\rho_2}, \ldots, \boldsymbol{\rho_\ell}$ behave just like random functions $\varphi_1, \varphi_2, \ldots, \varphi_\ell$ unless one of the following events occurs: (1) $\Delta_0 = 0^b$, or (2) A "collision" occurs among the inputs to ρ and $\boldsymbol{\rho_i}$. The probability for event (1) to occur is exactly $2^{-b} \leq 2^{-2n}$. For (2), if a collision occurs between inputs to ρ and $\boldsymbol{\rho_i}$, then it means that $j = m \oplus \Delta_i = m \oplus \gamma_i \Delta_0$ for some $j \in \{1, 2, \ldots, \lceil b/n \rceil\}$. This yields $(j \oplus m)/\gamma_i = \Delta_0$, and for a fixed (j, i) the probability of such an event is $2^{-b} \leq 2^{-2n}$. On the other hand, if a collision occurs between an input to $\boldsymbol{\rho_i}$ and an input to $\boldsymbol{\rho_j}$ for some $1 \leq i < j \leq \ell$, then it means that we have $m \oplus \Delta_i = m' \oplus \Delta_j$, or equivalently $m \oplus \gamma_i \Delta_0 = m' \oplus \gamma_j \Delta_0$.

This yields $(m \oplus m')/(\gamma_i \oplus \gamma_j) = \Delta_0$, and for a fixed (i, j) the probability that such an event occurs is $2^{-b} \leq 2^{-2n}$.

Since the values returned by $\boldsymbol{\rho}$ are random, adversary A learns nothing from them to bring about a collision. That is, we can assume that A is non-adaptive and outputs a sequence of fixed values $(i_1, m_1), (i_2, m_2), \ldots, (i_q, m_q)$, hoping that a collision occurs among them [16]. Now for the first type of collision there are at most $\lceil b/n \rceil \cdot q$ possible pairs, while for the second type there are at most $\binom{q}{2}$ pairs. Thus the advantage that A distinguish between $\boldsymbol{\rho_1}, \boldsymbol{\rho_2}, \ldots, \boldsymbol{\rho_\ell}$ and $\varphi_1, \varphi_2, \ldots, \varphi_\ell$ is at most

$$\frac{1}{2^{2n}} + \lceil b/n \rceil \cdot q \cdot \frac{1}{2^{2n}} + \binom{q}{2} \cdot \frac{1}{2^{2n}} \leq \frac{q^2}{2^{2n-1}}.$$

□

4 Description of the Proposed Mode

In this section we give the definition of our algorithm. Recall that our starting primitive is a compression function $f : \{0,1\}^{n+b} \rightarrow \{0,1\}^n$. We key it via $f_k(m) \stackrel{\text{def}}{=} f(k \| m)$ and tweak it via $f_i(m) \stackrel{\text{def}}{=} f_k(m \oplus \Delta_i)$, obtaining

$$f_1 \,,\, f_2 \,,\, \ldots \,,\, f_\ell \,,\, \bar{f}_1 \,,\, \bar{f}_2 \,,\, \bar{f}_3,$$

which should be (computationally) indistinguishable from random functions $\varphi_1, \varphi_2, \ldots, \varphi_\ell, \bar{\varphi}_1, \bar{\varphi}_2, \bar{\varphi}_3$ (Recall that f_i depends on the choice of key k, while $\bar{f}_i$ depends on the message length L, and so does $\bar{\varphi}_i$).

Now with these tweaked functions in hand, we first define a function (which depends on the choice of L)

$$\bar{f}_{123} : \{0,1\}^{b+2n} \rightarrow \{0,1\}^n,$$

from the three functions $\bar{f}_1$, $\bar{f}_2$ and $\bar{f}_3$. This function is used at the end of processing a message in our mode of operation

$$F_k : \{0,1\}^* \rightarrow \{0,1\}^n.$$

See Fig. 1 for precise definitions, as well as Fig. 2 for a pictorial description.

The construction of $\bar{f}_{123}$ may look unnatural at first glance. We note that this is not the only one that works. For example, the roles of S and $v_L \| s$ may be switched, or Two-Lane construction [9] may be used in the place. Our choice of $\bar{f}_{123}$ simply comes from considerations of efficiency.

The major feature of our mode of operation is the usage of message checksum $S = \bigoplus_{i=1}^{L} m_i$ and intermediate-value checksum $s = \bigoplus_{i=1}^{L} v_i$. The checksum construction is effectively combined with the tweaked compression functions, yielding security beyond the birthday barrier.

Algorithm $\bar{f}_{123}(S\|v_L\|s)$ // $S \in \{0,1\}^b$, $v_L, s \in \{0,1\}^n$
  Set $u \leftarrow v_L\|s$
  Compute $\Sigma_1 \leftarrow \bar{f}_1(S)$ and $\Sigma_2 \leftarrow \bar{f}_2(S)$
  Set $w \leftarrow (\Sigma_1\|\Sigma_2) \oplus u$
  Output $\tau \leftarrow \bar{f}_3(w\|0^{b-2n})$

Algorithm $F_k(M)$ // $M \in \{0,1\}^*$
  Pad $M \leftarrow M\|10^*$
  Divide $M = m_1\|m_2\|\cdots\|m_L$ so that $m_i \in \{0,1\}^b$
  Compute checksum $S \leftarrow \bigoplus_{i=1}^{L} m_i$
  Initialize $v_0 \leftarrow 0^n$
  Iterate $v_i \leftarrow f_i\big(m_i \oplus (v_{i-1}\|0^{b-n})\big)$ for $i = 1, 2, \ldots, L$
  Compute checksum $s \leftarrow \bigoplus_{i=1}^{L} v_i$
  Output $\tau \leftarrow \bar{f}_{123}(S\|v_L\|s)$

Fig. 1. Definitions of $\bar{f}_{123}$ and F_k

5 Proofs of Security beyond the Birthday Barrier

We want to prove that our mode of operation F_k is (computationally) indistinguishable from a truly random function $\Psi : \{0,1\}^* \to \{0,1\}^n$ in such a way as its security is still guaranteed when $q \approx 2^{n/2}$. Succinctly, we prove the following theorem:

Theorem 1. *Let $F_k : \{0,1\}^* \to \{0,1\}^n$ be the mode of operation as defined in Sect. 4. It is a PRF without the birthday barrier if the underlying compression function is a PRF. Concretely, we have*

$$\mathrm{Adv}_F^{\mathrm{prf}}(t, q, \ell) \le \mathrm{Adv}_f^{\mathrm{prf}}(t, q') + \frac{3q^2}{2^{2n}},$$

where $q \ge \lceil b/n \rceil$ and $q' = (\ell + 4)q$.

The proof is based on hybrid argument. In order to prove that F_k is a PRF via hybrid argument, we construct intermediate QRFs $\boldsymbol{\Phi}$ and $\boldsymbol{\Phi_\psi}$ as below.

The QRF $\boldsymbol{\Phi} : \{0,1\}^* \to \{0,1\}^n$ is constructed as follows. In the definition of F_k, we replace functions $f_1, f_2, \ldots, f_\ell$ with random functions $\varphi_1, \varphi_2, \ldots, \varphi_\ell$, where $\varphi_i : \{0,1\}^b \to \{0,1\}^n$ is drawn independently at random. We also replace $\bar{f}_1, \bar{f}_2, \bar{f}_3$ with random functions $\bar{\varphi}_1, \bar{\varphi}_2, \bar{\varphi}_3$ (The choice of these random functions depends on the value L). This gives us a (to-be-proven) QRF $\boldsymbol{\Phi}$. See Fig. 3 for an illustration of $\boldsymbol{\Phi}$.

The other QRF $\boldsymbol{\Phi_\psi}$ is obtained by modifying the last component in $\boldsymbol{\Phi}$. In the definition of $\boldsymbol{\Phi}$, note that we have a (to-be-proven) QRF

$$\boldsymbol{\bar{\varphi}_{123}} : \{0,1\}^{b+2n} \to \{0,1\}^n,$$

which is constructed of $\bar{\varphi}_1, \bar{\varphi}_2, \bar{\varphi}_3$ (Needless to say, $\boldsymbol{\bar{\varphi}_{123}}$ in $\boldsymbol{\Phi}$ corresponds to $\bar{f}_{123}$ in F_k). We replace this QRF $\boldsymbol{\bar{\varphi}_{123}}$ with a truly random function

$$\psi : \{0,1\}^{b+2n} \to \{0,1\}^n.$$

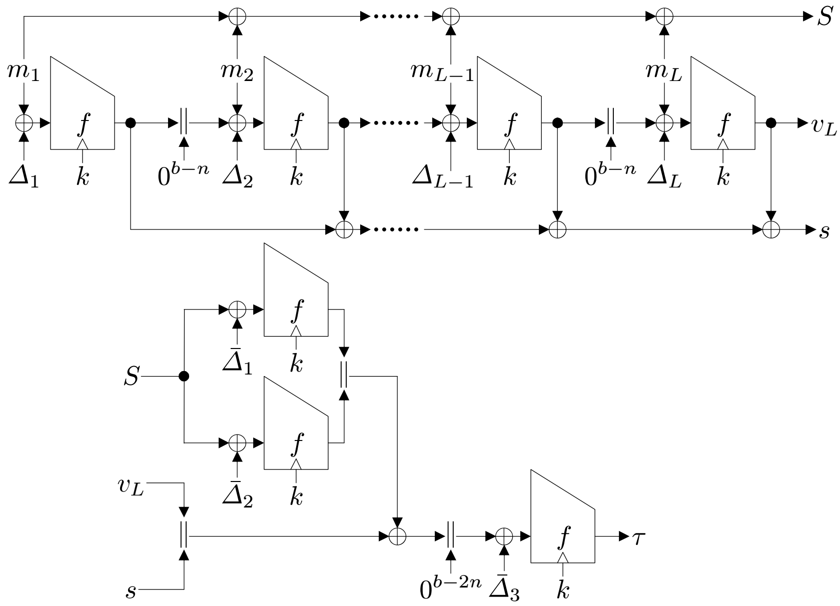

Fig. 2. Proposed mode of operation F_k (the lower half corresponding to $\bar{f}_{123}$)

That is to say, for each value of L, the function $\bar{\boldsymbol{\varphi}}_{\mathbf{123}}$ is replaced with a new random function $\psi = \psi_L$. We name the resulting scheme as $\boldsymbol{\Phi}_{\boldsymbol{\psi}} : \{0,1\}^* \to \{0,1\}^n$.

Now the hybrid argument works as follows. Let A be an adversary trying to distinguish between F_k and Ψ. Then

$$\begin{aligned}\mathrm{Adv}_F^{\mathrm{prf}}(A) &\stackrel{\mathrm{def}}{=} \Pr\big[A^F \Rightarrow 1\big] - \Pr\big[A^{\Psi} \Rightarrow 1\big] \\ &= \Pr\big[A^F \Rightarrow 1\big] - \Pr\big[A^{\boldsymbol{\Phi}} \Rightarrow 1\big] \\ &\quad + \Pr\big[A^{\boldsymbol{\Phi}} \Rightarrow 1\big] - \Pr\big[A^{\boldsymbol{\Phi}_{\boldsymbol{\psi}}} \Rightarrow 1\big] \\ &\qquad + \Pr\big[A^{\boldsymbol{\Phi}_{\boldsymbol{\psi}}} \Rightarrow 1\big] - \Pr\big[A^{\Psi} \Rightarrow 1\big].\end{aligned}$$

We bound the three differences in the rest of this section.

To evaluate the first difference, we note that it is rather straightforward to see that

$$\Pr\big[A^F \Rightarrow 1\big] - \Pr\big[A^{\boldsymbol{\Phi}} \Rightarrow 1\big] \le \mathrm{Adv}_f^{\mathrm{prf}}(t, q') + \frac{q^2}{2^{2n-1}},$$

where $q' \stackrel{\mathrm{def}}{=} (\ell+4)q$. This is because distinguishing between F and Φ essentially amounts to the security of tweaked functions $f_1, f_2, \ldots, f_\ell, \bar{f}_1, \bar{f}_2, \bar{f}_3$, where each $\bar{f}_i$ may vary upon each query (of varying length). So the above inequality follows from Lemma 1.

We next bound the second difference. It is again easy to see that

$$\Pr\big[A^{\boldsymbol{\Phi}} \Rightarrow 1\big] - \Pr\big[A^{\boldsymbol{\Phi}_{\boldsymbol{\psi}}} \Rightarrow 1\big] \le \mathrm{Adv}_{\bar{\boldsymbol{\varphi}}_{\mathbf{123}}}^{\mathbf{qrf}}(q).$$

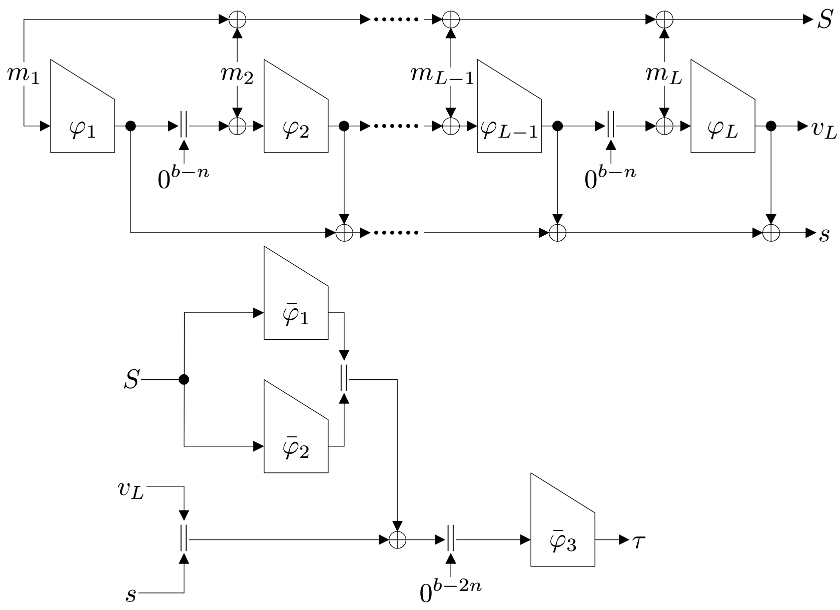

Fig. 3. Description of $\boldsymbol{\Phi}$ (the lower half corresponding to $\bar{\boldsymbol{\varphi}}_{\mathbf{123}}$)

This is because any adversary trying to distinguish between $\boldsymbol{\Phi}$ and $\boldsymbol{\Phi}_{\boldsymbol{\psi}}$ essentially amounts to distinguishing between $\bar{\boldsymbol{\varphi}}_{\mathbf{123}}$ and ψ. So it remains to evaluate the quantity $\mathrm{Adv}^{\mathbf{qrf}}_{\bar{\boldsymbol{\varphi}}_{\mathbf{123}}}(q)$. We do this in the following lemma:

Lemma 2. *Fix L. Then the function $\bar{\boldsymbol{\varphi}}_{\mathbf{123}}$ is a quasi-random function. More concretely, we have*

$$\mathrm{Adv}^{\mathbf{qrf}}_{\bar{\boldsymbol{\varphi}}_{\mathbf{123}}}(q) \leq \frac{q^2}{2^{2n+1}}.$$

Proof. Let B be an adversary trying to distinguish between $\bar{\boldsymbol{\varphi}}_{\mathbf{123}}$ and a truly random function $\psi : \{0,1\}^{b+2n} \rightarrow \{0,1\}^n$. Since $\bar{\varphi}_3$ is a random function, $\bar{\boldsymbol{\varphi}}_{\mathbf{123}}$ behaves just like a truly random function unless a collision occurs among the inputs to $\bar{\varphi}_3$. By a "collision" we mean an event $w = w'$ for distinct inputs $S\|u \neq S'\|u'$ (We carry over the symbols such as w, S, u from the definition of $\bar{f}_{123}$ in Fig. 1). We want to evaluate the probability that such an event occurs.

Since the values returned by $\bar{\boldsymbol{\varphi}}_{\mathbf{123}}$ are random, and B learns nothing from the values in order to bring about a collision, without loss of generality we can assume that B is non-adaptive [16]. That is to say, B just queries a sequence of fixed values $S_1\|u_1, S_2\|u_2, \ldots, S_q\|u_q$, hoping that a "collision" occurs between w_i and w_j for some $1 \leq i < j \leq q$.

So suppose $S\|u \neq S'\|u'$ and $w = w'$. We claim that the probability that such an event occurs is at most 2^{-2n}. To see this, we first observe that $S \neq S'$, for if $S = S'$, then $\Sigma_1\|\Sigma_2 = \Sigma_1'\|\Sigma_2'$ and $u \neq u'$, which implies $w \neq w'$ and hence never a collision. Thus we are looking at an event such that $\Sigma_1 \oplus v_L = \Sigma_1' \oplus v_L'$

and $\Sigma_2 \oplus s = \Sigma_2' \oplus s'$ for some fixed S, S', v_L, v_L', s, s'. Since $\bar{\varphi}_1$ and $\bar{\varphi}_2$ are random functions, the probability that each event occurs is 2^{-n}. Moreover, since $\bar{\varphi}_1$ and $\bar{\varphi}_2$ are independently random, the probability that both events occur is $2^{-n} \cdot 2^{-n} = 2^{-2n}$.

We have seen that the probability that $S_i \| u_i \neq S_j \| u_j$ and $w_i = w_j$ is at most 2^{-2n}. Since there are at most $\binom{q}{2}$ choices of values (i, j), we conclude that

$$\mathrm{Adv}^{\mathbf{qrf}}_{\bar{\boldsymbol{\varphi}}_{\mathbf{123}}}(q) \leq \binom{q}{2} \cdot \frac{1}{2^{2n}} \leq \frac{q^2}{2^{2n+1}}.$$

□

Now note that A's varying lengths L of its queries does not contribute to increasing the collision probability. So we obtain

$$\Pr\bigl[A^{\boldsymbol{\Phi}} \Rightarrow 1\bigr] - \Pr\bigl[A^{\boldsymbol{\Phi}_{\psi}} \Rightarrow 1\bigr] \leq \frac{q^2}{2^{2n+1}}.$$

Lastly, we bound the third difference. This is nothing but the quantity

$$\mathrm{Adv}^{\mathbf{qrf}}_{\boldsymbol{\Phi}_{\psi}}(A) \stackrel{\text{def}}{=} \Pr\bigl[A^{\boldsymbol{\Phi}_{\psi}} \Rightarrow 1\bigr] - \Pr\bigl[A^{\Psi} \Rightarrow 1\bigr],$$

by definition. Hence in the next lemma we show that $\boldsymbol{\Phi}_{\boldsymbol{\psi}}$ is indeed quasi-random:

Lemma 3. *The function $\boldsymbol{\Phi}_{\boldsymbol{\psi}}$ is quasi-random. More concretely, we have*

$$\mathrm{Adv}^{\mathbf{qrf}}_{\boldsymbol{\Phi}_{\psi}}(q, \ell) \leq \frac{q^2}{2^{2n+1}}.$$

Note that the quantity ℓ vanishes on the right-hand side.[2]

Proof. Since $\psi : \{0,1\}^{b+2n} \to \{0,1\}^n$ is a random function, $\boldsymbol{\Phi}_{\boldsymbol{\psi}}$ behaves just like a truly random function except when a collision occurs among the inputs to ψ. Here by a "collision" we mean an event that for two distinct queries $M = m_1 \| m_2 \| \cdots \| m_L$ and $M' = m_1' \| m_2' \| \cdots \| m_{L'}'$ the equality $S \| v_L \| s = S' \| v_{L'}' \| s'$ holds.

We want to evaluate the probability that such a collision occurs. We divide our proof into two cases, depending on the lengths L, L' of two messages.

Case A: $\boldsymbol{L \neq L'}$. There is nothing to prove in this case. That is, since the choice of ψ changes for different values of L, two independently random functions, say ψ_L and $\psi_{L'}$, are used for messages of different lengths. So there is no "collision" to consider here; the two outputs are truly random.

Case B: $\boldsymbol{L = L'}$. Observe that from the condition $M \neq M'$ there exists a unique $a \in \{1, 2, \ldots, L\}$ such that $(v_{a-1}, m_a) \neq (v_{a-1}', m_a')$ and $(v_{i-1}, m_i) = (v_{i-1}', m_i')$ holds for $i = a+1, a+2, \ldots, L$.

[2] An implicit assumption here is that upto $\ell \approx 2^n$ we have ℓ-many random functions for the place of ψ; *cf.* [22].

Case B-1: $(v_{a-1}\|0^{b-n}) \oplus m_a = (v'_{a-1}\|0^{b-n}) \oplus m'_a$. In this case we note that the rightmost $b-n$ bits of m_a and m'_a must be identical, and with the condition $(v_{a-1}, m_a) \neq (v'_{a-1}, m'_a)$ we see that $v_{a-1} \neq v'_{a-1}$ <u>and</u> $m_a \neq m'_a$. Since $v_{a-1} \neq v'_{a-1}$, the two inputs to the random function φ_{a-1} must differ, implying that v_{a-1} and v'_{a-1} are two independently random values. It means that the equality $(v_{a-1}\|0^{b-n}) \oplus m_a = (v'_{a-1}\|0^{b-n}) \oplus m'_a$ holds with a probability of 2^{-n}. Moreover, observe that since $s = s'$ and $v_i = v'_i$ for $a \leq i \leq L$ we must have $\bigoplus_{i=1}^{a-1} v_i = \bigoplus_{i=1}^{a-1} v'_i$. The condition $v_{a-1} \neq v'_{a-1}$ also tells us that $\bigoplus_{i=1}^{a-2} v_i \neq \bigoplus_{i=1}^{a-2} v'_i$. Now put $s_{a-2} \stackrel{\text{def}}{=} \bigoplus_{i=1}^{a-2} v_i$ and $s'_{a-2} \stackrel{\text{def}}{=} \bigoplus_{i=1}^{a-2} v'_i$. Then the values s_{a-2} and s'_{a-2} are created using random functions $\varphi_1, \varphi_2, \ldots, \varphi_{a-2}$, which are all independent from the random function φ_{a-1}. Therefore, the equality $s_{a-2} \oplus v_{a-1} = s'_{a-2} \oplus v'_{a-1}$ holds with a probability of 2^{-n}. This event is clearly independent from the previous equality, so this case occurs with a probability at most $2^{-n} \cdot 2^{-n} = 2^{-2n}$.

Case B-2: $(v_{a-1}\|0^{b-n}) \oplus m_a \neq (v'_{a-1}\|0^{b-n}) \oplus m'_a$. In this case the inputs to the random function φ_a are different, but their outputs are colliding (*i.e.*, $v_a = v'_a$). Clearly, the probability that such an event occurs is exactly 2^{-n}.

Case B-2-(i): $v_{a-1} \neq v'_{a-1}$. In this case we do an analysis similar to Case B-1. The condition $v_{a-1} \neq v'_{a-1}$ tells us that $\bigoplus_{i=1}^{a-2} v_i \neq \bigoplus_{i=1}^{a-2} v'_i$. Put $s_{a-2} \stackrel{\text{def}}{=} \bigoplus_{i=1}^{a-2} v_i$ and $s'_{a-2} \stackrel{\text{def}}{=} \bigoplus_{i=1}^{a-2} v'_i$. Then we have $s_{a-2} \neq s'_{a-2}$, and the equality $s_{a-2} \oplus v_{a-1} = s'_{a-2} \oplus v'_{a-1}$ holds with a probability of 2^{-n}.

Case B-2-(ii): $v_{a-1} = v'_{a-1}$. In this case we have $s_{a-1} = s'_{a-1}$. Since $M \neq M'$ and $S = \bigoplus_{i=1}^{L} m_i = S' = \bigoplus_{i=1}^{L'} m'_i$, there must exist at least two values of $i \in \{1, 2, \ldots, L\}$ such that $m_i \neq m'_i$. One of such values may be equal to the value a, but it still guarantees that there exists a $b \in \{1, 2, \ldots, a-1\}$ such that $(v_{b-1}, m_b) \neq (v'_{b-1}, m'_b)$ and $v_i = v'_i$, $s_i = s'_i$ for $i = b, b+1, \ldots, a-1$ and $m_i = m'_i$ for $i = b+1, b+2, \ldots, a-1$. Then we do an analysis at block b similar to Case B-1 and B-2 as done at block a, in order to prove that such an event happens at block b with a probability at most 2^{-n}.

In all events we see that the collision probability in Case B-2 is at most $2^{-n} \cdot 2^{-n} = 2^{-2n}$.

We have shown that in all cases the collision probability is at most 2^{-2n}. Since the values returned by ψ are random, and A learns nothing from these values in bringing about a collision, we can assume that A is non-adaptive. So assume that A makes a fixed sequence of queries $M_1, M_2, \ldots, M_q$, hoping that a collision occurs at some $1 \leq i < j \leq q$. We have just seen that for a pair (M_i, M_j) the probability that the two messages collide is at most 2^{-2n}. Since there are at most $\binom{q}{2}$ pairs, we conclude that

$$\mathrm{Adv}^{\mathbf{qrf}}_{\boldsymbol{\Phi}_{\psi}}(q, \ell) \leq \binom{q}{2} \cdot \frac{1}{2^{2n}} < \frac{q^2}{2^{2n+1}}.$$

□

Now we go back to proving our main theorem. We have

$$\begin{aligned}
\mathrm{Adv}_F^{\mathrm{prf}}(A) &\stackrel{\mathrm{def}}{=} \Pr\big[A^F \Rightarrow 1\big] - \Pr\big[A^{\Psi} \Rightarrow 1\big] \\
&= \Pr\big[A^F \Rightarrow 1\big] - \Pr\big[A^{\boldsymbol{\Phi}} \Rightarrow 1\big] \\
&\quad + \Pr\big[A^{\boldsymbol{\Phi}} \Rightarrow 1\big] - \Pr\big[A^{\boldsymbol{\Phi}_\psi} \Rightarrow 1\big] \\
&\qquad + \Pr\big[A^{\boldsymbol{\Phi}_\psi} \Rightarrow 1\big] - \Pr\big[A^{\Psi} \Rightarrow 1\big] \\
&\le \mathrm{Adv}_f^{\mathrm{prf}}(t, q') + \frac{q^2}{2^{2n-1}} + \frac{q^2}{2^{2n+1}} + \frac{q^2}{2^{2n+1}} \\
&= \mathrm{Adv}_f^{\mathrm{prf}}(t, q') + \frac{3q^2}{2^{2n}},
\end{aligned}$$

where $q' = (\ell + 4)q$. This proves our main theorem.

6 Optimization for Better Performance

In this section we introduce a couple of techniques to improve the performance of our mode. One is associated with the methods of setting up the masks, and the other is related to the ways of keying the compression function.

Mask Partition. The performance of our mode should be essentially as good as that of a naively-chained construction such as the Merkle-Damgård iteration (HMAC), except for the computational costs of workings outside the underlying primitive f. These include concatenation, XOR, mask setup (initialization) and its incrementation. The last calculation can be realized with a 1-bit (left-)shift operation plus a conditional XOR, because an incrementation corresponds to multiplying x to the mask in the field $\mathbb{F}_{2^b}$ (multiplication by $x + 1$ requires slightly more operations).

The 1-bit shift operation may be costly in software implementations, because we need to perform the operation on a long mask, say $b = 512$ bits, while the available size of registers may be much smaller, say 32 bits. The long-size mask causes another problem that we may be forced to store data outside registers, further lowering performance. These difficulties can be relaxed by dividing the b-bit mask into copies of a $2n$-bit mask. For example, consider the case $b = 512$ and $n = 128$. Then we can use $\varDelta_i \| \varDelta_i$ as the mask, where $\varDelta_i$ is a 256-bit mask (using for example $x^{256} + x^{16} + x^3 + x^2 + 1 \in \mathbb{F}_2[x]$ as an irreducible polynomial). Note that our proofs work with such a construction without significant changes.

Key-Length Flexibility. Our mode does not require re-keying, presenting a contrast to the classical Merkle-Damgård iteration that re-keys at every step. This does not have an impact on performance with compression functions such as `sha1` and `sha256`, but the situation would be quite different with block-cipher-like primitives equipped with heavy key-schedule algorithms.

We remark that our construction has no restriction on the key space, though so far we have assumed $k \in \{0,1\}^n$. In fact, our construction works with any finite PRF $f_k : \{0,1\}^b \to \{0,1\}^n$ with $k \in K$, where K can be an arbitrary type of key space, as long as f is a secure PRF. Hence using a key k shorter than n bits speeds up performance (*i.e.*, each invocation to f processes more bits of a message). This sort of situation may occur when the desired key length does not match the value n of a compression function in hand.

7 Open Problems

Case $b < 2n$ and Block-Cipher-Based Construction. Our construction requires that the underlying compression function $f : \{0,1\}^{n+b} \to \{0,1\}^n$ should meet the condition $b \geq 2n$. We leave it as an open problem whether we can construct a mode of operation, meeting our goals, with a compression function f with $b < 2n$. Since we utilize this condition essentially in two different places, our method does not seem to be feasible with such compression functions. In particular, the last process with $\bar{f}_{123}$ may be constructed by methods such as [12–14], but using the condition in tweaking f seems to face a hard problem.

A possibly more challenging problem is to construct a mode of operation using an f_k with $b = n$ and each f_k being a permutation, rather than a function (*i.e.*, a block cipher). This introduces the difficulty in handling the PRP $\leftrightarrow$ PRF Switching Lemma that causes the birthday security degradation.

Parallelizable Construction. Our construction is inherently serial, and thus not parallelizable. Parallelizability is one of the desirable properties in constructing a mode of operation.

Recall that PMAC [18] is a mode of operation for message authentication, which is fully parallelizable. Although usually constructed of a block cipher, PMAC can be based on a compression function $f : \{0,1\}^{n+b} \to \{0,1\}^n$ meeting the condition $b \geq 2n$. We can then modify such PMAC as "multilaned" in the ways described in [9]. This would yield a parallelizable construction (which would resist birthday attacks). The only problem with this construction is that it is not truly one-pass. We leave it as an open problem whether we can construct a mode of operation that enjoys all the seven properties in our construction as well as parallelizability.

Reducing the State Size. Our mode is based on three data flows, summing up to $b+n+n = b+2n$ bits of state size. This is larger than $2n$, the number of bits we expect to be necessary to preclude birthday attacks. It is an interesting problem to see how many of $b+2n$ bits we can reduce down to $2n$ bits with a new construction in future work.

8 Concluding Remarks

Remarks on Checksum Construction. The idea of message checksum and that of intermediate-value checksum appear in various scenarios, including CBC with Checksum [23, 24], 3GPP $f9$ [25] and O-NMAC [26]. The same techniques are also used in the context of keyless hash functions, the purpose being, among other things, to preclude multi-collision attacks [27]. However, many of these hash functions are broken subsequently after their introduction [28, 29].

On the other hand, checksum techniques are proven to be effective (among other things) for extending a distribution property of a compression function to the whole hash function [30]. Our construction presents another positive application of the techniques—providing a secure PRF without the birthday barrier.

Remarks on Masking Technique. The masking technique used in the present work might be contrasted to that in constructing target-collision-resistant (TCR) hash functions [31]. The difference lies in the number of necessary "randomness." In the case of TCR hash functions the construction requires fresh masks as many as logarithmic of the message length (for each message), whereas in our case all the masks are derived from a single mask (which is also derived from a single key) for all messages. Having or not having a "secret" key seems to be essential to making the difference here.

Acknowledgments. The author would like to thank anonymous FSE 2008 reviewers for their valuable comments. In particular, the author wishes to express his appreciation to one of the referees for pointing out an incomplete description of the methods for incrementing masks in an earlier version of the paper. The author is grateful to Kazumaro Aoki for having discussions to fix the problem.

The author received useful feedback from the participants in the FSE 2008 Workshop. Special thanks go to Tetsu Iwata, Kaisa Nyberg, Taizo Shirai and Thomas Shrimpton.

References

1. Bellare, M., Canetti, R., Krawczyk, H.: Keying hash functions for message authentication. In Koblitz, N., ed.: CRYPTO 1996. Volume 1109 of LNCS., Springer (1996) 1–15
2. Bellare, M., Kilian, J., Rogaway, P.: The security of cipher block chaining. In Desmedt, Y., ed.: CRYPTO 1994. Volume 839 of LNCS., Springer (1994) 341–358
3. Preneel, B., van Oorschot, P.C.: MDx-MAC and building fast MACs from hash functions. In Coppersmith, D., ed.: CRYPTO 1995. Volume 963 of LNCS., Springer (1995) 1–14
4. Preneel, B., van Oorschot, P.C.: On the security of iterated message authentication codes. IEEE Transactions on Information Theory **45**(1) (1999) 188–199
5. Bellare, M., Guérin, R., Rogaway, P.: XOR MACs: New methods for message authentication using finite pseudorandom functions. In Coppersmith, D., ed.: CRYPTO 1995. Volume 963 of LNCS., Springer (1995) 15–28

6. Jaulmes, É., Joux, A., Valette, F.: On the security of randomized CBC-MAC beyond the birthday paradox limit: A new construction. In Daemen, J., Rijmen, V., eds.: FSE 2002. Volume 2365 of LNCS., Springer (2002) 237–251
7. den Boer, B., Rompay, B.V., Preneel, B., Vandewalle, J.: New (two-track-)MAC based on the two trails of RIPEMD. In Vaudenay, S., Youssef, A.M., eds.: SAC 2001. Volume 2259 of LNCS., Springer (2001) 314–324
8. Lucks, S.: A failure-friendly design principle for hash functions. In Roy, B.K., ed.: ASIACRYPT 2005. Volume 3788 of LNCS., Springer (2005) 474–494
9. Yasuda, K.: Multilane HMAC—Security beyond the birthday limit. In Srinathan, K., Rangan, C.P., Yung, M., eds.: INDOCRYPT 2007. Volume 4859 of LNCS., Springer (2007) 18–32
10. Black, J., Halevi, S., Krawczyk, H., Krovetz, T., Rogaway, P.: UMAC: Fast and secure message authentication. In Wiener, M.J., ed.: CRYPTO 1999. Volume 1666 of LNCS., Springer (1999) 216–233
11. Bellare, M., Goldreich, O., Krawczyk, H.: Stateless evaluation of pseudorandom functions: Security beyond the birthday barrier. In Wiener, M.J., ed.: CRYPTO 1999. Volume 1666 of LNCS., Springer (1999) 270–287
12. Aiello, W., Venkatesan, R.: Foiling birthday attacks in length-doubling transformations – Benes: A non-reversible alternative to Feistel. In Maurer, U.M., ed.: EUROCRYPT 1996. Volume 1070 of LNCS., Springer (1996) 307–320
13. Patarin, J.: Improved security bounds for pseudorandom permutations. In: ACM Conference on Computer and Communications Security. (1997) 142–150
14. Patarin, J.: About Feistel schemes with six (or more) rounds. In Vaudenay, S., ed.: FSE 1998. Volume 1372 of LNCS., Springer (1998) 103–121
15. Lucks, S.: The sum of PRPs is a secure PRF. In Preneel, B., ed.: EUROCRYPT 2000. Volume 1807 of LNCS., Springer (2000) 470–484
16. Maurer, U.M.: Indistinguishability of random systems. In Knudsen, L.R., ed.: EUROCRYPT 2002. Volume 2332 of LNCS., Springer (2002) 110–132
17. Gligor, V.D., Donescu, P.: Fast encryption and authentication: XCBC encryption and XECB authentication modes. In Matsui, M., ed.: FSE 2001. Volume 2355 of LNCS., Springer (2001) 92–108
18. Black, J., Rogaway, P.: A block-cipher mode of operation for parallelizable message authentication. In Knudsen, L.R., ed.: EUROCRYPT 2002. Volume 2332 of LNCS., Springer (2002) 384–397
19. Jutla, C.S.: Encryption modes with almost free message integrity. In Pfitzmann, B., ed.: EUROCRYPT 2001. Volume 2045 of LNCS., Springer (2001) 529–544
20. Rogaway, P.: Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In Lee, P.J., ed.: ASIACRYPT 2004. Volume 3329 of LNCS., Springer (2004) 16–31
21. Brillhart, J., Lehmer, D.H., Selfridge, J.L., Tuckerman, B., Wagstaff, Jr., S.S.: Factorizations of $b^n \pm 1$, $b = 2, 3, 5, 6, 7, 10, 11, 12$ Up to High Powers. 3 edn. Volume 22 of Contemporary Mathematics. AMS (2002)
22. Dodis, Y., Pietrzak, K.: Improving the security of MACs via randomized message preprocessing. In Biryukov, A., ed.: FSE 2007. Volume 4593 of LNCS., Springer (2007) 414–433
23. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press (1996)
24. Schneier, B.: Applied Cryptography. 2nd edn. John Wiley & Sons (1996)
25. 3GPP: Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 1: $f8$ and $f9$ Specification. 3.1.1 edn. (2001)

26. Gauravaram, P., Millan, W., Nieto, J.G., Dawson, E.: 3C – A provably secure pseudorandom function and message authentication code. A new mode of operation for cryptographic hash function. Cryptology ePrint Archive (2005) Report 2005/390.
27. Gauravaram, P., Millan, W., Dawson, E., Viswanathan, K.: Constructing secure hash functions by enhancing Merkle-Damgård construction. In Batten, L.M., Safavi-Naini, R., eds.: ACISP 2006. Volume 4058 of LNCS., Springer (2006) 407–420
28. Joscák, D., Tuma, J.: Multi-block collisions in hash functions based on 3C and 3C+ enhancements of the Merkle-Damgård construction. In Rhee, M.S., Lee, B., eds.: ICISC 2006. Volume 4296 of LNCS., Springer (2006) 257–266
29. Gauravaram, P., Kelsey, J.: Cryptanalysis of a class of cryptographic hash functions. Cryptology ePrint Archive (2007) Report 2007/277.
30. Lei, D., Li, C.: Extended multi-property-preserving and ECM-construction. In Srinathan, K., Rangan, C.P., Yung, M., eds.: INDOCRYPT 2007. Volume 4859 of LNCS., Springer (2007) 361–372
31. Shoup, V.: A composition theorem for universal one-way hash functions. In Preneel, B., ed.: EUROCRYPT 2000. Volume 1807 of LNCS., Springer (2000) 445–452