A New Variant of PMAC: Beyond the Birthday Bound

Kan Yasuda

NTT Information Sharing Platform Laboratories,
NTT Corporation, Japan
yasuda.kan@lab.ntt.co.jp

Abstract. We propose a PMAC-type mode of operation that can be used as a highly secure MAC (Message Authentication Code) or PRF (Pseudo-Random Function). Our scheme is based on the assumption that the underlying n-bit blockcipher is a pseudo-random permutation. Our construction, which we call `PMAC_Plus`, involves extensive modification to PMAC, requiring three blockcipher keys. The `PMAC_Plus` algorithm is a first rate-1 (*i.e.*, one blockcipher call per n-bit message block) blockcipher-based MAC secure against $O(2^{2n/3})$ queries, increasing the $O(2^{n/2})$ security of PMAC at a low additional cost. Our analysis uses some of the security-proof techniques developed with the sum construction (Eurocrypt 2000) and with the encrypted-CBC sum construction (CT-RSA 2010).

Keywords: 64-bit blockcipher, PRP, sum construction, CBC vs. PMAC, game-playing technique.

1 Introduction

MACs (Message Authentication Codes) are frequently realized by making iterative use of blockciphers. They are called *blockcipher-based MACs* and specified in a large number of standardized documents including ISO 9797-1 [13].

The majority of blockcipher-based MACs iterate a blockcipher in the so-called *CBC* (*Cipher Block Chaining*) style [3,19,15], by xor-ing the current message block with the previous chaining value and then inputting the xor-ed result into the next blockcipher call. CBC-type MACs have a history of continuous updates, whose purpose is mainly to reduce the number of keys and to increase efficiency in the last message block. CMAC [18] *a.k.a.* OMAC [11], the first 1-key CBC MAC derived from XCBC [7], can be regarded as an outcome of such evolution.

On the other hand, PMAC (Parallelizable MAC) [8] is a distinctive, parallelizable blockcipher-based MAC. Its internal structure is completely different from CBC iteration, which is inherently sequential and not parallelizable. If sequentially implemented, then PMAC becomes slightly slower than CBC MACs, because PMAC requires an extra operation at every blockcipher call. The extra operation is typically a constant multiplication in the finite field, which is fast but still slower than a simple xor operation used by CBC MACs. However,

P. Rogaway (Ed.): CRYPTO 2011, LNCS 6841, pp. 596–609, 2011.

under parallel implementation, PMAC can possibly outperform CBC MACs significantly.

We believe it is worth re-evaluating PMAC-type constructions under the current trend of "parallelizable" (pipeline, superscalar, vector, and multi-core) CPUs (*e.g.*, see [14] for a state-of-the-art implementation of AES in this direction). It seems that today PMAC is still not as widespread as CBC MACs, perhaps because most of the computational environments commercially available so far have been increasing the clock rate and hence the speed of sequential operations.

In this paper we look at another advantageous aspect of PMAC-type constructions. That is its proof of security. The parallel construction has a structure easy to analyze and to obtain better bounds. Intuitively, the difference between PMAC-type and CBC-type iterations lies in the "long-message attacks" noted by Preneel and Oorschot [20]: Suppose, for the moment, that we iterate an n-bit function rather than permutation. Then two messages $M00\cdots0$ and $M'00\cdots0$ would collide at some point of the CBC iteration if they are long enough—say 2^n blocks—and the collision would propagate through the output of the MAC algorithm. An event of this sort does not happen to PMAC. This appealing aspect of PMAC-type constructions is already pointed out, though implicitly, by Minematsu [17] in obtaining an $O(\ell q^2/2^n)$-type bound for PMAC (where ℓ is the maximum length of a message, q the maximum number of queries and n the block size). Recall that obtaining such a bound for CBC MACs seems more troublesome [4].

Besides PMAC, there have been a few proposals of parallelizable MACs, for which a blockcipher can be used. These include XOR MAC by Bellare *et al.* [2] and PCS by Bernstein [6]. There have been also some improvements or alternatives to PMAC. These include PMAC1 by Rogaway [21] and iPMAC by Sarkar [22].

Birthday-Bound Problems and Our Contributions. We take the advantage of the provable-security aspect of PMAC-type constructions in solving the so-called *birthday-bound problem* of iterative MACs [20]. That is, a MAC construction having an n-bit size of intermediate values cannot be secure against more than $O(2^{n/2})$ queries—a forgery becomes possible after so many queries. Typical MAC constructions iterating an n-bit blockcipher suffer from this problem.

This is a sever problem particularly for 64-bit blockciphers. Not to mention the fact that the legacy Triple-DES is still widely used (especially in financial services), we also have new 64-bit blockciphers such as HIGHT [10] and PRESENT [9], possibly due to industrial demands for "lightweight" algorithms. The birthday bound may not be a serious problem for 128-bit blockciphers at the current moment. However, it contributes not only to existing 64-bit blockciphers but also to the longevity of 128-bit blockciphers in future use to construct an efficient MAC mode which is free of the birthday-bound problem.

Table 1. Summary of our result and comparison with previous constructions

	Rate	# of keys	Parallelizable?	Security bound	Ref.
Alg. 6 of ISO 9797-1	1/2	6		$O(\ell^4 q^3/2^{2n})$ or restricted $O(\ell^3 q^3/2^{2n})$	[13]
SUM-ECBC	1/2	4		$O(\ell^4 q^3/2^{2n})$ or restricted $O(\ell^3 q^3/2^{2n})$	[23]
`PMAC_Plus`	1	3	✓	$O(\ell^3 q^3/2^{2n} + \ell q/2^n)$	This work

The first attempt to solve this problem was made in ISO 9797-1 (without proofs of security).[1] Unfortunately, Algorithm 4 of ISO 9797-1 was attacked and shown insecure ("insecure" meaning secure only up to the $O(2^{n/2})$ birthday bound) by Joux *et al.* [12]. Algorithm 6 of ISO 9797-1, on the other hand, has been proven secure against $O(2^{2n/3})$ queries [23]. The $O(2^{2n/3})$ bound holds only with certain restrictions on the message length. The work [23] has also presented SUM-ECBC, the sum (xor) of two encrypted CBC MACs. SUM-ECBC has become a 4-key rate-1/2 (meaning two encryptions to process n-bits) blockcipher-based MAC having the same security bound as Algorithm 6 of ISO 9797-1. We improve over these MAC constructions by utilizing a PMAC-type iteration. We propose a new MAC algorithm `PMAC_Plus`. `PMAC_Plus` is an extensively modified version of PMAC. `PMAC_Plus` remains rate-1 but operates with three blockcipher keys. `PMAC_Plus` has an $O(2^{2n/3})$ bound without such a restriction on the message length as existed with the security bound for the previous constructions.[2] Table 1 summarizes our result.

Organization. Section 2 provides necessary background. In Section 3 we define our algorithm `PMAC_Plus` and state its security result. The entire Section 4 is devoted to proofs of security. The paper ends with brief discussion in Section 5.

2 Preliminaries

Symbols and Notation. We fix a block size n, which is typically 64 or 128. We write $\mathrm{Perm}(n)$ for the set of permutations $P : \{0,1\}^n \to \{0,1\}^n$. We also fix a key space $\mathscr{K}$. Usually $\mathscr{K} = \{0,1\}^\kappa$, where $\kappa = 80$, 128, 192 or 256. A blockcipher E is a function $E : \mathscr{K} \times \{0,1\}^n \to \{0,1\}^n$ such that for each key $K \in \mathscr{K}$ we have $E_K \in \mathrm{Perm}(n)$, where $E_K : \{0,1\}^n \to \{0,1\}^n$ is defined as $E_K(X) := E(K,X)$. We can then write E_K^{-1} for the inverse permutation.

The set $\{0,1\}^n$ can be regarded as a set of integers $\{0,1,\ldots,2^n-1\}$. This can be done by converting an n-bit string $a_{n-1}\cdots a_1 a_0 \in \{0,1\}^n$ to an

[1] See, for example, [1] for another direction (using random coins) of treating birthday-bound problems.

[2] Our new bound would become vacuous for very long messages, say $2^{2n/3}$ blocks.

integer $a_{n-1}2^{n-1} + \cdots + a_1 2 + a_0$, where multiplication and addition are integer arithmetic.

Let $GF(2^n)$ denote the finite field with 2^n elements. We regard $\{0,1\}^n$ as $GF(2^n)$. That is, we identify an n-bit string $a_{n-1}\cdots a_1 a_0 \in \{0,1\}^n$ with a formal polynomial $a_{n-1}\mathtt{x}^{n-1} + \cdots a_1\mathtt{x} + 1 \in GF(2)[\mathtt{x}]$. To do so we need to fix an irreducible polynomial $a(\mathtt{x}) = \mathtt{x}^n + a_{n-1}\mathtt{x}^{n-1} + \cdots + a_1\mathtt{x} + a_0 \in GF(2)[\mathtt{x}]$. We sometimes write $\oplus$ and $\odot$ to emphasize addition and multiplication in the field, respectively. So for example we have $2 \oplus 3 = \mathtt{x} + (\mathtt{x}+1) = 1$ and $3 \odot 3 = (\mathtt{x}+1)^2 = \mathtt{x}^2 + 1 = 5$ if $n \geq 3$.

We choose irreducible polynomials $a(\mathtt{x}) = \mathtt{x}^{64} + \mathtt{x}^4 + \mathtt{x}^3 + \mathtt{x} + 1$ for $n = 64$ and $a(\mathtt{x}) = \mathtt{x}^{128} + \mathtt{x}^7 + \mathtt{x}^2 + \mathtt{x} + 1$ for $n = 128$. These are actually *primitive* polynomials, meaning the element $2 = \mathtt{x}$ generates the entire multiplicative group $GF(2^n)^*$ of order $2^n - 1$.

Security Notions. An adversary $\mathscr{A}$ is an oracle machine. $\mathscr{A}$ has access to its oracle $O(\cdot)$ and, after interaction with the oracle, outputs a bit, 1 or 0. We write $\mathscr{A}^{O(\cdot)} = 1$ to denote the event that $\mathscr{A}$ outputs 1 after interacting with $O(\cdot)$. We measure the resources of $\mathscr{A}$ in terms of time and query complexities. We fix a model of computation and a method of encoding. The query complexity is measured in terms of the number of queries (usually denoted q) and also in terms of the maximum length of each query (denoted ℓ). The length of a query is measured in blocks (n bits).

We say that (informally) a block cipher E is a (*secure*) *pseudo-random permutation* (*PRP*) if it is indistinguishable from a random permutation $P \xleftarrow{\$} \mathrm{Perm}(n)$, where $\xleftarrow{\$}$ means uniformly random sampling. Specifically, we consider the advantage function

$$\mathbf{Adv}^{\mathrm{prp}}_E(\mathscr{A}) := \Pr\big[\mathscr{A}^{E_K(\cdot)} = 1; K \xleftarrow{\$} \mathscr{K}\big] - \Pr\big[\mathscr{A}^{P(\cdot)} = 1; P \xleftarrow{\$} \mathrm{Perm}(n)\big],$$

and if this quantity is "small enough" for a class of adversaries, then we say that E is a PRP. Here note that the probabilities are defined over internal coin tosses of $\mathscr{A}$, if any, as well as over the choices of K and P. We further define $\mathbf{Adv}^{\mathrm{prp}}_E(t,q) := \max_{\mathscr{A}} \mathbf{Adv}^{\mathrm{prp}}_E(\mathscr{A})$, where the max runs over all adversaries $\mathscr{A}$ whose running time is at most t, making at most q queries to its oracle.

With abuse of notation let $\{0,1\}^*$ denote the set of finite bit strings whose length is at most ℓq blocks. Let $\mathrm{Func}(*,n)$ denote the set of functions $G : \{0,1\}^* \to \{0,1\}^n$. Our goal is to construct a *pseudo-random function* (*PRF*) $F_K : \{0,1\}^* \to \{0,1\}^n$ having keys $K \in \mathscr{K}'$ (and preferably $\mathscr{K}'$ is not much larger than $\mathscr{K}$). Recall that any PRF can be used as a secure MAC. We say that F is a secure PRF if it is indistinguishable from a random function $G \xleftarrow{\$} \mathrm{Func}(*,n)$, or more precisely, we define

$$\mathbf{Adv}^{\mathrm{prf}}_F(\mathscr{A}) := \Pr\big[\mathscr{A}^{F_K(\cdot)} = 1; K \xleftarrow{\$} \mathscr{K}'\big] - \Pr\big[\mathscr{A}^{G(\cdot)} = 1; G \xleftarrow{\$} \mathrm{Func}(*,n)\big].$$

We also define $\mathbf{Adv}^{\mathrm{prf}}_F(t,q,\ell)$ to be the maximum advantage running over all adversaries $\mathscr{A}$ whose running time is at most t, making at most q queries to its oracle, each query being at most ℓ blocks.

Game-Playing Techniques. Our proofs of security largely depend on the so-called game-playing techniques [5]. In particular, we perform lazy sampling for a random permutation $P \xleftarrow{\$} \mathrm{Perm}(n)$. That is, P is initially set everywhere undefined, and when a value $P(X)$ becomes necessary at some point in the game, a corresponding range point Y is randomly sampled as $Y \xleftarrow{\$} \{0,1\}^n$, so that we have $P(X) = Y$. We implicitly maintain two sets, $\mathtt{Dom}\, P$ and $\mathtt{Ran}\, P$, which keep the record of already-defined domain points and that of range points, respectively.

3 PMAC_Plus: Specification and Security

There exist different versions of PMAC. The version that we use here is based on the discrete-log-based LFSR (linear feedback shift register) developed by Rogaway [21]. Our MAC construction PMAC_Plus is defined in Algorithm 1. It calls a subroutine Internal, which is described in Algorithm 2.

Algorithm 1. $\mathtt{PMAC\_Plus}[E_{K_1}, E_{K_2}, E_{K_3}](M)$

$(\Sigma, \Theta) \leftarrow \mathtt{Internal}[E_{K_1}](M)$
$T \leftarrow E_{K_2}(\Sigma) \oplus E_{K_3}(\Theta)$
return T

Algorithm 2. Subroutine $\mathtt{Internal}[E_K](M)$

$\Delta_0 \leftarrow E_K(0)$
$\Delta_1 \leftarrow E_K(1)$
$M \leftarrow M \| 10^*$
Partition M into $M[1] \,\|\, \cdots \,\|\, M[m]$
for $i = 1$ to m **do**
  $X[i] \leftarrow M[i] \oplus 2^i \cdot \Delta_0 \oplus 2^{2i} \cdot \Delta_1$
  $Y_i \leftarrow E_K\big(X[i]\big)$
end for
$\Sigma \leftarrow Y_1 \oplus \cdots \oplus Y_m$
$\Theta \leftarrow 2^{m-1} \cdot Y_1 \oplus 2^{m-2} \cdot Y_2 \oplus \cdots \oplus Y_m$
return (Σ, Θ)

In Algorithm 2, by "$M[m] \,\|\, 10^*$" we mean appending a bit 1 and then an appropriate number of bits 0 so that the bit length of $M[m] \,\|\, 10^*$ becomes n. By "partition M" we mean $M = M[1] \,\|\, \cdots \,\|\, M[m]$ so that $\big|M[1]\big| = \big|M[m]\big| = n$. See Fig. 1 for a pictorial representation of our construction PMAC_Plus.

The security of our PMAC_Plus construction is as follows:

Theorem 1 (Security of PMAC_Plus). *We have*

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathtt{PMAC\_Plus}}(t, q, \ell) \le \frac{27\ell^3 q^3}{2^{2n}} + \frac{3\ell q}{2^n} + 3\mathbf{Adv}^{\mathrm{prp}}_{E}(t', \ell q + 2),$$

where t' is about t plus a time complexity necessary to compute E for $\ell q + 2q + 2$ times.

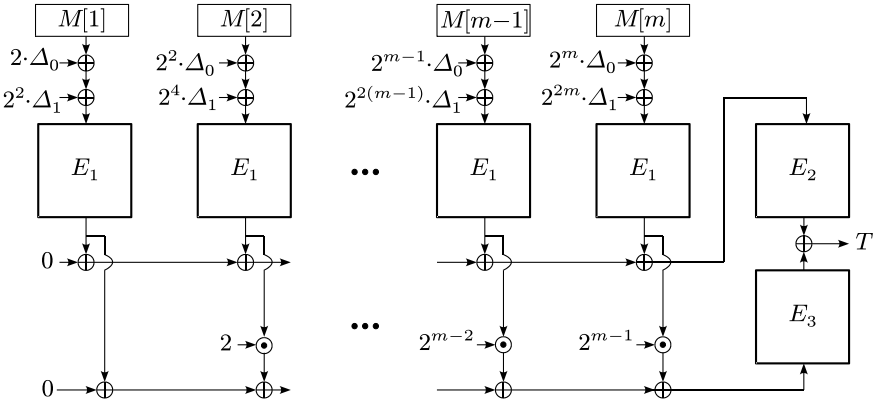

Fig. 1. Our PMAC_Plus algorithm using three blockcipher keys K_1, K_2 and K_3, where $E_1 = E_{K_1}$, $E_2 = E_{K_2}$, $E_3 = E_{K_3}$, $\Delta_0 = E_1(0)$ and $\Delta_1 = E_1(1)$

The additional term of $3\mathbf{Adv}^{\mathrm{prp}}_{E}(t', \ell q+2)$ comes from the standard argument of replacing actual blockciphers $E_1 = E_{K_1}$, $E_2 = E_{K_2}$ and $E_3 = E_{K_3}$ with random permutations P_1, P_2 and P_3, respectively.

4 Proofs of Security

We now prove that $\mathtt{PMAC\_Plus}[P_1, P_2, P_3]$ is an $O(2^{2n/3})$-secure PRF given random permutations P_1, P_2 and P_3.

4.1 Basic Ideas

Let $\mathscr{A}$ be an adversary that makes at most q queries, each query being at most ℓ blocks. The goal of $\mathscr{A}$ is to distinguish between the $\mathtt{PMAC\_Plus}[P_1, P_2, P_3](\cdot)$ oracle and a random function $G : \{0,1\}^* \to \{0,1\}^n$. We consider the game described in Fig. 2. In games, given a set $S \subset \{0,1\}^n$, we write for its complement $\overline{S} := \{0,1\}^n \setminus S$. Codes of subroutines are given in Figures 3, 4, 5 and 6. We observe that the game with single-boxed statements coincides with a random function G, whereas the game with double-boxed statements is exactly $\mathtt{PMAC\_Plus}[P_1, P_2, P_3]$. These two algorithms differ only when Bad events occur. Therefore, by the fundamental lemma of game-playing [5], we have

$$\Pr\left[\mathscr{A}^{\mathtt{PMAC\_Plus}[P_1,P_2,P_3](\cdot)} = 1\right] - \Pr\left[\mathscr{A}^{G(\cdot)} = 1\right] \le \Pr\left[\mathscr{A} \text{ sets } \mathtt{Bad}\right],$$

where the bad events are classified into five "winning" events as

$$\begin{aligned}
&\Pr[\mathscr{A} \text{ sets } \mathtt{Bad}] \\
\le\ &\Pr[\mathscr{A} \text{ sets } \mathtt{Zero}^*, \mathtt{Unfair}^*, \mathtt{UpLow}^*, \mathtt{LowUp}^* \text{ or } \mathtt{Coll}^*] \\
\le\ &\Pr[\mathtt{Zero}^*] + \Pr[\mathtt{Unfair}^*] + \Pr[\mathtt{UpLow}^*] + \Pr[\mathtt{LowUp}^*] + \Pr[\mathtt{Coll}^*].
\end{aligned}$$

```
 1: Δ_0, Δ_1 ←$ {0,1}^n          // sampling P_1(0) and P_1(1)
 2: if Δ_0 = 0 or Δ_1 = 0 then
 3:    Zero* ← true
 4:    Bad ← true  [Δ_* ←$ {0}-bar]
 5: end if
 6:
 7: upon a query M do
 8: (Σ, Θ) ← Internal[P_1](M)          // lazy sampling for P_1
 9: if Σ ∉ Dom P_2 and Θ ∉ Dom P_3 then
10:    go to Case A          // lazy sampling for P_2 and P_3
11: end if
12: if Σ ∈ Dom P_2 and Θ ∉ Dom P_3 then
13:    go to Case B          // lazy sampling for P_3
14: end if
15: if Σ ∉ Dom P_2 and Θ ∈ Dom P_3 then
16:    go to Case C          // lazy sampling for P_2
17: end if
18: if Σ ∈ Dom P_2 and Θ ∈ Dom P_3 then
19:    go to Case D          // a bad event
20: end if
21: return T
```

Fig. 2. Main game

The first term can be easily bounded; it remains to bound the last four terms. These correspond to Cases A, B, C and D, respectively, and they are described as subroutines. Up to this point we essentially follow the same framework as the proof of SUM-ECBC [23].

```
1: Choose a fair set R ⊂ Ran(P_2)-bar × Ran(P_3)-bar
2: (U, L) ←$ Ran(P_2)-bar × Ran(P_3)-bar
3: if (U, L) ∉ R then
4:    if ¬Bad then
5:       Unfair* ← true
6:    end if
7:    Bad ← true  [(U, L) ←$ R]
8: end if
9: T ← U ⊕ L
```

Fig. 3. Code for Case A

4.2 Bounding the Probability of Each Winning Event

Case A: Unfair*. This case can be essentially handled by the technique of fair sets developed by Lucks [16]. The proof is exactly the same as the case of SUM-ECBC [23].

1: $U \leftarrow P_2(\Sigma)$
2: $L \xleftarrow{\$} \{0,1\}^n$
3: **if** $L \in \mathtt{Ran}\, P_3$ **then**
4: **if** $\neg$`Bad` **then**
5: `UpLow`$^* \leftarrow$ **true**
6: **end if**
7: `Bad` $\leftarrow$ **true** $\boxed{L \xleftarrow{\$} \overline{\mathtt{Ran}\, P_3}}$
8: **end if**
9: $T \leftarrow U \oplus L$

Fig. 4. Code for Case B

1: $L \leftarrow P_3(\Theta)$
2: $U \xleftarrow{\$} \{0,1\}^n$
3: **if** $U \in \mathtt{Ran}\, P_2$ **then**
4: **if** $\neg$`Bad` **then**
5: `LowUp`$^* \leftarrow$ **true**
6: **end if**
7: `Bad` $\leftarrow$ **true** $\boxed{U \xleftarrow{\$} \overline{\mathtt{Ran}\, P_2}}$
8: **end if**
9: $T \leftarrow U \oplus L$

Fig. 5. Code for Case C

1: **if** $\neg$`Bad` **then**
2: `Coll`$^* \leftarrow$ **true**
3: **end if**
4: `Bad` $\leftarrow$ **true** $\boxed{T \xleftarrow{\$} \{0,1\}^n}$ $\boxed{\boxed{T \leftarrow P_2(\Sigma) \oplus P_3(\Theta)}}$

Fig. 6. Code for Case D

Lemma 1 (Case A). *We have*

$$\Pr[\mathscr{A} \text{ sets } \mathtt{Unfair}^*] \le \frac{2q^3}{2^{2n}},$$

for $q \le 2^{n-1}$.

Proof. The proof is given in [23], but for the sake of completeness we give one in Appendix. □

Case B: `UpLow`*. To treat this case we need the following:

Lemma 2. *For a pair of messages* (M, M') *such that* $M \neq M'$, *each being at most* ℓ *blocks, we have*

$$\Pr\big[\Sigma = \Sigma'; P \xleftarrow{\$} \mathrm{Perm}(n)\big] \le \frac{8\ell}{2^n},$$

where $(\Sigma, \Theta) \leftarrow \mathtt{Internal}[P](M)$ *and* $(\Sigma', \Theta') \leftarrow \mathtt{Internal}[P](M')$.

Proof. Consider lazy sampling for P. First draw a range point $\Delta := P(0)$. We assume that $\Delta \neq 0$; the probability that $\Delta = 0$ occurs is $1/2^n$. We next assume that none of the input blocks $X[a]$ or $X'[a]$ is 0 or 1; the probability that $X[a] = 0, 1$ or $X'[a] = 0, 1$ occurs for some $a \le \ell$ is at most $4\ell/2^n$.

Now without loss of generality we assume that $|M| \ge |M'|$. Let m, m' be the number of blocks in M and in M', respectively. Observe that we must have $m \ge 2$ for the above probability to be non-trivial. So assume $m \ge 2$.

Determine an index $i \in \{1, \ldots, m-1\}$ as follows. If $m > m'$, then set $i := m$. If $m = m'$ and the last blocks are the only blocks that differ, then the above probability becomes vacuous. So in the case of $m = m'$, let $i \le m$ be the maximum index such that $M[i] \neq M'[i]$.

We focus on the input block $X[i]$. Assume that $X[i]$ does not appear in any of $X[a]$ $(1 \le a \le m-1)$ or $X'[a']$ $(1 \le a \le m'-1)$ except for itself. The probability that $X[i] = X[a]$ or $X[i] = X'[a']$ occurs is at most $\big((\ell-1)+\ell\big)/2^n = (2\ell-1)/2^n$.

Finally, consider the condition $\Sigma = \Sigma'$. Now resume the lazy sampling for P. Sample the point $P(X[i])$ as $Y_i \xleftarrow{\$} \overline{\mathtt{Ran}\, P}$ after finishing sampling all other points; the point $P(X[i])$ gets always sampled according to the way of choosing the index i. In such a scenario the probability that $\Sigma = \Sigma'$ holds is at most $1/\big|\overline{\mathtt{Ran}\, P}\big| \le 1/\big(2^n - 1 - (\ell-2) - (\ell-1)\big)/2^n \le 1/(2^n - 2\ell) \le 2/2^n$, assuming $\ell \le 2^{n-2}$ (otherwise the desired inequality would become meaningless).

We sum up each terms. Overall, the probability can be bounded as

$$\frac{1}{2^n} + \frac{4\ell}{2} + \frac{2\ell-1}{2^n} + \frac{2}{2^n} \le \frac{8\ell}{2^n},$$

as desired. □

Now let $M^{(1)}, \ldots, M^{(q)}$ denote $\mathscr{A}$'s queries. Then the probability that $\mathscr{A}$ sets the $\mathtt{UpLow}^*$ flag can be bounded as

$$\begin{aligned}
&\sum_{i=2}^{q} \Pr\big[\Sigma^{(i)} \in \mathtt{Dom}\, P_2 \wedge L^{(i)} \in \mathtt{Ran}\, P_3; P_1, P_2, P_3 \xleftarrow{\$} \mathrm{Perm}(n)\big] \\
\le &\sum_{i=2}^{q} \sum_{j=1}^{i-1} \Pr\big[(\Sigma^{(i)} = \Sigma^{(j)}); P_1 \xleftarrow{\$} \mathrm{Perm}(n)\big] \cdot \Pr\big[L^{(i)} \in \mathtt{Ran}\, P_3; L^{(i)} \xleftarrow{\$} \{0,1\}^n\big] \\
\le &\sum_{i=2}^{q} \sum_{j=1}^{i-1} \frac{8\ell}{2^n} \cdot \frac{\big|\mathtt{Ran}\, P_3\big|}{2^n} \le \sum_{i=2}^{q} \sum_{j=1}^{i-1} \frac{8\ell}{2^n} \cdot \frac{2q}{2^n} \le \frac{q^2}{2} \cdot \frac{8\ell}{2^n} \cdot \frac{2q}{2^n} = \frac{8\ell q^3}{2^{2n}},
\end{aligned}$$

where we wrote $(\Sigma^{(i)}, \Theta^{(i)}) := \mathtt{Internal}[P_1]\big(M^{(i)}\big)$ and $L^{(i)}$ the sampling of L at the i-the query.

Case C: $\mathtt{LowUp}^*$. For this case we need the following lemma. Then the computation is similar to Case B, and we obtain the same bound of $8\ell q^3/2^{2n}$.

Lemma 3. *For a pair of messages (M, M') such that $M \neq M'$, each being at most ℓ blocks, we have*

$$\Pr\big[\Theta = \Theta'; P \xleftarrow{\$} \mathrm{Perm}(n)\big] \le \frac{8\ell}{2^n},$$

where $(\Sigma, \Theta) \leftarrow \mathtt{Internal}[P](M)$ and $(\Sigma', \Theta') \leftarrow \mathtt{Internal}[P](M')$.

Proof. Similar to Lemma 2. □

Case D: `Coll*`**.** Since P_1 is independent from P_2 and from P_3, we may fix $\mathscr{A}$'s (distinct) queries and let $M^{(1)}, \ldots, M^{(q)}$ denote them. We would like to compute the probability that at the i-th query $M^{(i)}$ we get $\Sigma^{(i)} \in \mathtt{Dom}\, P_2$ and $\Theta^{(i)} \in \mathtt{Dom}\, P_3$. The event implies that there exist some earlier queries $M^{(j)}$ and $M^{(k)}$ (j and k may be equal) such that $\Sigma^{(j)} = \Sigma^{(i)}$ and $\Theta^{(k)} = \Theta^{(i)}$.

Before evaluating the probability

$$\Pr\big[(\Sigma^{(j)} = \Sigma^{(i)}) \wedge (\Theta^{(k)} = \Theta^{(i)}); P_1 \xleftarrow{\$} \mathrm{Perm}(n)\big],$$

we first exclude the case that $Y_a \xleftarrow{\$} \overline{\mathtt{Ran}\, P_1}$ becomes zero (*i.e.*, $Y_a = 0$) in sampling range points of P_1 for messages $M^{(1)}, \ldots, M^{(q)}$. The overall probability that this event occurs is at most $\ell q/2^n$.

We then consider the case when an "input collision" occurs among $X^{(i)}[a]$, $X^{(j)}[a]$, $X^{(k)}[a]$. By an "input collision" we mean an event $X^{(*)}[a] = X^{(*)}[a']$ for some $a \neq a'$. If input collisions occur at indices $a < b < c$ such that $X^{(*)}[a] = X^{(*)}[b] = X^{(*)}[c]$, then this system of equations would determine the values Δ_0, Δ_1, so the probability that this occurs is at most $\binom{3\ell}{3} \cdot 1/2^n \leq 5\ell^3/2^n$.

Suppose no 3-collision occurs. The probability that a 2-collision happens upon sampling $\Delta \xleftarrow{\$} \{0,1\}^n$ for a fix set of $M^{(i)}$, $M^{(j)}$ and $M^{(k)}$ is at most $\binom{3\ell}{2} \cdot 1/2^n \leq 4.5\ell^2/2^n$. Under the event of an input (2-)collision, we focus on the equation $\Theta^{(i)} = \Theta^{(k)}$. We show that this provides a non-trivial equation for some random variable $Y_a^{(*)}$. Without loss of generality assume, for the moment, that $m^{(i)} \leq m^{(k)}$ where these are the number of blocks in the message $M^{(i)}$ and that in $M^{(k)}$, respectively. Let $\alpha \in \{1, \ldots, m^{(i)}\}$ be the largest index such that $M^{(i)}[\alpha] \neq M^{(k)}[\alpha]$, if such an index exists. Then we have $X^{(i)}[\alpha] \neq X^{(k)}[\alpha]$, so at least one of these input values is non-zero, which means that either $Y_\alpha^{(i)} = P(X^{(i)}[\alpha])$ or $Y_\alpha^{(k)} = P(X^{(k)}[\alpha])$ gets sampled. For that random variable the equation $\Theta^{(i)} = \Theta^{(k)}$ is non-trivial. On the other hand, if no such index α exists, then it means that $M^{(i)}[a] = M^{(k)}[a]$ for all $a \in \{1, \ldots, m^{(i)}\}$ and $m^{(i)}+1 \leq m^{(k)}$ (In such a case we say that $M^{(i)}$ is "contained" in $M^{(k)}$). Consider the values $X^{(k)}[m^{(i)}+1], \ldots, X^{(k)}[m^{(k)}]$. It cannot be the case that all of these input values are the same, as it would imply a zero value in the range of P_1. Therefore, we have $m^{(i)} + 2 \leq m^{(k)}$, and let $\beta \in \{m^{(i)}+1, \ldots, m^{(k)}\}$ be the largest index such that $X^{(k)}[\beta] \neq 0$. Then we see that $Y_\beta^{(k)} = P(X^{(k)}[\beta])$ gets always sampled. Therefore, the probability that this equation is satisfied is at most $1/\big|\mathtt{Ran}\, P_1\big| \leq 1/(2^n - 2\ell) \leq 2/2^n$ assuming $\ell \leq 2^{n-2}$.

We can now bound the probability that, for a fix set of $M^{(i)}$, $M^{(j)}$ and $M^{(k)}$, an input collision occurs and the equation $\Theta^{(i)} = \Theta^{(k)}$ holds. It would be at most $4.5\ell^2/2^n \cdot 2 \cdot /2^n \leq 9\ell^2/2^{2n}$.

Consider the case when no input collision occurs among $X^{(i)}[a]$, $X^{(j)}[a]$, $X^{(k)}[a]$. We start with the case $j = k$. Without loss of generality assume, for the moment, that $m^{(i)} \leq m^{(j)}$ where these are the number of blocks in the message $M^{(i)}$ and that in $M^{(j)}$, respectively. It can be directly verified

that we can choose indices $\alpha < \beta$ such that (a) $\beta \le m^{(i)}$ and $M^{(i)}[\alpha] \neq M^{(j)}[\alpha]$ and $M^{(i)}[\beta] \neq M^{(j)}[\beta]$, (b) $\alpha \le m^{(i)}$ and $M^{(i)}[\alpha] \neq M^{(j)}[\alpha]$ and $m^{(i)} + 1 \le \beta \le m^{(j)}$, or (c) $m^{(i)} + 1 \le \alpha < \beta \le m^{(j)}$. In any case, the system of equations $\Sigma^{(j)} = \Sigma^{(i)}$ and $\Theta^{(j)} = \Theta^{(i)}$ provides a unique solution set for the random variables $Y_\alpha^{(*)}$, $Y_\beta^{(*)}$. So the probability of this event is at most $1/\left|\mathtt{Ran}\, P_1\right| \cdot 1/\left|\mathtt{Ran}\, P_1\right| \le 1/(2^n - 2\ell)^2 \le 4/2^{2n}$, assuming $\ell \le 2^{n-2}$.

It remains to treat the case $j \neq k$. We consider the cases (a) $M^{(i)}$ is contained in $M^{(j)}$, (b) $M^{(j)}$ is contained in $M^{(i)}$, (c) $M^{(i)}$ is contained in $M^{(k)}$, or (d) $M^{(k)}$ is contained in $M^{(i)}$. For example, we discuss case (a). We can choose indices α, β such that (a1) $m^{(k)}+1 \le \alpha \le m^{(i)}$ and $m^{(i)}+1 \le \beta \le m^{(j)}$, (a2) $m^{(i)}+1 \le \alpha \le m^{(k)}$ and $m^{(k)}+1 \le \beta \le m^{(j)}$, (a3) $m^{(i)}+1 \le \alpha \le m^{(j)}$ and $m^{(j)}+1 \le \beta \le m^{(k)}$, (a4) $m^{(i)} + 1 \le \alpha \le m^{(j)}$ and $1 \le \beta \le m^{(i)}$ and $M^{(i)}[\beta] = M^{(j)}[\beta] \neq M^{(k)}[\beta]$, or (a5) $m^{(i)} + 1 \le \alpha \le m^{(j)}$ and $M^{(j)}[\alpha] \neq M^{(k)}[\alpha]$. In any case, the system of equations $\Sigma^{(j)} = \Sigma^{(i)}$ and $\Theta^{(j)} = \Theta^{(i)}$ provides a unique solution set for two random variables, so the probability of this event is at most $4/2^{2n}$, assuming $\ell \le 2^{n-2}$.

Lastly, assume that none of the containment (a) through (d) occurs. Then it means that there exist indices α, β such that $M^{(i)}[\alpha] \neq M^{(j)}[\alpha]$ and $M^{(i)}[\beta] \neq M^{(k)}[\beta]$. If $\alpha \neq \beta$, then we can simply choose $Y_\alpha^{(i)}$ and $Y_\beta^{(i)}$ to be the two variables. Suppose no such indices exist, that is, $\alpha = \beta$ and this is the only index that a difference occurs. If $M^{(j)}[\alpha] \neq M^{(k)}[\alpha]$, then we can choose two variables accordingly. If $M^{(j)}[\alpha] = M^{(k)}[\alpha]$, then since $M^{(j)} \neq M^{(k)}$, then $M^{(j)}$ is contained in $M^{(k)}$, or $M^{(k)}$ is contained in $M^{(j)}$, or there exists an index $\gamma > \alpha$ such that $M^{(j)}[\gamma] \neq M^{(k)}[\gamma]$. In any case, the system of equations $\Sigma^{(j)} = \Sigma^{(i)}$ and $\Theta^{(j)} = \Theta^{(i)}$ gives us a unique solution set for two random variables, and the probability of this event can be bounded as $4/2^{2n}$, again assuming $\ell \le 2^{n-2}$.

Now we are done with Case D. We just run indices i, j and k to get

$$\frac{\ell q}{2^n} + \sum_{i=2}^{q} \sum_{j=1}^{i-1} \sum_{k=1}^{i-1} \frac{5\ell^3 + 9\ell^2 + 4 + 4 + 4}{2^{2n}} \le \frac{\ell q}{2^n} + \frac{q^3}{3} \cdot \frac{26\ell^2}{2^{2n}} \le \frac{\ell q}{2^n} + \frac{9\ell^2 q^3}{2^{2n}}$$

for bounding the probability that case D happens.

4.3 Summing Up the Probabilities

We now bound the overall probability. The bound sums up to

$$\begin{aligned} &\Pr[\mathscr{A} \text{ sets } \mathtt{Zero}^*, \mathtt{Unfair}^*, \mathtt{UpLow}^*, \mathtt{LowUp}^* \text{ or } \mathtt{Coll}^*] \\ \le &\frac{2}{2^n} + \frac{2q^3}{2^{2n}} + \frac{8\ell q^3}{2^{2n}} + \frac{8\ell q^3}{2^{2n}} + \frac{\ell q}{2^n} + \frac{9\ell^3 q^3}{2^{2n}} \\ \le &\frac{27\ell^2 q^3}{2^{2n}} + \frac{3\ell q}{2^n}, \end{aligned}$$

which completes the proof.

5 Discussion

We have presented a 3-key rate-1 MAC construction based on a PMAC-type iteration. This raises a challenge to come up with a 1-key rate-1 MAC construction which is secure beyond the birthday bound.

After beating the birthday bound of $O(2^{n/2})$, we seem to be encountering another "bound problem" at the query complexity of $O(2^{2n/3})$. To beat this new bound efficiently is also a challenge for blockcipher-based message authentication.

Acknowledgments. The author would like to thank CRYPTO 2011 program committee members and reviewers for valuable feedback.

References

1. Bellare, M., Goldreich, O., Krawczyk, H.: Stateless evaluation of pseudorandom functions: Security beyond the birthday barrier. In: Wiener, M. J. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 270–287. Springer, Heidelberg (1999)
2. Bellare, M., Guérin, R., Rogaway, P.: XOR MACs: New Methods for Message Authentication Using Finite Pseudorandom Functions. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 15–28. Springer, Heidelberg (1995)
3. Bellare, M., Kilian, J., Rogaway, P.: The Security of Cipher Block Chaining. In: Desmedt, Y. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 341–358. Springer, Heidelberg (1994)
4. Bellare, M., Pietrzak, K., Rogaway, P.: Improved Security Analyses for CBC MACs. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 527–545. Springer, Heidelberg (2005)
5. Bellare, M., Rogaway, P.: The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Heidelberg (2006)
6. Bernstein, D.J.: How to stretch random functions: The security of Protected Counter Sums. J. Cryptology 12(3), 185–192 (1999)
7. Black, J., Rogaway, P.: CBC MACs for Arbitrary-Length Messages:The Three-Key Constructions. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 197–215. Springer, Heidelberg (2000)
8. Black, J., Rogaway, P.: A Block-Cipher Mode of Operation for Parallelizable Message Authentication. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 384–397. Springer, Heidelberg (2002)
9. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An Ultra-Lightweight Block Cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)
10. Hong, D., Sung, J., Hong, S.H., Lim, J.-I., Lee, S.-J., Koo, B.-S., Lee, C.-H., Chang, D., Lee, J., Jeong, K., Kim, H., Kim, J.-S., Chee, S.: HIGHT: A New Block Cipher Suitable for Low-Resource Device. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 46–59. Springer, Heidelberg (2006)

11. Iwata, T., Kurosawa, K.: OMAC: One-Key CBC MAC. In: Johansson, T. (ed.) FSE 2003. LNCS, vol. 2887, pp. 129–153. Springer, Heidelberg (2003)
12. Joux, A., Poupard, G., Stern, J.: New Attacks against Standardized MACs. In: Johansson, T. (ed.) FSE 2003. LNCS, vol. 2887, pp. 170–181. Springer, Heidelberg (2003)
13. JTC1. ISO/IEC 9797-1:1999 Information technology—Security techniques—Message Authentication Codes (macs)—Part 1: Mechanisms using a block cipher (1999)
14. Käsper, E., Schwabe, P.: Faster and Timing-Attack Resistant AES-GCM. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 1–17. Springer, Heidelberg (2009)
15. Kurosawa, K., Iwata, T.: TMAC: Two-Key CBC MAC. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 33–49. Springer, Heidelberg (2003)
16. Lucks, S.: The Sum of PRPs Is a Secure PRF. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 470–484. Springer, Heidelberg (2000)
17. Minematsu, K., Matsushima, T.: New Bounds for PMAC, TMAC, and XCBC. In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 434–451. Springer, Heidelberg (2007)
18. NIST. Recommendation for block cipher modes of operation: The CMAC mode for authentication. SP 800-38B (2005)
19. Petrank, E., Rackoff, C.: CBC MAC for real-time data sources. J. Cryptology 13(3), 315–338 (2000)
20. Preneel, B., van Oorschot, P.C.: MDx-MAC and Building Fast MACs from Hash Functions. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 1–14. Springer, Heidelberg (1995)
21. Rogaway, P.: Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 16–31. Springer, Heidelberg (2004)
22. Sarkar, P.: Pseudo-random functions and parallelizable modes of operations of a block cipher. IEEE Transactions on Information Theory 56(8), 4025–4037 (2010)
23. Yasuda, K.: The Sum of CBC MACs Is a Secure PRF. In: Pieprzyk, J. (ed.) CT-RSA 2010. LNCS, vol. 5985, pp. 366–381. Springer, Heidelberg (2010)

A Proof Lemma 1

Proof. The proof is almost exactly the same as the one for Lucks' SUM^2 construction $P_2(X) \oplus P_3(X)$ [16]. The fact that we have $\Sigma \neq \Theta$ does not have much effect on the computation of the probability. Specifically, we consider the following simulation of $P_2(\Sigma) \oplus P_3(\Theta)$.

The code without the boxed statement corresponds with $P_2(\Sigma) \oplus P_3(\Theta)$. The code with the boxed statement corresponds with a random oracle $\mathscr{R}$, because the set R is fair; that is, R is chosen so that the number of pairs $(U, L) \in R$ such that

$$T = U \oplus L$$

is the same for each value $T \in \{0,1\}^n$. In the code, we choose a fair set R as follows. Enumerate $\mathtt{Ran}\, P_2$ as $\{U_1, \ldots, U_\alpha\}$ and $\mathtt{Ran}\, P_3$ as $\{L_1, \ldots, L_\beta\}$. For each

1: $Y \leftarrow \overline{\mathtt{Ran}\, P_2}$, $Z \leftarrow \overline{\mathtt{Ran}\, P_3}$
2: Choose a fair set $R \subset Y \times Z$
3: $(U, L) \xleftarrow{\$} Y \times Z$
4: **if** $(U, L) \notin R$ **then**
5: $\quad \mathtt{Bad} \leftarrow$ **true** $\boxed{(U, L) \xleftarrow{\$} R}$
6: **end if**
7: $T \leftarrow U \oplus L$
8: **return** T

i and j such that $1 \le i \le \alpha$ and $1 \le j \le \beta$ we choose arbitrarily representatives $(U'_i, L'_j) \in Y \times Z$ such that $U'_i \oplus L'_j = U_i \oplus L_j$. We then define $R \leftarrow Y \times Z \setminus \bigcup_{i,j}\{(U'_i, L'_j)\}$. We see that, for each value $T \in \{0,1\}^n$,

$$\big|\{(U, L) \in R \mid U \oplus L = T\}\big| = 2^n - \alpha - \beta,$$

so R is indeed a fair set.

After q queries, the overall probability that the bad event occurs becomes

$$\begin{aligned}
\Pr\big[\mathtt{Bad}\big] &\le \sum_{i=1}^{q} \frac{\big|(Y \times Z) \setminus R\big|}{|Y \times Z|} \\
&= \sum_{i=1}^{q} \frac{\alpha\beta}{(2^n - \alpha)(2^n - \beta)} \\
&\le \sum_{i=0}^{q-1} \frac{i^2}{(2^n - q)^2} \\
&\le \frac{1}{(2^n - q)^2} \cdot \sum_{i=0}^{q-1} i^2 \\
&\le \frac{1}{(2^{n-1})^2} \cdot \frac{q(q-1)(2q-1)}{6} \\
&\le \frac{2q^3}{2^{2n}},
\end{aligned}$$

where we used the condition $q \le 2^{n-1}$. □