Designs, Codes and Cryptography (2023) 91:2237–2254
https://doi.org/10.1007/s10623-023-01199-4

Partition and mix: generalizing the swap-or-not shuffle

Nam-Su Jho[1] · **Jooyoung Lee**[2]

Received: 18 March 2022 / Revised: 21 October 2022 / Accepted: 10 February 2023 /
Published online: 2 March 2023

Abstract
Card shuffle algorithms have been studied from a cryptographic point of view with applications to format preserving encryption. In this work, we naturally extend the swap-or-not shuffle, proposed by Hoang, Morris and Rogaway at Crypto 2012, by replacing a perfect matching used in each round by a keyed partition with a certain uniform property. The resulting construction, dubbed the *partition-and-mix* (or simply PM) shuffle, is proved to be secure up to $(1-\delta)N$ queries for any $\delta > 0$ and the domain size N, while the number of rounds is significantly reduced compared to the swap-or-not. We give concrete examples of the keyed partitions that provide security as well as allow efficient implementation in practice. Such uniform keyed partitions seem of independent interest. The partition-and-mix shuffle might also be viewed as an alternative block cipher structure that extends the domain of a small block cipher operating on each block of the partition.

Keywords Card shuffle · Indistinguishability · Format preserving encryption · Block cipher

Mathematics Subject Classification 94A60

Communicated by M. Paterson.

Nam-Su Jho was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2021-0-00779, Development of high-speed encryption data processing technology that guarantees privacy based hardware).
Jooyoung Lee was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) Grant funded by the Korea government (MSIT) (No. 2022-0-01202, Regional strategic industry convergence security core talent training business).

✉ Jooyoung Lee
hicalf@kaist.ac.kr

Nam-Su Jho
nsjho@etri.re.kr

[1] Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea

[2] School of Computing, KAIST, Daejeon, Korea

1 Introduction

Format preserving encryption Suppose that we have a database that stores credit card numbers for a large number of customers, and for security reason, we would like to encrypt all of the credit card numbers. If we take a straightforward approach of using any well known block cipher such as AES, each credit card number, being 16-digits long, should be transformed into a 128-bit plaintext (by adding some dummy information), and then encrypted as a ciphertext of the same length. In order to accommodate all the ciphertexts as 128-bit strings, the database should be largely modified, causing a significant amount of extra cost. With this consideration, it would be desirable to encrypt the credit card numbers into ciphertexts of the same format, namely 16-digit numbers. This problem, called *format preserving encryption*, does not allow any solution as straightforward as one might expect. One should either design a novel mode of operation in order to use a block cipher operating on large-sized blocks such as AES [2–4], or construct a (dedicated) small block cipher from scratch.

Card shuffle-based encryption Focusing on the dedicated construction, a (balanced) Feistel cipher, for example, might not be a satisfactory solution at least from a point of provable security: no matter how carefully designed, the resulting block cipher provides only $n/2$-bit security for the block size n [12, 13]. This level of security might be acceptable for a large block size n, but not for a small size. Credit card numbers of 16 digits in the above example can be represented approximately by 54 bits, and 27-bit security level would be too low. To find an alternative block cipher structure to address this problem, card shuffle algorithms have begun to attract renewed interest that have a long history in probability theory. A card shuffle can be viewed as an encryption scheme when we think of the final position of a card at the end of the shuffle as the ciphertext of the initial position of the card.

In order for a card shuffle to be a computationally feasible block cipher, it should be oblivious, namely one should be able to trace the trajectory of a card without attending to lots of other cards in the deck. The Thorp shuffle is a well-known example of an oblivious card shuffle, where one first cuts a deck of cards into two equal piles, and then starts dropping the cards from either the left or right hand with probability $1/2$ [16]. Interpreted as a block cipher, a perfect matching is fixed on the set of positions for each round, and the two cards on each match is swapped or not according to a random coin of probability $1/2$, or equivalently according to the evaluation of a single-bit random function at the match. A representative of each match might be defined as the maximum of the two positions of the match. From this cryptographic point of view, the Thorp shuffle operating on $\{0,1\}^n$ has been proved to be secure up to $2^n/n$ queries for $O(n^2)$ rounds [10].

Afterwards, a randomized variant of the Thorp shuffle, named swap-or-not, has been proposed [8]. In this shuffle, a perfect matching is randomly chosen by an additional round key; a round key $K \in \{0,1\}^n$ defines a perfect matching on $\{0,1\}^n$ by the difference of K, namely position $x \in \{0,1\}^n$ is matched with $x \oplus K$. Then a single-bit round function is applied to each pair $\{x, x \oplus K\}$ and the cards at the two positions are swapped or not according to the round function value. Then the threshold number of queries is significantly improved up to $(1-\varepsilon)2^n$ for any $\varepsilon > 0$ for $O(n)$ rounds. Precisely, the adversarial distinguishing advantage is upper bounded by

$$\frac{8N^{3/2}}{r+4}\left(\frac{q+N}{2N}\right)^{r/4+1}$$

for the r-round swap-or-not shuffle, where N and q denote the size of the domain and the number of queries, respectively.[1] However it still requires a large number of rounds to achieve a sufficient level of security, for example more than 700 rounds for the domain size 2^{32} and the threshold number of queries $q = 2^{31}$.

1.1 Our results

Partition-and-mix In this work, we naturally extend the swap-or-not shuffle by replacing a perfect matching used in each round of the swap-or-not by a certain uniform keyed partition. Formally, fix a domain $[N] = \{0, \ldots, N-1\}$ for $N > 0$, the block size D of a keyed partition such that N is a multiple of D, and a certain key space $\mathcal{K}$. Let $(\mathcal{B}_K)_{K\in\mathcal{K}}$ be a keyed partition of $[N]$, where each key $K \in \mathcal{K}$ defines a partition of $[N]$

$$\mathcal{B}_K = \{B_K, B_K^2, \ldots, B_K^{\frac{N}{D}}\}$$

such that $|B_K^i| = D$ for $i = 1, \ldots, N/D$ and $\bigcup_{i=1}^{\frac{N}{D}} B_K^i = [N]$. For $\varepsilon > 0$, we will say the keyed partition $(\mathcal{B}_K)_{K\in\mathcal{K}}$ is *ε-almost D-uniform* if for every subset $U \in [N]$ such that $|U| = D$

$$\Pr\left[K \leftarrow_{\$} \mathcal{K} : U \in \mathcal{B}_K\right] \leq \frac{1+\varepsilon}{\binom{N-1}{D-1}}.$$

Remark 1 Fix a subset $U \in [N]$ of size D, and any single element a of U. When a partition of blocks of size D is chosen uniformly at random from the set of all possible partitions, the $D-1$ other elements of the block containing a are uniformly chosen from the set $[N] \setminus \{a\}$. The probability that they are $U\setminus\{a\}$ is exactly $1/\binom{N-1}{D-1}$. In other words, when a partition of blocks of size D is chosen uniformly at random from the set of all possible partitions, the probability of having U as its block is exactly $1/\binom{N-1}{D-1}$ for any subset $U \in [N]$ of size D.

Given an almost uniform keyed partition $(\mathcal{B}_K)_{K\in\mathcal{K}}$, the next step is to define an independent random permutation

$$\sigma_K^{i,t} : B_K^i \to B_K^i$$

for each key $K \in \mathcal{K}$, $i = 1, \ldots, N/D$ and $t = 1, \ldots, r$. Then the t-th round Ψ_t of the partition-and-mix shuffle, $t = 1, \ldots, r$, is defined as

$$\Psi_t(a) = \sigma_{K_t}^{i,t}(a) \tag{1}$$

for each $a \in \{0,1\}^n$, where $K_t \in \mathcal{K}$ is the t-th round key and $i \in \{1, \ldots, N/D\}$ is the index such that $a \in B_{K_t}^i$. Finally, the r-round partition-and-mix shuffle is defined as

$$\mathsf{PM}^r \stackrel{\text{def}}{=} \Psi_r \circ \cdots \circ \Psi_1.$$

As the entire domain is partitioned into blocks of a larger size $D \geq 2$ compared to the swap-or-not shuffle, and all the elements in each block are uniformly mixed, it would be natural to expect a faster mixing time, or a smaller number of rounds for a given level of security. We remark that the swap-or-not shuffle can be viewed as an instantiation of the partition-and-mix shuffle with $D = 2$ and $\mathcal{B}_K = \{\{x, x+K\} : x \in \{0,1\}^n\}$.

[1] The coefficient "4" appearing in the original upper bound in [8] should be corrected as "8".

The main contribution of this work is to prove the security of the partition-and-mix shuffle; for PM^r, we will prove

$$\mathbf{Adv}^{\mathrm{cca}}_{\mathsf{PM}^r}(q) \le \frac{4\,(1+\varepsilon)^{\frac{r}{4}}\,N^{\frac{r}{4}+\frac{1}{2}}}{(r-4)D^{\frac{r}{4}}(N-q)^{\frac{r}{4}-1}}.$$

In particular, if $q = (1-\delta)N$ for $\delta > 0$, then we have

$$\mathbf{Adv}^{\mathrm{cca}}_{\mathsf{PM}^r}((1-\delta)N) \le \frac{4\delta N^{\frac{3}{2}}}{r-4}\left(\frac{1+\varepsilon}{\delta D}\right)^{\frac{r}{4}}.$$

So, for a fixed number of adversarial queries, the number of rounds is reduced by $\frac{1}{\log D - \log(1+\varepsilon)}$ compared to the swap-or-not shuffle.

Uniform set partition In practice, the efficiency of the partition-and-mix shuffle would depend on the instantiation of the keyed partition. It seems of independent interest to find keyed partitions that allow efficient implementation. In this work, we propose two constructions of uniform random partitions.

The first construction is to use binary Hamming codes. For each integer $s \ge 2$, there is a binary Hamming code, denoted $\mathcal{C}_s$, with block length $2^s - 1$ and message length $2^s - s - 1$. In other words, $\mathcal{C}_s$ is a $(2^s - s - 1)$-dimensional subspace of $\{0,1\}^{2^s-1}$. Since a binary Hamming code is perfect, for any $\mathbf{x} \in \{0,1\}^{2^s-1}$, there is only one codeword $\mathbf{c} \in \mathcal{C}_s$ such that the Hamming distance of $\mathbf{c}$ and $\mathbf{x}$ is at most one. So the balls of radius one centered at the codewords partition the entire set $\{0,1\}^{2^s-1}$. With this observation, for $n \ge 2^s - 1$ and $D = 2^s$, we can construct an almost D-wise uniform keyed partition on $\{0,1\}^n$ by the following recipe.

1. Linearly independent keys $K_1, \ldots, K_{D-1} \in \{0,1\}^n$ are chosen uniformly at random. Then for a subspace
$$V = \langle K_1, \ldots, K_{D-1} \rangle$$
the entire domain $\{0,1\}^n$ is partitioned into the cosets of V.
2. Each coset can be identified as $\{0,1\}^{D-1}$. For example, one might choose a representative $\mathbf{a}$ for each coset, and define a bijection from $\{0,1\}^{D-1}$ to any coset by mapping
$$\mathbf{e} = (e_1, \ldots, e_{D-1}) \in \{0,1\}^{D-1} \mapsto \mathbf{a} + e_1 K_1 + \cdots + e_{D-1} K_{D-1}.$$
3. A $[2^s - 1, 2^s - s - 1, 3]$-Hamming code $\mathcal{C}_s$ and an additional round key
$$\mathbf{b} = (b_1, \ldots, b_{D-1}) \in \{0,1\}^{D-1}$$
defines a partition of the set $\{0,1\}^{D-1}$, and hence each coset of $\{0,1\}^n$, as follows.
$$\{0,1\}^{D-1} = \bigcup_{\mathbf{c} \in \mathcal{C}_s} \{\mathbf{c} + \mathbf{b} + \mathbf{e} : \mathbf{wt}(\mathbf{e}) \le 1\}.$$

This keyed partition is shown to be ε-almost uniform for $\varepsilon = 2^{D-n}$. We will discuss in detail the properties and the instantiation of the keyed partitions based on Hamming codes in Sects. 4 and 5.

Our second construction is recursive: for the block size $D > 0$, one can construct a D-uniform keyed partition of $X \times Y$ using a D-uniform keyed partition of X and a *D-wise independent function family* from X to Y. Notice that if a function family $(f_K)_{K \in \mathcal{K}_2}$ is D-wise independent, then for any distinct $x_1, \ldots, x_D \in X$ and any (not necessarily distinct)

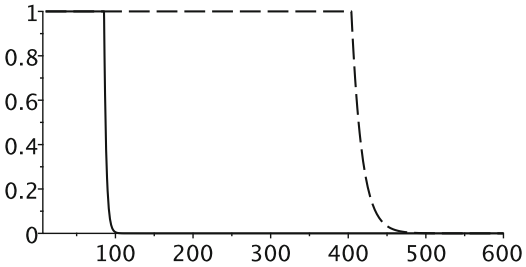

Fig. 1 Upper bounds on distinguishing advantages for the swap-or-not shuffle (in a dashed line) and the partition-and-mix shuffle (in a solid line) for $n = 32$, $q = 2^{31}$ given as a function of the number of rounds. The PM shuffle is based on a uniform keyed partition using a binary $[7, 4, 3]$-Hamming code

$y_1, \ldots, y_D \in Y$, the probability that $g(x_i) = y_i$ for all $i = 1, \ldots, D$ is the same, namely $1/|Y|^D$ over random choice of the key $K \in \mathcal{K}_2$.

Let $(\mathcal{B}'_K)_{K \in \mathcal{K}_1}$ be an ε-almost D-uniform keyed partition of X and let Y be an additive group. For a pair of keys $K = (K_1, K_2) \in \mathcal{K}_1 \times \mathcal{K}_2$, let

$$\mathcal{B}_K = \left\{\{(x, f_{K_2}(x) + c) : x \in B\} : B \in \mathcal{B}'_{K_1},\ c \in Y\right\} \subset X \times Y.$$

In Sect. 4, we prove that $(\mathcal{B}_K)_{K \in \mathcal{K}}$ is an ε'-almost D-uniform keyed partition of $X \times Y$ for

$$\varepsilon' = \varepsilon + \frac{D^2}{|X|} + \frac{\varepsilon D^2}{|X|}.$$

A D-wise independent function family is typically defined as a polynomial of degree at most $D-1$ over a finite field. This construction might be particularly useful when the domain size is not a power of two: for example, if we want to encrypt data (such as credit card numbers) within the domain $\{0, \ldots, 9\}^{16}$, then we can decompose the domain as $\{0, \ldots, 9\}^{16} = X \times Y$, where $X = \{0, 1\}^{16}$ and $Y = \{0, 1, 2, 3, 4\}^{16}$. Then we might use an almost uniform partition on the set X based on a binary Hamming code and any independent function family from X to Y to obtain a uniform keyed partition of $X \times Y$.

Comparison Figure 1compares the upper bounds on distinguishing advantages for the swap-or-not shuffle and the partition-and-mix shuffle based on a 8-uniform keyed partition for the domain size $N = 2^{32}$ and the threshold number of queries $q = N/2$. In this example, the partition-and-mix shuffle requires a family of random 3-bit permutations, while it provides the same level of security with approximately 1/4th of the number of rounds needed for the swap-or-not shuffle. Details on the instantiation of the partition-and-mix shuffle and its efficiency is discussed in Sect. 5.

1.2 Related work

The swap-or-not and the partition-and-mix shuffles asymptotically guarantee their security only up to $(1 - \varepsilon)N$ queries for any $\varepsilon > 0$, but not all the N possible queries for the domain size N. In [14], a new approach, called *mix-and-cut*, has been proposed turning one shuffle to another, where a deck of cards are randomly separated into two piles, and the shuffle algorithm is independently applied to each of the two piles. Within this framework, one obtains a shuffle achieving the full security by repeatedly applying the swap-or-not shuffle $O(\log^2 N)$ times. This approach has been further improved in [11], where they slightly modified mix-and-cut, and showed application of the underlying shuffle to only one of the

two piles is enough to achieve the full security. This framework, named *sometimes-recurse*, requires only $O(\log N)$ applications of the shuffle on average, significantly improving the efficiency over mix-and-cut.

As another line of research on block cipher construction, a substitution-permutation network is modeled as an iterated Even-Mansour cipher. The original single-round construction is shown to be secure only up to the birthday bound [7]. Iteration would naturally enhance its security, and indeed the r-round Even-Mansour cipher on $\{0,1\}^n$ has been proved to be secure up to $2^{\frac{rn}{r+1}}$ queries [5]. However we notice that the security model is incomparable to ours where the construction is based on independent random permutations whose size is the same as the entire construction as its underlying primitives, while an adversary is allowed to make queries to the inner permutations.

The partition-and-mix shuffle might be viewed as a mode of operation that extends the domain of a small block cipher operating on each block of the partition. The small block cipher might be constructed from a perfect random number generator, and again the random number generator constructed from any robust block cipher such as AES [15]. The domain extension of an ideal cipher has also been studied in [6], where they prove a 3-round Feistel cipher is a secure domain extender of an ideal cipher within the indifferentiability framework, while 2 rounds are enough to get a domain extender of a tweakable block cipher in the standard model.

2 Preliminaries

Notation For a fixed domain size $N > 0$, the set of all permutations on $[N]$ will be denoted $\mathcal{P}$. For a set T and an integer $s \geq 1$, T^{*s} denotes the set of all sequences that consists of s pairwise distinct elements of T. For integers $1 \leq s \leq t$, we will write $(t)_s = t(t-1)\cdots(t-s+1)$. If $|T| = t$, then $(t)_s$ becomes the size of T^{*s}.

For a binary string $\mathbf{w}$, the number of its nonzero components is called the *weight* of $\mathbf{w}$, denoted $\mathbf{wt}(\mathbf{w})$. For an element $x \in \{0,1,\ldots,2^s-1\}$, let $\langle x\rangle_s \in \{0,1\}^s$ denote the binary representation of x, namely, an s-bit string $(a_1,\ldots,a_s) \in \{0,1\}^s$ such that $x = 2^{s-1}a_s + \cdots + 2a_2 + a_1$, and let $\mathbf{e}(x)$ denote a (2^s-1)-bit string $(b_1,\ldots,b_{2^s-1}) \in \{0,1\}^{2^s-1}$ such that $b_i = 1$ if $i = x$, and $b_i = 0$ otherwise. So we have $\mathbf{wt}(\mathbf{e}(x)) = 0$ if $x = 0$, and $\mathbf{wt}(\mathbf{e}(x)) = 1$ otherwise.

Hamming code An $[n,k,d]_{2^e}$ linear error-correcting code $\mathcal{C}$ is a k-dimensional subspace of $\mathbb{F}_{2^e}^n$ with the minimum weight d, where $\mathbb{F}_{2^e}$ denotes a finite field of order 2^e. An $[n,k,d]_{2^e}$ code $\mathcal{C}$ can be represented by a $k \times n$ generator matrix G over $\mathbb{F}_{2^e}$ where every codeword of $\mathcal{C}$ is expressed as a linear combination of the row vectors of G, namely $w \cdot G$ for some $w \in \mathbb{F}_{2^e}^k$.

Hamming codes are a family of $[2^s-1, 2^s-k-1, 3]_2$ codes, where $s \geq 2$. For each Hamming code, the balls of Hamming radius one centered on the codewords exactly fill out the entire space $\{0,1\}^n$ where $n = 2^s-1$.

D-wise independent function family Let $(f_K)_{K\in\mathcal{K}}$ be a family of functions from X to Y with key space $\mathcal{K}$. For a positive integer D, $(f_K)_{K\in\mathcal{K}}$ is called *D-wise independent* if for any distinct $x_1,\ldots,x_D \in X$ and any (not necessarily distinct) $y_1,\ldots,y_D \in Y$, the probability that $g(x_i) = y_i$ for every $i = 1,\ldots,D$ is $1/|Y|^D$ over random choice of the key $K \in \mathcal{K}$.

Security definition Let E be a block cipher on $[N]$ that employs λ-bit keys. So each key $\mathbf{k} \in \{0,1\}^\lambda$ defines a permutation $\mathsf{E}_\mathbf{k}$ on $[N]$. In the adaptive chosen-ciphertext attack-indistinguishability (CCA-IND) model, an adversary $\mathcal{A}$ adaptively makes forward and

backward queries to either a permutation P or the blockcipher $\mathsf{E}_{\mathbf{k}}$ to tell apart $\mathsf{E}_{\mathbf{k}}$ and P, where $\mathsf{E}_{\mathbf{k}}$ uses a random secret key $\mathbf{k}$ and P is chosen uniformly at random from $\mathcal{P}$. Thus $\mathcal{A}$'s distinguishing advantage is formally defined by

$$\mathbf{Adv}^{\mathrm{cca}}_{\mathsf{E}}(\mathcal{A}) = \Pr\left[P \leftarrow_{\$} \mathcal{P} : \mathcal{A}^{P,P^{-1}} = 1\right] - \Pr\left[\mathbf{k} \leftarrow_{\$} \{0,1\}^{\lambda} : \mathcal{A}^{\mathsf{E}_{\mathbf{k}},\mathsf{E}_{\mathbf{k}}^{-1}} = 1\right].$$

In the non-adaptive chosen-plaintext attack (NCPA) model, an adversary $\mathcal{A}$ makes only non-adaptive forward queries. The advantage $\mathbf{Adv}^{\mathrm{ncpa}}_{\mathsf{E}}(\mathcal{A})$ is similarly defined in this model. For $\mathrm{atk} \in \{\mathrm{cca}, \mathrm{ncpa}\}$, and for $q > 0$, we define

$$\mathbf{Adv}^{\mathrm{atk}}_{\mathsf{E}}(q) = \max_{\mathcal{A}} \mathbf{Adv}^{\mathrm{atk}}_{\mathsf{E}}(\mathcal{A})$$

where the maximum is taken over all atk-adversaries making at most q queries. If the encryption and decryption algorithms are symmetric in their structures, we can lift the NCPA-security of the block cipher to CCA-security by doubling the number of rounds [9].

Lemma 1 *If F and G are block ciphers on the same message space, then for any $q > 0$,*

$$\mathbf{Adv}^{\mathrm{cca}}_{F \circ G^{-1}}(q) \le \mathbf{Adv}^{\mathrm{ncpa}}_{F}(q) + \mathbf{Adv}^{\mathrm{ncpa}}_{G}(q).$$

Total variation distance Given a finite event space Ω and two probability distributions μ and ν defined on Ω, the *total variation distance* between μ and ν, denoted $\|\mu - \nu\|$, is defined as

$$\|\mu - \nu\| \stackrel{\mathrm{def}}{=} \frac{1}{2} \sum_{x \in \Omega} |\mu(x) - \nu(x)| = \max_{S \subset \Omega} \{\mu(S) - \nu(S)\} = \max_{S \subset \Omega} \{\nu(S) - \mu(S)\}.$$

USEFUL LEMMAS. For a finite nonempty set Ω, let μ and ν be probability distributions supported on q-tuples of elements of Ω. If the first l elements $u_1^*, \ldots, u_l^*$ are fixed for $l = 0, \ldots, q-1$, then we can consider the distribution of μ restricted to the $(l+1)$-th element, conditioned on $(u_1^*, \ldots, u_l^*)$, namely

$$\mu(u | u_1^*, \ldots, u_l^*) = \Pr\left[X_{l+1} = u | X_1 = u_1^*, \ldots, X_l = u_l^*\right]$$

where $(X_1, \ldots, X_q) \sim \mu$. The distribution $\nu(\,\cdot\,|u_1^*, \ldots, u_l^*)$ is similarly defined, and hence

$$\|\mu(\,\cdot\,|u_1^*, \ldots, u_l^*) - \nu(\,\cdot\,|u_1^*, \ldots, u_l^*)\|.$$

Using this notation, given a set of random variables $(Z_1, \ldots, Z_q)$, we can define a new random variable

$$\|\mu(\,\cdot\,|Z_1, \ldots, Z_l) - \nu(\,\cdot\,|Z_1, \ldots, Z_l)\|$$

for $l = 0, \ldots, q-1$. Then the total variation distance $\|\mu - \nu\|$ is upper bounded by the sum of the conditional distances on average as follows.

Lemma 2 *Fix a finite nonempty set Ω and let μ and ν be probability distributions supported on q-tuples of elements of Ω, and suppose that $(Z_1, \ldots, Z_q) \sim \mu$. Then*

$$\|\mu - \nu\| \le \sum_{l=0}^{q-1} \mathbf{E}\left(\|\mu(\,\cdot\,|Z_1, \ldots, Z_l) - \nu(\,\cdot\,|Z_1, \ldots, Z_l)\|\right).$$

Note that the expectation is taken over the set of random variables $(Z_1, \ldots, Z_q)$.

Using the conventions $\binom{0}{0} = 1$ and $\binom{p}{q} = 0$ for $0 \le p < q$, the following lemma on binomial coefficients will be also useful later.

Lemma 3 *Let a, b, c be positive integers such that $b \leq c$. Then*

$$\sum_{j=0}^{a} \frac{\binom{b}{j}\binom{c-b}{a-j}}{(j+1)\binom{c}{a}} \leq \frac{c+1}{(a+1)(b+1)}. \tag{2}$$

Proof By integrating both sides of

$$(1+x)^b = \sum_{j=0}^{b} \binom{b}{j} x^j$$

we obtain

$$\frac{1}{b+1}\left((1+x)^{b+1} - 1\right) = \sum_{j=0}^{b} \frac{1}{j+1}\binom{b}{j} x^{j+1}.$$

Therefore the left-hand side of (2) is the coefficient of x^{a+1} in the polynomial

$$\frac{1}{\binom{c}{a}}\left(\sum_{j=0}^{b} \frac{1}{j+1}\binom{b}{j} x^{j+1}\right)\sum_{i=0}^{c-b}\binom{c-b}{i}x^i = \frac{\left((1+x)^{b+1}-1\right)(1+x)^{c-b}}{(b+1)\binom{c}{a}}$$
$$= \frac{(1+x)^{c+1} - (1+x)^{c-b}}{(b+1)\binom{c}{a}}$$

which is upper bounded by the coefficient of x^{a+1} in

$$\frac{(1+x)^{c+1}}{(b+1)\binom{c}{a}}. \tag{3}$$

The coefficient of x^{a+1} in (3) is

$$\frac{\binom{c+1}{a+1}}{(b+1)\binom{c}{a}} \leq \frac{c+1}{(a+1)(b+1)}.$$

□

3 Security of the partition-and-mix shuffle

The security of the r-round partition-and-mix shuffle PM^r defined by an ε-almost D-uniform keyed partition $(\mathcal{B}_K)_{K\in\mathcal{K}}$ and a set of independent random permutations $(\sigma_K^{i,t})_{(K,i,t)\in\mathcal{K}\times\{1,\ldots,\frac{N}{D}\}\times\{1,\ldots,t\}}$ is summarized as the following theorem.

Theorem 1 *Let PM^r be the r-round partition-and-mix shuffle defined by a keyed partition $(\mathcal{B}_K)_{K\in\mathcal{K}}$ and a set of mixing permutations $(\sigma_K^{i,t})$. If $(\mathcal{B}_K)_{K\in\mathcal{K}}$ is ε-almost D-uniform, $\sigma_K^{i,t}$ are all independent random, and round keys $K_1, \ldots, K_t$ are chosen independently and uniformly at random from $\mathcal{K}$, then*

$$\mathbf{Adv}^{\mathrm{cca}}_{\mathsf{PM}^r}(q) \leq \frac{4\,(1+\varepsilon)^{\frac{r}{4}}\,N^{\frac{r}{4}+\frac{1}{2}}}{(r-4)D^{\frac{r}{4}}(N-q)^{\frac{r}{4}-1}}.$$

3.1 Proof of Theorem 1

Fix q distinct elements $z_1, \ldots, z_q \in [N]$. For $j = 1, \ldots, q$ and $t = 1, \ldots, r$, let $X_t(j)$ denote the random variable that indicates the position of z_j at the end of the t-th round of PM^r, namely,

$$X_t(j) = \Psi_t \circ \cdots \circ \Psi_1(z_j)$$

where $\Psi_1, \ldots, \Psi_t$ are as defined in (1). Let τ_t be the distribution of

$$(X_t(1), \ldots, X_t(q))$$

and let π be the uniform random distribution on $[N]^{*q}$. So π is the distribution of q samples without replacement from $[N]$. The core of the security proof is to upper bound their statistical distance $\|\tau_r - \pi\|$ for reasonably small r since this is the distinguishing advantage of an NCPA-adversary that makes q queries $z_1, \ldots, z_q$.

Given a set of the first t round keys $K = (K_1, \ldots, K_t) \in \mathcal{K}^t$ for $t = 1, \ldots, r$, we can consider the distribution of $(X_t(1), \ldots, X_t(q))$ conditioned on a fixed set of partitions $(\mathcal{B}_{K_1}, \ldots, \mathcal{B}_{K_t})$, denoted τ_t^K. Then by the definition of the total variance distance and by the triangle inequality, we have

$$\begin{aligned}
\|\tau_r - \pi\| &= \frac{1}{2} \times \sum_{(u_1,\ldots,u_q)\in[N]^{*q}} \left| \left(\sum_{K\in\mathcal{K}^r} \frac{1}{|\mathcal{K}|^r} \tau_r^K(u_1, \ldots, u_q) \right) - \pi(u_1, \ldots, u_q) \right| \\
&= \frac{1}{2} \times \sum_{(u_1,\ldots,u_q)\in[N]^{*q}} \left| \sum_{K\in\mathcal{K}^r} \frac{1}{|\mathcal{K}|^r} \left(\tau_r^K(u_1, \ldots, u_q) - \pi(u_1, \ldots, u_q) \right) \right| \\
&\le \sum_{K\in\mathcal{K}^r} \frac{1}{|\mathcal{K}|^r} \left(\frac{1}{2} \times \sum_{(u_1,\ldots,u_q)\in[N]^{*q}} \left| \tau_r^K(u_1, \ldots, u_q) - \pi(u_1, \ldots, u_q) \right| \right) \\
&= \mathbf{E}\left(\|\tau_r^K - \pi\| \right) \qquad (4)
\end{aligned}$$

where the expectation is taken over random variable K (regarded as defined on $\mathcal{K}^r$ with the uniform distribution). Again, by Lemma 2, we have

$$\begin{aligned}
\mathbf{E}\left(\|\tau_r^K - \pi\| \right) &\le \mathbf{E}\left(\sum_{l=0}^{q-1} \mathbf{E}\left(\|\tau_r^K(\cdot\,|X_r(1), \ldots, X_r(l)) - \pi(\cdot\,|X_r(1), \ldots, X_r(l))\| \right) \right) \\
&= \sum_{l=0}^{q-1} \mathbf{E}\left(\|\tau_r^K(\cdot\,|X_r(1), \ldots, X_r(l)) - \pi(\cdot\,|X_r(1), \ldots, X_r(l))\| \right) \\
&= \sum_{l=0}^{q-1} \mathbf{E}\left(\|\tau_r^K(\cdot\,|X_r(1), \ldots, X_r(l)) - \frac{1}{m}\| \right) \qquad (5)
\end{aligned}$$

where the last expectation is taken over random variables $X_r(1), \ldots, X_r(l)$ and K, and $m = N - l$. For a fixed $l = 0, \ldots, q-1$, let

$$p_t(a) = \tau_t^K(a|X_t(1), \ldots, X_t(l)).$$

Then we have

$$\|\tau_t^K(\cdot|X_t(1),\ldots,X_t(l)) - \pi(\cdot|X_t(1),\ldots,X_t(l))\| = \frac{1}{2}\sum_{a\in S_t}|p_t(a) - 1/m|$$

where $S_t = [N] \setminus \{X_t(1),\ldots,X_t(l)\}$. By using the inequality $\mathbf{E}(X)^2 \le \mathbf{E}(X^2)$ (that holds for any random variable X) and the Cauchy-Schwarz inequality, we have

$$\left(\mathbf{E}\left(\sum_{a\in S_t}|p_t(a) - 1/m|\right)\right)^2 \le N \cdot \mathbf{E}\left(\sum_{a\in S_t}(p_t(a) - 1/m)^2\right). \tag{6}$$

Define $s_t = \sum_{a\in S_t}(p_t(a) - 1/m)^2$ for $t = 0,\ldots,r$. Since the initial positions of the elements $z_1,\ldots,z_q$ are deterministic, we have

$$\mathbf{E}(s_0) = \left(1 - \frac{1}{m}\right)^2 < 1.$$

Then we will express $\mathbf{E}(s_{t+1}|s_t)$ as a linear equation of s_t with small coefficients.

As s_t being a random variable defined by $X_t(1),\ldots,X_t(l)$ and $K_1,\ldots,K_t$, we fix the values of these variables, and consider the conditional expectation of s_{t+1}. Given a partition $\mathcal{B}_{K_{t+1}}$, we only determine the evolution of $X_t(1),\ldots,X_t(l)$ (not the other elements) to determine S_{t+1}. Then we can arbitrarily define a permutation

$$f : S_t \longrightarrow S_{t+1}$$

such that $f(B \cap S_t) = B \cap S_{t+1}$ for every $B \in \mathcal{B}_{K_{t+1}}$. (This is always possible since $|B \cap S_t| = |B \cap S_{t+1}|$.) Since

$$p_{t+1}(f(a)) = \begin{cases} \sum_{u\in B\cap S_t}\frac{p_t(u)}{|B\cap S_t|}, & \text{if } a \in B \cap S_t \\ p_t(a), & \text{if } a \notin B \cap S_t \end{cases}$$

for every $B \in \mathcal{B}_{K_{t+1}}$, it follows that

$$\begin{aligned}
&\mathbf{E}(s_{t+1}|s_t) \\
&= \mathbf{E}\left(\sum_{a\in S_t}(p_{t+1}(f(a)) - 1/m)^2 \,\middle|\, s_t\right) \\
&= \sum_{a\in S_t}\sum_{\substack{U\subset[N]\text{where} \\ a\in U \text{ and } |U|=D}} \Pr\left[K_{t+1} \leftarrow_{\$} \mathcal{K} : U \in \mathcal{B}_{K_{t+1}}\right]\left(\sum_{u\in U\cap S_t}\frac{p_t(u)}{|U\cap S_t|} - \frac{1}{m}\right)^2 \\
&\le (1+\varepsilon)\sum_{a\in S_t}\sum_{\substack{U\subset[N]\text{where} \\ a\in U \text{ and } |U|=D}}\frac{1}{\binom{N-1}{D-1}}\left(\sum_{u\in U\cap S_t}\frac{p_t(u)}{|U\cap S_t|} - \frac{1}{m}\right)^2.
\end{aligned}$$

For a fixed element $a \in S_t$, we can choose a set $U \subset [N]$ such that $a \in U$ and $|U| = D$ by the following process.

1. Fix $i = |(U \cap S_t)|$, where $1 \le i \le D$.
2. Choose $V = (U \cap S_t) \setminus \{a\} = \{v_1,\ldots,v_{i-1}\}$.
3. Choose $W = U \setminus S_t$ such that $|W| = D - i$.

4. Define $U = V \cup W \cup \{a\}$.

Since the number of ways of choosing sets W is $\binom{l}{D-i}$, we have

$$\begin{aligned}
&\sum_{\substack{U\subset[N]\text{where}\\ a\in U \text{ and } |U|=D}} \frac{1}{\binom{N-1}{D-1}} \left(\sum_{u\in U\cap S_t} \frac{p_t(u)}{|U\cap S_t|} - \frac{1}{m}\right)^2 \\
&= \sum_{i=1}^{D} \frac{\binom{l}{D-i}}{\binom{N-1}{D-1}} \sum_{\{v_1,\ldots,v_{i-1}\}\subset S_t\setminus\{a\}} \left(\frac{p_t(a)+p_t(v_1)+\cdots+p_t(v_{i-1})}{i} - \frac{1}{m}\right)^2 \\
&= \sum_{i=1}^{D} \frac{\binom{l}{D-i}}{\binom{N-1}{D-1}} \cdot \frac{1}{i^2(i-1)!} \\
&\times \sum_{(v_1,\ldots,v_{i-1})\subset(S_t\setminus\{a\})^{*(i-1)}} \left(\left(p_t(a)-\frac{1}{m}\right)+\cdots+\left(p_t(v_{i-1})-\frac{1}{m}\right)\right)^2.
\end{aligned}$$

We expand and simplify the inner summation using the following observations.

1.
$$\sum_{(v_1,\ldots,v_{i-1})\subset(S_t\setminus\{a\})^{*(i-1)}} \left(p_t(a)-\frac{1}{m}\right)^2 = (m-1)_{i-1}\left(p_t(a)-\frac{1}{m}\right)^2 \stackrel{\text{def}}{=} A_1.$$

2. For $1 \le j \le i-1$, since $\sum_{v\in S_t}\left(p_t(v)-\frac{1}{m}\right)=0$,
$$\begin{aligned}
&\sum_{(v_1,\ldots,v_{i-1})\subset(S_t\setminus\{a\})^{*(i-1)}} \left(p_t(a)-\frac{1}{m}\right)\left(p_t(v_j)-\frac{1}{m}\right) \\
&= (m-2)_{i-2}\left(p_t(a)-\frac{1}{m}\right)\sum_{v\in S_t\setminus\{a\}}\left(p_t(v)-\frac{1}{m}\right) \\
&= -(m-2)_{i-2}\left(p_t(a)-\frac{1}{m}\right)^2 \stackrel{\text{def}}{=} A_2
\end{aligned}$$
where we assume $m, i \ge 2$.

3. For $1 \le j \le i-1$,
$$\begin{aligned}
&\sum_{(v_1,\ldots,v_{i-1})\subset(S_t\setminus\{a\})^{*(i-1)}} \left(p_t(v_j)-\frac{1}{m}\right)^2 \\
&= (m-2)_{i-2}\sum_{v\in S_t\setminus\{a\}}\left(p_t(v)-\frac{1}{m}\right)^2 \\
&= (m-2)_{i-2}\left(s_t-\left(p_t(a)-\frac{1}{m}\right)^2\right) \stackrel{\text{def}}{=} A_3
\end{aligned}$$
where we assume $m, i \ge 2$.

4. For $1 \le j < h \le i-1$,
$$\sum_{(v_1,\ldots,v_{i-1})\subset(S_t\setminus\{a\})^{*(i-1)}} \left(p_t(v_j)-\frac{1}{m}\right)\left(p_t(v_h)-\frac{1}{m}\right)$$

$$
\begin{aligned}
&= (m-3)_{i-3}\left(\left(\sum_{v\in S_t\setminus\{a\}}\left(p_t(v)-\frac{1}{m}\right)\right)^2-\sum_{v\in S_t\setminus\{a\}}\left(p_t(v)-\frac{1}{m}\right)^2\right)\\
&= (m-3)_{i-3}\left(\left(p_t(a)-\frac{1}{m}\right)^2-\sum_{v\in S_t\setminus\{a\}}\left(p_t(v)-\frac{1}{m}\right)^2\right)\\
&= (m-3)_{i-3}\left(2\left(p_t(a)-\frac{1}{m}\right)^2-s_t\right)\overset{\text{def}}{=} A_4
\end{aligned}
$$

where we assume $m, i \geq 3$.

Since

$$
\begin{aligned}
\sum_{a\in S_t} A_1 &= (m-1)_{i-1}s_t,\\
\sum_{a\in S_t} A_2 &= -(m-2)_{i-2}s_t,\\
\sum_{a\in S_t} A_3 &= (m-2)_{i-2}\,(ms_t-s_t) = (m-1)_{i-1}s_t,\\
\sum_{a\in S_t} A_4 &= (m-3)_{i-3}\,(2s_t-ms_t) = -(m-2)_{i-2}s_t,
\end{aligned}
$$

we have

$$
\sum_{a\in S_t}(A_1+2(i-1)A_2+(i-1)A_3+(i-1)(i-2)A_4) = i(m-i)(m-2)_{i-2}s_t,
$$

and hence

$$
\begin{aligned}
\mathbf{E}\,(s_{t+1}|s_t) &= (1+\varepsilon)\sum_{i=1}^{D}\frac{\binom{l}{D-i}}{\binom{N-1}{D-1}}\cdot\frac{1}{i^2(i-1)!}\\
&\quad\times\sum_{a\in S_t}(A_1+2(i-1)A_2+(i-1)A_3+(i-1)(i-2)A_4)\\
&= (1+\varepsilon)\sum_{i=1}^{D}\frac{\binom{l}{D-i}\cdot i(m-i)(m-2)_{i-2}}{i^2(i-1)!\binom{N-1}{D-1}}s_t\\
&\leq (1+\varepsilon)\sum_{i=1}^{D}\frac{\binom{l}{D-i}\,(m-1)_{i-1}}{i!\binom{N-1}{D-1}}s_t\\
&\leq \frac{(1+\varepsilon)\,Ns_t}{Dm} \qquad (7)
\end{aligned}
$$

where the last inequality follows since by applying Lemma 3 with $a = D-1$, $b = m-1$ and $c = N-1$,

$$
\begin{aligned}
\sum_{i=1}^{D}\frac{\binom{l}{D-i}\,(m-1)_{i-1}}{i!\binom{N-1}{D-1}} &= \sum_{j=0}^{a}\frac{\binom{c-b}{a-j}\,(b)_j}{(j+1)!\binom{c}{a}} = \sum_{j=0}^{a}\frac{\binom{c-b}{a-j}\binom{b}{j}}{(j+1)\binom{c}{a}}\\
&\leq \frac{c+1}{(a+1)(b+1)} = \frac{N}{Dm}.
\end{aligned}
$$

By taking expectation on both sides of inequality (7), we have

$$\mathbf{E}(s_{t+1}) \leq \frac{(1+\varepsilon)N}{Dm}\mathbf{E}(s_t).$$

Since $\mathbf{E}(s_0) < 1$, we have

$$\mathbf{E}(s_r) \leq \left(\frac{(1+\varepsilon)N}{Dm}\right)^r.$$

Therefore by (4), (5) and (6), we have

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{ncpa}}_{\mathsf{PM}^r}(q) &= \|\tau_r - \pi\| \\
&\leq \frac{1}{2}\sum_{l=0}^{q-1}(N\mathbf{E}(s_r))^{\frac{1}{2}} \\
&\leq \frac{N^{\frac{1}{2}}}{2}\sum_{l=0}^{q-1}\left(\frac{(1+\varepsilon)N}{Dm}\right)^{\frac{r}{2}} \\
&\leq \frac{N^{\frac{3}{2}}}{2D^{\frac{r}{2}}}\sum_{l=0}^{q-1}\left(\frac{1+\varepsilon}{1-\frac{l}{N}}\right)^{\frac{r}{2}}\cdot\frac{1}{N} \\
&\leq \frac{N^{\frac{3}{2}}}{2D^{\frac{r}{2}}}\int_0^{\frac{q}{N}}\left(\frac{1+\varepsilon}{1-x}\right)^{\frac{r}{2}}dx \\
&\leq \frac{(1+\varepsilon)^{\frac{r}{2}}N^{\frac{r}{2}+\frac{1}{2}}}{(r-2)D^{\frac{r}{2}}(N-q)^{\frac{r}{2}-1}}.
\end{aligned}$$

By using Lemma 1, we complete the proof of Theorem 1.

4 Almost uniform partitions

In this section, we will describe in detail how keyed partitions can be defined based on binary Hamming codes, and efficiently implemented within the PM shuffle. We also analyze the property of the recursive construction given in Sect. 1.1.

4.1 Almost uniform partitions based on binary hamming codes

For each integer $s \geq 2$, let $\mathcal{C}_s$ be a binary $[2^s-1, 2^s-s-1, 3]$-Hamming code. Using the code $\mathcal{C}_s$, we can define a keyed partition $(\mathcal{B}_K)_{K\in\mathcal{K}}$ of $\{0,1\}^n$ for any $n \geq 2^s-1$ where each block is of size $D = 2^s$. The key space of this keyed partition is defined as

$$\begin{aligned}
\mathcal{K} = \{&(K_1,\ldots,K_{D-1}) \in (\{0,1\}^n)^{D-1} : K_1,\ldots,K_{D-1} \text{ are linearly independent}\} \\
&\times \{0,1\}^{D-1}.
\end{aligned}$$

Given a key $(K_1,\ldots,K_{D-1},\mathbf{b}) \in \mathcal{K}$, it determines a subspace of dimension $D-1$

$$V = \langle K_1,\ldots,K_{D-1}\rangle.$$

If we arbitrarily fix a set of representatives R for the quotient space $\{0,1\}^n/V$, then the entire set $\{0,1\}^n$ is partitioned as

$$\{0,1\}^n = \bigcup_{\mathbf{a}\in R} (\mathbf{a}+V).$$

Again, we partition each coset $\mathbf{a}+V$ as

$$\mathbf{a}+V = \bigcup_{\mathbf{c}\in\mathcal{C}_s} \{\mathbf{a} + (c_1+b_1+e_1)K_1 + \cdots + (c_{D-1}+b_{D-1}+e_{D-1})K_{D-1} : \mathbf{wt}(e_1,\ldots,e_{D-1}) \le 1\}.$$

where we write $\mathbf{c} = (c_1,\ldots,c_{D-1})$, $\mathbf{b} = (b_1,\ldots,b_{D-1})$. So for each codeword $\mathbf{c} = (c_1,\ldots,c_{D-1}) \in \mathcal{C}_s$ and the key $\mathbf{b} = (b_1,\ldots,b_{D-1})$, the element

$$\mathbf{a} + (c_1+b_1)K_1 + \cdots + (c_{D-1}+b_{D-1})K_{D-1} \tag{8}$$

becomes the *center* of the block containing the element itself in a sense that the other elements of the block are obtained by adding K_i, $i = 1,\ldots,D-1$, to the center. Given a key $(K_1,\ldots,K_{D-1},\mathbf{b})$, the center of each block is uniquely determined.

Let $U = \{\mathbf{u}_1,\ldots,\mathbf{u}_D\} \subset \{0,1\}^n$ be a subset of size D. Suppose that U is a block in a partition with key $(K_1,\ldots,K_{D-1},\mathbf{b})$. Then u_i should be the center of a ball for some $i = 1,\ldots,D$, which is of the form of (8). In this case, we have

$$(\mathbf{u}_1+\mathbf{u}_i,\ldots,\mathbf{u}_{i-1}+\mathbf{u}_i,\mathbf{u}_{i+1}+\mathbf{u}_i,\ldots,\mathbf{u}_D+\mathbf{u}_i) = (K_{g(1)},\ldots,K_{g(D-1)})$$

for some permutation g on $[D-1]$. Once i and g are fixed, then $V = \langle K_1,\ldots,K_{D-1}\rangle$ is determined, and hence a representative $\mathbf{a}$ such that $U \subset \mathbf{a}+V$. If we arbitrarily choose any codeword $\mathbf{c} \in \mathcal{C}_s$, then $\mathbf{b}$ is uniquely determined by $\mathbf{a}$, $\mathbf{c}$ and the center of the ball $u_i = \mathbf{a} + (c_1+b_1)K_1 + \cdots + (c_{D-1}+b_{D-1})K_{D-1}$.Since

$$|\mathcal{K}| = 2^{D-1}\cdot\prod_{i=0}^{D-2}(N-2^i),$$
$$|\mathcal{C}_s| = 2^{D-s-1},$$

and $D = 2^s$, we have

$$\begin{aligned}
\Pr\left[K \leftarrow_{\$} \mathcal{K} : U \in \mathcal{B}_K\right] &\le \frac{D\cdot(D-1)!\cdot|\mathcal{C}_s|}{|\mathcal{K}|} \\
&= \frac{(D-1)!}{\prod_{i=0}^{D-2}(N-2^i)} \\
&= \left(\prod_{i=0}^{D-2}\frac{1}{N-2^i}\right)\cdot\frac{(D-1)!\binom{N-1}{D-1}}{\binom{N-1}{D-1}} \\
&\le \left(\prod_{i=0}^{D-2}\frac{N}{N-2^i}\right)\cdot\frac{1}{\binom{N-1}{D-1}} \\
&= \left(\frac{1}{\prod_{i=0}^{D-2}\left(1-\frac{2^i}{N}\right)}\right)\cdot\frac{1}{\binom{N-1}{D-1}} \\
&\le \frac{1}{1-\frac{2^{D-1}}{N}}\cdot\frac{1}{\binom{N-1}{D-1}}
\end{aligned}$$

$$\leq \left(1 + \frac{2^D}{N}\right) \frac{1}{\binom{N-1}{D-1}}$$

if $N \geq 2^D$. Therefore this keyed partition is ε-almost D-uniform for $\varepsilon = 2^D/N$.

4.2 Extension of almost uniform partitions using random functions

Let $(\mathcal{B}'_K)_{K\in\mathcal{K}_1}$ be an ε-almost D-uniform keyed partition of X, let Y be an additive group, and let $(f_K)_{K\in\mathcal{K}_2}$ be a D-wise independent function family from X to Y. Then we can construct an ε'-almost D-uniform keyed partition $(\mathcal{B}_K)_{K\in\mathcal{K}}$ of $X \times Y$ with the key space being $\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2$, where

$$\varepsilon' = \varepsilon + \frac{D^2}{|X|} + \frac{\varepsilon D^2}{|X|}.$$

Given a key $K = (K_1, K_2) \in \mathcal{K}_1 \times \mathcal{K}_2$, the partition keyed with K is defined as

$$\mathcal{B}_K = \left\{\{(x, f_{K_2}(x) + c) : x \in B\} : B \in \mathcal{B}_{K_1},\ c \in Y\right\}.$$

Let $U = \{(x_1, y_1), \ldots, (x_D, y_D)\}$ be a subset of $X \times Y$ of size D. If there is a collision at the first position, namely $x_i = x_j$ for some $1 \leq i < j \leq D$, then

$$\Pr\left[K \leftarrow_{\$} \mathcal{K} : U \in \mathcal{B}_K\right] = 0.$$

Otherwise, for $M = |X|$, $M' = |Y|$ and $N = |X \times Y| = MM'$, we have

$$\begin{aligned}
\Pr\left[K \leftarrow_{\$} \mathcal{K} : U \in \mathcal{B}_K\right] &\leq \frac{(1+\varepsilon)}{\binom{M-1}{D-1}} \cdot \frac{1}{(M')^{D-1}} \\
&= \frac{(1+\varepsilon)}{\binom{N-1}{D-1}} \cdot \frac{\binom{N-1}{D-1}}{\binom{M-1}{D-1}(M')^{D-1}} \\
&\leq \frac{(1+\varepsilon)}{\binom{N-1}{D-1}} \cdot \frac{M^{D-1}}{(M-1)_{D-1}} \\
&= \frac{(1+\varepsilon)}{\binom{N-1}{D-1}} \prod_{i=1}^{D-1} \frac{1}{1 - \frac{i}{M}} \\
&\leq \frac{(1+\varepsilon)}{\binom{N-1}{D-1}} \cdot \frac{1}{1 - \frac{D^2}{2M}} \\
&\leq \frac{(1+\varepsilon)(1 + \frac{D^2}{M})}{\binom{N-1}{D-1}}
\end{aligned}$$

if $D^2 \leq M$.

5 Concrete instantiation of the PM shuffle

In this section, we present a concrete instantiation of an n-bit PM shuffle based on a binary $[2^s - 1, 2^s - s - 1, 3]$-Hamming code $\mathcal{C}_s$. Suppose that $n \geq 2^s - 1$ and let $D = 2^s$.

A single round of the resulting PM shuffle Given a key

$$K = (K_1, \ldots, K_{D-1}, \mathbf{b}) \in \mathcal{K}$$

then the $(D-1) \times n$ matrix L with the i-th row being K_i, $i = 1, \ldots, D-1$, can be transformed into a reduced row echelon form $H = (h_{ij})$, where we can also compute and record a $(D-1) \times (D-1)$ invertible matrix $M = (m_{ij})$ such that

$$ML = H.$$

This computation, using the elementary row operations, would not be costly in general, and might be precomputed prior to encryption of data. Let $j_1, \ldots, j_{D-1}$ denote the column indices of the leading ones in H. So $h_{\alpha, j_\alpha} = 1$ for $\alpha = 1, \ldots, D-1$.

Given an input $\mathbf{u} = (u_1, \ldots, u_n) \in \{0,1\}^n$, the representative of the coset containing $\mathbf{u}$ is defined by setting the elements at the positions of the leading ones to *zero*. Namely, the representative $\mathbf{a}$ is computed by

$$\mathbf{a} = \mathbf{u} + u_{j_1} H_1 + \cdots + u_{j_{D-1}} H_{D-1}$$

where H_i denotes the i-th row of H. Since

$$H_i = m_{i1} K_1 + \cdots + m_{i,D-1} K_{D-1}$$

for $i = 1, \ldots, D-1$, we can also compute $p_1, \ldots, p_{D-1} \in \{0,1\}$ such that

$$\mathbf{a} = \mathbf{u} + p_1 K_1 + \cdots + p_{D-1} K_{D-1}$$

or equivalently,

$$\mathbf{u} = \mathbf{a} + \mathbf{b} + (b_1 + p_1) K_1 + \cdots + (b_{D-1} + p_{D-1}) K_{D-1}.$$

Precisely, for $i = 1, \ldots, D-1$,

$$p_i = u_{j_1} m_{1,i} + u_{j_2} m_{2,i} + \cdots + u_{j_{D-1}} m_{D-1,i}.$$

By decoding the word $(b_1 + p_1, \ldots, b_{D-1} + p_{D-1})$ using the Hamming code $\mathcal{C}_s$, we can obtain a codeword $\mathbf{c} = (c_1, \ldots, c_{D-1})$ and the corresponding error vector

$$\mathbf{e} = (e_1, \ldots, e_{D-1}) = (b_1 + p_1 + c_1, \ldots, b_{D-1} + p_{D-1} + c_{D-1})$$

such that $\mathbf{wt}(\mathbf{e}) \leq 1$. This step is essentially to compute the syndrome of the word $(b_1 + p_1, \ldots, b_{D-1} + p_{D-1})$ using the parity check matrix of $\mathcal{C}_s$. Then we have

$$\mathbf{u} = \mathbf{a} + (b_1 + c_1 + e_1) K_1 + \cdots + (b_{D-1} + c_{D-1} + e_{D-1}) K_{D-1}$$

and the block containing $\mathbf{u}$ is labeled as $(\mathbf{a}, \mathbf{c}) \in \{0,1\}^n \times \{0,1\}^{D-1}$. The position of one in $\mathbf{e}$ can be encoded as an element of $\{0,1\}^s$, with no error being regarded as $(0, \ldots, 0) \in \{0,1\}^s$.

By applying the round permutation $\sigma_{\mathbf{a},\mathbf{c}}$ to $\mathbf{e}$,[2] a new error vector $\mathbf{e}' = (e'_1, \ldots, e'_{D-1})$ such that $\mathbf{wt}(\mathbf{e}') \leq 1$ is obtained, and finally the element $\mathbf{u}$ is mapped to

$$\mathbf{u}' = \mathbf{a} + (b_1 + c_1 + e'_1) K_1 + \cdots + (b_{D-1} + c_{D-1} + e'_{D-1}) K_{D-1}.$$

PSEUDOCODE. Suppose that the r-round PM^r cipher uses an s-bit *tweakable* permutation

$$\sigma : \Big(\{0,1\}^n \times \{0,1\}^{D-1} \times \{1, \ldots, r\} \Big) \times \{0,1\}^s \longrightarrow \{0,1\}^s$$

[2] When we look at the security proof, the permutation family σ do not need to be independent for every distinct key K; they are required to be independent only for every block once a partition is fixed.

Encryption PM($\mathbf{w}$)

1: $\mathbf{u} \leftarrow \mathbf{w}$ where $\mathbf{u} = (u_1, \ldots, u_n)$
2: **for** $t \leftarrow 1$ to r **do**
3: set L to the matrix such that the i-th row is $K_{t,i}$ for $i = 1, \ldots, D-1$
4: compute $M = (m_{i,j})$ such that ML is in reduced row echelon form
5: $H \leftarrow ML$ where the i-th row of H is denoted H_i for $i = 1, \ldots, D-1$
6: compute the positions $j_1, \ldots, j_{D-1} \in [n]$ of the leading one's in H
7: $\mathbf{a} \leftarrow \mathbf{u} + u_{j_1} H_1 + \cdots + u_{j_{D-1}} H_{D-1}$
8: **for** $i \leftarrow 1$ to $D-1$ **do**
9: $p_i \leftarrow u_{j_1} m_{1,i} + u_{j_2} m_{2,i} + \cdots + u_{j_{D-1}} m_{D-1,i}$
10: $\mathbf{p} \leftarrow (p_1, \ldots, p_{D-1})$
11: decode $\mathbf{b} + \mathbf{p}$ obtaining the error vector $\mathbf{e}$ such that $\mathbf{wt}(\mathbf{e}) = 1$
12: $\mathbf{c} \leftarrow \mathbf{b} + \mathbf{p} + \mathbf{e}$ where $\mathbf{c} \in \mathcal{C}_s$
13: **if** $\exists\, j \in \{1, \ldots, D-1\}$ such that $e_j = 1$ **then**
14: $j^* \leftarrow j$
15: **else**
16: $j^* \leftarrow 0$
17: $\mathbf{x} \leftarrow \langle j^* \rangle_s$
18: $\mathbf{y} \leftarrow \sigma_{\mathbf{a},\mathbf{c},t}(\mathbf{x})$ where $\mathbf{y} = (y_1, \ldots, y_{s-1})$
19: $j^{**} \leftarrow 2^s y_{s-1} + \cdots + 2y_2 + y_1$
20: $\mathbf{e}' \leftarrow \mathbf{e}(j^{**})$
21: $\mathbf{u} \leftarrow (\mathbf{b} + \mathbf{c} + \mathbf{e}')L + \mathbf{a}$
22: **return** $\mathbf{u}$

Fig. 2 The r-round PM shuffle based on a binary $[2^s - 1, 2^s - s - 1, 3]$-Hamming code

as its underlying primitive. Then PM^r encrypts $\mathbf{w} \in \{0,1\}^n$ using a set of t round keys

$$\left(K_{t,1}, \ldots, K_{t,D-1}, \mathbf{b}_t\right)_{t \in [r]} \in \left(\left(\{0,1\}^n\right)^{D-1} \times \{0,1\}^{D-1}\right)^r$$

as described in Fig. 2.

NUMERICAL EXAMPLE. Let $s = 3$, $n = 32$ and $r = 512$. Then one needs a 3-bit block cipher using 48-bit tweaks for the underlying primitive σ. This small block cipher can be instantiated using a tweakable block cipher, e.g., Skinny-128-256 [1]. For each round, one makes a single call to Skinny-128–256 with a fixed plaintext using a 256-bit tweakey containing the 48-bit tweak, obtaining a 128-bit random string, from which one can construct a random permutation on 3 bits. A straightforward way of constructing such a permutation is to parse the 128-bit string into a sequence of eight 16-bit blocks. If there is no collision between the blocks, then the sequence defines a permutation on $\{0,1\}^3$. The probability of collision is upper bounded by $\binom{8}{2}/2^{16}$, which is smaller than $\frac{1}{2^{11}}$.

Lines 3 to 6 in the pseudocode can be precomputed for every round $t \in [r]$ if a sufficient amount of memory is available. Line 11 can be executed using the syndrome decoding: the generator matrix of the $[7, 4, 3]$-Hamming code (for $s = 3$) is given as

$$G = \begin{bmatrix} 1\,0\,0\,0\,1\,1\,0 \\ 0\,1\,0\,0\,1\,0\,1 \\ 0\,0\,1\,0\,0\,1\,1 \\ 0\,0\,0\,1\,1\,1\,1 \end{bmatrix}$$

and its parity-check matrix is defined as

$$G^* = \begin{bmatrix} 1\,1\,0\,1\,1\,0\,0 \\ 1\,0\,1\,1\,0\,1\,0 \\ 0\,1\,1\,1\,0\,0\,1 \end{bmatrix}.$$

By computing $(\mathbf{b}+\mathbf{p})(G^*)^T$, one obtains the 3-bit *syndrome* of $\mathbf{b}+\mathbf{p}$, where $(G^*)^T$ denotes the transpose of G^*. The syndrome of $\mathbf{b}+\mathbf{p}$ specifies the exact position of the single bit error in $\mathbf{b}+\mathbf{p}$ (if any), allowing one to recover the corresponding codeword $\mathbf{c}$ and the error vector $\mathbf{e}$ such that $\mathbf{c}+\mathbf{e}=\mathbf{b}+\mathbf{p}$.

References

1. Beierle C., Jean J., Kšlbl S., Leander G., Moradi A., Peyrin T., Sasaki Y., Sasdrich P., Sim S.M.: The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Advances in Cryptology-CRYPTO 2016, pp. 123–153. Springer, Berlin Heidelberg (2016).
2. Bellare M., Ristenpart T., Rogaway P., Stegers T.: Format-preserving encryption. In: Selected Areas in Cryptography, pp. 295–312. Springer, Berlin Heidelberg (2009).
3. Bellare M., Rogaway P., Spies T.: The FFX mode of operation for format-preserving encryption. Unpublished NIST proposal (2010)
4. Brier E., Peyrin T., Stern J. BPS: a format-preserving encryption proposal. Submission to NIST, available from their website (2010).
5. Chen S., Steinberger J.: Tight security bounds for key-alternating ciphers. In: Advances in Cryptology-EUROCRYPT 2014, pp. 327–350. Springer, Berlin Heidelberg (2014).
6. Coron J.S., Dodis Y., Mandal A., Seurin Y.: A domain extender for the ideal cipher. In: Theory of Cryptography, pp. 273–289. Springer, Berlin Heidelberg (2010).
7. Even S., Mansour Y.: A construction of a cipher from a single pseudorandom permutation. J. Cryptol. **10**(3), 151–161 (1997).
8. Hoang V.T., Morris B., Rogaway P.: An enciphering scheme based on a card shuffle. In: Advances in Cryptology-CRYPTO 2012, pp. 1–13. Springer, Berlin Heidelberg (2012).
9. Maurer U., Pietrzak K., Renner R.: Indistinguishability amplification. In: Advances in Cryptology-CRYPTO 2007, pp. 130–149. Springer, Berlin Heidelberg (2007).
10. Morris B., Rogaway P., Stegers T.: How to encipher messages on a small domain. In: Advances in Cryptology-CRYPTO 2009, pp. 286–302. Springer, Berlin Heidelberg (2009).
11. Morris B., Rogaway P.: Sometimes-Recurse Shuffle. In: Advances in Cryptology-EUROCRYPT 2014, pp. 311–326. Springer, Berlin Heidelberg (2014).
12. Patarin J.: Luby-Rackoff: 7 rounds are enough for $2n(1-\varepsilon)$ security. In: Advances in Cryptology-CRYPTO 2003, pp. 513–529. Springer, Berlin Heidelberg (2003).
13. Patarin J.: Security of random Feistel schemes with 5 or more rounds. In: Advances in Cryptology-CRYPTO 2004, pp. 106–122. Springer, Berlin Heidelberg (2004).
14. Ristenpart T., Yilek S.: The mix-and-cut shuffle: small-domain encryption secure against N queries. In: Advances in Cryptology-CRYPTO 2013, pp. 392–409. Springer, Berlin Heidelberg (2013).
15. Stefanov E., Shi E.: FastPRP: fast pseudo-random permutations for small domains. IACR Cryptol. **2012**, 254 (2012).
16. Thorp E.O.: Nonrandom shuffling with applications to the game of Faro. J. Am. Stat. Assoc. **68**(344), 842–847 (1973).