An Enciphering Scheme Based on a Card Shuffle

Viet Tung Hoang[1] Ben Morris[2] Phillip Rogaway[1]

[1] Dept. of Computer Science, University of California, Davis, USA
[2] Dept. of Mathematics, University of California, Davis, USA

Abstract. We introduce the *swap-or-not shuffle* and show that the technique gives rise to a new method to convert a pseudorandom function (PRF) into a pseudorandom permutation (PRP) (or, alternatively, to directly build a confusion/diffusion blockcipher). We then prove that swap-or-not has excellent quantitative security bounds, giving a Luby-Rackoff type result that ensures security (assuming an ideal round function) to a number of adversarial queries that is nearly the size of the construction's domain. Swap-or-not provides a direct solution for building a small-domain cipher and achieving format-preserving encryption, yielding the best bounds known for a practical scheme for enciphering credit-card numbers. The analysis of swap-or-not is based on the theory of mixing times of Markov chains.

Keywords: Blockciphers, Feistel network, Luby-Rackoff, Markov chain, PRF-to-PRP conversion, pseudorandom permutations, swap-or-not.

1 Introduction

Overview. Despite the diversity of proposed blockciphers, only two approaches underlie the construction of real-world designs: essentially everything looks like some sort of Feistel network (e.g., DES, FEAL, MARS, RC6) or SP-network (e.g., Rijndael, Safer, Serpent, Square). Analogously, in the literature on constructing pseudorandom permutations (PRPs) from pseudorandom functions (PRFs), we have provable-security analyses for Feistel variants (e.g., [12–14, 18, 21]), as well as modes of operation (e.g., [10, 11, 18, 19]) that can again be construed as SP-networks, now on a large domain. Perhaps there just are not that many fundamentally different ways to make a blockcipher. Or perhaps we might have failed to notice *other* possibilities.

In this short paper we describe a very different way to make a blockcipher. We call it a *swap-or-not* network (or cipher or shuffle). Besides introducing the construction, we evidence its cryptographic utility. We do this by showing that swap-or-not provides the quantitatively best mechanism known, in terms of concrete security bounds, to convert a PRF into a PRP. We also show that swap-or-not provides a practical solution for the problem of format-preserving encryption (FPE) on domains of troublesome size, such as enciphering credit-card numbers.

```
proc $E_{KF}(X)$                    //**swap-or-not**
for $i \leftarrow 1$ to $r$ do
    $X' \leftarrow K_i \oplus X$
    $\hat{X} \leftarrow \max(X, X')$
    if $F_i(\hat{X}) = 1$ then $X \leftarrow X'$
return $X$
```

Fig. 1. Cipher $E = \mathrm{SN}[r, n]$ encrypts $X \in \{0,1\}^n$ using a key KF naming $K_1, \ldots, K_r \in \{0,1\}^n$ and round functions $F_1, \ldots, F_r \colon \{0,1\}^n \to \{0,1\}$.

Construction. Suppose we aim to encipher n-bit strings; our message space is the set $\mathcal{X} = \{0,1\}^n$. Assume we will use r rounds, and that the blockcipher's key KF names subkeys $K_1, \ldots, K_r \in \{0,1\}^n$ as well as round functions $F_1, \ldots, F_r$, each of which maps n-bits to a single bit, so $F_i : \{0,1\}^n \to \{0,1\}$. Then we encipher $X \in \{0,1\}^n$ as shown in Fig. 1. The reason that this works, that one gets a permutation, is simply that $X \mapsto K_i \oplus X$ is an involution, and our round function depends on the set $\{X, K_i \oplus X\}$. The inverse direction for swap-or-not is identical to the forward direction shown above except for having i run from r down to 1.

Restating the algorithm in English, at each round i we pair the current value of $X \in \{0,1\}^n$ with a "partner" point $X' = K_i \oplus X$. We either replace X by its partner or leave it alone. Which of these two things we do is determined by applying the boolean-valued F_i to the two-element set $\{X, X'\}$. Actually, in order to give F_i a more conventional domain, we select a canonical representative from $\{X, X'\}$, say $\hat{X} = \max(X, X')$, and apply F_i to it. Note that each plaintext maps to a ciphertext by xoring into it some subset of the subkeys $\{K_1, \ldots, K_r\}$. This might sound linear, but it most definitely is not.

Card shuffling view. The swap-or-not construction was invented, and will be analyzed, by regarding it as a way to shuffle a deck of cards. Seeing a blockcipher as a card shuffle enables one to exploit a large body of mathematical techniques, these dating back to the first half of the twentieth century. In addition, some ways to shuffle cards give rise to enciphering schemes that cryptographers did not consider. Swap-or-not is such a case.

One can always see a card shuffle as an enciphering scheme, and vice versa. If you have some method to shuffle N cards, this determines a corresponding way to encipher N points: place a card at each position $X \in [N]$, where $[N] = \{0, 1, \ldots, N-1\}$; shuffle the deck; then look to see the position where the card initially at position X ended up. Call that position the ciphertext Y for X. The randomness used in the shuffle corresponds the cipher's key.

The first thing needed for a card shuffle to give rise to a computationally feasible blockcipher is that the shuffle be *oblivious*, an idea suggested by Moni Naor [18, p. 62], [23, p. 17]. In an oblivious shuffle one can trace the trajectory of a card without attending to lots of *other* cards in the deck. Most conventional shuffles, such as the riffle shuffle, are not oblivious. The Thorp shuffle [26] is oblivious—and so is swap-or-not. As a shuffle, here's how it looks.

```
$K \xleftarrow{\$} \{0,1\}^n$            //swap-or-not as a shuffle
for each pair of positions $\{X, K \oplus X\}$
    $b \xleftarrow{\$} \{0,1\}$
    if $b = 1$ then swap the cards
        at positions $X$ and $K \oplus X$
```

Fig. 2. Mixing a deck of $N = 2^n$ cards, each at a position $X \in \{0,1\}^n$. The code shows one shuffle. For better mixing, the shuffle is repeated r times.

Recasting swap-or-not as a way to shuffle cards, suppose we have N cards, one at each position $X \in [N]$, where $N = 2^n$. To shuffle the deck, choose a random $K \in \{0,1\}^n$ and then, for each pair of card positions X and $K \oplus X$, flip a fair coin. If it lands heads, swap the cards at the indicated positions; if it lands tails, leave them alone. See Fig. 2. The process can be repeated any number r times, using independent coins (both the K-values and the b-values) for each shuffle.

When the swap-or-not shuffle of Fig. 2 is translated back into the language of encryption, one recovers the swap-or-not cipher of Fig. 1; these are different views of precisely the same process. The random pairing-up of cards specified by K for the ith shuffle corresponds to the subkey K_i. The random bit b flipped at the shuffle's round i for the pair $\{X, K \oplus X\}$ corresponds $F_i(\hat{X})$.

```
proc $E_{KF}(X)$                //Generalized domain
for $i \leftarrow 1$ to $r$ do
    $X' \leftarrow K_i - X$
    $\hat{X} \leftarrow \max(X, X')$
    if $F_i(\hat{X}) = 1$ then $X \leftarrow X'$
return $X$
```

Fig. 3. Cipher $E = \mathrm{SN}[r, N, +]$ encrypts $X \in [N]$ using a key KF naming $K_1, \ldots, K_r \in [N]$ and round functions $F_1, \ldots, F_r\colon [N] \to \{0,1\}$.

Generalizing. It is useful to be a bit more general here, working in a finite abelian group $G = ([N], +)$ instead of the group $(\{0,1\}^n, \oplus)$ of bit strings under xor. (For convenience, we have assumed that the group elements are named $[N] = \{0, \ldots, N-1\}$.) In this way we won't need the number of points N in the message space $\mathcal{X} = [N]$ to be a power of two—we'll be able to encipher points on any set $\mathcal{X} = [N]$, just by naming a group operator, say addition modulo N. For generalizing the shuffle of Fig. 2, the value K is uniformly drawn from $[N]$ rather than from $\{0,1\}^n$, and we consider the pair of positions $\{X, K - X\}$ rather than $\{X, K \oplus X\}$. For the generalized cipher—see Fig. 3—the key KF will name subkeys $K_1, \ldots, K_r \in [N]$ and round functions $F_1, \ldots, F_r\colon [N] \to \{0,1\}$. We set $X' \leftarrow K_i - X$ rather than $X' \leftarrow K_i \oplus X$. The inverse remains what one gets by iterating from r down to 1.

Results. As with Luby and Rackoff's seminal paper [14], we can analyze the swap-or-not construction by regarding its constituent parts as uniformly random. Formally, let us write $\mathrm{SN}[r, N, +]\colon \mathcal{K} \times [N] \to [N]$ for the blockcipher E specified in Fig. 3 that is swap-or-not with r rounds, a message space of $[N]$, the indicated group operator, and where the key space names all possible subkeys $K_1, \ldots, K_r \in [N]$ and all possible round functions $F_1, \ldots, F_r : [N] \to \{0,1\}$.

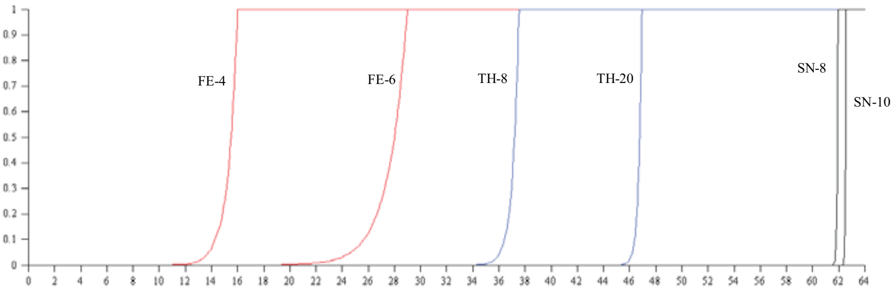

Fig. 4. Illustration of results. The message space has $N = 2^{64}$ points. The graphs show established upper bounds on CCA advantage when the adversary asks q queries, where $\log_2(q)$ labels the x-axis. **Rightmost two graphs**: the new results—the swap-or-not cipher for either eight passes (512 rounds) (SN-8) or 10 (SN-10), as given by Theorem 4. (One pass is defined as $\lceil \lg N \rceil$ rounds.) For comparison, the **leftmost two graphs** are for balanced Feistel, both the classical 4-round result of Luby and Rackoff [14, 20] (LR-4) and then a six-round result of Patarin (LR-6) [22, Th. 7]. The **middle two graphs** are for the Thorp shuffle, either with eight passes (TH-8) or 20 (TH-20), as given by [17, Th. 5].

Thus a random key KF for this cipher has the K_i and F_i values uniformly chosen. We define the CCA (also called the "strong-PRP") advantage of an adversary A attacking E by dropping it into one of two worlds. In the first, the adversary gets an oracle for $E_{KF}(\cdot)$, for a random KF, and also an oracle for its inverse, $E_{KF}^{-1}(\cdot)$. Alternatively, the adversary is given a uniformly random permutation $\pi\colon [N] \to [N]$, along with its inverse, $\pi^{-1}(\cdot)$. Define

$$\mathbf{Adv}^{\mathrm{cca}}_{\mathrm{SN}[r,N,+]}(q) \;=\; \max_{A} \left\{ \Pr[A^{E_{KF}(\cdot),\, E^{-1}_{KF}(\cdot)} \Rightarrow 1] - \Pr[A^{\pi(\cdot),\, \pi^{-1}(\cdot)} \Rightarrow 1] \right\},$$

the maximum over all adversaries that ask at most q total queries. Our main result is that

$$\mathbf{Adv}^{\mathrm{cca}}_{\mathrm{SN}[r,N,+]}(q) \;\le\; \frac{4N^{3/2}}{r+4} \left(\frac{q+N}{2N} \right)^{r/4+1} . \qquad (1)$$

Roughly said, you need $r = 6 \lg N$ rounds of swap-or-not to start to see a good bound on CCA-security. After that, the adversary's advantage drops off inverse exponentially in r. The summary explanation of formula (1) just given assumes that the number of adversarial queries is capped at $q = (1-\epsilon)N$ for some fixed $\epsilon > 0$.

The quantitative guarantee above is far stronger than anything a balanced Feistel network can deliver. The only remotely comparable bound we know, retaining security to $N^{1-\epsilon}$ queries instead of $(1-\epsilon)N$ queries, is the Thorp shuffle [26] (or, equivalently, a maximally-unbalanced Feistel network [17]). But the known result, establishing $\mathbf{Adv}^{\mathrm{cca}}_{E'}(q) \le (2q/r+1)(4nq/N)^r$ if one shuffles $N = 2^n$ points for $r(4n-2)$ rounds [17], vanishes by the time that $q \ge \frac{N}{4 \lg N}$.

Numerically, the Thorp-shuffle bounds come out much weaker for most r, q, and N. See Fig. 4 for sample graphs comparing known bounds on balanced Feistel, the Thorp shuffle, and swap-or-not.

As a simple numerical example, swap-or-not enciphering 64-bit strings for 1200 rounds using a random round function will yield a maximal CCA advantage of less than 10^{-10}, even if the adversary can ask $q = 2^{63}$ queries. While the number of rounds is obviously large, no other construction can deliver a comparable guarantee, achieving security even when q is close to N.

For a more complexity-theoretic discussion of swap-or-not, see Section 4.

Format-preserving encryption. Swap-or-not was originally invented as a solution for *format-preserving encryption* (FPE) [1, 3, 5], where it provides the best known solution, in terms of proven-security bounds, when N is too big to spend linear time computing, yet too small for conventional constructions to deliver desirable bounds. This landscape has not much changed with the recent work of Stefanov and Shi [24], who, following Granboulan and Pornin [9], show how to speed up (e.g., to $\tilde{\Theta}(N^{0.5})$ time) determining where a card goes in a particular N-card shuffle after spending $\tilde{\Theta}(N)$ time at key-setup. For more discussion of swap-or-not and its use in FPE, see Section 5.

2 Preliminaries

Total variation distance. Let μ and ν be probability distributions on Ω. The *total variation distance* between distributions μ and ν is defined as

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

Blockciphers. Let $E\colon \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ be a blockcipher, meaning that $\mathcal{K}$ and $\mathcal{M}$ are finite and each $E_K(\cdot) = E(K, \cdot)$ is a permutation on $\mathcal{M}$. We emphasize that $\mathcal{K}$ and $\mathcal{M}$ need not consist of binary strings of some particular length, as is often assumed to be the case. For any blockcipher E, we let E^{-1} be its inverse blockcipher.

For blockcipher $E\colon \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ and adversary A the *advantage* of A in carrying out an (adaptive) chosen-ciphertext attack (CCA) on E is

$$\mathbf{Adv}_E^{\mathrm{cca}}(A) = \Pr[K \xleftarrow{\$} \mathcal{K}\colon A^{E_K(\cdot), E_K^{-1}(\cdot)} \Rightarrow 1] - \Pr[\pi \xleftarrow{\$} \mathrm{Perm}(\mathcal{M})\colon A^{\pi(\cdot), \pi^{-1}(\cdot)} \Rightarrow 1].$$

Here $\mathrm{Perm}(\mathcal{M})$ is the set of all permutations on $\mathcal{M}$. We say that A carries out an (adaptive) chosen-plaintext attack (CPA) if it asks no queries to its second oracle. Adversary A is *non-adaptive* if it asks the same queries on every run. Let $\mathbf{Adv}_E^{\mathrm{cca}}(q)$ be the maximum advantage of any (adaptive) CCA adversary against E subject to the adversary asking at most q total oracle queries. Similarly define $\mathbf{Adv}_E^{\mathrm{ncpa}}(q)$ for nonadaptive CPA attacks (NCPA).

For blockciphers $F, G\colon \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ let $F \circ G$ denote their cascade, with F's output fed into G's input; formally, $F \circ G\colon \mathcal{K}^2 \times \mathcal{M} \to \mathcal{M}$ is defined by $(F \circ G)_{(K,K')} = G_{K'}(F_K(X))$.

Lifting ncpa to cca security. We bound the CCA-security of a Feistel network from its NCPA-security by using the following result of Maurer, Pietrzak, and Renner [15, Corollary 5]. It is key to our approach, effectively letting us assume that our adversaries are of the simple, NCPA breed. Recall that in writing $F \circ G$, the blockciphers are, in effect, independently keyed.

Lemma 1 (Maurer-Pietrzak-Renner) *If F and G are blockciphers on the same message space then, for any q, $\mathbf{Adv}^{\mathrm{cca}}_{F \circ G^{-1}}(q) \le \mathbf{Adv}^{\mathrm{ncpa}}_{F}(q) + \mathbf{Adv}^{\mathrm{ncpa}}_{G}(q)$.*

3 Security of Swap-Or-Not

Fix a finite abelian group $G = ([N], +)$ where $[N] = \{0, 1, \ldots, N-1\}$. We define the swap-or-not shuffle $\mathrm{SN}[r, N, +]$ of r rounds over the elements of G. The shuffling at round t is as follows. Initially, each of N distinct cards is at a position in the set $[N]$. To shuffle during this round, choose $K_t \xleftarrow{\$} [N]$, the *subkey* at round t. Then, for each set $\{X, K_t - X\}$ with $X \in G$, choose $b \xleftarrow{\$} \{0, 1\}$ and then swap the cards at positions X and $K_t - X$ if $b = 1$.

Let $\{W_t : t \ge 0\}$ be the Markov chain representing the swap-or-not shuffle with N cards. More formally, let $\mathcal{C}$ be a set of cardinality N, whose elements we call *cards.* The state space of $\{W_t\}$ is the set of bijections from $\mathcal{C}$ to $\{0, \ldots, N-1\}$. For a card $z \in \mathcal{C}$, we interpret $W_t(z)$ as the position of card z at time t.

Let A be a deterministic adversary that makes exactly q queries. Our proof is based on an analysis of the mixing rate of the swap-or-not shuffle. However, since A makes only $q \le N$ queries, we need only bound the rate at which some q-element subset of the cards mixes. So let $z_1, \ldots, z_q$ be distinct cards in $\mathcal{C}$, and let X_t be the vector of positions of cards $z_1, \ldots, z_q$ at time t. For j in $\{1, \ldots, q\}$ we write $X_t(j)$ for the position of card z_j at time t, and define $X_t(1, \ldots, j) = (X_t(1), \ldots, X_t(j))$. We shall call X_t the *projected swap-or-not shuffle.* Note that the stationary distribution of X_t, which we denote by π, is uniform over the set of distinct q-tuples of elements from G. Equivalently, π is the distribution of q samples without replacement from G. Let τ_t denote the distribution of X_t.

Theorem 2 (Rapid mixing). *Consider the swap-or-not shuffle* $\mathrm{SN}[r, N, +]$ *for $r, N \ge 1$, and let $q \in \{1, \ldots, N\}$. Fix $z_1, \ldots, z_q$ and let $\{X_t \colon t \ge 0\}$ be the corresponding projected swap-or-not shuffle, let π be its stationary distribution, and let τ_t be the distribution of X_t. Then*

$$\|\tau_r - \pi\| \le \frac{2N^{3/2}}{r+2} \left(\frac{q+N}{2N} \right)^{r/2+1} .$$

Proof. Let τ_t^k be the conditional distribution of X_t given the subkeys $K_1, \ldots, K_r$. (Here we consider $K_1, \ldots, K_r$ random variables, and we condition on the σ-algebra of these random variables.) We will actually show that $\mathbf{E}(\|\tau_r^k - \pi\|)$ satisfies the claimed inequality. Note that since $K_1, \ldots, K_r$ are random variables, so is τ_r^k, and hence so is $\|\tau_r^k - \pi\|$. This implies the theorem since $\tau_r = \mathbf{E}(\tau_r^k)$ and hence

$$\|\tau_r - \pi\| = \|\mathbf{E}(\tau_r^k - \pi)\| \leq \mathbf{E}\Big(\|\tau_r^k - \pi\|\Big),$$

by Jensen's inequality, since for distributions μ and τ, the total variation distance $\|\mu - \tau\|$ is half the L^1-norm of $\mu - \tau$, and the L^1-norm is convex. For a distribution ν on q-tuples of Ω, define

$$\begin{aligned} \nu(u_1, \ldots, u_j) &= \Pr[Z_1 = u_1, \ldots, Z_j = u_j] \text{ and} \\ \nu(u_j \mid u_1, \ldots, u_{j-1}) &= \Pr[Z_j = u_j \mid Z_1 = u_1, \ldots, Z_{j-1} = u_{j-1}] \end{aligned}$$

where $(Z_1, \ldots, Z_q) \sim \nu$. For example, $\tau_t(u_1, \ldots, u_j)$ is the probability that, in the swap-or-not shuffle, cards $z_1, \ldots, z_j$ land in positions $u_1, \ldots, u_j$ at time t, while $\tau_t(u_j \mid u_1, \ldots, u_{j-1})$ is the probability that at time t card z_j is in position u_j given that cards $z_1, \ldots, z_{j-1}$ are in positions $u_1, \ldots, u_{j-1}$. On the other hand, $\pi(u_j \mid u_1, \ldots u_{j-1})$ is the probability that, in a uniform random ordering, card z_j is in position u_j given that cards $z_1, \ldots, z_{j-1}$ land in positions $u_1, \ldots, u_{j-1}$.

Each of the conditional distributions $\tau_t^k(\,\cdot \mid u_1, \ldots, u_{j-1})$ converges to uniform as $t \to \infty$. When all of these distributions are "close" to uniform, then τ_t^k will be close to π. In fact, we only need the conditional distributions to be close "on average," as is formalized in the following lemma, which is easily established using coupling. For a proof, see [17, Appendix A].

Lemma 3 *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*

$$\|\mu - \nu\| \;\leq\; \sum_{l=0}^{q-1} \mathbf{E}\Big(\|\mu(\,\cdot \mid Z_1, \ldots, Z_\ell) - \nu(\,\cdot \mid Z_1, \ldots, Z_\ell)\|\Big)\,. \tag{2}$$

Note that in the above lemma, since $Z_1, \ldots, Z_q$ are random variables (whose joint distribution is given by μ), so is $\|\mu(\,\cdot \mid Z_1, \ldots, Z_\ell) - \nu(\,\cdot \mid Z_1, \ldots, Z_\ell)\|$ for every $\ell < q$; each summand in the right-hand side of (2) is the expectation of one of these random variables.

Recall that τ_t^k is the conditional distribution of X_t given $K_1, \ldots, K_r$. Fix $\ell \in \{0, \ldots, q-1\}$. We wish to bound the expected distance between the distribution $\tau_t^k(\,\cdot \mid X_t(1), \ldots, X_t(\ell))$ and $\pi(\,\cdot \mid X_t(1), \ldots, X_t(\ell))$ (i.e., the uniform distribution on $G \setminus \{X_t(1), \ldots, X_t(\ell)\}$).

For $t \geq 0$, let $S_t = G \setminus \{X_t(1), \ldots, X_t(\ell)\}$. Thus S_t is the set of positions that card $z_{\ell+1}$ could be located in at time t, given the positions of cards $z_1, ..., z_\ell$. For $a \in S_t$, let $p_t(a) = \tau_t^k(a \mid X_t(1), \ldots, X_t(\ell))$. Then we have

$$\|\tau_t^k(\,\cdot \mid X_t(1, \ldots, \ell)) - \pi(\,\cdot \mid X_t(1, \ldots, \ell))\| = \frac{1}{2} \sum_{a \in S_t} |p_t(a) - 1/m|, \tag{3}$$

where $m = |S_t| = N - \ell$. Using the Cauchy-Schwarz inequality twice gives

$$\begin{aligned}\left(\mathbf{E}\left[\sum_{a\in S_t} |p_t(a) - 1/m|\right]\right)^2 &\le \mathbf{E}\left[\left(\sum_{a\in S_t} |p_t(a) - 1/m|\right)^2\right]\\ &\le m\cdot\mathbf{E}\left[\sum_{a\in S_t}(p_t(a)-1/m)^2\right]\\ &\le N\cdot\mathbf{E}\left[\sum_{a\in S_t}(p_t(a)-1/m)^2\right] \ . \qquad (4)\end{aligned}$$

We shall prove, by induction on t, that

$$\mathbf{E}\left[\sum_{a\in S_t}(p_t(a)-1/m)^2\right] \le \left(\frac{\ell+N}{2N}\right)^t \qquad (5)$$

for every $t \le r$. Then, substituting $t = r$ to (3), (4), and (5), we have

$$\begin{aligned}&\mathbf{E}\Big(\|\tau_r^k(\,\cdot \mid X_r(1,\ldots,\ell)) - \pi(\,\cdot \mid X_r(1,\ldots,\ell))\|\Big)\\ &\le \frac{1}{2}\left(N\cdot\mathbf{E}\left[\sum_{a\in S_r}(p_r(a)-1/m)^2\right]\right)^{1/2} \le \frac{\sqrt{N}}{2}\left(\frac{\ell+N}{2N}\right)^{r/2} \ .\end{aligned}$$

Substituting this into Lemma 3 gives

$$\begin{aligned}\mathbf{E}\Big(\|\tau_r^k - \pi\|\Big) &\le \sum_{\ell=0}^{q-1}\mathbf{E}\Big(\|\tau_r^k(\,\cdot \mid X_r(1,\ldots,\ell)) - \pi(\,\cdot \mid X_r(1,\ldots,\ell))\|\Big)\\ &\le \sum_{\ell=0}^{q-1}\frac{\sqrt{N}}{2}\left(\frac{\ell+N}{2N}\right)^{r/2}\\ &\le N^{3/2}\int_0^{q/2N}(1/2+x)^{r/2}dx \le \frac{2N^{3/2}}{r+2}\left(\frac{q+N}{2N}\right)^{r/2+1} \ .\end{aligned}$$

We now verify equation (5). First, consider the base case $t = 0$. Since the initial positions of the cards are deterministic,

$$\mathbf{E}\Big[\sum_{a\in S_0}(p_0(a)-1/m)^2\Big] = (1-1/m)^2 + (m-1)\cdot(0-1/m)^2 = 1 - 1/m < 1 \ .$$

Now suppose that equation (5) holds for t. We prove that it also holds for $t+1$. Define $s_t = \sum_{a\in S_t}(p_t(a)-1/m)^2$. It is sufficient to show that

$$\mathbf{E}(s_{t+1} \mid s_t) = \left(\frac{\ell+N}{2N}\right)s_t. \qquad (6)$$

Define $f : S_t \to S_{t+1}$ by

$$f(a) = \begin{cases} a & \text{if } a \in S_{t+1}; \\ K_{t+1} - a & \text{otherwise.} \end{cases}$$

Note that f is a bijection from S_t to S_{t+1}: it sends S_t to S_{t+1} because if $a \in S_t$ then either a or $K_{t+1} - a$ must be in S_{t+1}, and it has an inverse $f^{-1} : S_{t+1} \to S_t$ defined by

$$f^{-1}(b) = \begin{cases} b & \text{if } b \in S_t; \\ K_{t+1} - b & \text{otherwise.} \end{cases}$$

Furthermore, note that

$$p_{t+1}(f(a)) = \begin{cases} p_t(a) & \text{if } K_{t+1} - a \notin S_t; \\ \frac{1}{2} p_t(a) + \frac{1}{2} p_t(K_{t+1} - a) & \text{otherwise.} \end{cases}$$

Since K_{t+1} is independent of the process up to time t, for every $y \in G$, we have $\Pr[K_{t+1} - a = y \mid s_t] = 1/N$. Hence, since $|S_t| = m$, conditioning on the value of $K_{t+1} - a$ gives

$$\mathbf{E}\Big(\Big[p_{t+1}(f(a)) - \tfrac{1}{m}\Big]^2 \mid s_t\Big) = \frac{\ell}{N}\left(p_t(a) - \tfrac{1}{m}\right)^2 + \frac{1}{N}\sum_{y \in S_t}\Big[\frac{p_t(a) + p_t(y)}{2} - \tfrac{1}{m}\Big]^2. \quad (7)$$

The sum can be rewritten as

$$\sum_{y \in S_t} \tfrac{1}{4}\Big[(p_t(y) - 1/m) + (p_t(a) - 1/m)\Big]^2$$
$$= \tfrac{1}{4}\sum_{y \in S_t}(p_t(y) - 1/m)^2 + \tfrac{1}{2}(p_t(a) - 1/m)\sum_{y \in S_t}(p_t(y) - 1/m) + \tfrac{1}{4}\sum_{y \in S_t}(p_t(a) - 1/m)^2$$
$$= \tfrac{1}{4}s_t + \tfrac{m}{4}(p_t(a) - 1/m)^2,$$

since $\sum_{y \in S_t}(p_t(y) - 1/m) = 0$. Combining this with (7) gives

$$\mathbf{E}\Big(\Big[p_{t+1}(f(a)) - 1/m\Big]^2 \mid s_t\Big) = \frac{s_t}{4N} + \frac{4\ell + m}{4N}(p_t(a) - 1/m)^2. \quad (8)$$

Note that

$$\begin{aligned} \mathbf{E}(s_{t+1} \mid s_t) &= \sum_{b \in S_{t+1}} \mathbf{E}\Big(\Big[p_{t+1}(b) - 1/m\Big]^2 \mid s_t\Big) \\ &= \sum_{a \in S_t} \mathbf{E}\Big(\Big[p_{t+1}(f(a)) - 1/m\Big]^2 \mid s_t\Big). \end{aligned}$$

Evaluating each term in the sum using (8) gives

$$\begin{aligned} \mathbf{E}(s_{t+1} \mid s_t) &= \frac{m s_t}{4N} + \frac{4\ell + m}{4N}\sum_{a \in S_t}(p_t(a) - 1/m)^2 \\ &= \frac{m s_t}{4N} + \frac{(4\ell + m)s_t}{4N} \\ &= \frac{\ell + N}{2N} s_t, \end{aligned}$$

where the last line holds because $m + \ell = N$. It follows that $\mathbf{E}(s_{t+1} \mid s_t) = \left(\frac{\ell+N}{2N}\right)s_t$, which verifies (6) and hence (5). This completes the proof. □

CCA-SECURITY. Observe that if $E = \mathrm{SN}[r, N, +]$ for some abelian group $G = ([N], +)$ then E^{-1} is also $\mathrm{SN}[r, N, +]$. Employing Lemma 1 we conclude our main theorem.

Theorem 4. *Let* $E = \mathrm{SN}[2r, N, +]$. *Then* $\mathbf{Adv}_E^{\mathrm{cca}}(q) \le \dfrac{4N^{3/2}}{r+2}\left(\dfrac{q+N}{2N}\right)^{r/2+1}$.

4 Complexity-Theoretic Interpretation

While Theorem 4 is information-theoretic, it should be clear that the result applies to the complexity-theoretic setting too, in exactly the same manner as Luby-Rackoff [14] and its successors. Namely, from a PRF $F\colon \mathcal{K} \times \{0,1\}^* \to \{0,1\}$ and a number n, define n-bit round functions $F_i(X)$ whose jth bit is $F(\langle i, j, n, X\rangle)$. Also define n-bit round keys K_i whose jth bit is $F(\langle i, j, n\rangle)$. Using these components, apply the swap-or-not construction for, say, $r = 7n$ rounds, yielding a PRP E on n bits. Translating the information-theoretic result into this setting, the PRP-security of E is the PRF-security of F minus a term that remains negligible until $q = (1-\epsilon)2^n$ adversarial queries, for any $\epsilon > 0$. That is, from the asymptotic point of view, the swap-or-not construction preserves essentially all of a PRF's security in the constructed PRP.

We emphasize that our security results only cover the (strong) PRP notion of security. An interesting question we leave open is whether the swap-or-not cipher is indifferentiable from a random permutation [16]. Following Coron, Patarin, and Seurin [6], Holenstein, Künzler, and Tessaro show that the 14-round Feistel construction is indifferentiable from a random permutation [12]. But their proof is complex and delivers very poor concrete-security bounds. It would be desirable to have a construction supporting a simpler proof with better bounds.

5 Format-Preserving Encryption

In the *format-preserving encryption* (FPE) problem, one wants to encipher on an arbitrary set $\mathcal{X}$, often $\mathcal{X} = [N]$ for some number N. Usually constructions are sought that start from a conventional blockcipher, like AES. The problem has attracted increasing interest [1–5, 8, 9, 17, 24, 25, 27], and is the subject of ongoing standardization work by NIST and the IEEE.

When N is sufficiently small that one can afford $\tilde{\Omega}(N)$-time to encrypt, provably good solutions are easy, by directly realizing a random shuffle [3]. And when N is sufficiently large that no adversary could ask anything near $N^{1/2}$ queries, nice solutions are again easy, using standard cryptographic constructions like multi-round Feistel. But for intermediate-size domains, like those with 2^{30}–2^{60} points, the bounds associated to well-known construction are disappointing, even if known attacks are not remotely feasible, and spending time proportional to the domain size, even in key-setup phase, is not attractive.

With these problematic-size domains in mind, suppose we use swap-or-not to encipher 9-digit social security numbers ($N \approx 2^{30}$). Employing Theorem 4, if we use 340 rounds we are guaranteed a maximal CCA advantage of less than 10^{-10} even if the adversary can ask $q = 10^8$ queries. Similarly, suppose we use swap-or-not to encipher 16-digit credit cards ($N \approx 2^{53}$). If we use 500 rounds we are guaranteed a maximal CCA advantage of less than 10^{-10} even if the adversary can ask $q = 10^{15}$ queries. (Of course these numbers assume random round functions; if one bases the construction on AES, say, one will have to add in a term for its insecurity.) The round counts are obviously high, yet the rounds are fast and the guarantees are strong. (We note too that, at least for the binary-string setting and AES as a starting point, there are tricks to reduce the number of blockcipher calls by a factor of five, as shown in prior work [17]. But this is probably not helpful in the presence of good AES support, as with recent Intel processors.)

A very different approach to small-domain FPE is taken by Granboulan and Pornin [9], who show how to realize a particular shuffle on N cards in $O(\lg^3 N)$ encryption time and $O(\lg N)$ space. But the method seems to be impractical, requiring extended-precision arithmetic to sample from a hypergeometric distribution. Stefanov and Shi go on to show how to exploit preprocessing to realize a different N-card shuffle [24]. Their method is applicable when the key-setup cost of $\tilde{\Theta}(N)$ is feasible, as is key storage and per-message encryption cost of $\tilde{\Theta}(N^{1/2})$. Near or beyond $N \approx 2^{30}$, these assumptions seem unlikely to hold in most settings. That said, the approach allows an adversary to query all N points, whereas the shuffle of this paper has only been proven to withstand $(1 - \epsilon)N$ queries. (We conjecture that swap-or-not works well for N queries and reasonable r—that its mixing time is fast—but no such result is proven here.)

6 Confusion/Diffusion Ciphers

Swap-or-not can also be construed as an approach for making a confusion/diffusion blockcipher. In doing this one would instantiate round functions $F_i : \{0,1\}^n \to \{0,1\}$ by a fast, concrete construction. Perhaps the simplest plausible instantiation is have F_i be specified by an n-bit string L_i, letting $F_i(\hat{X}) = L_i \odot \hat{X} = L_i[1]\hat{X}[1] \oplus \cdots \oplus L_i[n]\hat{X}[n]$ be the inner-product of L_i and $\hat{X}$. This concrete realization of swap-or-not is shown in Fig. 5. (We comment that for

```
proc $E_{KL}(X)$     //**inner-product realization**
for $i \leftarrow 1$ to $r$ do
    $X' \leftarrow K_i \oplus X$
    $\hat{X} \leftarrow \max(X, X')$
    if $L_i \odot \hat{X} = 1$ then $X \leftarrow X'$
return $X$
```

Fig. 5. Cipher $E = \mathrm{SN}[r, n, \odot]$ encrypts a string $X \in \{0,1\}^n$ using a key KL that specifies subkeys $K_1, \ldots, K_r, L_1, \ldots, L_r \in \{0,1\}^n$.

this instantiation it is necessary to use “max” instead of “min” in selecting a canonical one of $\{X, X'\}$; otherwise, we’d have $X = 0^n$ always encrypting to 0^n.)

We do not know how many rounds to suggest such that the construction of Fig. 5 should be a good blockcipher. It is incorrect to think that the theoretical analysis suggests a value like $r = 6n$; for one thing, there is an enormous gap between computing a random round function $F_i(\hat{X})$ and an inner product $L_i \odot \hat{X}$. We leave it as a problem for cryptanalysts to investigate how large r needs to be, to ascertain if inner product with L_i is actually a good choice for F_i, and to understand what other choices might work well.

Acknowledgments

The authors gratefully acknowledge comments from Mihir Bellare and Terence Spies. This work was supported under NSF grants DMS-1007739 and CNS-0904380.

References

1. M. Bellare, T. Ristenpart, P. Rogaway, and T. Stegers. Format-preserving encryption. Selected Areas in Cryptography (SAC) 2009. Springer, pp. 295–312, 2009.
2. M. Bellare, P. Rogaway, and T. Spies. The FFX mode of operation for format-preserving encryption. February 2010. Submission to NIST, available from their website.
3. J. Black and P. Rogaway. Ciphers with arbitrary finite domains. *Topics in Cryptology – CT-RSA 2002*, LNCS vol. 2271, Springer, pp. 114–130, 2002.
4. E. Brier, T. Peyrin, and J. Stern. BPS: a format-preserving encryption proposal. Submission to NIST, available from their website.
5. M. Brightwell and H. Smith. Using datatype-preserving encryption to enhance data warehouse security. 20th National Information Systems Security Conference Proceedings (NISSC), pp. 141–149, 1997.
6. J. Coron, J. Patarin, and Y. Seurin: The random oracle model and the ideal cipher model are equivalent. *CRYPTO 2008*, LNCS 5157, Springer, pp. 1–20, 2008.
7. P. Diaconis and J. Fill. Strong stationary times via a new form of duality. *Annals of Probability*, 18(4), pp. 1483–1522, 1990.
8. FIPS 74. U.S. National Bureau of Standards (U.S). Guidelines for implementing and using the NBS Data Encryption Standard. U.S. Dept. of Commerce, 1981.
9. L. Granboulan and T. Pornin. Perfect block ciphers with small blocks. *Fast Software Encryption* (FSE 2007), LNCS vol. 4593, Springer, pp. 452–465, 2007.
10. S. Halevi. EME*: Extending EME to handle arbitrary-length messages with associated data. *INDOCRYPT 2004*, Springer, pp. 315–327, 2004.
11. S. Halevi and P. Rogaway. A tweakable enciphering mode. *CRYPTO 2003*, Springer, pp. 482-499, 2003.
12. T. Holenstein, R. Künzler, and S. Tessaro. The equivalence of the random oracle model and the ideal cipher model, revisited. *STOC 2011*, pp. 89–98, 2011. Full version at arXiv:1011.1264
13. V. Hoang and P. Rogaway. On generalized Feistel networks. *CRYPTO 2010*, pp. 613–630, 2010.

14. M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. on Computing*, 17(2), pp. 373–386, 1988.
15. U. Maurer, K. Pietrzak, and R. Renner. Indistinguishability amplification. *CRYPTO 2007*, LNCS vol. 4622, Springer, pp. 130–149, 2007.
16. U. Maurer, R. Renner, and C. Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. *TCC 2004*, LNCS 2951, Springer, pp. 21–39, 2004.
17. B. Morris, P. Rogaway, and T. Stegers. How to encipher messages on a small domain: deterministic encryption and the Thorp shuffle. CRYPTO 2009, pp. 286–302, 2009.
18. M. Naor and O. Reingold. On the construction of pseudo-random permutations: Luby-Rackoff revisited. *J. of Cryptology*, 12(1), pp. 29-66, 1999.
19. M. Naor and O. Reingold. A pseudo-random encryption mode. Manuscript, 1997.
20. J. Patarin. Pseudorandom permutations based on the DES scheme. *EUROCODE'90*, LNCS 514, Springer, pp. 193–204, 1990.
21. J. Patarin. Luby-Rackoff: 7 rounds are enough for $2^{n(1-\varepsilon)}$ security. *CRYPTO 2003*, LNCS vol. 2729, Springer, pp. 513–529, 2003.
22. J. Patarin. Security of balanced and unbalanced Feistel schemes with linear non equalities. Cryptology ePrint report 2010/293.
23. S. Rudich. Limits on the provable consequences of one-way functions. Ph.D. Thesis, UC Berkeley, 1989.
24. E. Stefanov and E. Shi. FastPRP: Fast pseudo-random permutations for small domains. Cryptology ePrint Report 2012/254.
25. T. Stütz and A. Uhl. Efficient format-compliant encryption of regular languages: block-based cycle-walking. *Communications and Multimedia Security*, Springer, pp. 82–92, 2010.
26. E. Thorp. Nonrandom shuffling with applications to the game of Faro. *Journal of the American Statistical Association*, 68, pp. 842–847, 1973.
27. J. Wen, M. Severa, W. Zeng, M. Luttrell, and W. Jin. Circuits and systems for video technology. *IEEE Transactions on Circuits & Systems for Video Technology*, 12(6), pp. 545–557, Jun 2002.