Appears in *Advances in Cryptology – CRYPTO 2009*

How to Encipher Messages on a Small Domain
Deterministic Encryption and the Thorp Shuffle

Ben Morris[1], Phillip Rogaway[2], and Till Stegers[2]

[1] Dept. of Mathematics, University of California, Davis, California 95616, USA
[2] Dept. of Computer Science, University of California, Davis, California 95616, USA

Abstract. We analyze the security of the Thorp shuffle, or, equivalently, a maximally unbalanced Feistel network. Roughly said, the Thorp shuffle on N cards mixes any $N^{1-1/r}$ of them in $O(r \lg N)$ steps. Correspondingly, making $O(r)$ passes of maximally unbalanced Feistel over an n-bit string ensures CCA-security to $2^{n(1-1/r)}$ queries. Our results, which employ Markov-chain techniques, enable the construction of a practical and provably-secure blockcipher-based scheme for deterministically enciphering credit card numbers and the like using a conventional blockcipher.

Key words: card shuffling, coupling, modes of operation, symmetric encryption, Thorp shuffle, unbalanced Feistel network.

1 Introduction

Small-space encryption. Suppose you want to encrypt a 9-decimal-digit plaintext, say a U.S. social-security number, into a ciphertext that is again a 9-decimal-digit number. A shared key K is used to control the encryption. Syntactically, you seek a cipher $E\colon \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ where $\mathcal{M} = \{0, 1, \ldots, N-1\}$, $N = 10^9$, and $E_K = E(K, \cdot)$ is a permutation for each key $K \in \mathcal{K}$. You aim to construct your scheme from a well-known primitive, say AES, and to prove your scheme is as secure as the primitive from which you start.

The problem is harder than it sounds. You can't just encode each plaintext $M \in \mathcal{M}$ as a 128-bit string and then apply AES, say, as that will return a 128-bit string and projecting back onto $\mathcal{M}$ will destroy permutivity. Standard blockcipher modes of operation are of no use, and constructions like balanced Feistel [17, 31] or Benes [1, 30] have security that falls off by, at best, the square root of the size of the domain, N. Here N is so small that such a result may provide no practically-useful guarantee.

The above *small-space encryption* problem was first investigated by Black and Rogaway [6], but those authors could find no practical and provably-secure solution for N-values where $q > \sqrt{N}$ queries are feasible but having an encryption take N computational steps is not—values like $2^{20} \leq N \leq 2^{50}$. This paper provides a solution for these troublesome domains.

Thorp shuffle. Our approach is based on the *Thorp shuffle* [41], which works like this. Suppose you want to shuffle N cards, where N is even. Cut the deck

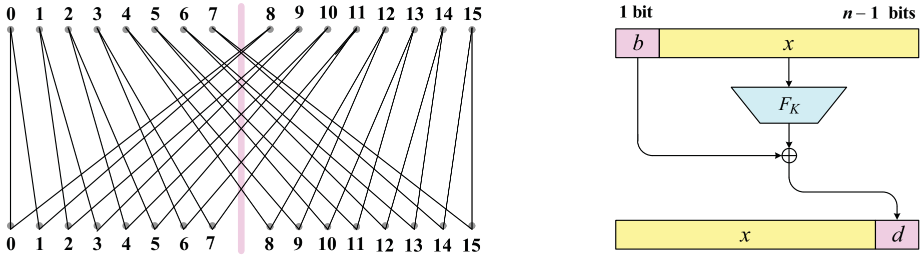

Fig. 1. Two views of the Thorp shuffle (one round). *Left*: each card is paired with the one $N/2$ positions away. For cards at positions x and $x + N/2$, a random bit c (not shown) determines if the cards get mapped to $2x$ and $2x+1$ or to $2x+1$ and $2x$. *Right*: each card is regarded as an n-bit string X (assume $N = 2^n$). Now card $b \parallel x$ gets sent to $x \parallel b \oplus F_K(x)$ for a uniform (and round-dependent) random function F_K.

into two equal piles. Drop the bottom card from either the left or right pile according to the outcome of a fair coin flip, and then drop the card from the bottom of the other pile. Continue in this way, flipping $N/2$ independent coins and using each to decide if you drop cards left-then-right or right-then-left. This is one *round* of the shuffle; repeat for as many rounds as you like. Expressed a bit more algebraically, for each round $r = 1, 2, \ldots, R$ the cards at positions x and $x + N/2$, where $x \in \{0, \ldots, N/2 - 1\}$, are moved either to positions $2x$ and $2x+1$ or else to positions $2x+1$ and $2x$, which ones being determined by a uniform coin flip $c \in \{0, 1\}$. See the left-hand side of Fig. 1. Let $\mathrm{Th}[N, R]$ denote the Thorp shuffle with message space $\mathcal{M} = \{0, \ldots, N-1\}$ and R rounds.

The potential utility of the Thorp shuffle to cryptography and complexity theory was first noticed by Naor some 20 years ago [27, p. 62], [34, p. 17]. He observed that the Thorp shuffle is *oblivious* in the following sense: one can trace the route of any given card in the deck without attending to the remaining cards in the deck. If the Thorp shuffle mixes cards quickly enough, this property would make it suitable for small-space encryption. Namely, the random bit c used for cards x and $x + N/2$ at round r could be determined by applying a pseudorandom function F, keyed by some underlying key K, to x and r. Conceptually, the string K compactly names all of the $(N/2) \cdot R$ random bits that would be needed to shuffle the entire deck. But because the Thorp shuffle is oblivious, only R of these bits, so that many PRF calls, would be needed to encipher a message x.

Feistel connection. There are a variety of alternative views of what goes on in the Thorp shuffle. The one most resonant to cryptographers is this. Suppose that $N = 2^n$ is a power of two. In this case the Thorp shuffle coincides with a maximally unbalanced Feistel network. In an unbalanced Feistel network [18, 36], the left and right portions in the n-bit string that is acted on may have different lengths. Throughout this paper, "maximally unbalanced Feistel" means that the round function takes in $n - 1$ bits and outputs a single bit, a "source-heavy" scheme. See the right-hand side of Fig. 1. A moment's reflection will make clear that, if the round function F_K provides uniform random bits, independently selected for each round, then unbalanced Feistel *is* the Thorp shuffle.

As it takes n rounds of maximally unbalanced Feistel until each bit gets its turn in being replaced, we term n rounds of maximally unbalanced Feistel (or $\lceil \lg N \rceil$ rounds of Thorp) a *pass*. One might hope that the Thorp shuffle mixes the deck well after a small number of passes.

Our results. Assume $N = 2^n$ is a power of two, $r \geq 1$, and let $E = \mathrm{Th}[N, R]$ be the Thorp shuffle with $R = 2rn$ rounds (that is, $2r$ passes). We will show that an adversary mounting a nonadaptive chosen-plaintext attack and making q queries will have advantage that is at most $(q/(r+1)) \cdot (4nq/N)^r$ at distinguishing E from a random permutation on n bits. We prove this bound by regarding the Thorp shuffle of a designated q out of N cards as a Markov chain and applying a coupling argument. To the best of our knowledge, this is the first time that coupling has been used to prove security for a symmetric cryptographic primitive. Using a result of Maurer, Pietrzak, and Renner [21], we can infer that $4r$ passes are enough so that a q-query adversary making an adaptive chosen-ciphertext attack will have advantage at most $(2q/(r+1)) \cdot (4nq/N)^r$ at distinguishing E from a random permutation and its inverse. Put in asymptotic terms, one can construct an n-bit permutation that is CCA-secure to $2^{n(1-1/r)}$ queries by making $4r$ passes of a maximally unbalanced Feistel (its round function being a uniformly random function from $n-1$ bits to 1 bit). This far exceeds what balanced Feistel can achieve, providing a demonstrable separation between the security of balanced and unbalanced Feistel. Finally, we consider a weaker notion of security than customary—withstanding a (nonadaptive) *designated-point attack*. For achieving this, just two passes of unbalanced Feistel are already enough.

In applying the results above to solve the small-space encryption problem using a blockcipher like AES, the number of rounds R becomes the number of blockcipher calls. We describe a trick to reduce this by a factor of five (for a 128-bit blockcipher). We sketch other such "engineering" improvements, like making the constructed cipher tweakable [16], and we tabulate the number of blockcipher calls needed for various provable-security guarantees.

Further related work. Morris proved that the mixing time for the Thorp shuffle—roughly, the number of steps until all $q = N$ cards are ordered nearly uniformly—is polylogarithmic: it is $O(\lg^{44} N)$ [25]. This was subsequently improved to $O(\lg^{19} N)$ [22] and then to $O(\lg^4 N)$ [23].

Naor and Reingold analyzed unbalanced Feistel constructions, showing, in particular, that one pass over a maximally unbalanced Feistel network that operates on n bits remains secure to nearly $2^{n/2}$ queries.

For *balanced* Feistel, the classical analysis by Luby and Rackoff [17] shows that three rounds provide CPA-security (four rounds for CCA-security) to nearly $2^{n/4}$ queries. This was improved by Maurer and Pietrzak [20], who showed that r rounds of balanced Feistel could withstand about $2^{n/2-1/r}$ queries (in the CCA setting). Patarin [29, 31] went on to show that a constant number of rounds (six for CCA-security) was already enough to withstand about $2^{n/2}$

queries. He also suggested that enough rounds of maximally unbalanced Feistel ought to achieve security for up to $2^{n(1-\varepsilon)}$ queries [29, p. 527], a conjecture that our work now proves.

Granboulan and Pornin [12] describe a method to perfectly realize a random permutation using a clever shuffling procedure due to Czumaj, Kanarek, Kutyłowski, and Loryś [8]. The shuffle requires one to repeatedly sample in a hypergeometric distribution using parameters that are large and vary during the shuffle. In an implementation, Granboulan and Pornin employ an arbitrary-precision floating-point package to help achieve the needed sampling. In the end, about 10^9 machine cycles are used to encipher on a space of $N < 2^{32}$ points. While improvements may come [42], the approach is currently impractical.

Kaplan, Naor, and Reingold describe a method to reduce the number of bits needed to specify a permutation that will appear uniform against some number q of queries [14]. They do this by derandomizing a construction such as the Thorp shuffle. They discuss this case, invoking the result of Morris [25].

Håstad analyzes the mixing time of the following *square lattice shuffle*: given an $m \times m$ array, uniformly permute the entries in each row, and then uniformly permute the entries in each column [13]. He shows that a constant number of such passes are enough to mix well. The shuffle is oblivious, and a recursive realization of it would give rise to another solution to the small-space encryption problem.

The problem of enciphering on a small or unusual-size domain is a special case of *format-preserving encryption*, a goal informally described by Brightwell and Smith [7], named by Spies [39], and recently formalized by Bellare and Ristenpart [5] and by Rogaway [33].

In a recent proposal to NIST, Spies [38] describes a blockcipher mode of operation, FFSEM, to encipher on an arbitrary intermediate-size domain $\mathcal{M} = \{0, 1, \ldots, N-1\}$. The mechanism combines the use of a balanced Feistel network and the folklore cycle-walking approach.[3]

Finally, we mention that one could always solve small-domain encryption by *de novo* construction, creating a confusion/diffusion primitive with an unusually rich domain. This is what Schroeppel did with Hasty Pudding [37], anticipating by several years even a formulation of the general problem.

The problem with balanced Feistel. It seems likely that, for any even n, enough rounds of balanced Feistel using a pseudorandom round function yield a computationally-secure small-domain encryption scheme, even up to $q = 2^n - 2$ queries (recall that a Feistel-determined permutation is always even [28, Th 6.1]). No remotely practical attack is known [28], and the construction is of course quite old. But proofs of security for ciphers made from pseudorandom functions invariably work by proving information-theoretic security and then passing to the complexity-theoretic setting. Since balanced Feistel is information-theoretically *insecure* beyond $2^{n/2}$ queries, such an approach is inherently doomed. More

[3] Cycle-walking works like this. To construct a cipher E_K that enciphers on $\mathcal{M} = \{0, 1, \ldots, N-1\}$ using a cipher E'_K that works on $\mathcal{M}' = \{0, 1, \ldots, N'-1\}$, where $N' \geq N$, iterate $E'_K(X)$ until the first point is found that lies in $\mathcal{M}$. Return this. The method is efficient if E' is and N' is not too much larger than N.

precisely, if the adversary may ask $q = 2^{\theta+n/2}$ queries for some $\theta \geq 0$, then, to have any chance of information-theoretic security, one will need a number of rounds that is at least $r = 2^{\theta+1}$. See Appendix B for a simple analysis giving this bound.

2 Preliminaries

Ciphers. By a *cipher* we mean a map E: $\mathcal{K} \times \mathcal{M} \to \mathcal{M}$ where $\mathcal{K}$ and $\mathcal{M}$ are finite nonempty sets (the *key space* and the *domain*) and $E_K(\cdot) = E(K, \cdot)$ is a permutation on $\mathcal{M}$ for every $K \in \mathcal{K}$. Let $\mathcal{A}$ be an adversary, meaning an algorithm with access to an oracle. For the game used to define E's indistinguishability from a random permutation, the oracle will depend on a permutation f: $\mathcal{M} \to \mathcal{M}$. It will respond to a query (enc, x) with $f(x)$ and it will respond to a query (dec, y) with $f^{-1}(y)$. Queries outside of $\{\mathsf{enc}, \mathsf{dec}\} \times \mathcal{M}$ are ignored. Define $\mathbf{Adv}_E^{\mathrm{cca}}(\mathcal{A}) = \mathbf{P}\,(K \overset{\$}{\leftarrow} \mathcal{K}:\ \mathcal{A}^{\pm E_K} \Rightarrow 1) - \mathbf{P}\,(\pi \overset{\$}{\leftarrow} \mathrm{Perm}(\mathcal{M}):\ \mathcal{A}^{\pm\pi} \Rightarrow 1)$ where $\mathcal{A}^{\pm f}$ denotes $\mathcal{A}$ interacting with the f-dependent oracle just described and $\mathcal{A}^f \Rightarrow 1$ is the event that it outputs a 1.

We say that adversary $\mathcal{A}$ is *nonadaptive* if its queries are the same on each and every run. It carries out a *chosen-plaintext* attack if each query is an encryption query, and a *chosen-ciphertext* attack if queries may be either encryption or decryption queries. Let $\mathbf{Adv}_E^{\mathrm{ncpa}}(q) = \max_{\mathcal{A}} \mathbf{Adv}_E^{\mathrm{cca}}(\mathcal{A})$ where the maximum is taken over all nonadaptive adversaries that ask at most q encryption queries and no decryption queries. By the standard averaging argument, the notion is unchanged if nonadaptive adversaries are assumed to be *deterministic*: they statically choose their queries $x_1, \ldots, x_q$. Let $\mathbf{Adv}_E^{\mathrm{cca}}(q) = \max_{\mathcal{A}} \mathbf{Adv}_E^{\mathrm{cca}}(\mathcal{A})$ where the maximum is taken over all adversaries that ask at most q queries.

Markov chains. The next section assumes some familiarity with Markov chains and how to show rapid mixing for them using coupling arguments. See any text on the subject, such as Levin, Peres, and Wilmer [15], for some background on this topic.

Let Ω be a finite nonempty set and let μ, ν be probability distributions on Ω. A *coupling* of μ and ν is a pair of random variables (X, Y), defined on the same probability space, such that the marginal distributions of X and Y are μ and ν, respectively. Let

$$\|\mu - \nu\| = \max_{S \subset \Omega} \mu(S) - \nu(S) = \min_{X \sim \mu,\, Y \sim \nu} \mathbf{P}(X \neq Y) \tag{1}$$

be the total variation distance between μ and ν, where $Z \sim \tau$ means that Z has distribution τ. The minimum in the right-hand side of (1) is taken over all couplings (X, Y) of μ and ν. We shall call a coupling that achieves the minimum an *optimal coupling* of μ and ν.

3 Variational Distance of the Projected Thorp Shuffle

Fix $N = 2^n$. Let $\{\mathrm{Th}_t : t \geq 0\}$ be the Markov chain representing the Thorp shuffle with N cards. More formally, let $\mathcal{C}$ be a set of cardinality N, whose elements we call *cards*. For concreteness, $\mathcal{C} = \{0,1\}^n$. The state space of $\{\mathrm{Th}_t\}$ is the set of bijections from $\mathcal{C}$ to $\{0,1\}^n$. For a card $z \in \mathcal{C}$, we interpret $\mathrm{Th}_t(z)$ as the position of card z at time t.

Let $\mathcal{A}$ be a deterministic adversary that makes exactly q queries. Our proof is based on an analysis of the mixing rate of the Thorp shuffle. However, since $\mathcal{A}$ 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 Thorp shuffle*. Note that since the Thorp shuffle is a random walk on a group (see, e.g., [35]), it has uniform stationary distribution. Hence the stationary distribution of X_t, which we denote by π, is uniform over the set of distinct q-tuples of elements from $\{0,1\}^n$. Equivalently, π is the distribution of q samples without replacement from $\{0,1\}^n$. Let τ_t denote the distribution of X_t.

Theorem 1 (Rapid mixing). *Let $N = 2^n$ and $q \in \{1, \ldots, N\}$, $\{X_t \colon t \ge 0\}$ the corresponding projected Thorp shuffle, π its stationary distribution, and τ_t the distribution of X_t. Then, for any $r \ge 1$,*

$$\|\tau_{r(2n-1)} - \pi\| \le \frac{q}{r+1}\left(\frac{4nq}{N}\right)^r .$$

Proof. For a distribution ν on distinct q-tuples of Ω, define

$$\begin{aligned} \nu(u_1, \ldots, u_j) &= \mathbf{P}\left(Z_1 = u_1, \ldots, Z_j = u_j\right) \\ \nu(u_j \mid u_1, \ldots, u_{j-1}) &= \mathbf{P}\left(\, Z_j = u_j \mid Z_1 = u_1, \ldots, Z_{j-1} = u_{j-1} \,\right) \end{aligned}$$

where $(Z_1, \ldots, Z_q) \sim \nu$. For example, $\tau_t(u_1, \ldots, u_j)$ is the probability that, in the Thorp 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 $\tau_t(\,\cdot \mid u_1, \ldots, u_{j-1})$ converges to uniform as $t \to \infty$. When all of these distributions are "close" to uniform, then τ_t 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 proved in Appendix A.

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*

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

Note that in the above lemma, since $Z_1, \ldots, Z_q$ are random variables (whose joint distribution is governed by μ), so is $\|\mu(\,\cdot \mid Z_1, \ldots, Z_l) - \nu(\,\cdot \mid Z_1, \ldots, Z_l)\|$

for every $l \leq q$; each summand in the right-hand side of (2) is the expectation of one of these random variables.

Coupling arguments. Later in the proof, we will be using a coupling argument to bound $\mathbf{E}\big(\|\mu(\,\cdot \mid Y_1, \ldots, Y_l) - \nu(\,\cdot \mid Y_1, \ldots, Y_l)\|\big)$. Typically, such arguments are used in the following way. There is a Markov chain with transition matrix P and stationary distribution π, started from state x. One wants to estimate the total variation distance $\|P^t(x, \cdot) - \pi\|$ between the distribution of the chain at time t and the stationary distribution. To do so, one constructs a pair process $\{(X_t, Y_t) : t \geq 0\}$, the *coupling*, that satisfies the following conditions:

1. Individually, $\{X_t\}$ and $\{Y_t\}$ are Markov chains with transition matrix P.
2. For every $t \geq 0$, if $X_t = Y_t$ then $X_{t+1} = Y_{t+1}$.
3. We have $X_0 = x$ and $Y_0 \sim \pi$.

The random variable $T = \min\{t : X_t = Y_t\}$ is called the *coupling time.* Note that condition (3) implies that $Y_t \sim \pi$ for all $t \geq 0$. Hence equation (1) implies

$$\begin{aligned} \|P^t(x,\,\cdot\,) - \pi\| &\leq \mathbf{P}\left(X_t \neq Y_t\right) \\ &= \mathbf{P}\left(T > t\right). \end{aligned}$$

The idea is to define the coupling in such a way that T is unlikely to be large.

Defining the coupling. Let τ_t be the distribution of X_t. We wish to use coupling to bound the expected distance between $\tau_t(\,\cdot \mid X_t(1), \ldots, X_t(l))$ and the uniform distribution on $\{0,1\}^n \setminus \{X_t(1), \ldots, X_t(l)\}$, for each $l \in \{1, \ldots, q-1\}$.

Our approach will be as follows. For each value of l we will construct a process $\{U_t\}$ on the same probability space as $\{X_t\}$, to get a coupling $\{(X_t, U_t) \colon t \geq 0\}$. The process $\{U_t\}$ will satisfy the following conditions.

- The positions of the first l cards in U_t always agree with X_t.
- For every t, the distribution of the position of card z_{l+1} at time t, given the positions of cards $z_1, \ldots, z_l$, is uniform.

We begin with a key definition. Say that two cards are *adjacent* at time t if their positions (viewed as elements of $\{0,1\}^n$) are the same except for the first bit (or, viewed as elements of $\{0, \ldots, N-1\}$, they differ by $N/2$).

Let X_t be the projected Thorp shuffle. It will be convenient to use a rule for generating the evolution of X_t that uses q fair coins, $c^1, \ldots, c^q$, each of which is flipped at each step. Formally, each c^j is a sequence $\{c^j_t \colon t \geq 0\}$ of Bernoulli(1/2) random variables, where we interpret c^j_t as the outcome of coin c^j at time t. We assume that all of the c^j_t are independent.

Note that for a given step, it is enough to describe, for each pair z_i, z_j of adjacent cards, $i < j$, how the position of z_i is updated (since this dictates how the position of z_j must be updated). We shall use the following update rule:

> **Update rule** For each pair of cards z_i, z_j with $i < j$ that are adjacent at time t, we determine the position of z_i at time $t+1$ using coin c^i and coin flip c^i_t as follows:

1. the first (leftmost) bit of the position of z_i is set to c_t^i, and then
2. the position of z_i undergoes a cyclic left bit shift.

Thus if c_i is at position x at time t then at time $t+1$ it will be at position $2(x \bmod N/2) + c_t^i$, or, in string-oriented notation, at position $x[2..N] \,\|\, c_t^i$. We claim that if $t \geq n-1$ then for any pair of cards z_i and z_j we have

$$\mathbf{P}\left(z_i \text{ and } z_j \text{ are adjacent at time } t\right) \leq 2^{1-n} \ . \tag{3}$$

To verify this claim, note that (by reordering if necessary) we may assume that $i = 1, j = 2$, and the evolution of X_t is governed by the update rule described above. Let E be the event that z_1 and z_2 are adjacent at time t. In order for E to happen, at each step during times $t-1, \ldots, t-n+1$, when their bits are changed (in step 1 of the update rule), the same change must occur for both z_1 and z_2. Thus $E = A \cap B$, where A is the event that z_1 and z_2 were not adjacent at any of the times $t-1, \ldots, t-n+1$, and B is the event that coins c^i and c^j had the same outcomes at times $t-1, \ldots, t-n+1$. It follows that $\mathbf{P}(E) \leq \mathbf{P}(B) = 2^{1-n}$, and the claim is verified.

We are now ready to describe $\{U_t \colon t \geq 0\}$. The starting state U_0 is constructed as follows.

1. We set $U_0(1, \ldots, l) = X_0(1, \ldots, l)$. That is, cards $z_1, \ldots, z_l$ have the same initial positions in U_0 as X_0.
2. The distribution of $U_0(l+1)$ is uniform over $\{0,1\}^n \setminus \{U_0(1), \ldots, U_0(l)\}$.

(We may assume without loss of generality that the probability space on which $\{X_t \colon t \geq 0\}$ is defined is rich enough to allow us to construct such a U_0.) Note that the final condition implies that for every time t the conditional distribution of $U_t(l)$ given $U_t(1, \ldots, l)$ is uniform over $\{0,1\}^n \setminus \{U_t(1), \ldots, U_t(l)\}$.

We now describe the rule for generating (X_{t+1}, U_{t+1}) from (X_t, U_t). Note that the rule for generating $\{X_t \colon t \geq 0\}$ using coins $c^1, \ldots, c^q$ leads to a natural way to generate the evolution of $\{(X_t, U_t) : t \geq 0\}$. Namely, we use the same coins $c^1, \ldots, c^q$ to update both X_t and U_t in each step. Since the positions of cards $z_1, \ldots, z_l$ initially agree in both X_0 and U_0, and we are using the same coin flips $c_t^1, \ldots, c_t^l$ to update them each step, the positions of these cards remain matched for all times t. Furthermore, note that if at any point the position of card z_{l+1} becomes matched, then it remains matched from then on. Recall that τ_t is the distribution of X_t, and let $Z_1, \ldots, Z_l$ be the positions of cards $z_1, \ldots, z_l$ at time t. (Note that these positions are the same in both X_t and U_t.) By (1), we have

$$\begin{aligned}\|\tau_t(\,\cdot \mid Z_1, \ldots, Z_l) - \pi(\,\cdot \mid Z_1, \ldots, Z_l)\| &\leq \mathbf{P}\left(\, X_t(l+1) \neq U_t(l+1) \mid Z_1, \ldots, Z_l \,\right) \\ &= \mathbf{P}\left(T > t \mid Z_1, \ldots, Z_l\right),\end{aligned}$$

where $T = \min\{t : X_t(l+1) = U_t(l+1)\}$ is the coupling time. Taking expectations gives

$$\mathbf{E}\Big(\|\tau_t(\,\cdot \mid Z_1, \ldots, Z_l) - \pi(\,\cdot \mid Z_1, \ldots, Z_l\|\Big) \leq \mathbf{P}\left(T > t\right). \tag{4}$$

We claim that

$$\mathbf{P}\left(T > 2n-1\right) \leq p\,, \tag{5}$$

where $p = nl2^{2-n}$. Let A be the event that at some time in $\{n-1, n, \ldots, 2n-2\}$ card z_{l+1} is adjacent to some card of smaller index in the Y or Z process. Unless A occurs, coupling occurs by time $2n-1$. Summing equation (3) over 2 processes, n timesteps, and l smaller indices verifies the claim by showing that

$$\mathbf{P}\left(A\right) \leq 2nl \cdot 2^{1-n} = p\,. \tag{6}$$

Note that equation (5) holds regardless of the initial state (X_0, U_0), and that the process $\{(X_t, U_t) : t \geq 0\}$ is itself a Markov chain. Now imagine that we have a sequence of trials where in each trial we run the coupling for an additional $2n-1$ steps. The probability that card z_{l+1} remains unmatched after the first trial is at most p. Furthermore, by the memoryless property of Markov chains, given that card z_{l+1} remained unmatched after the first $r-1$ trials, the conditional probability that it remains unmatched after the r-th trial is again at most p. Hence, by induction, $\mathbf{P}\left(\text{card } z_{l+1} \text{ remains unmatched after } r \text{ trials}\right) \leq p^r = (nl2^{2-n})^r$, that is,

$$\mathbf{P}\left(T > r(2n-1)\right) \leq \left(nl2^{2-n}\right)^r. \tag{7}$$

Summing over $l \in \{0, \ldots, q-1\}$ and using Lemma 2 gives

$$\begin{aligned} \|\tau_{r(2n-1)} - \pi\| \quad &\leq \quad \sum_{l=0}^{q-1} (nl2^{2-n})^r \leq \int_0^q (n2^{2-n})^r x^r \; dx \\ &\leq \quad \frac{q^{r+1}}{r+1} \cdot n^r 2^{2r-nr} = \frac{q}{r+1} \left(\frac{4nq}{N}\right)^r . \qquad \square \end{aligned}$$

4 Pseudorandomness of the Thorp Shuffle

CPA-SECURITY. The total variation distance is identical to the advantage with respect to a (deterministic) nonadaptive chosen-plaintext attack. So, reformulating Theorem 1 in cryptographic terms, what we have shown is the following.

Theorem 3 (nCPA-security, concrete). *Let $N = 2^n$ and $1 \leq q \leq N$. Then, for any $r \geq 1$,*

$$\mathbf{Adv}^{\text{ncpa}}_{\text{Th}[N, r(2n-1)]}(q) \leq \frac{q}{r+1} \left(\frac{4nq}{N}\right)^r .$$

TIME REVERSAL. Let $\text{Th}^{-1}[N, R] = (\text{Th}[N, R])^{-1}$ denote the time-reverse Thorp shuffle on N cards with R rounds: in round $r \in \{1, \ldots, R\}$ it sends cards $2x$ and $2x+1$, where $0 \leq x < N/2$, either to x and $x + N/2$, or to $x + N/2$ and x, depending on a random bit $c(x, r)$. For N a power of two the forward and reverse Thorp shuffle are "isomorphic" Markov chains in the sense that there is a relabeling of states from $\text{Th}[N, R]$ to $\text{Th}^{-1}[N, R]$ that preserves the transition rule. As a consequence, the bound of Theorem 1 applies to the reverse Thorp shuffle as well, giving us the following.

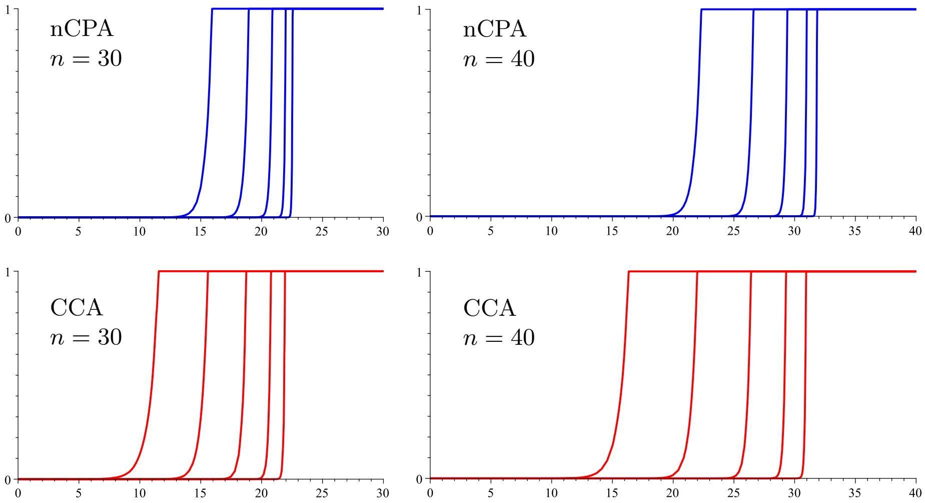

Fig. 2. Proven security of the Thorp shuffle. The x-axis gives the log (base 2) of the number of queries. The y-axis gives an upper bound on an adversary's nCPA advantage by Theorem 3 (top) or its CCA advantage by Theorem 5 (bottom). The curves are for $N = 2^{30}$ points (left) or $N = 2^{40}$ points (right). The curves, from left to right, are for 4, 8, 16, 32, and 64 passes.

Corollary 4 (nCPA-security, reverse Thorp) *Let $N = 2^n$ and $1 \le q \le N$. Then, for any $r \ge 1$, $\mathbf{Adv}^{\mathrm{ncpa}}_{\mathrm{Th}^{-1}[N,r(2n-1)]}(q) \le (q/(r+1)) \cdot (4nq/N)^r$.*

CCA-security. A lovely result of Maurer, Pietrzak, and Renner [21] allows us to easily extend Theorem 3 to a larger class of adversaries, namely, we can trade our nCPA-adversaries for CCA ones. The cost of doing so will be a doubling in the number of rounds, as well as in the advantage bound.

Theorem 5 (CCA-security, concrete). *Let $N = 2^n$ and $1 \le q \le N$. Then, for any $r \ge 1$,*

$$\mathbf{Adv}^{\mathrm{cca}}_{\mathrm{Th}[N,r(4n-2)]}(q) \le \frac{2q}{r+1}\left(\frac{4nq}{N}\right)^r .$$

Proof. We use the second half of Corollary 5 from Maurer et al. [21]. In their notation, we have that $\mathbf{F} = \mathrm{Th}[N, R/2]$, which is stateless, $\mathbf{G} = \mathrm{Th}^{-1}[N, R/2]$, which also stateless, and thus $\varDelta_q(\langle \mathbf{F} \triangleright \mathbf{G}^{-1} \rangle, \langle \mathbf{P} \rangle) = \mathbf{Adv}^{\mathrm{cca}}_{\mathrm{Th}[N,R]}(q)$ is bounded above by $\varDelta_q^{\mathsf{NA}}(\mathbf{F}, \mathbf{P}) + \varDelta_q^{\mathsf{NA}}(\mathbf{G}, \mathbf{P}) = \mathbf{Adv}^{\mathrm{ncpa}}_{\mathrm{Th}[N,R/2]}(q) + \mathbf{Adv}^{\mathrm{ncpa}}_{\mathrm{Th}^{-1}[N,R/2]}(q)$. Note that nonadaptive adversaries in [21] may be probabilistic, but that the best deterministic adversary must do at least as well. Applying Theorem 3 and Corollary 4 to bound the last two summands yields the result. □

Graphical illustration. The bounds of Theorems 3 and 5 are illustrated in Fig. 4. For example, for 16 passes and $N = 2^{40}$ points (third curve on the bottom right), an adversary must ask at least $2^{26.2}$ queries to have CCA advantage 0.5. For comparison, when applied to a maximally unbalanced Feistel network, the earlier analysis of Naor and Reingold [27, Th 6.2] would have topped out—with one pass—at $2^{16.8}$ queries. Had we enciphered strings using a balanced Feistel network instead, then the result of Maurer and Pietrzak [20, Th 1], would give a family of curves (depending, like ours, on how many rounds were performed) that would top out by $2^{18.5}$ queries. Patarin's result for six-round Feistel [31] would apparently be similar, but the concrete security is not explicitly given in that work, and the quantitative bounds are difficult to infer.

Asymptotic interpretation. For an asymptotic interpretation of Theorem 3, fix $r > 0$ and suppose that $q = N^{1-1/r}$ and where, as before, $N = 2^n$. Then

$$\mathbf{Adv}^{\mathrm{ncpa}}_{\mathrm{Th}[N,2rn]}(q) \le \frac{q}{r+1}\left(\frac{4nq}{N}\right)^r = \frac{4^r n^r}{r+1} \cdot \frac{1}{N^{1/r}} \; .$$

In other words, we have upper-bounded the advantage by an expression of the form $(a \log^b N)/N^{1/r}$ for r-dependent constants a and b. Since this goes to 0 as $n \to \infty$, we conclude the following.

Corollary 6 (nCPA-security, asymptotic) *Let $r \ge 1$ be an integer. Then*

$$\lim_{n\to\infty} \mathbf{Adv}^{\mathrm{ncpa}}_{\mathrm{Th}[2^n,2rn]}\left(2^{n(1-1/r)}\right) = 0 \; .$$

In English, a maximally-unbalanced Feistel network on n bits employing $2r$ passes maintains security to nearly 2^n queries: specifically, to $2^{n(1-1/r)}$ queries for large enough n. Said differently, you can achieve security up to $N^{1-\varepsilon}$ nonadaptive queries, for any $\varepsilon > 0$, provided you make at least $2 \cdot \lceil 1/\varepsilon \rceil$ passes. This is far better than what a balanced Feistel network can achieve (see Appendix B). The asymptotic version of Theorem 5 is similar.

Corollary 7 (CCA-security, asymptotic) *Let $r \ge 1$ be an integer. Then*

$$\lim_{n\to\infty} \mathbf{Adv}^{\mathrm{cca}}_{\mathrm{Th}[2^n,4rn]}\left(2^{n(1-1/r)}\right) = 0 \; .$$

Designated-point security. The PRP notion of security formalizes an adversary's inability to detect non-uniform behavior when it sees a *population* of plaintext/ciphertext pairs. Many security notions instead demand that the adversary figure something out about a designated point that it selects: the customary formulations for find-then-guess security, semantic security, unforgeability, and non-malleability are all this way. Weakening the security notion along these lines facilitates a stronger bound for the Thorp shuffle.

Let $E\colon \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ be a cipher and let $\mathcal{A}$ be an adversary. We measure the effectiveness of $\mathcal{A}$ in carrying out a *designated-point attack* on E by $\mathbf{Adv}^{\mathrm{dpa}}_{E}(\mathcal{A}) = \mathbf{P}\left(\mathcal{A}^G \Rightarrow 1\right) - \mathbf{P}\left(\mathcal{A}^H \Rightarrow 1\right)$ where oracles G and H behave like this. Both begin by choosing $K \overset{\$}{\leftarrow} \mathcal{K}$ and then answering queries (enc, x) by $E_K(x)$. Oracle G answers the same way for a query (test, x), but H answers

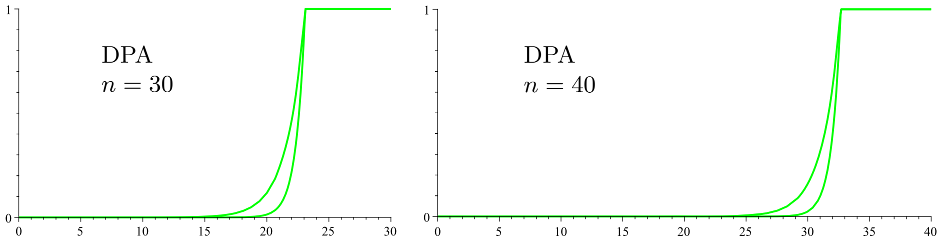

Fig. 3. Proven security of the Thorp shuffle, continued. The x-axis gives the log (base 2) of the number of queries. The y-axis gives an upper bound on an adversary's DPA advantage by Theorem 8, both for $N = 2^{30}$ points (left) and $N = 2^{40}$ points (right). The curves, from left to right, are for two passes and then four.

such a query by a uniformly chosen value that has not yet been returned to $\mathcal{A}$. No other types of queries are allowed. The adversary may ask a single test query, its last: once a test query is asked, any subsequent query returns $\perp$. Let $\mathbf{Adv}_E^{\mathrm{dpa}}(q) = \max_{\mathcal{A}} \mathbf{Adv}_E^{\mathrm{dpa}}(\mathcal{A})$ where the maximum is taken over all deterministic nonadaptive adversaries that ask exactly q enc queries. The DPA notion is similar to, but weaker than, the IUP notion investigated by Desai and Miner [9].

Theorem 8 (Designated-point security). *Let $N = 2^n$ and $1 \leq q \leq N$. Then, for any $r \geq 1$,*

$$\mathbf{Adv}^{\mathrm{dpa}}_{\mathrm{Th}[N,r(2n-1)]}(q) \leq \left(\frac{4nq}{N}\right)^r .$$

The proof follows immediately from equations (4) and (7). The bounds are illustrated in Fig. 3. An asymptotic counterpart for the result is as follows.

Corollary 9 (Designated-point security, asymptotic) *For any $\varepsilon > 0$,*

$$\lim_{n\to\infty} \mathbf{Adv}^{\mathrm{dpa}}_{\mathrm{Th}[2^n,2n-2)]}\left(2^{n(1-\varepsilon)}\right) = 0 .$$

More general message spaces. We emphasize that our results on the Thorp shuffle have assumed that the size of the message is a power of two. By using the cycle-walking construction [6], this suffices to encipher messages on any message space $\{0, \ldots, N-1\}$. But the cost of applying this domain transformation can be nearly as bad as an expected doubling in the encryption and decryption time. It would be more desirable for the results to directly apply to Thorp-enciphering for any even N. We expect the full version of this paper to report on such results.

5 Efficiently Realizing the Thorp Shuffle

Encrypting data in such a way that its format is preserved can lead to a simpler migration path when encryption is added to legacy systems; for example,

$p =$	$n = 20$				$n = 30$				$n = 40$			
#passes	#AES	dpa	ncpa	cca	#AES	dpa	ncpa	cca	#AES	dpa	ncpa	cca
2	8	12.7	6.9	—	12	22.1	11.6	—	16	31.7	16.4	—
4	16	13.2	9.4	6.4	24	22.6	15.7	11.2	32	32.2	22.1	15.9
8	32	13.4	11.3	9.1	48	22.8	18.8	15.3	64	32.4	26.5	21.7
16	64	13.6	12.4	11.1	96	23.0	20.8	18.6	128	32.6	29.3	26.3
32	128	13.6	13.4	13.0	192	23.1	22.5	20.0	256	32.6	31.8	30.9
64	256	13.6	13.4	13.0	384	23.1	22.6	21.9	512	32.6	31.8	30.9

Fig. 4. Security and its cost. The columns indicate the domain size $N = 2^n$, the number of passes p, the number of AES calls per encryption (with 5x-speedup), and values $\lg q$ such that our bound on $\mathbf{Adv}^{\mathrm{xxx}}_{\mathrm{Th}[2^n,np]}(q)$ is about 0.5, for xxx $\in$ {dpa, ncpa, cca}.

a database's schema can remain unchanged when some of its fields get enciphered [7]. Small-space encryption enables a realization of this idea. The problem has already attracted the interest of industry, with companies like Voltage Security already making products in this space.

In this section we sketch a practical realization of Thorp-shuffle encryption. We assume a pseudorandom function $f \colon \mathcal{K} \times \Sigma^* \to \{0,1\}^{128}$. In the analysis, $\rho = f(K, \cdot)$ is regarded as a uniform random function. The translation to the complexity-theoretic setting is standard, the PRF's key K naming a particular ρ. A natural instantiation of f would be the CBC MAC of AES (after a prefix-free encoding of the input [32]). Typically, only one AES call would be needed per PRF invocation.

Update rule. Our realization of $\mathrm{Th}[N, R]$ will effectively use a different update rule than that of Section 3: to each $x, x + N/2 \in \{0, \ldots, N-1\}$ and each round r we associate a coin flip $c(x, r)$ that is used to map the card occupying position x in round r to position $2x + c(x, r)$ if $x < N/2$ and to $2x - N + 1 - c$ if $x \geq N/2$. When N is a power of two and $c(x, r) = \rho(x \bmod N/2, r)$, this corresponds to an unbalanced Feistel network.

Five rounds at once. We now describe a technique that lets one compute several rounds of the Thorp shuffle with a single call to the underlying PRF. With a PRF outputting 128 bits, one call will let us do five rounds of enciphering or deciphering instead of just one. We call this the 5x trick. With it, one needs $\lceil R/5 \rceil$ PRF calls to realize $\mathrm{Th}[N, R]$. See Fig. 4 for a representation of how many AES calls would be needed to make various numbers of passes over domains of various sizes, and our proven security bounds in each case.

To explain the idea behind the 5x trick, assume for now that we are enciphering $N = 2^n$ points for some $n \geq 5$. We will use the following notation. If $X \in \{0,1\}^\ell$ and $i, j \in \{0, \ldots, \ell - 1\}$, we write $X[i]$ for its i-th bit, where the leftmost bit of X is $X[0]$. The substring consisting of the i-th through j-th bit

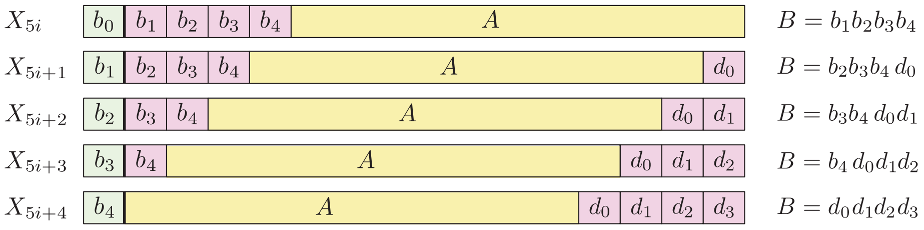
X_{5i}
b_0 b_1 b_2 b_3 b_4 A
$B = b_1b_2b_3b_4$
X_{5i+1}
b_1 b_2 b_3 b_4 A d_0
$B = b_2b_3b_4\,d_0$
X_{5i+2}
b_2 b_3 b_4 A d_0 d_1
$B = b_3b_4\,d_0d_1$
X_{5i+3}
b_3 b_4 A d_0 d_1 d_2
$B = b_4\,d_0d_1d_2$
X_{5i+4}
b_4 A d_0 d_1 d_2 d_3
$B = d_0d_1d_2d_3$

Fig. 5. The 5x trick. The lines show successive n-bit strings X_j as we encipher (going down) or decipher (going up) using $\mathrm{Th}[2^n, R]$. For any $A \in \{0,1\}^{n-5}$ and round j divisible by 5, a single call computes the coins associated to all $(n-1)$-bit strings $\star\star\star\star A$ for round j, $\star\star\star A\star$ for round $j+1$, $\star\star A\star\star$ for round $j+2$, $\star A\star\star\star$ for round $j+3$, and $A\star\star\star\star$ for round $j+4$, where each $\star$ may be 0 or 1.

of X is written $X[i..j]$. It is empty if $i > j$. If $v_1, \ldots, v_k$ are bitstrings or integers, $\langle v_1, \ldots, v_k\rangle$ is the tuple $(v_1, \ldots, v_k)$ encoded as a bitstring in some fixed way.

Denote the ciphertext of $X \in \{0,1\}^n$ after i rounds of Thorp-enciphering by X_i, with $X_0 = X$. Instead of evaluating ρ at $\langle X_i[1\,..\,n-1], i\rangle$ and using only one of the resulting 128 bits as $c(X_i[1\,..\,n-1], i)$, we will instead extract a sufficient number of bits to determine all coin flips $c(U, r)$ that may be needed in the same group of five consecutive rounds. Realizing this idea is slightly tricky because it is essential that each coin $c(U, r)$ taken from ρ's output be well-defined no matter how this value may arise, and, at the same time, that it be independent from $c(V, s)$ unless $(U, r) = (V, s)$.

Our strategy is illustrated by Fig. 5. We group the round into runs of five which we call *phases*, beginning with the first. (The last group of five may be shorter.) We exploit the fact that, for $j \in \{0, 1, 2, 3, 4\}$, the strings X_{5i} and X_{5i+j} have at least an $(n-5)$-bit substring A in common. We evaluate ρ only on $\langle A, i\rangle$, obtaining 128 random bits. The coin flip $c(X_t[1..n-1], t)$ used to encrypt X_t in round $t = 5i + j$ is then picked out of these 128 bits using $\langle B, j\rangle$ where B is the concatenation of bits $1\,..\,(4-j)$ and $(n-j+4)\,..\,(n-1)$ of the string X_{5i+j}. For example, in round 2 we have $A = X_2[3\,..\,n-3]$ and $B = X_2[1..2] \,\|\, X_2[n-2\,..\,n-1]$, whereas in round 14 we have $B = X_{14}[n-4\,..\,n-1]$. The independence requirement is satisfied since the tuple (A, B, i, j) uniquely determines $(X_{5i+j}[1\,..\,n-1], 5i+j)$.

The real complexity comes when N is *not* a power of 2. Carefully generalizing the idea just sketched by replacing string operations with modulo arithmetic gives rise to the same 5x speedup for Thorp-enciphering any number of points N provided that N is a multiple of 32. We specify this case in Fig. 6. The cycle-walking trick (see footnote 3 on p. 4) can then be used to extend the domain to arbitrary N while never enciphering on a domain that exceeds the size of the original one by more than 31 points. The 5x trick uses $5 \cdot 2^4 = 80$ of the 128 bits output by the PRF. It generalizes to yield a k-fold speedup if the PRF outputs at least $k \cdot 2^{k-1}$ bits, though this necessitates rounding N to the next multiple of 2^k.

Theorem 10. *Suppose* $32 \mid N$ *and* $R \geq 1$. *If* $\rho\colon \Sigma^* \to \{0,1\}^{128}$ *is a uniform random function then the permutation* Enc_ρ *defined in Fig. 6 realizes* $\mathrm{Th}[N, R]$.

Also, Dec_ρ *is its inverse. Furthermore, computing* $\mathrm{Enc}_\rho(x)$ *or* $\mathrm{Dec}_\rho(x)$ *requires* ρ *to be evaluated on at most* $\lceil R/5 \rceil$ *points.*

Proof. For the first claim, it suffices to show that if N, R are positive integers and $32 \,|\, N$, then F_ρ is a random function on pairs (r, x) where $r \in \{0, \ldots, R-1\}$, and $0 \le x < N/2$. Let $(r', x') \ne (r'', x'')$ be tuples of this form. If v is a variable in the code of F_ρ in Fig. 6, we denote its value at the end of executing $F_\rho^{r'}(x')$ and $F_\rho^{r''}(x'')$ by v' and v'', respectively. If $Y' \ne Y''$ or $16j' + b' \ne 16j'' + b''$ then the bits c', c'' output at line 41 are independent as random variables over the choice of ρ and we are done.

So suppose $Y' = Y''$ and $16j' + b' = 16j'' + b''$. Since we have $x < N/2$ in any execution of F_ρ we get $\mathit{hi} = (x \operatorname{div} 2^j) \operatorname{div} N/2^5 \le (N/2^{j+1})/(N/2^5) = 2^{4-j}$; clearly $\mathit{lo} < 2^j$. It follows that $b < 16$, in particular $b', b'' < 16$. Hence the index $16j' + b' = 16j'' + b''$ uniquely determines $(j', b') = (j'', b'')$ and thus $(\mathit{lo}', \mathit{hi}') = (\mathit{lo}'', \mathit{lo}'')$. From $Y' = Y''$ we get $N' = N''$, $i' = i''$, and $a' = a''$. Since the values of a, hi, lo uniquely determine the value of x and i, j determine r, we conclude $(r', x') = (r'', x'')$.

Let $x \in \{0, \ldots, N-1\}$. Denote by Y_i and a_i the value of the variables Y and a, respectively, when executing F_ρ in the i-th round of $\mathrm{Enc}_\rho(x)$. We prove the second claim of the theorem by showing that $a_{5i+j} = a_{5i}$ whenever $j \in \{0, \ldots, 4\}$ and $0 \le 5i+j < R$, since this and the definition of i in line 31 imply $Y_{5i+j} = Y_{5i}$. Note that if x_k is the encryption of x after k rounds then by lines 13–14

$$x_{k+1} \operatorname{div} 2 \equiv x_k \pmod{N/2}\,. \tag{8}$$

Since $32 \mid N$ we have $2^j \mid N$ for $j \in \{0, \ldots, 4\}$. Thus $y \equiv y' \pmod{N/2^j}$ implies $y \operatorname{div} 2 \equiv y' \operatorname{div} 2 \pmod{N/2^{j+1}}$ for any integers y, y'. Using (8) and this fact, we obtain $x_{5i+1} \operatorname{div} 2 \equiv x_{5i+2} \operatorname{div} 2^2 \pmod{N/2^2}$, and since (8) is also valid mod $N/2^5$, we get $x_{5i} \equiv x_{5i+2} \operatorname{div} 2^2 \pmod{N/2^5}$. Continuing this way, we get $x_{5i} \equiv x_{5i+j} \operatorname{div} 2^j \pmod{N/2^5}$, which implies $a_{5i} = a_{5i+j}$. □

TWEAKING. A practical realization for small-space encryption should be *tweakable*, a notion formalized by Liskov, Rivest, and Wagner [16]. The syntax of the cipher is extended to take an additional argument, the *tweak*, and each tweak effectively names a random independent cipher. The algorithm of Fig. 6 is easily modified to accommodate a tweak by adding it to the tuple Y in line 37. As a simple example, an application might need to encipher the upper five digits of a U.S. social security number using a tweak that is the lower four digits.

VARIABLE INPUT LENGTH. In Fig. 6, we included the domain size N within the scope of ρ's input (lines 37–38). This makes the scheme secure in the sense of a variable-input-length (VIL) cipher. The property allows to, for example, encipher under a common key database fields that have different domain sizes.

Acknowledgments

Many thanks to Terence Spies, and Voltage Security, for renewing the second author's interest in the small-space encryption problem; without their interest, this work would not exist.

```
10 algorithm $\mathrm{Enc}_\rho(x)$
11 for $r \leftarrow 0$ to $R-1$ do
12     $c \leftarrow F^r_\rho(x \bmod N/2)$
13     if $x < N/2$ then $x \leftarrow 2x + c$
14     else $x \leftarrow 2(x \bmod N/2) + 1 - c$
15 return $x$
```

```
20 algorithm $\mathrm{Dec}_\rho(y)$
21 for $r \leftarrow R-1$ downto $0$ do
22     $c \leftarrow F^r_\rho(y \operatorname{div} 2)$
23     if $c = y \bmod 2$ then $y \leftarrow y \operatorname{div} 2$
24     else $y \leftarrow y \operatorname{div} 2 + N/2$
25 return $x$
```

```
30 algorithm $F^r_\rho(x)$
31 $i \leftarrow r \operatorname{div} 5$
32 $j \leftarrow r \bmod 5$
33 $a \leftarrow (x \operatorname{div} 2^j) \bmod N/32$
34 $hi \leftarrow (x \operatorname{div} 2^j) \operatorname{div} N/32$
35 $lo \leftarrow x \bmod 2^j$
36 $b \leftarrow hi \cdot 2^j + lo$
37 $Y \leftarrow \langle N, i, a \rangle$
38 $table \leftarrow \rho(Y)$
39 $k \leftarrow 16j + b$
40 $c \leftarrow table[k]$
41 return $c$
```

Fig. 6. Realization of $\mathrm{Th}[N, R]$ that incorporates the 5x trick. We assume that $32 \mid N$ and $\rho\colon \{0,1\}^* \to \{0,1\}^{128}$. Line 38 need only be evaluated once every five rounds.

References

1. W. Aiello and R. Venkatesan. Foiling birthday attacks in length-doubling transformations: Benes: a non-reversible alternative to Feistel. *EUROCRYPT 1996*, LNCS vol. 1070, Springer, pp. 307–320, 1996.
2. D. Aldous and P. Diaconis. Shuffling cards and stopping times. *American Mathematical Monthly*, 93, pp. 333–348, 1986.
3. D. Aldous and P. Diaconis. Strong uniform times and finite random walks. *Advances in Applied Mathematics*, 8(1), pp. 69–97, 1987.
4. D. Bayer and P. Diaconis. Tracing the dovetail shuffle to its lair. *Annals of Applied Probability*, 2(2), pp. 294–313, 1992.
5. M. Bellare and T. Ristenpart. Format-preserving encryption. Cryptology ePrint report 2009/251.
6. J. Black and P. Rogaway. Ciphers with arbitrary finite domains. *Topics in Cryptology – CT-RSA 2002*, LNCS vol. 2271, Springer, pp. 114–130, 2002.
7. 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.
8. A. Czumaj, P. Kanarek, M. Kutyłowski, and K. Loryś. Fast generation of random permutations via networks simulation. *Algorithmica*, 21(1), Springer, May 1998.
9. A. Desai and S. Miner. Concrete security characterizations of PRFs and PRPs: reductions and applications. *ASIACRYPT 2000*, LNCS vol. 1976, Springer, pp. 503–516, 2000.
10. P. Diaconis. *Group representations in Probability and Statistics*, vol. 11 of Lecture Notes—Monograph series. Institute of Mathematical Statistics, Hayward, California, 1988.
11. P. Diaconis and J. Fill. Strong stationary times via a new form of duality. *Annals of Probability*, 18(4), pp. 1483–1522, 1990.
12. L. Granboulan and T. Pornin. Perfect block ciphers with small blocks. *Fast Software Encryption* (FSE 2007), LNCS vol. 4593, Springer, pp. 452-465, 2007.
13. J. Håstad. The square lattice shuffle. *Random Structures and Algorithms*, 29(4), pp. 466–474, 2006.

14. E. Kaplan, M. Naor, and Omer Reingold. Derandomized constructions of k-wise (almost) independent permutations. *Randomization and Computation* (RANDOM 2005), LNCS vol. 3624, Springer, pp. 354–365, 2005.
15. D. Levin, Y. Peres, and E. Wilmer. *Markov chains and mixing times.* American Mathematical Society, 2008.
16. M. Liskov, R. Rivest, and D. Wagner. Tweakable block ciphers. *CRYPTO 2002*, LNCS vol. 2442, Springer, pp. 31–46, 2002.
17. M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. on Computing*, 17(2), pp. 373–386, 1988.
18. S. Lucks. Faster Luby-Rackoff ciphers. *Fast Software Encryption* (FSE 1996), LNCS vol. 1039, Springer, pp. 180–203, 1996.
19. U. Maurer. A simplified and generalized treatment of Luby-Rackoff pseudorandom permutation generators. *EUROCRYPT 1992*, LNCS vol. 658, pp. 239–255, 1992.
20. U. Maurer and K. Pietrzak. The security of many-round Luby-Rackoff pseudorandom permutations. *EUROCRYPT 2003*, LNCS vol. 2656, Springer, pp. 544–561, 2003.
21. U. Maurer, K. Pietrzak, and R. Renner. Indistinguishability amplification. *CRYPTO 2007*, LNCS vol. 4622, Springer, pp. 130–149, 2007.
22. R. Montenegro and P. Tetali. Mathematical aspects of mixing times in Markov chains. *Foundations and Trends in Theoretical Computer Science*, 1(3), Now Publishers, 2006.
23. B. Morris. Improved mixing time bounds for the Thorp shuffle and L-reversal chain. February 4, 2008. arXiv:0802.0339
24. B. Morris. The mixing time for simple exclusion. *Annals of Applied Probability*, 16(2), 2006.
25. B. Morris. The mixing time of the Thorp shuffle. *SIAM J. on Computing*, 38(2), pp. 484–504, 2008. Earlier version in *STOC 2005*.
26. B. Morris and Y. Peres. Evolving sets, mixing and heat kernel bounds. *Probability Theory and Related Fields*, 133(2), pp. 245–266, 2005.
27. M. Naor and O. Reingold. On the construction of pseudo-random permutations: Luby-Rackoff revisited. *J. of Cryptology*, 12(1), pp. 29-66, 1999.
28. J. Patarin. Generic attacks on Feistel schemes. Cryptology ePrint report 2008/036.
29. J. Patarin. Luby-Rackoff: 7 rounds are enough for $2^{n(1-\varepsilon)}$ security. *CRYPTO 2003*, LNCS vol. 2729, Springer, pp. 513–529, 2003.
30. J. Patarin. A proof of security in $O(2^n)$ for the Benes scheme. *Progress in Cryptology – AFRICACRYPT 2008*, LNCS vol. 5023, Springer, pp. 209–220, 2008.
31. J. Patarin. Security of random Feistel schemes with 5 or more rounds. *CRYPTO 2004*, LNCS vol. 3152, Springer, pp. 106–122, 2004.
32. E. Petrank and C. Rackoff. CBC MAC for real-time data sources. *J. of Cryptology*, 13(3), pp. 315–338, 2000.
33. P. Rogaway. A synopsis of format-preserving encryption. Manuscript, Sept. 2008.
34. S. Rudich. Limits on the provable consequences of one-way functions. Ph.D. Thesis, UC Berkeley, 1989.
35. L. Saloff-Coste. Random walks on finite groups. In *Probability on Discrete Structures*, *Encyclopedia of Mathematical Sciences*, vol. 110, H. Kesten, editor, Springer, pp. 263–346, 2004.
36. B. Schneier and J. Kelsey. Unbalanced Feistel networks and block-cipher design. *Fast Software Encryption* (FSE 1996), LNCS vol. 1039, Springer,pp. 121–144, 1996.
37. R. Schroeppel. Hasty Pudding Cipher specification. Manuscript available at http://richard.schroeppel.name:8015/hpc/hpc-spec. June 1998 (revised 5/99).

38. T. Spies. Feistel finite set encryption. NIST submission, February 2008. Available at csrc.nist.gov/groups/ST/toolkit/BCM/modes_development.html.
39. T. Spies. Personal communications, February 2009.
40. W. Steiger. A best possible Kolmogoroff-type inequality for martingales and a characteristic property. *Annals of Mathematical Statistics*, 40, pp. 764–769, 1969.
41. E. Thorp. Nonrandom shuffling with applications to the game of Faro. *Journal of the American Statistical Association*, 68, pp. 842–847, 1973.
42. H. Zechner. Efficient sampling from continuous and discrete distributions. Ph.D. Thesis, Institute for Statistics, TU Graz, 1997.

A Lemma about Total Variation Distance

The following Lemma was used in the proof of Theorem 1. It is essentially Lemma 12 in [24]. We reproduce it here for the convenience of the reader.

Lemma 2. Fix a finite 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

$$\|\mu - \nu\| \le \sum_{l=0}^{q-1} \mathbf{E}\Big(\|\mu(\,\cdot \mid Z_1, \ldots, Z_l) - \nu(\,\cdot \mid Z_1, \ldots, Z_l)\| \Big). \tag{9}$$

Proof. For probability distributions $\widehat{\mu}$ and $\widehat{\nu}$, the total variation distance is

$$\|\widehat{\mu} - \widehat{\nu}\| = \min_{W_1 \sim \widehat{\mu},\, W_2 \sim \widehat{\nu}} \mathbf{P}(W_1 \neq W_2)\,. \tag{10}$$

Thus for every l and $z_1, \ldots, z_l$, we can construct $W_1 \sim \mu(\,\cdot \mid z_1, \ldots, z_l)$ and $W_2 \sim \nu(\,\cdot \mid z_1, \ldots, z_l)$ such that

$$\mathbf{P}(W_1 \neq W_2) = \|\mu(\,\cdot\, | z_1, \ldots, z_l) - \nu(\,\cdot \mid z_1, \ldots, z_l)\|\,.$$

We couple $Z \sim \mu$ with $Y \sim \nu$ as follows. Choose (X_1, Z_1) according to the optimal coupling (i.e., a coupling that achieves the minimum in the RHS of (10)), and subsequently for all l with $1 \le l \le q-1$, if $(Z_1, \ldots, Z_l) = (X_1, \ldots, X_l)$, then choose (Z_{l+1}, X_{l+1}) according to the optimal coupling of $\mu(\,\cdot \mid Z_1, \ldots, Z_l)$ and $\nu(\,\cdot \mid Z_1, \ldots, Z_l)$; else couple (X_{l+1}, Z_{l+1}) in an arbitrary way. Note that

$$\mathbf{P}(Z \neq Y) = \sum_{l=0}^{q-1} \mathbf{P}\Big((Z_1, \ldots, Z_l) = (Y_1, \ldots, Y_l),\ Z_{l+1} \neq Y_{l+1} \Big). \tag{11}$$

But on the event that $(Z_1, \ldots, Z_l) = (Y_1, \ldots, Y_l)$, the pair (Z_{l+1}, Y_{l+1}) is chosen according to the optimal coupling of $\mu(\,\cdot \mid Z_1, \ldots, Z_l)$ and $\nu(\,\cdot \mid Z_1, \ldots, Z_l)$, so the RHS of (11) is at most $\sum_{l=0}^{q-1} \mathbf{E}\Big(\|\mu(\,\cdot \mid Z_1, \ldots, Z_l) - \nu(\,\cdot \mid Z_1, \ldots, Z_l)\| \Big)$. □

B An Attack on Balanced Feistel

Let us describe an attack on balanced Feistel that takes $q = 2^{\theta+n/2}$ queries, where $\theta \geq 0$, and see what it implies. The attack is simply to ask q random but distinct queries $x_1, \ldots, x_q \in \{0,1\}^n$, getting responses $y_1, \ldots, y_q \in \{0,1\}^n$, and then see if there is any sequence of round functions $f_1, \ldots, f_r : \{0,1\}^{n/2} \to \{0,1\}^{n/2}$ that are consistent with the gathered query/response pairs. If there is one, output 1, else output 0. Obviously this is not efficient, but we are only trying to evidence that an information-theoretic proof is impossible, not that a practical attack is in hand. To lower-bound the adversary's effectiveness, note that for a random permutation $\pi : \{0,1\}^n \to \{0,1\}^n$ there are $N!/(N-q)!$ possible tuples $(x_1, y_1), \ldots, (x_q, y_q)$ that the adversary might observe, where $N = 2^n$, whereas an n-bit, r-round balanced Feistel cipher, $\Phi[n,r]$, encompasses at most $2^{r\cdot(n/2)\cdot 2^{n/2}}$ possible permutations. So the adversary $\mathcal{A}$ we have described has at least advantage

$$\mathbf{Adv}^{\mathrm{prp}}_{\Phi[n,r]}(\mathcal{A}) \;\geq\; 1 - \frac{2^{\,r\cdot\frac{n}{2}\cdot 2^{n/2}}}{N!/(N-q)!} \tag{12}$$

To illustrate numerically, when $n = 30$ and $q = 2^{20}$, one needs $r = 64$ rounds of Feistel so that $\mathcal{A}$'s advantage will not be large; in fact, with $r = 63$ rounds the adversary's advantage is essentially 1 (it exceeds $1 - 10^{-10000}$). To verify the earlier claim that you need $r = 2^{\theta+1}$ rounds to withstand $q = 2^{\theta+n/2}$ queries, set $r = 2^\theta$ and notice that $2^{n/2} \leq 2^n - q = 2^n - 2^{\theta+n/2}$ for $n \geq 2$, so

$$2^{r\cdot\frac{n}{2}\cdot 2^{n/2}} = (2^{n/2})^q \leq 2^n \cdot 2^{n-1} \cdot \cdots \cdot (2^n - q + 1) = \frac{N!}{(N-q)!} \leq (2^n)^q = 2^{2r\cdot\frac{n}{2}\cdot 2^{n/2}}$$

Therefore, letting $r^*(n,q)$ denote the minimum $r \geq 1$ such that such that the lower bound in (12) is 0, we have $2^\theta \leq r^*(n,q) \leq 2^{\theta+1}$.