COMMUNICATIONS IN INFORMATION AND SYSTEMS
Vol. 1, No. 2, pp. 181–204, April 2001

004

A POLYNOMIAL-TIME UNIVERSAL SECURITY AMPLIFIER IN THE CLASS OF BLOCK CIPHERS*

JOHN O. PLIAM†

Abstract. By construction, we establish the existence of an efficient block cipher with the property that whenever it is composed with any non-perfect cipher, the resulting product is strictly more secure, against an ideal adversary, than the original cipher. We call this property universal security amplification, and note that it holds for at least one trivial stream cipher (notably, the one-time pad). However, as far as we are aware, our construction is the first efficient block cipher possessing this property. Several practical implications of this result are considered.

1. Introduction. It is often asked in cryptography whether the product of two ciphers might be more or less secure than one of the ciphers by itself. An *amplification* of security doesn't happen in general and important counterexamples have been identified. For example, if the permutations of a block cipher form a group (or more precisely, are uniformly distributed on a subgroup of the symmetric group on the set of message blocks), then two-key double encryption is no better than single encryption. Thus, it is important to rule out this pathology in the case of DES (as was done in [5]). Furthermore, the security of a product cipher may actually be less than that of the second cipher when the plaintext statistics are ill-behaved with respect to the permutations of the first cipher [16]. Nevertheless, depending on how security is measured and how the ciphers are modeled, other affirmative results have been advanced [23, 9, 1].

In this paper, we take a novel approach to this problem, raising a strong existence question about the security of product ciphers. Specifically we ask: Is there an efficient block cipher which amplifies the security, against an ideal adversary (one who has unbounded time and data complexities as described below), of every non-perfect cipher with which it is composed? By construction, we answer this question affirmatively.

The constructed cipher, as presented, would not be widely viewed as practical because it requires a variable length key which grows with the amount of plaintext encrypted (much like a one-time pad). On the other hand, if a cryptographically strong substitute for the key were used (such as a key schedule, hash, or pseudo-random function), then the strength of the security amplification would be no worse than the strength of the key substitute.

There are other practical implications of our result. First of all, the techniques used here could facilitate the construction of computationally efficient primitives (such as dynamic S-boxes) with provably strong security properties. More generally, if we are to understand, in more than purely heuristic terms, the security convergence of modern iterated cryptosystems, then our result establishes new limits on what

*Received May 1, 2000; accepted for publication Feb 5, 2001.
†Laboratory for Security and Cryptography (LASEC), Swiss Federal Institute of Technology, Lausanne (EPFL), LASEC–DSC–EPFL, CH–1015 Lausanne, Switzerland, E-mail: pliam@atbash.com.

can be accomplished in polynomial-time. Our construction might be modified and compromised to obtain faster ciphers with complementary security results.

2. Preliminaries. A basic familiarity with random variables and probability spaces [12] is assumed. Some group theory [22] is also assumed, but in the next subsection, we shall review some important terminology about permutation groups [8].

2.1. Permutation Groups. Let $\mathscr{X}$ be any set. The collection of all invertible functions on $\mathscr{X}$ forms the *symmetric group* $\mathfrak{S}_{\mathscr{X}}$. Any subgroup $G \leq \mathfrak{S}_{\mathscr{X}}$ is called a *permutation group*, and we also say that G *acts on* $\mathscr{X}$ and that $\mathscr{X}$ is a *G-set*. The subgroup of G which fixes a point $x \in \mathscr{X}$ is called the *(point) stabilizer* of x, and is given by $\mathrm{Stab}_G(x) = \{h \in G \mid hx = x\}$.

When studying n-bit block ciphers, the finite set $\mathscr{M} = \{0,1\}^n$ of all n-bit binary strings (or equivalently the integers $\{0, 1, \ldots, 2^n - 1\}$) is the most natural G-set for some permutation group $G \leq \mathfrak{S}_{\mathscr{M}}$. In this paper, we will consider two other actions of G on related sets. By $\mathscr{M}^{(\ell)}$ we mean the set of tuples of size ℓ with distinct elements in $\mathscr{M}$, G acting elementwise. If $p = (p_1, \ldots, p_\ell) \in \mathscr{M}^{(\ell)}$, the point stabilizer $\mathrm{Stab}_G(p)$ is sometimes written as $\mathrm{Stab}_G(p_1, \ldots, p_\ell)$. By $\mathscr{M}^{\{m\}}$ we mean the set of subsets of $\mathscr{M}$ of size m, where $g \in G$ acts on $S \in \mathscr{M}^{\{m\}}$ by taking $S \mapsto gS$. The point stabilizer of $S \in \mathscr{M}^{\{m\}}$ is sometimes written as $\mathrm{Stab}_G\{S\}$.

2.2. Majorization. Given two n-dimensional positive vectors $x, y \in \mathbb{R}_+^n$, we write $x \preceq y$ and say that x is *majorized* by y or that y *majorizes* x if

$$\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i, \tag{2.1}$$

$$\sum_{i=1}^{k} x_{[i]} \leq \sum_{i=1}^{k} y_{[i]}, \quad 1 \leq k \leq n, \tag{2.2}$$

where square brackets in $x_{[i]}, y_{[i]}$ indicate rearrangements of x_i, y_i (respectively) into non-increasing order. Of course, (2.1) holds automatically for discrete probabilities. Note that here we stray from common notation for majorization. We must reserve the usual notation $x \prec y$ for strict majorization which is defined in the obvious way. If $x \preceq y$ while $y \not\preceq x$, then we say that x is *strictly majorized* by y, and write $x \prec y$. Strict majorization will turn out to be useful for establishing strict security inequalities.

Of fundamental importance are two theorems (see [15]) which, in a sense, characterize majorization algebraically. The *Hardy-Littlewood-Pólya theorem* [13] states that $x \preceq y$ iff $x = Dy$, for some $n \times n$ doubly stochastic matrix D. *Birkhoff's theorem* [3] states that every doubly stochastic matrix D is a convex sum of permutation matrices. Thus majorization $x \preceq y$ is equivalent to

$$x = \left(\sum_j p_j \Pi_j \right) y, \tag{2.3}$$

where $p_j \geq 0$, $\sum_j p_j = 1$ and each Π_j is a permutation matrix. In the sequel, we shall see that majorization fits nicely with Shannon's model of ciphers.

2.3. Shannon's Model and Product Ciphers. Following Shannon [23], we model an n-bit *block cipher* as a $\mathfrak{S}_{\mathscr{M}}$-valued random variable. If a cipher X only takes values in a subgroup $G \leq \mathfrak{S}_{\mathscr{M}}$, then X may be called a *G-cipher*. We may model a stream cipher in the same spirit (cf. [17]). Let $\{0,1\}^*$ denote the (infinite) set of finite binary strings, and let $H \leq \mathfrak{S}_{\{0,1\}^*}$ be the subgroup of length-preserving permutations. We shall call an H-valued random variable a *stream cipher*[1]. By a *cipher* we mean either a block cipher or a stream cipher.

Given two independent ciphers X and Y acting on the same message space, the cipher XY is called a *product cipher*, Y is called its *first component* and X is called its *second component*. The distribution of the product of two block ciphers is given by the *convolution*,

$$\mathsf{Pr}[XY = g] = x * y(g) \stackrel{\triangle}{=} \sum_{h \in G} x(gh^{-1})y(h), \tag{2.4}$$

where $x(g) = \mathsf{Pr}[X = g]$ and $y(g) = \mathsf{Pr}[Y = g]$. This representation of a product cipher will prove useful in the sequel.

The cipher U which is uniformly distributed on $\mathfrak{S}_{\mathscr{M}}$ is called the *perfect cipher*. For any subgroup $G \leq \mathfrak{S}_{\mathscr{M}}$ the G-cipher U_G which is uniformly distributed on G is called the *uniform G-cipher*. Given an infinite sequence of independent and uniformly random bits, $z_0, z_1, \ldots$, we may form a simple stream cipher, called the *one-time pad*, by mapping plaintext word m into $z(|m|) \oplus m$, where $z(|m|)$ is the word of random bits $z_0 \cdots z_{|m|}$.

2.4. The Computational Model. Shannon's model is a purely probabilistic one; it says very little about how a computer might transform plaintext into ciphertext and back. For a cipher X to be practical, there should be effective procedures for encryption (computing the action of X on plaintext) and decryption (computing the action of X^{-1} on ciphertext).

One natural choice for the computational model is the standard *Turing machine* model [10]. Informally, we have an encryption algorithm Enc, which has as input arguments the plaintext m and the random key k, and which outputs the ciphertext c. The corresponding decryption algorithm Dec is similarly defined. Formally in this model, we require a pair of deterministic Turing machines E and D, such that (under suitable encoding) $m = D(k, E(k, m))$, for all m and k. Notice that under this model, all randomness enters as an argument to the encryption and decryption algorithms, or equivalently as input data on the Turing machine tapes. Our view is that this model of computation is unnecessarily restrictive, because it fails to capture the simple idea that some ciphers (like the one-time pad) are "computationally efficient" even though they may require impractical amounts of key material to encrypt *every* possible plaintext.

Alternatively, we consider encryption and decryption algorithms which access key material as an auxiliary subroutine call. Formally, such a subroutine call is idealized by

[1] In practice, a stream cipher will typically also have consistent *block prefix action*, i.e. for some integer n, it will be confined to permutations $h \in H$ such that when $|u| = |u'| \in n\mathbb{Z}$, $h(uw) = u'w'$ implies that for all v of length $|w|$, $h(uv) = u'v'$ for some v'.

an *oracle function* $f : \{0,1\}^* \longrightarrow \{0,1\}$, and we are thus invoking the computational model of an *oracle Turing machine (OTM)* [10]. An OTM is a deterministic Turing machine augmented by an *oracle tape* and additional logic so that at any time, the oracle tape with input α written on it can, in one step of computation, be transformed to have $f(\alpha)$ written on it. An OTM M with specific oracle function f will be denoted by M^f, and its time complexity is computed in the usual way (with oracle evaluation counting as one step). We may model uncertainty about the oracle function by treating it as an instance of a *random oracle function* $F : \{0,1\}^* \longrightarrow \{0,1\}$.

The next two definitions capture our intuitive notion of efficient encryption /decryption for block and stream ciphers, respectively.

DEFINITION 2.1 (Efficient Block Ciphers). *An ensemble of block ciphers* $\{X_n\}$, $n \in \mathbb{N}$ *is called* **computable in polynomial-time** *if there exists a random oracle function* F *and a pair of polynomial-time OTM's,* E *and* D*, such that for each* $n \in \mathbb{N}$*: (i). for each* $p \in \{0,1\}^n$, $p = D^F(E^F(p))$*, (ii). the distribution of* E^F*, restricted to strings of length* n*, is identical to that of* X_n*, and (iii). the distribution of* D^F*, restricted to strings of length* n*, is identical to that of* X_n^{-1}.

By a mild but common abuse of notation, a block cipher X acting on $\{0,1\}^n$ is called *computable in polynomial-time* if it is one of an ensemble of such ciphers, and any important properties hold for each representative.

DEFINITION 2.2 (Efficient Stream Ciphers). *A stream cipher* X *is called* **computable in polynomial-time** *if there exists a random oracle function* F *and a pair of polynomial-time OTM's,* E *and* D*, such that: (i). for each* $p \in \{0,1\}^*$, $p = D^F(E^F(p))$*, (ii). the distribution of* E^F *is identical to that of* X*, and (iii). the distribution of* D^F *is identical that of* X^{-1}.

Note that by Definitions 2.1 and 2.2, both the one-time pad and the Luby-Rackoff construction [17] are efficient. In fact, each is computable in linear time. Notice also that being computable in polynomial-time does not preclude that exponentially many bits may be necessary to completely describe the cipher's action on the entire message space. For example, each round of the Luby-Rackoff construction (a Feistel cipher with a perfectly random function acting on half-words) takes on one of

$$\left(2^{\frac{n}{2}}\right)^{2^{\frac{n}{2}}}$$

distinct permutations of an n-bit message space. Thus, for the common 3-round version of the construction, there must be $3n2^{\left(\frac{n-2}{2}\right)}$ bits to entirely describe it.

However, neither the one-time pad nor the Luby-Rackoff construction meets our objective. The one-time pad is not a block cipher. Furthermore, every permutation of the Luby-Rackoff construction is even and hence is confined to a proper subgroup (the alternating group, $\mathfrak{A}_{\mathscr{M}} \leq \mathfrak{S}_{\mathscr{M}}$), and we shall see from Lemma 3.2 below that it cannot be a universal security amplifier.

2.5. Optimal Chosen Plaintext Attacks. We now introduce the measure of security in terms of which strict security inequalities will be derived. Informally, it is just the average cost of the optimal (non-adaptive) chosen plaintext attack for an adversary in possession of an oracle which will answer the question, "is $X = g$?".

Given such an oracle for any random variable Z the average cost of guessing its value is called the *guesswork*[2] of Z which is given by

$$W(Z) \triangleq \sum_{i=1}^{n} p_{[i]} i, \tag{2.5}$$

where Z takes on n values, and the probabilities of Z have been arranged according to $p_{[i]} \geq p_{[j]}$ for all $i < j$. The r.h.s. of (2.5) extends to any positive vector $p = (p_1, \ldots, p_n) \in \mathbb{R}_+^n$, thus defining $W(p)$.

Now, there are two stages in the optimal strategy for guessing X in a chosen plaintext attack. First the adversary discards all permutations which are inconsistent with the acquired plaintext-ciphertext pairs. Then among the remaining permutations, he queries the oracle for the exact permutation in order of non-increasing probability. The adversary will obviously choose the plaintexts such that the average cost of this strategy is minimized. The difficulty of this attack is a direct and meaningful measure of the cipher's security. To formally quantify its cost, let us assume that the adversary has collected ℓ plaintexts and their corresponding ciphertexts into tuples $p, c \in \mathscr{M}^{(\ell)}$, respectively. The ciphertext tuple c is an instance of the random variable $C^\ell = Xp$, whose uncertainty is due exclusively to the uncertainty about X.

For fixed p and c, the *conditional guesswork* $W(X|c,p)$ is the guesswork of X as in (2.5) after discarding all permutations $g \in G$ such that $c \neq gp$, and then rearranging and rescaling the probabilities accordingly. Now we must still account for the uncertainty about C^ℓ. Evidently, for a particular choice of plaintext tuple p, the cost of the attack must be weighted by the *a posteriori* probabilities $\omega(c|p) = \mathsf{P}\left[C^\ell = c \mid p\right]$, yielding

$$W(X|C^\ell, p) = \sum_{c \in \mathscr{M}^{(\ell)}} W(X|c,p)\omega(c|p). \tag{2.6}$$

The minimum value of $W(X|C^\ell, p)$ is the *optimal chosen plaintext attack work factor*, which will be denoted

$$\theta_\ell(X) = \min_{p \in \mathscr{M}^{(\ell)}} W(X|C^\ell, p). \tag{2.7}$$

For continuity we take $\theta_0(X)$ to be $W(X)$.

3. The Main Result.

3.1. The Existence Theorem. We shall prove by construction the following theorem.

THEOREM 3.1. *There is a cipher X, computable in polynomial-time, such that for each $0 \leq \ell \leq 2^n$ and every independent cipher Y, $\theta_\ell(XY) \geq \theta_\ell(Y)$. Furthermore, equality holds iff $\theta_\ell(Y) = \theta_\ell(U)$.*

It is easily seen (see e.g. [19]) that no non-perfect cipher Y can have $\theta_\ell(Y) = \theta_\ell(U)$, for all ℓ. Thus this theorem tells us in a very meaningful way, that every non-perfect cipher is brought closer to the perfect cipher by left multiplication by X.

[2] Guesswork has sometimes been called *guessing entropy*, cf. [4] and [20].

The proof of Theorem 3.1 relies on three lemmas which treat different aspects of the problem. To express these lemmas succinctly, we define the *support* of a G-cipher (or indeed any random variable) as $\operatorname{supp} X = \{g \in G \mid \mathsf{Pr}[X = g] \neq 0\}$.

The first lemma treats the case $\ell = 0$ and establishes an algebraic equivalent condition for universal amplification with zero data complexity. Thus it is quite powerful, and for example, can provide a quick way to show that a cipher is *not* a universal amplifier.

LEMMA 3.2. *Given $G \leq \mathfrak{S}_{\mathscr{M}}$, let X be a G-cipher. Every independent non-uniform G-cipher Y satisfies $W(XY) > W(Y)$, iff for each $g \in G$ and each subgroup $H \neq G$, $\operatorname{supp} X \not\subseteq gH$.*

The next lemma provides sufficient conditions for nearly universal amplification ($\ell > 0$) for ciphers in any permutation group.

LEMMA 3.3. *For a permutation group $G \leq \mathfrak{S}_{\mathscr{M}}$, let X be a G-cipher such that $\operatorname{supp} X = G$. Then for each $1 \leq \ell \leq 2^n$ and every independent G-cipher Y, $\theta_\ell(XY) \geq \theta_\ell(Y)$. Furthermore, equality holds iff $\theta_\ell(Y) = \theta_\ell(U_G)$.*

The final lemma asserts the existence of a cipher suitable to translate Lemmas 3.2 and 3.3 into Theorem 3.1.

LEMMA 3.4. *There exists a cipher X which is computable in polynomial-time and satisfies $\operatorname{supp} X = \mathfrak{S}_{\mathscr{M}}$.*

Assuming the validity of the above lemmas, the proof of Theorem 3.1 is immediate.

The three lemmas are proved in sections 4.3, 5.3 and 5.4. Before diving into the details, we first take a slightly informal look at the ideas underlying the construction of the cipher X, as the somewhat counterintuitive property of Lemma 3.4 plays a central role in this paper.

3.2. An Intuitive Glimpse at the Construction. The symmetric group on the message space is truly enormous. Its size is approximated by

$$\log\log(2^n!) \approx n + \log(n) = O(n).$$

Because it takes two logarithms to bring $2^n!$ down to the polynomial n, our construction will exhibit two distinct sources of algorithmic efficiency:

1. *Recursion:* The cipher X is recursively defined as the product of simpler ciphers. More precisely, the encryption algorithm Enc is itself recursive but also calls another recursive algorithm invSort. The decryption algorithm Dec is similarly defined. The time complexity and recursion depth of each algorithm is a polynomial in n.
2. *Oblivious Action*[3]*:* The cipher X is represented as a product of a large number of random powers of transpositions (i.e. permutations of message blocks two at a time). So Enc and Dec, the defining algorithms of X, make use of only polynomially many transpositions for every block encrypted.

Let $G \leq \mathfrak{S}_{\mathscr{M}}$ be any permutation group. There are many ways to construct a product cipher PQ which achieves every permutation in G, even though *both* P and Q

[3]We borrow this term from [18] where it is used in the same context.

are sparse on G. Indeed for any subgroup $H \leq G$, we may take Q which achieves every permutation in H and P which achieves one permutation in every left coset of H in G. It is easy to see that PQ achieves every permutation in G. For many large groups, it is possible to find subgroups satisfying $|G| \gg |H|$ and $|G| \gg [G:H]$. Formally, we have an amplification of support: $|\operatorname{supp} PQ| \gg |\operatorname{supp} P|$, and $|\operatorname{supp} PQ| \gg |\operatorname{supp} Q|$. Thus by exploiting the algebraic structure of the group, we may construct a densely distributed cipher as a product of very sparsely distributed ciphers.

Let's try to carry this idea even further. Consider a chain of subgroups of G

$$\{1\} = H_0 \leq H_1 \leq \cdots \leq H_m = G,$$

and for each i, an H_i-cipher P_i which contains one permutation in every left coset of H_{i-1} in H_i. Then by simple induction, the product cipher $P_m \cdots P_2 P_1$, would have complete support on G. For example, in the symmetric group on 2^n symbols, consider the subgroups $H_i = \operatorname{Stab}(1, \ldots, 2^n - i)$, $0 \leq i \leq 2^n$. On the one hand, this choice of subgroups is promising because the number of cosets in $\mathfrak{S}_{2^n}$ of the largest proper subgroup is the polynomial n. Unfortunately however, there are 2^n subgroups in this chain, and so the number of terms in the product $P_m \cdots P_2 P_1$ grows exponentially with n. If we are to employ this technique, it may be inconvenient to use a chain of subgroups which fix collections of words in $\mathscr{M}$, either as tuples or as sets, because any hierarchy of such collections would typically be as large as $\mathscr{M}$ itself.

It thus makes more sense to define subgroups which fix some feature of the words in $\mathscr{M}$. To that end define K_i to be the subgroup consisting of the permutations of $\mathfrak{S}_{\mathscr{M}}$ which preserve the first $n - i$ bits of each message block. We shall call K_i the $(n-i)$-*bit prefix stabilizer subgroup* of $\mathfrak{S}_{\mathscr{M}}$, and as i ranges from 0 to n these form the chain of subgroups

$$\{1\} = K_0 \leq K_1 \leq \cdots \leq K_n = \mathfrak{S}_{\mathscr{M}}. \tag{3.1}$$

We will construct, for each $1 \leq i \leq n$, a K_i-cipher P_i which contains one permutation in every left coset of K_{i-1} in K_i. In this way, the cipher of Lemma 3.4 is defined as

$$X = P_n \cdots P_2 P_1. \tag{3.2}$$

Let us compute the minimal support required by P_n. That is to say, let us count the number of left cosets of K_{n-1} in $\mathfrak{S}_{\mathscr{M}}$. Since K_{n-1} permutes all but the most significant bit of words in $\mathscr{M}$, the left cosets of K_{n-1} are characterized by the rearrangements of $\mathscr{M}$ with distinct patterns of the most significant bit. There are precisely

$$[\mathfrak{S}_{\mathscr{M}} : K_{n-1}] = \begin{pmatrix} 2^n \\ 2^{n-1} \end{pmatrix}$$

rearrangements. Observe that while we have reduced the number of permutations by a large number (by $(2^{n-1}!)^2$ in fact), on a doubly logarithmic scale we still have

$$\log\log \begin{pmatrix} 2^n \\ 2^{n-1} \end{pmatrix} \approx n + 1 = O(n).$$

It may appear that we are right back where we started. But as we shall see, we have transported the problem onto fertile new ground.

The efficiency in our algorithms for P_i has its heritage in the closely related problem of card shuffling. In fact, both the security of a product cipher [19] and the fairness of a shuffled deck of cards [2, 7] are related to the uniformity of convolutions as in (2.4). In their analysis of riffle shuffles, Aldous and Diaconis remarked that "the lovely new idea here is to consider shuffling as inverse sorting." [2, Remark (a), p. 344]. Indeed it is quite natural to consider *encryption* as inverse sorting because the rearrangements of $\mathscr{M}$ which characterize the left cosets of $K_{n-1} \leq \mathfrak{S}_{\mathscr{M}}$ correspond precisely to the permutations which would be used in the *first* step of the obvious recursive sorting algorithm. In the reverse order, we may achieve all permutations of $\mathscr{M}$ by first achieving all rearrangements of the most significant bit, and then proceeding recursively with the less significant bits. What we claim is that sorting and inverse sorting on the most significant bit can be done in polynomial-time using both recursion and the oblivious action of transpositions. The rest is gravy.

Let us demonstrate this efficiency in a simple example with $n = 3$ and thus $\mathscr{M} = \{0, 1, \ldots, 7\}$. We start with a random arrangement $(6, 3, 5, 0, 7, 1, 4, 2)$ of the elements of $\mathscr{M}$, and attempt to sort this tuple on the most significant bit by the application of $n = 3$ rounds of involutions (recall that every involution is a product of disjoint transpositions). For reasons of efficiency we shall restrict ourselves to transpositions of the form $(j, j \oplus 2^i)$, with i constant for every round. The allowable round involutions are $(01)^{b_1}(23)^{b_2}(45)^{b_3}(67)^{b_4}$, $(02)^{b_5}(13)^{b_6}(46)^{b_7}(57)^{b_8}$ and $(04)^{b_7}(15)^{b_9}(26)^{b_{10}}(37)^{b_{11}}$, for rounds 0, 1 and 2, respectively. Table 3.1 below shows that we can indeed sort on the most significant bit of 2^n integers by carefully choosing the powers b_i in only n rounds.

		round (transp./arrangemnt.)					
		0		**1**		**2**	
7	$2 = 010$	(67)	100	()	100	()	100
6	$4 = 100$	(67)	010	(46)	111	()	111
5	$1 = 001$	()	001	()	001	(15)	101
4	$7 = 111$	()	111	(46)	010	(04)	110
3	$0 = 000$	(23)	101	(13)	011	()	011
2	$5 = 101$	(23)	000	()	000	()	000
1	$3 = 011$	()	011	(13)	101	(15)	001
0	$6 = 110$	()	110	()	110	(04)	010

TABLE 3.1

A randomly chosen arrangement of $\{0, 1, \ldots, 7\}$ *is sorted with respect to the most significant bit after the application of only* 3 *rounds of disjoint transpositions. The first two columns indicate the initial arrangement (position, value). The next three columns give, for each round, the transposition affecting the value at that position and subsequent arrangement.*

To overcome the limitations of having so few permutations, our strategy is as follows: the goal at the end of round 1, is to collect integers with leading 1 into the lowest part of the bottom half (those positions ≤ 3), and to collect integers

with leading 0 into the lowest part of the top half (those positions ≥ 4). Then the powers of the transpositions in the final round (round 2) are determined by the sorting requirement. We claim that this strategy will work for all n.

4. The Construction: Lemma 3.4. In this section, we formally describe the universal security amplifier X and prove Lemma 3.4.

4.1. Algebraic Details. We may encrypt by inverting the sorting procedure described in the previous section. Formally, for any j, define $R_i^{(j)}$ to be the product of independent and uniformly random powers of the 2^{n-1} distinct transpositions of the form $(k, k \oplus 2^i)$, with $0 \leq k \leq 2^n - 1$. Let

$$P_i = R_0^{(i)} R_1^{(i)} \cdots R_{i-1}^{(i)}.$$

Then, as before, $X = P_n \cdots P_2 P_1$. Each random involution $R_i^{(j)}$ corresponds to a "round" as shown in Fig. 4.1 below. Note that while there is repetition (e.g. $R_i^{(j_1)}$ and $R_i^{(j_2)}$ are i.i.d. random variables), X is *not* a traditional iterated cryptosystem because adjacent rounds are not identical and the specific sequence of rounds is carefully chosen.

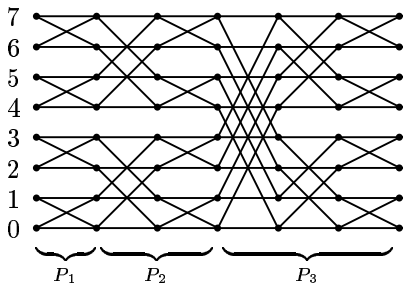

FIG. 4.1. *The structure of the cipher $X = P_3 P_2 P_1$ for $n = 3$. The rounds are applied left to right, and each round corresponds to a random involution shown as a vertical column of butterflies. Each butterfly in the diagram represents a random transposition of the form $(k, k \oplus 2^i)^b$, where parallel lines indicate $b = 0$ and a crossover indicates $b = 1$.*

4.2. Algorithm Details. It is clear that we may recursively affect the actions of X and X^{-1} on any block, if we can carry out the rounds $R_i^{(j)}$ in the correct order and in such a way that the powers of all relevant transpositions are independent and equiprobable. Moreover, if we encounter the the same butterfly in two different executions, we must be able to reproduce the same random power of the corresponding transposition. This is easily accomplished if we consider the random bits to be indexed by $\mathscr{M} \times \mathbb{Z}$. The resulting function $f : \mathscr{M} \times \mathbb{Z} \longrightarrow \{0,1\}$ is easily transformed into a random oracle function $F : \{0,1\}^* \longrightarrow \{0,1\}$ appropriate for Definition 2.1. We employ the convention that the power of any transposition $(k, k \oplus 2^i)$ is $f(m, r)$, where $m = \min\{k, k \oplus 2^i\}$ and r is the round. In other words, f is applied to the lower left hand corner of every butterfly in Fig. 4.1.

The next algorithm implements encryption. To encrypt a single plaintext block, the computational complexity will be $n(n+1)/2$ or $O\left(\frac{1}{2}n^2\right)$. For a block size of $n = 128$, this yields about $8,256$ operations.

ALGORITHM 1. *This algorithm defines recursive encryption functions* Enc *and* invSort. *The action of* $X = P_n \cdots P_2 P_1$ *on* $p \in \mathscr{M}$ *is affected by* $(q, r) = \mathsf{Enc}(p, n, 1)$ *so that* $q = Xp$. *The action of* P_i *on* $p \in \mathscr{M}$ *is affected by* $(q, r) = \mathsf{invSort}(p, i-1, *)$ *so that* $q = P_i p$.

```
function Enc(p, i, r):
  if i > 1 then
    (q, r) = Enc(p, i − 1, r).
  endif
  return invSort(q, i − 1, r).
```

```
function invSort(p, j, r):
  q = p ⊕ 2^j.
  if p < q then
    b = f(p, r).
  else
    b = f(q, r).
  endif
  if b = 0 then
    q = p.
  endif
  if j > 0 then
    return invSort(q, j − 1, r + 1).
  else
    return (q, r + 1).
  endif
```

The decryption algorithm is easily obtained by performing the the transpositions in the reverse order. The necessary modifications are immediate, and we shall call the "reverse" of inverse-sorting fwdSort.

ALGORITHM 2. *This algorithm defines recursive decryption functions* Dec *and* fwdSort. *The action of* $X^{-1} = P_1^{-1} P_2^{-1} \cdots P_n^{-1}$ *on* $p \in \mathscr{M}$ *is affected by* $(q, r) = \mathsf{Dec}(p, n, \frac{1}{2}n(n+1))$ *so that* $q = X^{-1}p$. *The action of* P_i^{-1} *on* $p \in \mathscr{M}$ *is affected by* $(q, r) = \mathsf{fwdSort}(p, i-1, *)$ *so that* $q = P_i^{-1} p$.

```
function Dec(p, i, r):
  (q, r) = fwdSort(q, i − 1, r).
  if i > 1 then
    return Dec(q, i − 1, r).
  else
    return (q, r).
  endif
```

```
function fwdSort(p, j, r):
  if j > 0 then
    (p, r) = fwdSort(p, j − 1, r).
  endif
  q = p ⊕ 2^j.
  if p < q then
    b = f(p, r).
  else
    b = f(q, r).
  endif
  if b = 0 then
    return (p, r − 1).
  else
    return (q, r − 1).
  endif
```

REMARK 1. *Notice how the round information is explicitly carried by input–output argument r through the entire recursion processed. During the execution of* Enc, *it is incremented, while during the execution of* Dec *it is decremented. This is necessary because encryption and decryption must agree on the random bits $f(p,r)$ which determine the appropriate powers of the various transpositions involved.*

4.3. The Proof of Lemma 3.4. To prove Lemma 3.4 we must first develop some terminology and prove some preliminary results. Recall that the integers in $\mathscr{M}$ will have a dual role as n-bit strings. When treating prefixes and other substrings it is useful to have a *padding function* $\pi_i : \mathbb{Z} \longrightarrow \{0,1\}^i$ taking j to the binary representation of $j \mod 2^i$ padded up to i bits. Also define a *prefix truncation function* $\tau_i : \{0,1\}^* \longrightarrow \{0,1\}^i$ taking binary word w to its first i bits (the most significant i bits).

It is natural for us to recursively partition $\mathscr{M}$ into disjoint subsets which share the same prefix. For example, let $S_0 = \{i \in \mathscr{M} \mid \tau_1(i) = 0\}$ and $S_1 = \{i \in \mathscr{M} \mid \tau_1(i) = 1\}$, so that $\mathscr{M}$ is the disjoint union $S_0 \cup S_1$. More generally, let $S_{\pi_i(j)} = \{k \in \mathscr{M} \mid \tau_i(k)\}$ $= \pi_i(j)$ with $1 \le j \le 2^i - 1$, and again we partition $\mathscr{M}$ into disjoint subsets

$$\mathscr{M} = \bigcup_{j=0}^{2^i-1} S_{\pi_i(j)}.$$

The prefix stabilizers are naturally expressed in terms of these subsets. For example, $K_{n-1} = \mathrm{Stab}_{K_n}\{S_0\} \cap \mathrm{Stab}_{K_n}\{S_1\}$, and more generally

$$K_{n-i} = \bigcap_{j=0}^{2^i-1} \mathrm{Stab}_{K_n}\{S_{\pi_i(j)}\}.$$

The following proposition characterizes the left cosets of $K_{n-1} \le K_n$.

PROPOSITION 4.1. *A left coset of K_{n-1} in K_n is completely determined by the image of S_0 under the action of any left coset representative.*

Proof. First of all $K_{n-1} = \mathrm{Stab}_{K_n}\{S_0\} \cap \mathrm{Stab}_{K_n}\{S_1\} = \mathrm{Stab}_{K_n}\{S_0\}$, because anything which fixes S_0 must also fix S_1. Now $K_n = \mathfrak{S}_{\mathscr{M}}$ acts transitively on the set $\mathscr{M}^{\{2^{n-1}\}}$ of all subsets of $\mathscr{M}$ of half its size. By standard group action arguments [22, 8], the left cosets $\{gK_{n-1}\}$ are in one-to-one correspondence with the images $\{gS_0\}$, in a well-defined way. □

We shall derive presently a similar characterization of the left cosets of K_{n-i-1} in K_{n-i}. First, whenever $A \subset B$ we will treat $\mathfrak{S}_A$ to be a subgroup of $\mathfrak{S}_B$. Recall [22] that if a group G factors into product $G = HK$ of normal subgroups H and K, with $H \cap K = \{1\}$, then G is a *direct product* of H and K (it is literally isomorphic to the Cartesian product with the obvious group law). Clearly whenever B is a disjoint union of A_1 and A_2, $\mathfrak{S}_B$ contains the direct product $\mathfrak{S}_{A_1}\mathfrak{S}_{A_2}$. Visibly, $K_{n-1} = \mathfrak{S}_{S_0}\mathfrak{S}_{S_1}$,

and if we write $\mathfrak{S}_{\pi_i(j)} = \mathfrak{S}_{S_{\pi_i(j)}}$, we also have that K_{n-i} is the direct product

$$K_{n-i} = \prod_{j=0}^{2^i-1} \mathfrak{S}_{\pi_i(j)}.$$

PROPOSITION 4.2. *A left coset of K_{n-i-1} in K_{n-i} is completely determined by the images of $S_{\pi_i(j)0}$, $0 \leq j \leq 2^i - 1$, under the action of any left coset representative.*

Proof. Since K_{n-i} is the direct product given above, a left coset gK_{n-i-1} factors into a product of left cosets

$$\prod_{j=0}^{2^i-1} g_j \left(\mathrm{Stab}_{\mathfrak{S}_{\pi_i(j)}}\{S_{\pi_i(j)0}\} \cap \mathrm{Stab}_{\mathfrak{S}_{\pi_i(j)}}\{S_{\pi_i(j)1}\} \right).$$

However, we again have

$$\mathrm{Stab}_{\mathfrak{S}_{\pi_i(j)}}\{S_{\pi_i(j)0}\} \cap \mathrm{Stab}_{\mathfrak{S}_{\pi_i(j)}}\{S_{\pi_i(j)1}\} = \mathrm{Stab}_{\mathfrak{S}_{\pi_i(j)}}\{S_{\pi_i(j)0}\}.$$

Finally, 2^i invocations of Proposition 4.1 obtains the desired result. □

With this machinery in place, we may now prove Lemma 3.4.

Proof. [of Lemma 3.4.] Recall that in order to facilitate the induction argument of Section 3.2, thereby establishing that $\mathrm{supp}\, X = \mathfrak{S}_{\mathscr{M}}$, we must show that (for each i) $\mathrm{supp}\, P_{n-i}$ contains a representative of each left coset of K_{n-i-1} in K_{n-i}.

What we'll actually show, by an inner induction argument, is that for every subset $S \subset \mathscr{M}$ contiguous on each $S_{\pi_i(j)}$ $(0 \leq j \leq 2^i - 1)$ and every possible image T of S under the action of K_{n-i} (i.e., every T of the form gS for some $g \in K_{n-i}$), $\mathrm{supp}\, P_{n-i}$ contains a permutation g taking $S \mapsto T$. Since each $S_{\pi_i(j)0}$ is trivially a contiguous subset of $S_{\pi_i(j)}$, we have the desired result by Proposition 4.2. Note also that if we can take an arbitrary contiguous set to an arbitrary image, then we can also take an arbitrary complement of a contiguous set to an arbitrary image.

Induction Base: Clearly K_1 is isomorphic to the direct product of 2^{n-1} symmetric groups on 2 elements (cyclic groups of order 2), and thus has size $|K_1| = 2^{2^n-1}$. Since $|\mathrm{supp}\, P_1| = 2^{2^n-1}$ also, the induction hypothesis holds trivially.

Induction Step: Without loss of generality, we consider the case $i = 0$. By hypothesis, $\mathrm{supp}\, P_{n-1}$ contains an element of K_{n-1} taking any contiguous subset of S_0 to a desired image ($\subset S_0$, and of the same size), while simultaneously taking any contiguous subset of S_1 to a desired image (again $\subset S_1$, and of the same size). Choose arbitrary sets $U \subset S_0, V \subset S_1$, let $T = U \cup V$, and choose any contiguous set $S \subset \mathscr{M}$ of size $|T|$. Again without loss of generality, assume that $|S \cap S_0| \geq |U|$ (because otherwise $|S \cap S_1| \geq |V|$ and a completely symmetric argument applies). We must show that $\mathrm{supp}\, P_n = \mathrm{supp}\, P_{n-1}\, \mathrm{supp}\, R^{(n)}_{n-1}$ contains a g such that $gS = T$. Write $g = hk\alpha$, with $h \in \mathfrak{S}_{S_0}, k \in \mathfrak{S}_{S_1}$, and where α is some product of transpositions of the form $(j, j \oplus 2^{n-1})$. Evidently the real job of α is to send elements of $S \cap S_0$ in excess of $|U|$ across the most significant bit boundary into S_1, because $h, k \in \mathrm{Stab}_{K_n}\{S_0\}$ cannot do this later on. The transpositions in $\mathrm{supp}\, R^{(n)}_{n-1}$, which flip the most significant bit,

are perfect for this task. Let α be the product of the transpositions $(j, j \oplus 2^{n-1})$, with $j \in J$, where J consists of the highest $|S \cap S_0| - |U|$ elements of $S \cap S_0$. We claim that $(\alpha S) \cap S_0$ is a contiguous subset of S_0, and that $(\alpha S) \cap S_1$ is either a contiguous subset or the complement of a contiguous subset of S_1. Assuming that is true, then by the induction hypothesis, we may choose h taking $(\alpha S) \cap S_0 \mapsto U$ and k taking $(\alpha S) \cap S_1 \mapsto V$, so that $gS = hk(\alpha S) = T$.

Two cases naturally arise. (*Case 1:*) If S doesn't intersect with S_1 then α takes J contiguously to some image in the middle of S_1, and α leaves $S - J$ contiguously in the middle of S_0. (*Case 2:*) If S intersects non-trivially with S_1, then because S is contiguous, J is precisely the highest $|S \cap S_0| - |U|$ elements of S_0 itself, and furthermore $S \cap S_1$ consists of the lowest $|S \cap S_1|$ elements of S_1 which are left fixed by α. Therefore, $(\alpha S) \cap S_1$ consists of the complement of a contiguous set (those elements between $S \cap S_1$ and αJ). But again α leaves $(S \cap S_0) - J$ contiguously in the middle of S_0. This completes the induction step for $i = 0$.

Applying this same argument within the appropriate direct product subgroups when $i > 0$ yields the inner induction step and thus completes the proof. □

REMARK 2. *The previous proof seems harrowing with 2 cases nested inside 2 w.l.o.g.'s nested inside of 2 layers of induction. But, it is in essence just a rigorous form of the more intuitive sorting example given in the previous section (which may have seemed simpler at first glance).*

5. Security Amplification: Lemmas 3.2 and 3.3. With Lemma 3.4, we have an efficient cipher X which achieves every permutation of its message space with nonzero probability. It is clear that, for any cipher Y,

$$\operatorname{supp} Y \subseteq \operatorname{supp} XY = \mathfrak{S}_{\mathscr{M}},$$

with equality only when Y achieves every permutation as well. Based on this amplification of support, we would intuitively expect the product cipher XY to be strictly more secure than Y. In this section, we translate our intuition into precise statements about the cost $\theta_\ell(XY)$ of the optimal attack against XY defined in Section 2.5. We begin by developing some elementary relationships between majorization and guesswork inequalities.

5.1. An Elementary Theory of Guesswork. When dealing with guesswork, it is useful to introduce for $x \in \mathbb{R}_+^n$, the *marginal guesswork*

$$w_\alpha(x) = \min\left\{ k \,\middle|\, \sum_{i=1}^{k} x_{[i]} \geq \alpha \|x\|_1 \right\}.$$

When $\|x\|_1 = 1$, $w_\alpha(x)$ can be interpreted as the worst case number of guesses necessary to be guaranteed a chance of success α for determining a random variable X, whose probability distribution is x (see [19, 21]). Because as a function of α, $w_\alpha(x)$

is piecewise-constant, the area under its curve reduces to the sum

$$\int_0^1 w_\alpha(x)\,d\alpha = \frac{1}{\|x\|_1}\sum_{i=1}^{n} i x_{[i]} = \frac{1}{\|x\|_1} W(x). \tag{5.1}$$

Also we say that two vectors $x, y \in \mathbb{R}_+^n$ are *similarly ordered* if the same permutation matrix Π brings both into the *canonical ordering* $z_1 \geq z_2 \geq \ldots \geq z_n$. Finally, let u be the constant vector with $\|u\|_1 = 1$, i.e. u corresponds to the uniform probability distribution $u = (1/n, \ldots, 1/n)$.

PROPOSITION 5.1. *With n, m finite, assume $j \leq m$, $\gamma, \omega, \omega_j \in \mathbb{R}$, $x, y, x^{(j)}, y^{(j)} \in \mathbb{R}_+^n$, D, D_j, Π are $n \times n$ doubly stochastic matrices with Π a permutation matrix and $S \subset \mathbb{R}_+^n$ is the positive span of a set of similarly ordered vectors. We have the following:*

(i). *W is permutation invariant, i.e. $W(\Pi y) = W(y)$.*
(ii). *W is linear on S, i.e. $W(\gamma x + \omega y) = \gamma W(x) + \omega W(y)$, when $x, y \in S$.*
(iii). *$x \preceq y \implies W(x) \geq W(y)$.*
(iv). *$x \prec y \implies W(x) > W(y)$.*
(v). *$Dy \prec y \iff W(Dy) > W(y)$.*
(vi). *$W(Dy) \geq W(y)$ with equality iff D acts as a permutation on y.*
(vii). *$W(x) \leq \|x\|_1 (n+1)/2$, with equality iff x is constant.*
(viii). (Day, 1972) *If $y^{(j)} \in S$ and $x^{(j)} \preceq y^{(j)}$, then $\sum_j x^{(j)} \preceq \sum_j y^{(j)}$.*
(ix). *$W\left(\sum_j \omega_j D_j y^{(j)}\right) \geq \sum_j \omega_j W(y^{(j)})$.*
(x). *The inequality in* (ix) *becomes strict if for any j, $D_j y^{(j)} \prec y^{(j)}$.*

Proof. (i). By definition, $W(y)$ depends only on the values y_i and not on their order.

(ii). For canonically ordered vectors, guesswork takes the form of the linear function $W(z) = a \cdot z$, so *(ii)* follows from *(i)* and the definition of similar ordering.

(iii). From (2.1) we may assume w.l.o.g. $\|x\|_1 = \|y\|_1 = 1$. Equation (2.2) tells us that for $k = w_\alpha(x)$, $\sum_{i=1}^k y_{[i]} \geq \sum_{i=1}^k x_{[i]} \geq \alpha$. Thus $w_\alpha(x) \geq w_\alpha(y)$, and so

$$\int_0^1 w_\alpha(x)\,d\alpha \geq \int_0^1 w_\alpha(y)\,d\alpha.$$

Now from (5.1) it follows that $W(x) \geq W(y)$.

(iv). Again assume w.l.o.g. $\|x\|_1 = \|y\|_1 = 1$. If $x \preceq y$ but $y \not\preceq x$, then there must be an $m < n$ for which $\sum_{i=1}^m x_{[i]} < \sum_{i=1}^m y_{[i]}$. If we take $\lambda = \sum_{i=1}^m y_{[i]}$, then $w_\lambda(x) > m \geq w_\lambda(y)$. Indeed for some $\varepsilon > 0$ and $\lambda \leq \alpha \leq (\lambda + \varepsilon)$, we have $w_\alpha(x) > w_\alpha(y)$. Thus

$$\int_0^1 w_\alpha(x)\,d\alpha > \int_0^1 w_\alpha(y)\,d\alpha,$$

and so by (5.1), $W(x) > W(y)$.

(v). ($\Rightarrow$:) Follows the Hardy-Littlewood-Pólya theorem and *(iv)*. ($\Leftarrow$:) Assume $Dy \not\prec y$. Then D acts as a permutation on y and by *(i)*, $W(Dy) \not> W(y)$.

(vi). Tautologically equivalent to *(v)*.

(vii). Assume w.l.o.g. $\|x\|_1 = 1$, and let J be the doubly stochastic matrix with $1/n$ in each entry. Then $u = Jx \preceq x$ so by *(vi)*, $W(x) \leq W(u) = (n+1)/2$, and equality holds iff J acts as a permutation on x, which happens iff $x = u$.

(viii). The result is due to Day [6] (see also [15, §5.A.6, p.121]).

(ix). We may permute the $y^{(j)}$ into canonically ordered vectors $z^{(j)} = \Pi_j y^{(j)}$, by appropriate choices of Π_j. Let us define doubly stochastic matrices $\widehat{D_j} = D_j \Pi_j^{-1}$, so that $\sum_j \omega_j D_j y^{(j)} = \sum_j \omega_j \widehat{D_j} z^{(j)}$. Now by the Hardy-Littlewood-Pólya theorem, for each j, $\widehat{D_j} z^{(j)} \preceq z^{(j)}$, and thus $\omega_j \widehat{D_j} z^{(j)} \preceq \omega_j z^{(j)}$ as well. Now applying *(viii)* to the similarly ordered vectors $\omega_j z^{(j)}$ we obtain $\sum_j \omega_j D_j y^{(j)} \preceq \sum_j \omega_j z^{(j)}$. The inequality of *(iii)*, the linearity of *(ii)* and the identity of *(i)* yield

$$\begin{aligned} W\left(\sum_j \omega_j D_j y^{(j)}\right) &\geq W\left(\sum_j \omega_j z^{(j)}\right) \\ &= \sum_j \omega_j W(z^{(j)}) \\ &= \sum_j \omega_j W(y^{(j)}). \end{aligned}$$

(x). The guesswork of the sum may be manipulated as follows.

$$W\left(\sum_j \omega_j D_j y^{(j)}\right) \geq \sum_j \omega_j W(D_j y^{(j)}) \tag{5.2}$$

$$> \sum_j \omega_j W(y^{(j)}), \tag{5.3}$$

where (5.2) follows from *(ix)* using identity matrices, and (5.3) follows from *(iii)* and at least one instance of *(iv)*. □

REMARK 3. *Notice that unlike the simpler case of Proposition 5.1.(v), the sufficient condition $D_j y^{(j)} \prec y^{(j)}$ of 5.1.(x) is* not *necessary. Indeed if x is nonzero and uniform on* half *of its values then it is easy to find Π such that $u = (x + \Pi x)/2$. But then $W(Ix/2 + \Pi x/2) > W(x)$ while no strict majorization is present.*

5.2. Conditions for Strict Majorization. Our goal is to compare a variety of expressions involving guesswork, and we saw in Proposition 5.1.(v) that the strict inequality $W(Dy) > W(y)$ is equivalent to the strict majorization $Dy \prec y$, with $y \in \mathbb{R}^n_+$ and D a doubly stochastic $n \times n$ matrix. Now from the algebraic characterization of majorization given in Section 2.2, it is clear that D must be a non-trivial convex sum of permutation matrices (otherwise D is a permutation and $y \preceq Dy$). However, the converse is not in general true, because any doubly stochastic matrix satisfies $u = Du$ and $u \not\prec u$. Nevertheless, we seek a weakened converse, i.e. when y is *not* constant, what group-theoretic properties of the permutation matrices Π_j are required

so that

$$\left(\sum_j p_j \Pi_j\right) y \prec y?$$

Answering this question leads naturally to a proof of Lemma 3.2.

Throughout this section, we treat permutations abstractly as elements of a subgroup $G \leq \mathfrak{S}_n$ with the usual action $\sigma \cdot (x_1, \ldots, x_n) = (x_{\sigma^{-1}1}, \ldots x_{\sigma^{-1}n})$. Given this action, Birkhoff's theorem tells us that any doubly stochastic matrix can also be represented abstractly as

$$D = \sum_{g \in G} d(g) g, \quad \text{with} \quad \sum_{g \in G} d(g) = 1, \; d(g) \geq 0. \tag{5.4}$$

Though the decomposition of (5.4) is not necessarily unique, the algebraic properties of $\operatorname{supp} d = \{g \in G \mid d(g) \neq 0\}$ play a crucial role in the sequel.

PROPOSITION 5.2. *Let d, D be as in equation (5.4) and let $x \in \mathbb{R}^n_+$ satisfy*

$$x_{[1]} = \ldots = x_{[\ell]} > x_{[\ell+1]} \geq \ldots \geq x_{[n]},$$

with $\pi \cdot x$ canonically ordered. Then the following two statements are equivalent:

(i). *The vector $y = Dx$ satisfies $y_{[i]} = x_{[1]}$, for $1 \leq i \leq \ell$.*

(ii). *For some $\sigma \in G$, $\operatorname{supp} d \subseteq \sigma \operatorname{Stab}_G\{\pi^{-1}U\}$, where $U = \{1, \ldots, \ell\}$.*

Proof. We may w.l.o.g. treat only the case in which x has canonical ordering:

$$x_1 = \ldots = x_\ell > x_{\ell+1} \geq \ldots \geq x_n,$$

and thus $\pi = 1$. The general case amounts to renaming things in a straightforward way (the pedantic details are worked out in [19]).

(i) $\Longrightarrow$ (ii): We must show that for some $\sigma \in G$, $\operatorname{supp} d \subseteq \sigma \operatorname{Stab}_G\{1, \ldots, \ell\}$. Let $\mathscr{X} = \{1, \ldots, n\}$ and $U = \{1, \ldots, \ell\}$. As in Section 2.1, G acts on the set $\mathscr{X}^{\{\ell\}}$ of subsets of $\mathscr{X}$ with ℓ elements. The setwise stabilizer $H = \operatorname{Stab}_G\{U\}$, is merely the point stabilizer of U under this action. Now the j-th component of y is given by

$$y_j = \left(\sum_{g \in G} d(g) g x\right)_j = \sum_{g \in G} d(g)(gx)_j = \sum_{g \in G} d(g) x_{g^{-1}j}.$$

If $g^{-1}j$ were to lie outside of U for any $g \in \operatorname{supp} d$, then $x_{g^{-1}j} < x_1$, and $y_j < x_1$. So the assumption $y_{[i]} = y_{j_i} = x_1$, for all $i \in U$, really means that $g^{-1}j_i \in U$ for all $g \in \operatorname{supp} d$ and all $i \in U$. Let $V = \{j_1, \ldots, j_\ell\}$, so that we have $g^{-1}V = U$ for all $g \in \operatorname{supp} d$. Alternatively,

$$\operatorname{supp} d \subseteq \{g \in G \mid gU = V\} = \sigma H,$$

where $\sigma U = V$.

(ii) $\Longrightarrow$ *(i)*: We must show that $y = Dx$ satisfies $y_{[i]} = x_1$, for $1 \leq i \leq \ell$. By assumption, $\operatorname{supp} d \subseteq \sigma H$, we may write $D = \sigma E$, where

$$E = \sum_{h \in H} d(\sigma h) h.$$

Consider $z = Ex$, whose i-th component is given by

$$z_i = \sum_{h \in H} d(\sigma h) x_{h^{-1} i}.$$

Because H leaves U fixed as a set, $x_{h^{-1} i} = x_1$ for all $h \in H$ and all $i \in U$. Thus $z_i = x_1$ for all $i \in U$. But $y = \sigma E x = \sigma z$. Since y and z differ by a permutation we have $y_{[i]} = z_i = x_1$, for all $i \in U$. □

Any non-constant vector satisfies non-trivially the condition of Proposition 5.2. That simple observation leads to the following:

PROPOSITION 5.3. *Given d, D as in equation (5.4), then $Dx \prec x$ for all non-uniform $x \in \mathbb{R}^n_+$ if and only if*

$$\operatorname{supp} d \not\subseteq \sigma \operatorname{Stab}_G\{V\},$$

for each $\sigma \in G$ and each proper $V \subset \{1, \ldots, n\}$.

Proof.

($\Longrightarrow$): Suppose the support of d is contained in a coset of a setwise stabilizer of a proper subset $V \subset \{1, \ldots, n\}$. Writing $\ell = |V|$ and $U = \{1, \ldots, \ell\}$, there is a permutation $\pi \in \mathfrak{S}_n$, such that $V = \pi^{-1} U$. Furthermore, there is an $x \in \mathbb{R}^n_+$ satisfying

$$x_{[i]} = \begin{cases} x_{[1]} & 1 \leq i \leq \ell, \\ 0 & (\ell + 1) \leq i \leq n, \end{cases}$$

with $\pi \cdot x$ canonically ordered. Writing $y = Dx$, we have by Proposition 5.2, $y_{[i]} = x_{[1]}$ for $i \in U$. But also $\|y\|_1 = \|x\|_1$ so that x and y differ by a permutation and thus $y \not\prec x$.

($\Longleftarrow$): Conversely, suppose there is a non-uniform $x \in \mathbb{R}^n_+$ such that $y = Dx \not\prec x$. For some $\ell < n$, x takes the form

$$x_{[1]} = \ldots = x_{[\ell]} > x_{[\ell+1]} \geq \ldots \geq x_{[n]},$$

and hence for each i, $y_{[i]} = x_{[i]}$. Again by Proposition 5.2, the support of d is contained in a coset of a setwise stabilizer of a subset of $\{1, \ldots, n\}$ of size $\ell < n$. □

5.3. The Proof of Lemma 3.2. Recall from (2.4) that the probability distribution of a product cipher $Z = XY$ is given by a convolution, which may be thought of as multiplication by a doubly stochastic matrix. Preferring, as in the previous section, to work with permutations abstractly, we point out that the convolution $z(g)$

is just a way to represent products in the *group algebra* (the vector space $\mathbb{R}G$ with multiplication extended from the group law [14]):

$$\sum_{g\in G} z(g)g = \sum_{g\in G} x * y(g)g = \left(\sum_{f\in G} x(f)f\right)\left(\sum_{h\in G} y(h)h\right).$$

In this way, left multiplication by cipher X can be represented as left multiplication by the linear operator $D = \sum_g x(g)g$, and the random variables Y and Z can be represented as vectors $y, z \in \mathbb{R}G$, namely (as linear combinations of the basis vectors G) $z = \sum_g z(g)g$ and $y = \sum_g y(g)g$.

By Proposition 5.1.(iii), we immediately have $W(XY) \geq W(Y)$. But our characterization of strict majorization in the previous section allows us to establish strict inequalities, and finally to prove Lemma 3.2.

Proof. [of Lemma 3.2:] Take $Z = XY$ and $z = Dy$ as above. Then by definition, $W(Z) > W(Y)$ for every non-uniform Y iff $W(Dy) > W(y)$ for every non-uniform y. By Proposition 5.1.(v), this can only happen iff $Dy \prec y$ for every non-uniform y. By Proposition 5.3, the desired result follows if we can identify the set of subgroups $H \neq G$ with the set of setwise stabilizers (under action $L(G)$ of left multiplication) of proper subsets of G. In other words, we claim that $\{H \leq G \mid H \neq G\} = \{\mathrm{Stab}_{L(G)}\{U\} \mid U \subset G\}$.

($\subseteq$:) By definition, $\mathrm{Stab}_{L(G)}\{H\}$ is the point stabilizer of H under the action of G on $G^{\{|H|\}}$ by left multiplication. Now because $gH = H$ iff $g \in H$, it follows that $\mathrm{Stab}_{L(G)}\{H\} = H$. ($\supseteq$:) Given any proper set $U \subset G$, $H = \mathrm{Stab}_{L(G)}\{U\}$ is a subgroup, and we must show that it is not identically G. There is an $h \in G - U$, because U is a proper subset. Since the action $L(G)$ is transitive, there is a $k \in G$ such that $kh \in U$. In other words, $k^{-1}U \neq U$ so that $k^{-1} \notin H$ and $H \neq G$. □

5.4. The Proof of Lemma 3.3. In our general technique for quantifying and comparing the cost of the optimal chosen plaintext attack, we start by fixing the data complexity ℓ and the plaintext tuple $p \in \mathscr{M}^{(\ell)}$. We then study how a cipher's statistical and algebraic structure affects the conditional guesswork expression in (2.6). From that point, we proceed to deduce the desired comparison between two ciphers in terms their expressions for θ_ℓ given in (2.7).

A simple but useful observation is that the conditional guesswork $W(Y|c,p)$ is completely determined by the distribution of Y on some coset of the stabilizer $H = \mathrm{Stab}_G(p)$. This follows from Bayes' theorem [12] and the standard observation about group action that $\{g \in G \mid g \cdot p = c\}$ is a coset of H.

To facilitate a security analysis of Y it is again useful to treat its distribution as a vector in $\mathbb{R}G$, $y = \sum_g y(g)g$, with $y(g) = \mathsf{Pr}[Y = g]$. The group algebra $\mathbb{R}G$ is rich in structure, admitting many useful decompositions [7, 11]. For our purposes, we need only to decompose down to the cosets of H. To that end, let $k = [G:H]$ and fix a set $\{g_i\}_{i=1}^k$ of left coset representatives of H in G. There is a well-known decomposition

of $\mathbb{R}G$ which mirrors the group-theoretic structure

$$G = \bigcup_{i=1}^{k} g_i H.$$

A mathematically succinct way to express this decomposition is as a natural isomorphism of $\mathbb{R}G$, as a left $\mathbb{R}G$-module, to the *induced representation from H to G by $\mathbb{R}H$* given by the tensor product of modules[4]

$$\mathbb{R}G \cong \mathbb{R}G \otimes_{\mathbb{R}H} \mathbb{R}H = \bigoplus_{i=1}^{k} g_i \otimes \mathbb{R}H, \tag{5.5}$$

where the isomorphism takes $g_i h \mapsto g_i \otimes h$, and thus takes

$$\sum_{g\in G} y(g)g = \sum_{i=1}^{k}\sum_{h\in H} y(g_i h)g_i h \mapsto \widehat{y} = \sum_{i=1}^{k} g_i \otimes y^{(i)} = \sum_{i=1}^{k} g_i \otimes \left(\sum_{h\in H} y(g_i h)h\right). \tag{5.6}$$

Note that $\mathsf{Pr}[Y = g_i h] = y(g_i h) = y_h^{(i)}$, and so the vectors $\{y^{(i)}\}$ really describe the distribution of Y on the cosets of H, even though each $y^{(i)} \in \mathbb{R}H$. Now, it follows directly from the definitions that

$$W(Y|C^{\ell}, p) = \sum_{i=1}^{k} W(y^{(i)}). \tag{5.7}$$

Recall from the previous section that the distribution of the product cipher $Z = XY$ is expressible as the matrix multiplication $z = Dy$, with $D = \sum_g x(g)g$ and $z = \sum_g z(g)g$. Indeed using the direct sum decomposition of (5.5), we shall derive the block structure of the matrix D. (To be more precise, we again treat permutations abstractly in order to work as much as possible in a basis-free way. A true block structure decomposition for D would follow from a suitable ordering of the basis H of $\mathbb{R}H$. However, due to the explicit form of (5.7), a choice of ordering of H will not be required). Using this structure we shall compare the distribution within the appropriate cosets of H for XY vs. Y. The key is to represent Y and Z by $\widehat{y}, \widehat{z} \in \mathbb{R}G \otimes_{\mathbb{R}H} \mathbb{R}H$ as in (5.6), but to leave X as a convex sum of permutations.

Now any $g \in G$ acts by left multiplication on any $g_j \otimes v \in \mathbb{R}G \otimes_{\mathbb{R}H} \mathbb{R}H$ according to $g(g_j \otimes v) = g_i \otimes hv$, where $gg_jH = g_iH$, so that $h \in H$ is uniquely determined by

[4]Our treatment of induced representations follows Jacobson's [14] and is aimed at succinctness. Briefly, given a ring B, a right B-module U, and a left B-module V, one forms the *tensor product of modules* $T = U \otimes_B V$ in a way which is completely analogous to the case of vector spaces, except that now we have $ub \otimes v = u \otimes bv$, $b \in B$. In general, T is only a $\mathbb{Z}$-module, but when U is a A-B-bimodule for some ring A (i.e. U is a left A-module as well as a right B-module), T becomes a left A-module with left multiplication defined by $a(u \otimes v) = au \otimes v$. In this way, for any $\mathbb{R}H$-module V, $\mathbb{R}G \otimes_{\mathbb{R}H} V$ becomes a $\mathbb{R}G$-module called the *representation induced from H to G by V*.

The reader who is unfamiliar with the module approach to representations is encouraged to begin with a more constructive definition of the induced representation (such as in [11]) and work out the relatively inelegant details for left multiplication by a cipher X.

$gg_j = g_i h$. Thus we have that

$$\begin{aligned}\widehat{z} &= \sum_{i=1}^{k} g_i \otimes z^{(i)} = \left(\sum_{g\in G} x(g)g\right)\left(\sum_{j=1}^{k} g_j \otimes y^{(j)}\right) \\ &= \sum_{j=1}^{k}\left(\sum_{g\in G} x(g)g\right)(g_j \otimes y^{(j)}).\end{aligned}$$

For any particular i, we may collect together contributions to direct summand $g_i \otimes \mathbb{R}H$,

$$\begin{aligned}g_i \otimes z^{(i)} &= \sum_{j=1}^{k} \sum_{g\, g_j H = g_i H} x(g)\, g\,(g_j \otimes y^{(j)}) \\ &= \sum_{j=1}^{k} \sum_{g\in\Gamma_{ij}} x(g)(g_i \otimes h_{ij}(g) y^{(j)}) \\ &= g_i \otimes \left(\sum_{j=1}^{k}\left(\sum_{g\in\Gamma_{ij}} x(g) h_{ij}(g)\right) y^{(j)}\right),\end{aligned}$$

where $\Gamma_{ij} = \{g \in G \mid g\, g_j H = g_i H\}$ and $h_{ij}(g) = g_i^{-1} g g_j$. Thus,

$$z^{(i)} = \sum_{j=1}^{k} \omega_{ij} D_{ij} y^{(j)}, \tag{5.8}$$

where

$$D_{ij} = \sum_{g\in\Gamma_{ij}} \frac{x(g)}{x(\Gamma_{ij})} h_{ij}(g), \tag{5.9}$$

and where the $\omega_{ij} = x(\Gamma_{ij})$ represent transition probabilities between the cosets of H in G, and thus are easily seen to be the coefficients of a doubly stochastic matrix. Notice that the sum in (5.9) is a convex sum of permutations in H, hence each D_{ij} takes the form of a doubly stochastic matrix (under a suitable ordering of H). Also note that for each i, the diagonal sets Γ_{ii} are subgroups conjugate to H, i.e. $\Gamma_{ii} = H^{g_i} \leq G$. The core of Lemma 3.3 is the following proposition, which reduces the case of arbitrary data complexity to the case of zero data complexity, namely Lemma 3.2.

PROPOSITION 5.4. *For a permutation group $G \leq \mathfrak{S}_{\mathscr{M}}$, let X and Y be independent G-ciphers such that* $\operatorname{supp} X = G$. *For any $p \in \mathscr{M}^{(\ell)}$ such that Y is non-uniform on at least one left coset of* $\operatorname{Stab}_G(p)$, *we have $W(XY|C^\ell, p) > W(Y|C^\ell, p)$.*

Proof. Let p satisfy the assumption of the proposition, and let $\widehat{z}$ represent the distribution of the product $Z = XY$ as above. Let $y^{(j)}$ be non-uniform and consider $D_{jj} y^{(j)}$. That is to say, let us focus on this one submatrix block on the diagonal of the larger doubly stochastic matrix representing the convolution $z = x * y$.

Since $\Gamma_{jj} = H^{g_j}$, we may rewrite D_{jj} as

$$D_{jj} = \sum_{g\in\Gamma_{jj}} \frac{x(g)}{x(\Gamma_{jj})} g^{g_j^{-1}} = \sum_{h\in H} \frac{x(h^{g_j})}{x(H^{g_j})} h.$$

Thus by scaling appropriately, $D_{jj}y^{(j)}$ has the form of a product of two independent H-ciphers $\widetilde{X}\widetilde{Y}$, with $\mathsf{Pr}\left[\widetilde{X}=h\right] = \mathsf{Pr}\left[X=h^{g_j}\right]/x(H^{g_j})$, and $\mathsf{Pr}\left[\widetilde{Y}=h\right] = \mathsf{Pr}\left[Y=g_jh\right]/y(g_jH)$. Since $\operatorname{supp} X = G$ and conjugation by g_j yields an isomorphism of $H \longleftrightarrow \Gamma_{jj}$, $\operatorname{supp}\widetilde{X}$ is not confined to any proper coset of H, and we may invoke Lemma 3.2 to obtain $W(\widetilde{X}\widetilde{Y}) > W(\widetilde{Y})$, and thus $W(D_{jj}y^{(j)}) > W(y^{(j)})$. From Proposition 5.1.(v), we have $D_{jj}y^{(j)} \prec y^{(j)}$.

Using Proposition 5.1.(ix), we may bound any $W(z^{(i)})$

$$W(z^{(i)}) \;=\; W\left(\sum_{m=1}^{k}\omega_{im}D_{im}y^{(m)}\right) \;\geq\; \sum_{m=1}^{k}\omega_{im}W(y^{(m)}),$$

and using Proposition 5.1.(x), we may *strictly* bound $W(z^{(j)})$

$$W(z^{(j)}) \;=\; W\left(\sum_{m=1}^{k}\omega_{jm}D_{jm}y^{(m)}\right) \;>\; \sum_{m=1}^{k}\omega_{jm}W(y^{(m)}).$$

Combining these bounds on $W(z^{(i)})$ we obtain a strict bound on $W(Z|C^\ell,p)$ as follows

$$\begin{aligned} W(Z|C^\ell,p) \;&=\; \sum_{i=1}^{k}W(z^{(i)}) > \sum_{i=1}^{k}\sum_{m=1}^{k}\omega_{im}W(y^{(m)}) \\ &= \sum_{m=1}^{k}W(y^{(m)})\sum_{i=1}^{k}\omega_{im} \\ &= \sum_{m=1}^{k}W(y^{(m)}) \;=\; W(Y|C^\ell,p), \end{aligned}$$

which was to be proved. □

REMARK 4. *Evidently in the previous proposition, we could have weakened the condition* $\operatorname{supp} X = G$ *to: For every* $p \in \mathscr{M}^{(\ell)}$, $\operatorname{supp} X \cap \operatorname{Stab}_G(p)$ *is not confined to a proper coset of* $\operatorname{Stab}_G(p)$. *However, for our purposes, this was not necessary.*

The next proposition provides an important interpretation of the situation when a cipher is uniform on every coset of an ℓ-message stabilizer.

PROPOSITION 5.5. *Let* Y *be a* G*-cipher, for a permutation group* $G \leq \mathfrak{S}_{\mathscr{M}}$. *For any* $p \in \mathscr{M}^{(\ell)}$, *write* $H = \operatorname{Stab}_G(p)$ *and we have*

$$W(Y|C^\ell,p) \leq \frac{1+|H|}{2},$$

with equality holding iff Y *is uniform on each coset of* H.

Proof. For $c \in \mathscr{M}^{(\ell)}$ with $\omega(c|p) \neq 0$,

$$1 \leq W(Y|c,p) \leq \frac{1+|H|}{2},$$

because $W(Y|c,p)$ is the guesswork on a coset of size $|H|$. Furthermore by Proposition 5.1.(vii), equality in the upper bound is achieved iff Y has constant probability on

that particular coset. Now since $\sum_{c \in \mathscr{M}^{(\ell)}} \omega(c|p) = 1$, the sum of (2.6) is convex and therefore achieves its maximum of $\frac{1}{2}(1 + |H|)$ iff Y is constant on each coset of H (Y will of course have the constant probability 0 on those cosets corresponding to $\omega(c|p) = 0$). □

By tying together the previous two propositions, we may finally prove Lemma 3.3.

Proof. [of Lemma 3.3.] Again we write $Z = XY$, and let us denote the size of the smallest ℓ-message stabilizer of G by

$$M_G(\ell) \stackrel{\triangle}{=} \min_{p \in \mathscr{M}^{(\ell)}} |\mathrm{Stab}_G(p)|.$$

From Proposition 5.1.(vii), it is easy to see that $\theta_\ell(U_G) = \frac{1}{2}[1 + M_G(\ell)]$.

Now suppose there is a $p \in \mathscr{M}^{(\ell)}$ such that $\theta_\ell(Z) = W(Z|C^\ell, p)$ and Y is non-uniform on at least one coset of $\mathrm{Stab}_G(p)$. Then we may invoke Proposition 5.4 to obtain

$$\theta_\ell(Z) = W(Z|C^\ell, p) > W(Y|C^\ell, p) \geq \theta_\ell(Y).$$

On the other hand, suppose that for every $p \in \mathscr{M}^{(\ell)}$ satisfying $\theta_\ell(Z) = W(Z|C^\ell, p), Y$ is uniform on each coset of $\mathrm{Stab}_G(p)$. Let $H = \mathrm{Stab}_G(p)$ for any such p. By (5.8), Z is uniform on each coset of H as well, and by Proposition 5.5,

$$\theta_\ell(Z) = W(Z|C^\ell, p) = \frac{1 + |H|}{2}$$

Now choose any $\widehat{p}$ with $|\mathrm{Stab}_G(\widehat{p})| = M_G(\ell)$ and hence

$$\theta_\ell(Z) = \frac{1 + |H|}{2} \leq W(Z|C^\ell, \widehat{p}) \leq \frac{1 + M_G(\ell)}{2},$$

forcing $|H| = M_G(\ell)$, and thus $\theta_\ell(Z) = \theta_\ell(U_G)$. Then, either $\theta_\ell(Y) \neq \theta_\ell(U_G)$, in which case $\theta_\ell(Z) > \theta_\ell(Y)$, or $\theta_\ell(Y) = \theta_\ell(U_G)$.

To summarize what we have proved thus far, $\theta_\ell(Z) \geq \theta_\ell(Y)$ and if equality holds then $\theta_\ell(Y) = \theta_\ell(U_G)$. However conversely, if $\theta_\ell(Y) = \theta_\ell(U_G)$, then $\theta_\ell(U_G) \geq \theta_\ell(Z) \geq \theta_\ell(Y) = \theta_\ell(U_G)$, forcing equality $\theta_\ell(Z) = \theta_\ell(Y)$, which completes the proof. □

6. Conclusion. The issue of security amplification by product composition remains a complex one. In this paper, we have added to the number of situations where a definite answer can be given. Specifically, Theorem 3.1 asserts that there exists an efficient cipher X such that the security of XY is strictly greater than Y unless Y is perfect. There is room for further improvement in this result. For example, a more efficient cipher might be constructed which makes use of a weakened form of Lemma 3.3 as discussed in Remark 4. Additionally, our implementation might be optimized for bulk encryption.

The cipher we construct to prove Theorem 3.1 is costly in some ways but has other desirable properties. Unlike a one-time pad, if the key were replaced by a pseudo-random source, a known plaintext-ciphertext block would *not* trivially betray the key used for that block. This property could be useful in constructing provably secure practical encryption systems. Also observe that our construction is *not* an iterated cryptosystem but rather a product of independent rounds with a *carefully chosen order*. The techniques employed here might be a useful new paradigm for practical cryptosystems with key schedules instead of a truly random source of key material.

Finally we note that due to Lemma 3.3 and the nature of our existence question we have been content to focus on strict inequalities and strict amplification of support alone. While an infinitesimally small increase from $\theta_\ell(Y)$ to $\theta_\ell(XY)$ is possible, techniques beyond the scope of this paper have been developed to establish much stronger claims of amplification. Ongoing research suggests that the cipher X of Lemma 3.4 has stronger security properties than required by Theorem 3.1.

Acknowledgments. I would like to thank Serge Vaudenay for his many insightful comments, and in particular for suggesting the formal computational model of Section 2.

REFERENCES

[1] W. Aiello, M. Bellare, G. Di Crescenzo, and R. Venkatesan, *Security amplification by composition: The case of doubly-iterated, ideal ciphers.* In H. Krawczyk, editor, Advances in Cryptology - CRYPTO '98, Berlin, 1998. Springer-Verlag.

[2] David J. Aldous and Persi Diaconis, *Shuffling cards and stopping times.* Amer. Math. Monthly, 93(1986), pp. 333–348.

[3] G. Birkhoff, *Tres observaciones sobre el algebra lineal.* University Nac. Tucuman Rev. Ser. A, 5(1946), pp. 147–150.

[4] Christian Cachin, *Entropy Measures and Unconditional Security in Cryptography.* PhD thesis, ETH Zürich, 1997.

[5] Keith W. Campbell and Michael J. Wiener, *DES is not a group.* In Ernest F. Brickell, editor, Advances in Cryptology - CRYPTO'92, pp. 512–517, Berlin, 1992. Springer-Verlag.

[6] P. W. Day, *Rearrangement inequalities.* Canad. J. Math., 24(1972), pp. 930–943.

[7] Persi Diaconis, *Group Representations in Probability and Statistics.* Institute of Mathematical Statistics, Hayward, CA, 1988.

[8] John D. Dixon and Brian Mortimer, *Permutation Groups.* Springer-Verlag, New York, 1996.

[9] S. Even and O. Goldreich, *On the power of cascade ciphers.* ACM Transactions on Computer Systems, 3:2, 1985.

[10] Michael R. Garey and David S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, New York, 2nd edition, 1979.

[11] Roe Goodman and Nolan R. Wallach, *Representations and Invariants of the Classical Groups*, volume 68 of Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge, 1998.

[12] G. R. Grimmett and D. R. Stirzaker, *Probability and Random Processes.* Oxford University Press, Oxford, 2nd edition, 1992.

[13] Godfrey H. Hardy, John E. Littlewood, and George Pólya, *Some simple inequalities satisfied by convex functions.* Messenger Math., 58(1929), pp. 145–152.

[14] Nathan Jacobson, *Basic Algebra II.* W. H. Freeman and Company, New York, 2nd edition, 1980.

[15] Albert W. Marshall and Ingram Olkin, *Inequalities: Theory of Majorization and Its Applications*. Academic Press, San Diego, 1979.

[16] Ueli M. Maurer and James L. Massey, *Cascade ciphers: The importance of being first*. Journal of Cryptology, 6(1993), pp. 55–61.

[17] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, Boca Raton, 1997.

[18] Moni Naor and Omer Reingold, *On the construction of pseudorandom permutations: Luby-Rackoff revisited*. Journal of Cryptology, 12(1999), pp. 29–66.

[19] John O. Pliam, *Ciphers and their Products: Group Theory in Private Key Cryptography*. PhD thesis, University of Minnesota, July 1999. URL: `http://www.ima.umn.edu/~pliam/doc`.

[20] John O. Pliam, *Guesswork and variation distance as measures of cipher security*. In Selected Areas in Cryptography - SAC'99, LNCS1758, pp. 62–77, Berlin, 2000. Springer-Verlag.

[21] John O. Pliam, *On the incomparability of entropy and marginal guesswork in brute-force attacks*. In Proceedings of Indocrypt 2000, LNCS1977, pp. 67–79, Berlin, 2000. Springer-Verlag.

[22] Joseph J. Rotman, *An Introduction to the Theory of Groups*. Wm. C. Brown, Dubuque, IA, 3rd edition, 1988.

[23] Claude E. Shannon, *Communication theory of secrecy systems*. Bell System Tech. Jour., 28(1949), pp. 656–715.