BitFlip: A Randomness-Rich Cipher

Gideon Samid[*] Serguei Popov[†]

April 26, 2017

Abstract

We present a cipher that represents a novel strategy: replacing algorithmic complexity with computational simplicity while generating cryptographic efficacy through large as desired quantities of randomness. The BitFlip cipher allows its user to defend herself with credibly appraised mathematical intractability, well-hinged on solid combinatorics. This is the situation when the amount of randomness is small relative to the accumulated amount of processed plaintext. Deploying more randomness, BitFlip will frustrate its cryptanalyst with terminal equivocation among two or more plausible message candidates. This equivocation defense can be increased by simply increasing the amount of deployed randomness, coming at-will close to Vernam's perfect secrecy. BitFlip is structured as a super polyalphabetic cipher where a letter comprised of $2n$ bits is pointed-to by any $2n$ bits string with a Hamming distance of n from it. When a passed $2n$ bits string is found to have no n-valued Hamming distance from any letter in the reader's alphabet, it is regarded as null. This allows for co-encryption of several messages each over its respective alphabet; thereby offering a powerful equivocation defense because the ciphertext does not indicate which alphabet the intended reader is using. BitFlip becomes increasingly timely and practical, exploiting the advent of high quality non-algorithmic randomness, as well as the effect of Moore's law on

[*] Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland OH, U.S.; BitMint, LLC. E-mail: gideon@bitmint.com

[†] Department of Statistics, Institute of Mathematics, Statistics and Scientific Computation, University of Campinas – UNICAMP, rua Sérgio Buarque de Holanda 651, 13083–859, Campinas SP, Brazil. E-mail: popov@ime.unicamp.br

the cost of handling large amounts of memory. BitFlip is a natural fit for what fast emerges as the biggest customer of cryptography: the Internet of Things.

1 Introduction

Vernam’s famous ”One Time Pad” cipher is one hundred years old this year: elegant, simple, and unbreakable in as much as possession of the ciphertext confers no entropic advantage over a cryptanalyst aware of its existence and size, but not of its contents. Despite this perfect secrecy, Vernam’s cipher per se never caught on because it required very large amounts of top-quality randomness to insure its theoretical capability. Hundred years ago there was no convenient way to generate the required amounts of randomness, nor to store, and much less to communicate the same. Mostly, then, cryptography ventured into a fundamentally different strategy: achieving the desired secrecy with small manageable keys which are thoroughly mixed with the plaintext, using ever more ingenious complexity-generating algorithms. Some tried to approach Vernam’s security by using pseudo-randomness, which in turn was generated from very small keys through planned algorithmic complexity.

In the intervening one hundred years technology progressed and now offers (i) convenient generation of large amounts of high-quality randomness, and (ii) increasingly affordable means to store, and communicate ever larger quantities of random bits. It is time to revisit the hundred years old debate: is it better to aspire for secrecy through greater and greater algorithmic complexity over limited size randomness (keys), or perhaps it is better to secure data through randomness-rich ciphers, operating with simple “Vernam-like” protocols. Regardless of how this debate will fare in the coming years, the new randomness-generating and randomness handling technology breath new life in the old Vernam idea.

In particular one identifies a “cryptographic desert” between the message-long Vernam keys, and the message-size-independent short keys prevailing today. We propose to consider the notion of Vernam-inspired ciphers (Trans-Vernam ciphers) where randomness and plaintexts are mixed in such a way that the computational effort of generating the ciphertext will be at most polynomial, and at best ’flat’, allowing their user to secure their data with as much randomness as they care to ’throw in’. This will extend to users the power to gauge the provided security to the sensitivity and operational criticality of the secured data. Being novel, this strategy is likely to offer additional benefits, not yet envisioned. As we discuss ahead, randomness-rich ciphers offer a large unicity distance, which implies fundamental equivocation – the concept behind Vernam's perfection.

The here presented BitFlip cipher is a Trans-Vernam super-polyalphabetic cipher that illustrates this new strategy for effective cryptographic secrecy.[1]

2 Notations

First, we introduce some notations. Fix $s_0 \in \mathbb{N}$ and denote $\mathfrak{A} = \{1, \dots, s_0\}$, $\mathfrak{A}^* = \{0, 1, \dots, s_0\}$. In the following, $\mathfrak{A}$ will stand for the alphabet we use for writing texts to be transmitted, and $\mathfrak{A}^*$ is the "extended alphabet" which contains also a "meaningless" letter 0. We use the word "message" for any string of elements of $\mathfrak{A}$, and "plaintext" for any string of elements of $\mathfrak{A}^*$. Note that from any plaintext one obtains a message in a unique way, simply by removing all zeros.

For $k \in \mathbb{N}$ let $\mathcal{C}_k$ be the unit (hyper)cube, $\mathcal{C}_k = \{0, 1\}^k$; the elements of $\mathcal{C}_k$ are thus binary words of length k. We use notations 0_k and 1_k for the all-zero and all-one binary words. For $\eta, \zeta \in \mathcal{C}_k$, we define the operation of bitwise addition modulo 2, i.e.,

$$\eta \oplus \zeta = (\eta^{(1)} + \zeta^{(1)} \mod 2, \dots, \eta^{(k)} + \zeta^{(k)} \mod 2), \tag{1}$$

being $\eta = (\eta^{(1)}, \dots, \eta^{(k)})$, $\zeta = (\zeta^{(1)}, \dots, \zeta^{(k)})$. Note that

$$\eta \oplus \eta = 0_k \qquad \text{for all } \eta \in \mathcal{C}_k. \tag{2}$$

Define $\bar{\eta} = \eta \oplus 1_k$ to be the word with all bits flipped. Then, define

$$\|\eta\| = \sum_{j=1}^{k} \eta^{(j)}$$

to be the number of 1's in η, and

$$\mathcal{H}(\eta, \zeta) = \|\eta \oplus \zeta\|$$

to be the so-called *Hamming distance* between η and ζ, i.e., the number of positions where their corresponding bits are different.

[1] For other Trans-Vernam ciphers see references [7, 8, 9, 10].

We can equip $\mathcal{C}_k$ with a (non-oriented) graph structure by declaring η and ζ neighbours whenever $\mathcal{H}(\eta,\zeta)=1$ (i.e., they differ in only one bit). It is clear that $\mathcal{C}_k$ is a bipartite graph, and we denote the two classes by

$$\begin{aligned}\mathcal{C}_k^0 &= \{\eta : \|\eta\| \text{ is even}\},\\ \mathcal{C}_k^1 &= \{\eta : \|\eta\| \text{ is odd}\}.\end{aligned}$$

We state the following simple fact without proof:

Proposition 2.1. *Let $j\in\{0,1\}$. Then,*

(i) for any $\eta,\zeta\in\mathcal{C}_k^j$ we have that $\mathcal{H}(\eta,\zeta)$ is even, and

(ii) for any $\eta\in\mathcal{C}_k^j$,$\zeta\in\mathcal{C}_k^{1-j}$ we have that $\mathcal{H}(\eta,\zeta)$ is odd.

The above means that changing an even number of bits keeps the word in the same class, while changing an odd number of bits changes the class.

For $\eta\in\mathcal{C}_{2k}$, we define the set

$$\mathsf{FR}(\eta)=\{\zeta\in\mathcal{C}_{2k}:\mathcal{H}(\eta,\zeta)=k\}, \tag{3}$$

i.e., the set of binary words that differ from η in exactly half of the positions. We call $\mathsf{FR}(\eta)$ the *flip range* of η. Let us stress that we use this definition only for the binary words of *even* length. We summarize the basic properties of $\mathsf{FR}(\cdot)$ in the following

Proposition 2.2. *It holds that*

(i) $\mathsf{FR}(\eta)=\mathsf{FR}(\bar\eta)$ for all $\eta\in\mathcal{C}_{2k}$;

(ii) $\zeta\in\mathsf{FR}(\eta)$ if and only if $\eta\in\mathsf{FR}(\zeta)$;

(iii) if k is even, then $\eta\in\mathcal{C}_{2k}^j$ implies $\zeta\in\mathcal{C}_{2k}^j$ for all $\zeta\in\mathsf{FR}(\eta)$; if k is odd, then $\eta\in\mathcal{C}_{2k}^j$ implies $\zeta\in\mathcal{C}_{2k}^{1-j}$ for all $\zeta\in\mathsf{FR}(\eta)$;

(iv) (with card(A) *denoting the cardinality of the set A)*

$$\operatorname{card}\big(\mathsf{FR}(\eta)\big)=\binom{2k}{k}=\frac{(2k)!}{(k!)^2}\sim\frac{2^{2k}}{\sqrt{\pi k}}. \tag{4}$$

(v) $\mathsf{FR}(\eta)\cap\mathsf{FR}(\zeta)\neq\emptyset$ if and only if $\mathcal{H}(\eta,\zeta)$ is even.

We note, in particular, that the approximation in (4) works rather good; for example, for $k = 10$ the relative error is only a bit larger than 1%. Due to Proposition 2.1, (v) means that the flip ranges of η and ζ have nonempty intersection only in the case when both η and ζ belong to the same class, $\mathcal{C}_{2k}^0$ or $\mathcal{C}_{2k}^1$. Notice also that (iii) implies that the whole set $\mathsf{FR}(\eta)$ is contained either in $\mathcal{C}_{2k}^0$ or in $\mathcal{C}_{2k}^1$.

Proof of Proposition 2.2. The proof of (i)–(iii) is quite straightforward, and one readily obtains the last approximation in (4) using (12). Note that one can write an even better approximation using (13) or (14). As for the part (v), first, one obtains from (iii) that $\mathsf{FR}(\eta) \cap \mathsf{FR}(\zeta) = \emptyset$ in case $\mathcal{H}(\eta, \zeta)$ is odd. Assume now that $\mathcal{H}(\eta, \zeta)$ is even, and denote by $A = \{j : \eta^{(j)} \neq \zeta^{(j)}\}$ the set where the two words disagree. By assumption, $\operatorname{card}(A) = \mathcal{H}(\eta, \zeta)$ is even, and therefore A can be divided into two disjoint sets A_1 and A_2 with equal cardinality; also, the set $B = \{1, \dots, 2k\} \setminus A$ can be divided into two disjoint sets B_1 and B_2 with equal cardinality. To construct a binary word which belongs to both $\mathsf{FR}(\eta)$ and $\mathsf{FR}(\zeta)$, just flip the bits of η on A_1 and B_1; it is straightforward to see that the same word is also the result of flipping the bits of ζ on A_2 and B_1 (note that $\operatorname{card}(A_j \cup B_1) = k$ for $j = 1, 2$). $\square$

Next, for $k \in \mathbb{N}$ let us denote by $\mathbb{Z}_k^d = \mathbb{Z}^d / k\mathbb{Z}^d$ the d-dimensional discrete torus of size k (and of volume k^d). It is a transitive graph with the neighbourhood relation inherited from $\mathbb{Z}^d$. It holds, by the way, that $\mathcal{C}_m = \mathbb{Z}_2^m$.

For $\eta = (\eta^{(1)}, \dots, \eta^{(k)}) \in \mathcal{C}_k$ and $\zeta = (\zeta^{(1)}, \dots, \zeta^{(m)}) \in \mathcal{C}_m$ we introduce the binary word $\eta \curlywedge \zeta \in \mathcal{C}_{k+m}$ by

$$(\eta \curlywedge \zeta)^{(\ell)} = \begin{cases} \eta^{(\ell)}, & \text{for } 0 \leq \ell \leq k, \\ \zeta^{(\ell-k)}, & \text{for } k+1 \leq \ell \leq k+m; \end{cases}$$

that is, $\eta \curlywedge \zeta = (\eta^{(1)}, \dots, \eta^{(k)}, \zeta^{(1)}, \dots, \zeta^{(m)})$ is the *concatenation* of the two binary words. Clearly, it holds that $\|\eta \curlywedge \zeta\| = \|\eta\| + \|\zeta\|$.

3 Description of the protocol

Now, we are ready to describe the transmission protocol. There is one sender, Alice, and m_0 recipients, $\mathrm{Bob}_1, \dots, \mathrm{Bob}_{m_0}$. The Alice's goal is to transmit m_0 messages $\mu_1 \in \mathfrak{A}^{\ell_1}, \dots, \mu_{m_0} \in \mathfrak{A}^{\ell_{m_0}}$ as one ciphertext via a common channel in such a way that, for $j = 1, \dots, m_0$, after decryption Bob_j gets a plaintext $\wp_j$

that reduces to the intended message μ_j after throwing zeros away. For this, she first constructs a plaintext $\hat{\wp} \in \mathcal{C}^{k_0}$, where $k_0 \geq \ell_1 + \cdots + \ell_{m_0}$, composed of her messages and (possibly) zeros in the following way: let $f(j,i)$ be the position of the ith letter from the jth message in $\hat{\wp}$, so that $\hat{\wp}_{f(j,i)} = \mu_j^{(i)}$. Then, we require that the function

$$f : \bigcup_{j=1}^{m_0} \{j\} \times \{1, \dots, \ell_j\} \longrightarrow \{1, \dots, k_0\},$$

is an injection such that $f(j,i_1) < f(j,i_2)$ for all $1 \leq i_1 < i_2 \leq \ell_j$ and all $j = 1, \dots, m_0$. We also require that $\hat{\wp}^{(i)} = 0$ for all $i = 1, \dots, k_0$ such that $f^{-1}(i) = \emptyset$. In words, she "mixes" the messages in such a way that each individual message is written in order, and then, possibly, also adds zeros arbitrarily.

The next step for Alice is to produce a ciphertext with the desired properties. Let $n_0, N, v_0, w_0 \geq 2$ be integer parameters. For each $j = 1, \dots, m_0$, Alice shares with Bob$_j$ the following information:

(i) binary words $\eta_{1,j} \dots, \eta_{s_0,j} \in \mathcal{C}_{2n_0}$;

(ii) a function $g_j : \mathbb{Z}_{w_0}^{d_0} \to \mathcal{C}_{v_0}$, where $d_0 = 2^{v_0 - 1}$. Notice that g_j can be extended (periodically) to the whole $\mathbb{Z}^{d_0}$ in a natural way;

(iii) a site $x_j \in \mathbb{Z}_{w_0}^{d_0} \setminus \{0\}$.

Informally, the binary words $\eta_{1,j} \dots, \eta_{s_0,j}$ correspond to the s_0 letters of the alphabet $\mathfrak{A}$, and the role of the "obfuscation matrix" g_i will become clear later. The total size of the key that Alice needs to share with each of Bobs is $2s_0 n_0 + 2^{v_0} w_0^{d_0} + d_0 \lceil \log_2 w_0 \rceil$ bits (the first term accounts to the letter's encoding, the second term is for the obfuscation matrix g_j, and the third one is for the "shift vector" x_j). Also, we suggest that in practice $v_0 = d_0 = 2$ (when g_j is really a square matrix) may be already a good choice.

Next, let $e_1, \dots, e_{d_0}$ be the canonical coordinate vectors of $\mathbb{Z}^{d_0}$, and fix a bijection[2] $h : \mathcal{C}_{v_0} \to \{\pm e_1, \dots, \pm e_{d_0}\}$ (recall that $2d_0 = 2^{v_0}$, so such a bijection exists). Assume for simplicity that v_0 divides $2n_0$. We describe the ciphertext's construction in an inductive way. It is a concatenation of Nk_0

[2] in fact, this bijection may be also a part of the shared key; however, to keep the things simple, we suppose for now that it is chosen in some convenient way and known to everybody

binary words of length $2n_0$, that is, N binary words correspond to one letter. Assume that we have already constructed $N(j-1)$ binary words of the ciphertext, $\zeta_1, \dots, \zeta_{N(j-1)}$, which encode $\hat{\wp}^{(1)}, \dots, \hat{\wp}^{(j-1)}$. Consider now the jth letter $\hat{\wp}^{(j)}$ of Alice's plaintext. There can be two cases: it can be 0 (when $f^{-1}(j) = \emptyset$), or it can be, say, the ith letter of mth message, that is, $\mu_m^{(i)} \in \mathfrak{A}$ (in other words, $f^{-1}(j) = (m, i)$). Let us first deal with the latter case.

Recall that we assumed that v_0 divides $2n_0$; abbreviate $\alpha = 2n_0/v_0$. Abbreviate also $a = \mu_m^{(i)}$, the letter to be encoded in such a way that only Bob$_m$ can read it. Let $y_{j-1,k} \in \mathbb{Z}_{w_0}^{d_0}$ be the "current random walk's position" for Bob$_k$, $k = 1, \dots, m_0$. We set $y_{0,k} = 0$ for all k.

Alice then wants to construct $\zeta_{(j-1)N+1}, \dots, \zeta_{jN}$, the N binary words encoding the letter a for Bob$_m$ (and meaningless for the others). For this, she takes N words $\theta_1, \dots, \theta_N \in \mathsf{FR}(\eta_{a,m})$, in such a way that

$$\Big\{ j : \eta_{j,m} \in \bigcap_{n=1}^{N} \mathsf{FR}(\theta_n) \Big\} = \{a\}; \tag{5}$$

that is, $\eta_{a,m}$ is the only one among $\eta_{1,m}, \dots, \eta_{s_0,m}$ whose flip range contains θ_n for all $n = 1, \dots, N$ (recall Proposition 2.2 (ii)).

Next, we divide the binary word $\theta_1 \curlywedge \dots \curlywedge \theta_N$ into αN pieces of length v_0, that is, we write

$$\theta_1 \curlywedge \dots \curlywedge \theta_N = \delta_1 \curlywedge \dots \curlywedge \delta_{\alpha N},$$

where $\delta_t \in \mathcal{C}_{v_0}$ for all $t = 1, \dots, \alpha N$.

Abbreviate $z_k^0 = y_{j-1,k}$. For $t = 0, \dots, \alpha N - 1$ set

$$\hat{\delta}_{t+1} = g_m(z_m^t) \oplus \delta_{t+1}, \tag{6}$$

and

$$z_k^{t+1} = z_k^t + h\big(\hat{\delta}_{t+1} \oplus g_k(z_k^t + x_k)\big). \tag{7}$$

That is, she transforms $\delta_1, \dots, \delta_{\alpha N}$ to $\hat{\delta}_1, \dots, \hat{\delta}_{\alpha N}$ using the words (of equal length) contained in the sites of $\mathbb{Z}_{w_0}^{d_0}$, and the sites used for that lie on a random walk's trajectory. Then, Alice sets

$$\zeta_{(j-1)N+1} = \hat{\delta}_1 \curlywedge \dots \curlywedge \hat{\delta}_\alpha, \quad \dots, \quad \zeta_{jN} = \hat{\delta}_{\alpha(N-1)+1} \curlywedge \dots \curlywedge \hat{\delta}_{\alpha N},$$

and $y_{j,k} = z_k^{\alpha N}$, $k = 1, \dots, m_0$.

For $k = 1, \dots, m_0$, Bob_k then does the following: first, since $(\hat{\delta}_t, t = 1, \dots, \alpha N)$ are known to everybody, he is able to calculate $(z_k^t, t = 1, \dots, \alpha N)$ using (7). Then, he calculates

$$\sigma_k^t = \hat{\delta}_t \oplus g_k(z_k^t) \quad \text{for } k = 1, \dots, \alpha N,$$

and sets

$$\begin{aligned} \chi_k^1 &= \sigma_k^1 \curlywedge \dots \curlywedge \sigma_k^\alpha, \\ &\dots \\ \chi_k^N &= \sigma_k^{\alpha(N-1)+1} \curlywedge \dots \curlywedge \sigma_k^{\alpha N}. \end{aligned}$$

He then verifies if there is $a \in \mathfrak{A}$ such that (5) holds with $(\chi_k^1, \dots, \chi_k^N)$ on the place of $(\theta_1, \dots, \theta_N)$. Now, observe that, by (2), Bob_m receives what Alice intended to transmit (that is, $\sigma_m^t = \delta_t$, so $\theta_1 = \chi_m^1, \dots, \theta_N = \chi_m^N$), that is, $\theta_1 \curlywedge \dots \curlywedge \theta_N$. So, he is able to identify that he received the letter a. On the other hand, we require that, for all $k \neq m$, (5) does not hold, so all other Bobs receive zeros.

Finally, in the case when $\hat{\wp}^{(j)} = 0$, Alice chooses $(\hat{\delta}_t, t = 1, \dots, 2\alpha)$ directly (and, as before, sets $\zeta_{(j-1)N+1} = \hat{\delta}_1 \curlywedge \dots \curlywedge \hat{\delta}_\alpha, \dots, \zeta_{jN} = \hat{\delta}_{\alpha(N-1)+1} \curlywedge \dots \curlywedge \hat{\delta}_{\alpha N}$) in such a way that all Bobs get zeros after doing the above procedure (note that everybody can still apply (7)).

3.1 Generalizations and modifications

First, let us explain why we need the obfuscation matrix. For this, let us consider the protocol without it, i.e., we simply encode each letter by N (random) words from the corresponding flip range. Next, assume that the attacker is allowed to feed a text of his choice into the cipher; or he discovers somehow the exact way a sufficiently large known text is encoded. This may mean that the attacker could identify at least $2n_0 - 1$ different words belonging to $\mathsf{FR}(\eta_{j,m})$ for some j and m. Although the set $\mathsf{FR}(\eta_{j,m})$ has huge cardinality (recall (4)), nevertheless knowing relatively few its elements may already be sufficient to identify $\eta_{j,m}$ and thus compromise the cipher.

Indeed, assume that the attacker knows that $z_1, \dots, z_{2n_0-1} \in \mathcal{C}_{2n_0}$ all belong to $\mathsf{FR}(x)$ for some (unknown to the attacker) $x \in \mathcal{C}_{2n_0}$. Let $\hat{x}$, $\hat{z}_1, \dots, \hat{z}_{2n_0-1}$ be the corresponding words with all 0's substituted by (-1)'s, regarded as vectors in $\mathbb{R}^{2n_0}$. It is immediate to observe (being $a \cdot b$ the usual scalar product of $a, b \in \mathbb{R}^{2n_0}$) that

$$\hat{z}_j \cdot \hat{x} = 0, \text{ for all } j = 1, \dots, 2n_0 - 1, \tag{8}$$

that is, $\hat{x}$ is orthogonal to all the vectors $\hat{z}_1, \dots, \hat{z}_{2n_0-1}$. Note that it is computationally easy to solve a system of linear equations (one can do it in $O(n_0^3)$ steps); so, if the vectors $\hat{z}_1, \dots, \hat{z}_{2n_0-1}$ are linearly independent, we can obtain $\hat{x}$ up to sign, and therefore we can find x or $\bar{x}$ (recall Proposition 2.2 (i)). Of course, in principle, $\hat{z}_1, \dots, \hat{z}_{2n_0-1}$ are not necessarily linearly independent, but we have assumed that the attacker can obtain many z-words, so he is likely to be able to find a sufficient number of linearly independent $\hat{z}$-vectors anyway. The above explains the necessity of the "obfuscation" step, that prevents the attacker to collect many words from the same flip range.

At this point let us observe that the problem of discovering the underlying scenery by seeing it at a random walker's location (known as *the scenery reconstruction problem*) is known to be very difficult even in two dimensions (see [3]; it was proved it is possible, but with millions of colors; with just a few, this should be hardly possible). Probably, the "random walk" method alone would already provide a decent cipher; we feel, however, that first using the "flip range" approach greatly increases the security.

In fact, for addressing the above potential vulnerability, one could consider another modification of the protocol, that may work even without the obfuscation matrix. For $M \in \mathbb{N}$ let us define the alphabets

$$\begin{aligned} \mathfrak{A}_{(M)} &= \{1, \dots, Ms_0\}, \\ \mathfrak{A}^*_{(M)} &= \{0, 1, \dots, Ms_0\}; \end{aligned}$$

that is, one can interpret that in $\mathfrak{A}_{(M)}$ each original letter is repeated M times, and (as before) the alphabet $\mathfrak{A}^*_{(M)}$ also contains a meaningless letter 0. The letter i of the original alphabet $\mathfrak{A}$ is represented by the letters $i, i+s_0, \dots, i+(M-1)s_0$ of the alphabet $\mathfrak{A}_{(M)}$. (Of course, we can also repeat different letters different number of times.) Then, for all $i \in \mathfrak{A}$, each time Alice wants to encode the letter i, she first chooses $j \in \{i, i+s_0, \dots, i+(M-1)s_0\}$ at random and then encodes it according to the above protocol. If she does the encoding without the obfuscation matrix, then the security of the cipher depends on the following general question. Let $\hat{x}_1, \dots, \hat{x}_M$ be vectors in $\mathbb{R}^{2n_0}$ (unknown to the attacker), with all coordinates being equal to ± 1. Assume that we have a large (at least $(2n_0-1)M$) number of known vectors $\hat{z}_1, \dots, \hat{z}_h \in \mathbb{R}^{2n_0}$, also with all coordinates being equal to ±1, and such that for all $j = 1, \dots, h$ we are guaranteed that there exist $r_j \in \{1, \dots, M\}$ such that $\hat{z}_j \cdot \hat{x}_{r_j} = 0$. Can we determine at least one of the unknown vectors up to sign, *in an efficient way*[3]? We are assuming, of course, that for some $i \in \{1, \dots, M\}$ there is a subset $\{t_1, \dots, t_{2n_0-1}\}$ of $\{1, \dots, h\}$ such that

$$\hat{z}_{t_j} \cdot \hat{x}_i = 0, \quad j = 1, \dots, 2n_0 - 1$$

and $z_{t_1}, \dots, z_{t_{2n_0-1}}$ are linearly independent, so the above system of linear equations determines $\hat{x}_i$ up to sign; we may even assume that the above holds for all i. The general difficulty is, of course, that we cannot quickly check all subsets of $\{1, \dots, h\}$ of size $2n_0 - 1$. Still, the authors are unsure if a more efficient solution of the above problem exists. This indicates the (relative?) necessity of the "obfuscation matrix" step. Still, probably it is a good idea to combine them all, i.e., use the approach with $\mathfrak{A}_{(M)}$ and the obfuscation matrix.

4 Some tools

Notice that if X is a random element of $\mathcal{C}_k$, then $\|X\| \sim \text{Binom}(k, \frac{1}{2})$. Also, $\mathcal{H}(X, \eta) \sim \text{Binom}(k, \frac{1}{2})$ for any fixed η, which implies that $\mathcal{H}(X, Y) \sim \text{Binom}(k, \frac{1}{2})$ for independent $X, Y \in \mathcal{C}_k$.

We recall the Chernoff's bound for the binomial distribution[4]: let $Y \sim \text{Binom}(k, q)$. Then, for any k and a with $0 < a < q < 1$, we have

$$\mathbb{P}[Y \leq ak] \leq \exp\big(-kH(a, q)\big), \tag{9}$$

where

$$H(a, q) = a \ln \frac{a}{q} + (1 - a) \ln \frac{1 - a}{1 - q} > 0.$$

The same inequality holds for $\mathbb{P}[Y \geq ak]$ when $0 < q < a < 1$. Note that, in particular,

$$H(a, \tfrac{1}{2}) = a \ln(2a) + (1 - a) \ln(2(1 - a)), \tag{10}$$

see Figure 1.

[3] that is, in polynomial(n_0) time, for fixed $M \geq 2$

[4] see e.g. Proposition 5.2 of Chapter 8 of [6], or Section 6 of Chapter I of [12]; also, the inequality in (9) is, in some sense, "almost equality", but more advanced tools are needed to justify that, see [2]

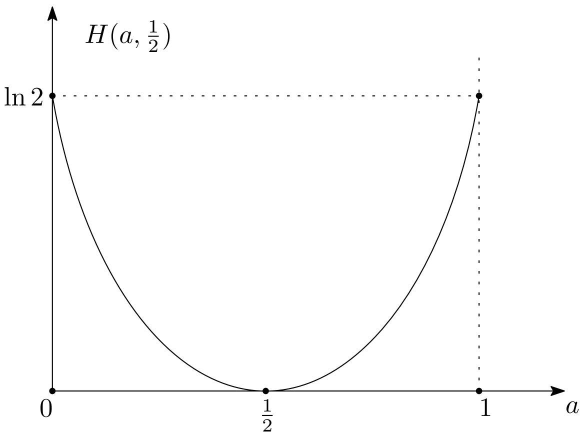

Figure 1: The graph of $H(a, \frac{1}{2})$.

As a corollary, note that, if X, Y are two independent random elements of $\mathcal{C}_{2k}$, and $b < 1$, then

$$\mathbb{P}\big[\mathcal{H}(X,Y) < bk\big] = \mathbb{P}[\mathcal{H}(X,Y) > (2-b)k\big] \le \exp\big(-2kH\big(\tfrac{b}{2},\tfrac{1}{2}\big)\big). \qquad (11)$$

We also make use of the *Stirling's approximation* of the factorial:

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n, \qquad (12)$$

or a refined version of the above

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n\Big(1 + \frac{1}{12n} + O(n^{-2})\Big), \qquad (13)$$

see e.g. [5]. In fact, in [5] a stronger result was proved, namely

$$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\frac{1}{12n+1}} < n! < \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\frac{1}{12n}} \qquad (14)$$

for all $n \in \mathbb{N}$.

Now, we need the following simple result about flip ranges:

Proposition 4.1. *Let $\eta \in \mathcal{C}_{2k}$, X is a randomly chosen binary word from $\mathcal{C}_{2k}$, and Y is a randomly chosen binary word from $\mathsf{FR}(\eta)$. Let $\zeta \in \mathcal{C}_{2k}$ be such that $\mathcal{H}(\eta,\zeta) = 2s$. Then*

$$\mathbb{P}[X \in \mathsf{FR}(\eta)] = 2^{-2k}\binom{2k}{k} \sim \frac{1}{\sqrt{\pi k}}, \qquad (15)$$

and

$$\mathbb{P}[X \in \mathsf{FR}(\zeta) \mid X \in \mathsf{FR}(\eta)] = \mathbb{P}[Y \in \mathsf{FR}(\zeta)] = \frac{\binom{2s}{s}\binom{2(k-s)}{k-s}}{\binom{2k}{k}} \sim \frac{1}{\sqrt{\pi k\alpha(1-\alpha)}}, \tag{16}$$

where $\alpha := s/k$ (we assume in the above equivalence that both k and s are large).

Note that the probability in (16) equals 0 by Proposition 2.2 (v) in case $\mathcal{H}(\eta, \zeta)$ is odd.

Proof. First, observe that (15) immediately follows from (4). Now, recall the proof of item (v) of Proposition 2.2; it is straightforward to see that the method we used to construct a binary word that belongs to both flip ranges is *the only* possible one. This gives the exact formula in (16), and the asymptotic expression is again obtained from (12) after some elementary calculations. □

5 "Choose-it-at-random" works!

In this section we address the question about how Alice chooses the ciphertext words in the above described algorithm, and also how she chooses the keys. Basically, we are going to show that choosing N words at random (from the corresponding flip range) on each step works with probability close to 1. This is, of course, a very classical approach, cf. e.g. the beautiful book [1].

First, recall (11) and let us use it e.g. with $b = 1/2$; it tells us that the Hamming distance between two randomly chosen binary words of length $2k$ is at least $k/2$ and at most $3k/2$ with probability at least $1-2\exp\left(-2kH(\frac{1}{4}, \frac{1}{2})\right)$, where $H(\frac{1}{4}, \frac{1}{2}) \approx 0.1308$. By the union bound, if Alice just chooses $s_0 m_0$ words from $\mathcal{C}_{2n_0}$ at random, then *all* pairs of them will be separated (in Hamming distance) by at least $n_0/2$ and at most $3n_0/2$, with probability at least

$$1 - 2s_0 m_0 \exp\Big(- 2n_0 H\Big(\frac{1}{4}, \frac{1}{2}\Big)\Big).$$

For example, with $s_0 = 26$, $m_0 = 10$, and $n_0 = 30$, the last formula gives approximately 0.7969; that is, with at least that probability Alice will be able to choose the codewords at the first try. In any case, for *reasonable* values of the parameters (i.e., n_0 should be large enough; the large is $s_0 m_0$, the large n_0 must be), Alice can choose the codewords $(\eta_{i,j})$ in such a way that the Hamming distance between any pair of these words is at least $n_0/2$ and at most $3n_0/2$ simply by choosing them independently at random (in the unlikely event that the above did not happen, Alice just repeats the whole procedure once more). From now on we assume that the codewords were chosen in this way.

Now, we address the question about how Alice chooses the words $\theta_1, \dots, \theta_N$ in a way that ensures that the decoding works correctly (i.e., the "right" Bob can read the corresponding letter, while the others not). For $m = 1, \dots, m_0$, let

$$\mathcal{K}_m = \{\eta_{j,m}, j = 1, \dots, s_0\}$$

be the set of codewords shared between Alice and Bob$_m$, and denote by

$$\mathcal{K} := \bigcup_{m=1}^{m_0} \mathcal{K}_m$$

the set of all Alice's codewords. Now, fix any $\eta \in \mathcal{K}_m$, and assume that Alice wants to encrypt the corresponding (to η) letter for the corresponding Bob (i.e., Bob$_m$). We want to argue that a good strategy for her is simply to choose N elements from $\mathsf{FR}(\eta)$ independently at random: with probability close to 1 this procedure would lead to intended result (and if not, Alice can just repeat). Let us denote these elements by $Y_1, \dots, Y_N$. If $\zeta \neq \eta$ is another word from $\mathcal{K}_m$ such that $\mathcal{H}(\eta, \zeta)$ is even, observe that Proposition 4.1 gives us that

$$\mathbb{P}\big[Y_j \in \mathsf{FR}(\zeta) \text{ for all } j = 1, \dots, N\big] \lesssim \Big(\frac{4}{\sqrt{6\pi n_0}}\Big)^N = \big(4/\sqrt{6\pi}\big)^N n_0^{-N/2} \quad (17)$$

(note that the parameter α in Proposition 4.1 will be between $\frac{1}{4}$ and $\frac{3}{4}$, so $\alpha(1-\alpha) \geq \frac{3}{16}$). Next, for $k \neq m$, Bob$_k$ will receive "transformed" words $Y_1^k, \dots, Y_N^k$, which can be assumed to be roughly uniformly distributed on $\mathcal{C}_{2n_0}$. Let us introduce the event (recall that $(Y_1^m, \dots, Y_N^m) = (Y_1, \dots, Y_N)$)

$$G = \big\{\text{for any } \zeta \in \mathcal{K}_\ell \setminus \{\eta\} \text{ there exists } j \text{ such that } Y_j^\ell \notin \mathsf{FR}(\zeta), \\ \text{for all } \ell = 1, \dots, m_0\big\}.$$

Using also (15), we then apply the union bound[5] to obtain that

$$\mathbb{P}[G] \gtrsim 1 - s_0 m_0 (C n_0)^{-N/2}, \quad (18)$$

[5] observe that with the union bound we are on the safe side

for a universal constant $C > 0$. Notice that the above event G guarantees that the right Bob will receive the right letter, and all other Bobs just receive zeros.

In the same way we can bound (obtaining essentially the same estimate) the probability that all Bobs receive zeros if Alice just chooses N words totally at random (in case she wants to transmit a meaningless letter).

Essentially, (18) suggests that the maximal cardinality of the codeword set[6] is $O(n_0^{N/2})$; that is, with that many codewords Alice is able to "encrypt with randomness". A significantly larger number of codewords could make the encryption process difficult (as well as possibly compromise the security): the set $\mathcal{C}_{2n_0}$ would be, in a way, "overcrowded" with codewords.

6 Perfect functional secrecy

Loosely speaking, the prevailing ciphers lock the contents of the message behind mathematical locks, which are nonetheless vulnerable to 'weaponized math' of greater depth. By contrast, Vernam cipher protects its secret by hiding it among all the possible messages of the same bit length. Claude Shannon proved that the Vernam ciphertext contains no means to distinguish the encrypted message from the $2^n - 1$ "decoy" messages (n is the bit count of the ciphertext). To overcome this obfuscation a reader needs to possess a copy of the same randomness that generated this ciphertext. Come to think about it, Vernam is an "over-kill". In practical situations there is no need for all possible decoy messages to be viable candidates. In fact any small number of plausible decoy messages, if they are packed together with the actual message such that the ciphertext contains no clue for identifying the true message, will offer functional equivocation, and will doom their cryptanalyst to terminal ambiguity. For example, a stock adviser may communicate to his client one of three options for handling a financial instrument: "buy!"[7], "sell" or "hold". A cipher that will pack all three messages into a single ciphertext will be functionally equivalent to Vernam. Using the BitFlip cipher Alice will communicate "buy!" to Bob$_1$, communicate "sell" to Bob$_2$, and communicate "hold" to Bob$_3$. Say, Bob$_2$ and Bob$_3$ are virtual, and only Bob$_1$ exists. Bob$_1$ then will interpret as zeros all the letters

[6] observe that $\text{card}(\mathcal{K}) = s_0 m_0$

[7] The exclamation mark is added to "buy" to make it comprised of four ASCII symbols, like the other options, just to validate the subsequent comparison with Vernam cipher

communicating to Bob_2 and Bob_3 (ignore them), and read his message unequivocally. However, a cryptanalyst without the possession of Bob_1's key will either be confounded by the intractability of the cipher, or, at best, will dig out all three messages: "sell", "buy!" and "hold" and will not be the wiser.

Had that financial adviser used Vernam, say, with ASCII coding ($8 \times 4 = 32$ bits), the cryptanalyst would have faced a much large equivocation: 2^{32} message candidates. This theoretical advantage over the BitFlip user is of little practical value since the cryptanalyst, aware of the circumstances, would expect the message range to be "buy!", "sell" or "hold".

7 Document management protocol

Large projects conducted by highly structured organizations are documented through D_1 category data designed to be exposed to all project handlers. On top of D_1 the organization will develop D_2 category data that is designed to be hidden from certain project handlers who are cleared for D_1. Iteratively, such a project develops data D_i to be read and be written by category i project handlers. D_i is designed to be hidden from project handlers of categories $1, 2, \ldots, i-1$. The challenge of managing data exposure may be alleviated by deploying BitFlip with i sets of binary words: $\mathsf{Key}_j = \eta_{1,j} \ldots, \eta_{s_0,j}$ for $j = 1, 2, \ldots, i$ and assigning to readers of category j the keys $\mathsf{Key}_1, \mathsf{Key}_2, \ldots, \mathsf{Key}_j$. This will allow the organization to keep one updated copy of the project document (one master copy), distributed to all project handlers. That master copy will be encrypted to insure that individuals from each project handling category are exposed only to the parts of the master file that is designated for them. A great administrative relief compared to the standard protocols where a myriad of documents must be managed, and one must insure that all updates flow through all the versions. With BitFlip a low level project handler will be able to send the encrypted version of the single master project document to a higher level project handler which will see in it what the sender does not. All project handlers will use their keys to read and write in that single master project document.

8 Security

Let $H(M)$ be the Shannon entropy of the message space, M, as evaluated by the attacker before he captured the user's ciphertext, and $H'(M)$, the entropy as it is evaluated by the attacker after exhausting his cryptanalysis of the ciphertext c. We regard $\rho = (H(M) - H'(M))/H(M) \in [0, 1]$ as the efficacy of the attack.

We discuss two boundary situations: Large $H(M)$, and Small $H(M)$. The first case may be exemplified by the sender communicating a secret password. The attacker, a-priori faces a very large plausible message space where the probability of each element in M is $1/|M|$ or just about it. The second case may be exemplified by the sender communicating a stock handling recommendation, where M is compromised of three elements: $M = \{$"buy", "sell", "hold"$\}$.

The security of the case of large entropy is diminished when the ratio between the size of the encrypted material, and the key space, $|m|/|K|$, is growing. This is because of the inherent nature of BitFlip. Given any ciphertext word, w, its Flip Range, $\mathsf{FR}(w)$ determines the scope of the alphabet letter it represents. And the larger $|m|$, there more words there are, and the more linear equations may be written between a proposed key and the given ciphertext. Every proposed key for which these equations have no solution is the wrong key. Obviously for $|m| \to \infty$ there remains only one key that satisfies the growing number of linear equations, and only the computation intractability stands between the secret message and its successful cryptanalysis ($\rho = 1$).

Operationally this implies that the user may wish to replace the key before the cryptanalyst has enough information to identify unambiguously the encryption key. The combinatorics computations of this strategy are a bit complex, and will be given in a subsequent publication. What is important in this case is the fact that the vulnerability of the BitFlip cipher for large $H(M)$ is credibly anticipated by the user, and it can be remedied by either replacing the key, or by adjusting its size.

The case of low $H(M)$ may be handled, surprisingly, in the opposite way: the more message material that is processed with a given key, the more $H'(M)$ approaches a well calculated low boundary, which in turn keeps the attack efficacy well bounded. Which in turn is the guaranteed security for the user. See below.

Security for low A-Priori Message Entropy. Given an arbitrary natural number $h \in \mathbb{N}$, a message writer will identify a subset M_h of the message space M, comprised of h messages $m_1, m_2, \ldots m_h$ such that each m_i is more plausible than the statistical average (as dictated by the prevailing circumstances.) Namely:

$$\mathbb{P}[M = m_i \mid m_i \in M_h] > 1/|M| \qquad \text{for } i = 1, 2, \ldots, h.$$

Let us equate the size of the h messages to $|m| = |m_1| = |m_2| = \ldots$ by adding null characters, if necessary.

Using the BitFlip cipher via the obfuscation matrix the message writer will pick a uniformly selected key $k_i \in \mathcal{K}$ to encrypt m_i to c_i $(i = 1, 2, \ldots, h)$. The writer will then mix the c_i to a combined ciphertext c.

Lemma 8.1. *For $|m| \to \infty$, there is no key $k^* \neq k_i$ that decrypts c to m_i.*

Proof. For a given message size $|m_i|$ let $k^* \neq k_i$ decrypt: $m_i = Dec_{k^*}(c)$. As $|m|$ grows (more message material is encrypted via $k_1, k_2, \ldots, k_h$), the chances for a given letter l in m_i for which the word expression in k^* (l_i^*) is different from the word expression in k (l_i), to be encrypted to a word l' which while $l_i \in \mathsf{FR}(l)$, it does not belong to the Flip Range of l_i^*; $\mathsf{FR}(l_i^*)$, is getting larger. That is because, as has been shown above, the Flip Ranges of two non identical strings, $x \neq y$, are not the same, $\mathsf{FR}(x) \neq \mathsf{FR}(y)$ (unless $y = \bar{x}$), and hence sooner or later a random selection of a member of the Flip Range of x will not qualify as a member of the Flip Range of y. And hence, k^* will not decrypt c to l. □

Loosely speaking, as more and more message material is encrypted through $k_1, k_2, \ldots, k_h$, there is a diminishing chance that any other key will decrypt c to the corresponding messages $m_1, m_2, \ldots, m_h$.

Lemma 8.2. *For $|m| \to \infty$, for any plausible message m^* that does not belong to M_h, there is no key $k \in \mathcal{K}$ that decrypts c to m^*.*

Proof. This is the same situation as in all the common ciphers: since the size of combined key $\{k_1, k_2, \ldots, k_h\}$ is fixed, and since for $|m| \to \infty$ the proportion of plausible messages relative to all possible messages is fast shrinking, then the chance for any key k^* to decrypt c to a plausible messages shrinks too, and becomes negligible. □

We summarize: as more and more message material is encrypted via the randomly selected h keys, the chance for the encrypted material to be decrypted from c via different keys, diminishes, and the chance for non selected plausible messages to be decrypted from c through any key are equally diminishing.

So for large enough $|m|$, we may write:

$$\mathbb{P}[M = m_i \mid C = c] \approx 1/h \qquad \text{for all } i = 1, 2, \ldots, h.$$

Now suppose that k_1 is the key that was shared with the intended reader of the message. She will readily decrypt $m_1 = Dec_{k_1}(c)$. Alas, an attacker will face a probability $1/h$ for the right message m_1, and a probability of $(h-1)/h$ for some other message to be the valid one. In the simplest case where $h = 2$, the cryptanalyst will face a 50:50 chance to identify the right message. This chance diminishes for larger h.

Conclusion: We identified a use methodology for BitFlip in situations where the a-priori entropy is small. The methodology is comprised of co-encrypting the secret message with plausible decoy messages using the obfuscation matrix. We have shown that this procedure will deny even an unbound cryptanalyst an unambiguous determination of the encrypted message.

Note: the decoy plausible messages may be worked out automatically using modern AI techniques. Writer and recipient may share all the K_h keys, and switch as to which key they use each time, or each day, or otherwise, through a pre-agreed randomized schedule. This will prevent a cryptanalyst from learning which key counts by analyzing the reaction of the recipient to the read messages.

9 More on security

We have seen above that, even without the obfuscation matrix, the number of different cipherstrings (of length $2n_0$) representing a given letter is very large (about $2^{2n_0}/\sqrt{\pi n_0}$, recall (4)), which rules out any possibilities of e.g. using frequency analysis or similar methods. Also, the size of the space of all possible keys (with only one Bob, i.e., $m_0 = 1$, and without the obfuscation matrix) is $O(2^{2s_0 n_0})$, which makes the complete search in the key space impossible even for moderately large values of n_0.

Although the precise calculations seem to be very difficult in the case when the obfuscation matrix is involved, nevertheless, in the following we are going to present a (not completely rigorous) argument that shows that the ciphertext produced by BitFlip is practically indistinguishable from a random bit string (at least when the binary logarithm of the size of the ciphertext is much less than $2n$, which seems to be a reasonable assumption).

The key idea is to observe that large chunks of the ciphertext we create are *almost completely* random. To formalize this, let us recall the notion of *total variation distance* between two probability measures P and Q on a measurable space $(\Omega, \mathcal{F})$:

$$\|P - Q\|_{TV} = \sup_{A \in \mathcal{F}} |P(A) - Q(A)|. \tag{19}$$

It is elementary to obtain that $\|P - Q\|_{TV} \in [0, 1]$ for all P and Q, and also

$$\|P - Q\|_{TV} = \frac{1}{2} \sum_{x} |p(x) - q(x)| \tag{20}$$

in case P and Q are discrete with weight functions p and q, and

$$\|P - Q\|_{TV} = \frac{1}{2} \int_{-\infty}^{+\infty} |f_1(x) - f_2(x)| \, dx \tag{21}$$

in case P and Q are (absolutely) continuous with densities f_1 and f_2. In general, it also holds that $1 - \|P - Q\|_{TV}$ equals the probability of the *coupling event* under the *maximal coupling*, cf. [13].

Next, define the funcion

$$\varphi(t) = \begin{cases} \dfrac{1}{2\sqrt{2\pi}} \displaystyle\int_{-\infty}^{+\infty} \left| e^{-\frac{x^2}{2}} - \frac{1}{\sqrt{1-t}} e^{-\frac{x^2}{2(1-t)}} \right| dx, & \text{for } t \in [0, 1), \\ 1, & \text{for } t = 1, \end{cases} \tag{22}$$

see Figure 2. It is elementary to see that φ is a continuous increasing function on the interval $[0, 1]$, with $\varphi(0) = 0$, $\varphi(1) = 1$. Also, since for any fixed x and $t \to 0$

$$\begin{aligned} \left| 1 - \frac{1}{\sqrt{1-t}} e^{-\frac{x^2}{2}\left(\frac{1}{1-t} - 1\right)} \right| &= \left| 1 - \left(1 + \frac{t}{2} + O(t^2)\right)\left(1 - \frac{x^2}{2} t + O(t^2)\right) \right| \\ &= \frac{t}{2} \times \left| 1 - x^2 + O(t) \right|, \end{aligned}$$

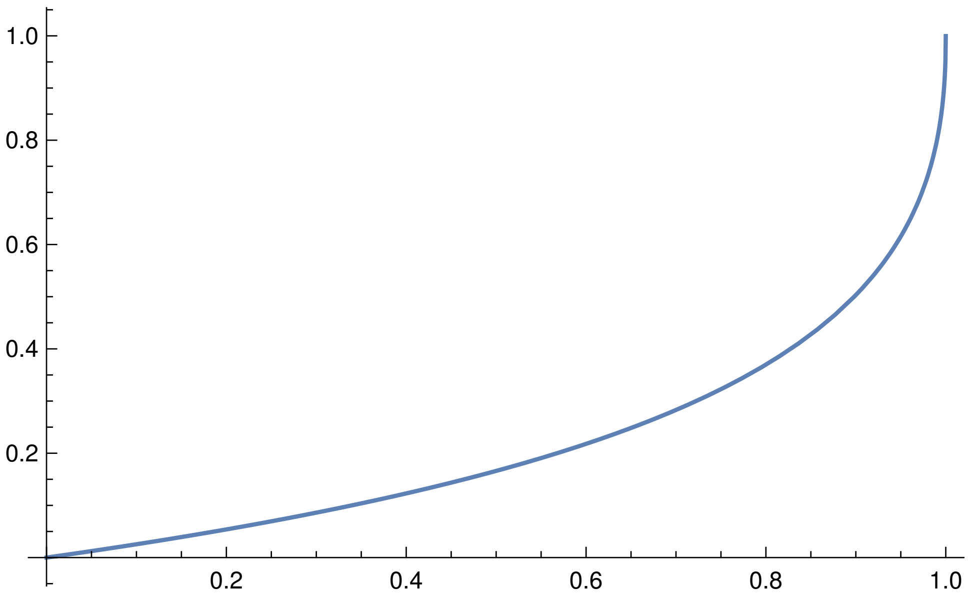

Figure 2: The graph of φ.

we have

$$\varphi'(0) = \frac{\mathbb{E}|1-Z^2|}{4} = \frac{1}{\sqrt{2e\pi}} \approx 0.241971, \tag{23}$$

where Z is a standard Normal random variable.

Notice that (recall (21)) $\varphi(t)$ is the total variation distance between the centered Normal law with variance $(1-t)$ and the standard Normal law. Moreover, an easy change-of-variable argument shows that $\varphi(t)$ is also the total variation distance between $\mathcal{N}(\mu,\sigma^2)$ and $\mathcal{N}(\mu,(1-t)\sigma^2)$, for *any* $\mu \in \mathbb{R}$ and $\sigma > 0$.

Now, we are ready to formulate our "almost-independence" result. For $\eta \in \mathcal{C}_{2n}$ and $k \le 2n$ let $\mathcal{L}_k^{(2n)}(\eta)$ be the law of the first k bits of a randomly chosen (with uniform distribution) configuration from $\mathsf{FR}(\eta)$, and denote also by $\widetilde{\mathcal{L}}_k$ the law of random independent bits (i.e., a sequence of k Bernoulli($\frac{1}{2}$) trials). We have

Proposition 9.1. *For any sequence of binary words* $(\eta_n \in \mathcal{C}_{2n}, n \ge 1)$ *and any* $t \in [0,1]$, *we have*

$$\left\|\mathcal{L}_{[2tn]}^{(2n)}(\eta_n) - \widetilde{\mathcal{L}}_{[2tn]}\right\|_{TV} \to \varphi(t) \tag{24}$$

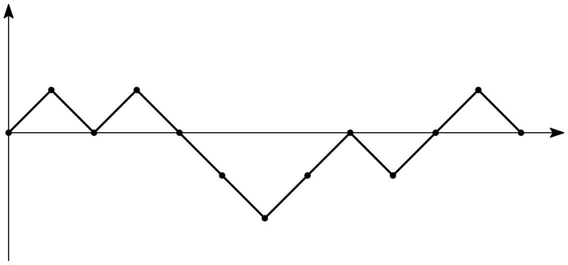

Figure 3: The random walk's trajectory generated by the binary word $101000110110 \in \mathsf{FR}(0_{12})$.

as $n \to \infty$.

In words, the above result means that large pieces (of size αn_0 for a not-so-large $\alpha > 0$) of the ciphertext are "almost indistinguishable" from the Bernoulli trials. Also, it is clear that the same result applies to the bits on any $[2tn]$ *fixed* positions. For example, since $\varphi(1/3)$ is only approximately 0.1, in at least 9 cases of 10 the set of bits on $2n_0/3$ positions cannot be distinguished from the set of completely random bits.

Proof of Proposition 9.1. Note that, by symmetry, it is enough to consider the case $\eta_n = 0_{2n}$, for all $n \geq 1$. Now, the key idea to represent a binary word $\eta \in \mathcal{C}_{2n}$ as a *random walk*: we interpret every 1 as a step up, and every 0 as a step down, see Figure 3. Then, the completely random element of $\mathcal{C}_{2n}$ (i.e., $2n$ Bernoulli($\frac{1}{2}$) trials) corresponds to a trajectory of a *simple random walk*, i.e., the walk that steps in both directions with equal probabilities. On the other hand, it is clear that a random element of $\mathsf{FR}(0_{2n})$ can be interpreted as a simple random walk *conditioned* on being at the origin at time $2n$. The task of estimating the total variational distance between $\mathcal{L}^{(2n)}_{[2tn]}(\eta_n)$ and $\widetilde{\mathcal{L}}_{[2tn]}$ then amounts to constructing a coupling between conditioned and unconditioned simple random walks up to time $[2tn]$. Since all paths of the same length have apriori the same weight, it is clear that it's enough to couple the positions of the walkers at time $[2tn]$.

Now, instead of witing the formal proof, we present a heuristic argument that shows the validity of (24). Indeed, it is well known that, under the scaling $(m, \ell) \mapsto (\frac{m}{2n}, \frac{\ell}{\sqrt{2n}})$, the simple random walk converges to the Brownian motion, and the conditioned simple random walk converges to the

Brownian bridge[8] (see e.g. [4]). Notice that, in the limit, the discrete time $[2tn]$ becomes the continuous time t under this scaling. Now, it holds that $W_t \sim \mathcal{N}(0,t)$ and $B_t \sim \mathcal{N}(0, t(1-t))$; the total variation distance between then is therefore $\varphi(t)$.

Strictly speaking, that scaling limit argument does not imply the convergence of the total variation distances we want to obtain. However, for random walks the calculations are essentially the same if one uses a suitable version of the *local* Central Limit Theorem[9]. We omit the details. □

As Proposition 9.1 shows, there is a kind of "mesoscopic independence" in the ciphertext. There is still "global dependence", but it is hidden by the random walks on the g-matrices: since the "close" steps are virtually independent, it is likely that the "raw" ciphertext will be transformed by that random walk in a completely impredictable way (note also the huge number of possible random walk's trajectories). It is then reasonable to believe that this renders any "linear equations attacks" (as described in the end of Section 3.1) nearly impossible.

The above justifies the following conjecture: a successfull attempt to break the BitFlip cipher amounts to a (more-or-less) complete search in the space of possible keys, which is, of course, not computationally feasible for reasonably large values of n_0.

References

[1] N. Alon, J.H. Spencer (2016) *The Probabilistic Method.* (4th ed.) Wiley.

[2] A. Dembo, O. Zeitouni (2010) *Large Deviations Techniques and Applications.* Springer.

[3] M. Löwe, H. Matzinger (2002) Scenery reconstruction in two dimensions with many colors. *Ann. Appl. Probab.* **12** (4), 1322–1347.

[8] informally, the Brownian bridge on the interval $[0,1]$ is the standard Brownian motion conditioned on being in 0 at time 1; we wrote "informally" because this event has zero probability and therefore one must take some care to properly define the conditioned process

[9] e.g. the theorem of de Moivre-Laplace

[4] D. Revuz, M. Yor (1999) *Continuous Martingales and Brownian Motion* (2nd ed.). Springer, New York.

[5] H. Robbins (1955) A remark on Stirling's formula. *Amer. Math. Monthly* **62** (1), 26–29.

[6] Sheldon M. Ross (2009) *A First Course in Probability.* 8th ed.

[7] Gideon Samid (2002) At-Will Intractability Up to Plaintext Equivocation Achieved via a Cryptographic Key Made As Small, or As Large As Desired - Without Computational Penalty. International Workshop on CRYPTOLOGY AND NETWORK SECURITY, San Francisco, California, USA September 26 – 28, 2002.

[8] Gideon Samid (2004) Denial Cryptography Based on Graph Theory. US Patent 6,823,068.

[9] Gideon Samid (2015) Equivoe-T: Transposition Equivocation Cryptography. International Association of Cryptology Research, ePrint Archive https://eprint.iacr.org/2015/510

[10] Gideon Samid (2016) Cryptography of Things (CoT): Enabling Money of Things (MoT), kindling the Internet of Things. The 17th International Conference on Internet Computing and Internet of Things, Las Vegas, July 2016.

[11] Claude Shannon (1949) *Communication Theory of Secrecy Systems.* http://netlab.cs.ucla.edu/wiki/files/shannon1949.pdf

[12] A.N. Shiryaev (1996) *Probability.* Springer, New York.

[13] H. Thorisson (2000) *Coupling, Stationarity, and Regeneration.* Springer, New York.

[14] Gilbert S. Vernam (1918) *Secret Signaling System.* US Patent 1310719A.