On Data Complexity of Distinguishing Attacks vs. Message Recovery Attacks on Stream Ciphers

Goutam Paul · **Souvik Ray**

Abstract We revisit the different approaches used in the literature to estimate the data complexity of distinguishing attacks on stream ciphers and analyze their inter-relationships. In the process, we formally argue which approach is applicable (or not applicable) in what scenario. To our knowledge, this is the first kind of such an exposition. We also perform a rigorous statistical analysis of the message recovery attack that exploits a distinguisher and show that in practice there is a significant gap between the data complexities of a message recovery attack and the underlying distinguishing attack. This gap is not necessarily determined by a constant factor as a function of the false positive and negative rate, as one would expect. Rather this gap is also a function of the number of samples of the distinguishing attack. We perform a case study on RC4 stream cipher to demonstrate that the typical complexities for message recovery attack inferred in the literature are but under-estimates and the actual estimates are quite larger.

Keywords Data Complexity · Distinguisher · Distinguishing Attack · Message Recovery · Stream Cipher

Mathematics Subject Classification (2010) 94A60

The second author worked for this paper during the summer and winter breaks of 2015 in his Bachelor of Statistics course.

Indian Statistical Institute, Kolkata 700 108, India.
E-mail: goutam.paul@isical.ac.in, souvikr974@gmail.com

1 Introduction

A stream cipher generates a long pseudo-random keystream from a short secret key to encrypt a message by bitwise XOR operation with the keystream. Since the sender and the receiver share the same secret key and the keystream generation algorithm is deterministic, the identical keystream is generated at the receiver side, which when bitwise XOR-ed with the ciphertext recovers the message.

For a stream cipher, if there is an event such that the probability of occurrence of the event is different from the same event in case of a uniformly random sequence of bits, the event is said to be *biased*. If there exists a biased event based only on the bits of the keystream sequence, then such an event gives rise to a *distinguisher* for the cipher. A distinguisher can computationally differentiate between the keystream output of the stream cipher and a truly random sequence of bits.

Very often, a distinguisher is directly used in mounting a message recovery attack on stream ciphers. A famous example is the attack [20] on broadcast RC4. Let Z_r be the r-th keystream byte of RC4 and $N = 256$ be the standard state array size of RC4. It was proved in [20] that $\Pr(Z_2 = 0) \approx \frac{2}{N}$ for RC4, whereas the same event in an uniformly random bitstream would occur with probability $\frac{1}{N}$. In the broadcast scenario, the same plaintext is encrypted using multiple secret keys, and then the ciphertexts are broadcast to a group of recipients, possessing the corresponding secret keys. For every encryption key, the second message byte M_2 has the probability $\approx \frac{2}{N}$ to be XOR-ed with 0, and the probability $\approx \frac{1}{N}$ to be XOR-ed with each of the other possible bytes (subject to the obvious constraint that the probabilities sum up to 1). Thus, a fraction of $\frac{2}{N}$ of the second ciphertext bytes C_2 are expected to have the same value as M_2, and thus the most frequent value of C_2 across all the samples is the mostly likely value of M_2. The above approach has been adopted by [14] in mounting message recovery attack on every individual message bytes 3 to 255 based on the distinguishers for RC4 keystream Z_r, $3 \leq r \leq 255$. Later, the work [1] considered the collection of all the biases in all the keystream byte together to perform joint message recovery. Similar message recovery attacks can be performed based on the distinguishers on other stream ciphers as well such as HC-128, Spritz etc. [18,24,5]. However, as case study, we focus on RC4 only, as its description is comparatively shorter and easier.

The efficiency of a distinguisher is measured by two complexities - the *sample complexity*, i.e., the number of samples (of the involved keystream bits) required to identify the bias, and the *data complexity*, i.e., the number of keystream bits required to identify the bias. For an attacker to successfully identify and exploit a bias, one requires to inspect a certain length of the output sequence so that one can collect sufficient number of samples for the event under consideration. The less the number of samples or keystream bits required to mount the distinguisher or to perform the message recovery attack, the more is the efficiency of the distinguisher or the message recovery attack. In general, the sample complexity and the data complexity need not be the same. For many distinguishers, the keystream from one single key can be used to mount the attack, but it is also possible to use different keystreams, with new IV. For the message recovery attack, one always must use different keystreams since one needs to look at one message bit/byte. Moreover, the biased bit/byte might not be in the start of the sequence and this may affect the amount of keystream required as well. Without loss of generality, in this paper we focus on the sample complexity, since the goal is to compare the complexities of a distinguishing attack and the corresponding message recovery attack. In the rest of the paper, whenever we use the term data complexity, it actually means sample complexity.

In all the above examples, the number of samples required to mount the message recovery attack is considered to be of the same order as that of the underlying distinguisher. However, we observe that in practice it is not always so. For example, for the broadcast attack on RC4 second byte, the complexity of message recovery attack for a success probability of 70% is around 8 times higher than that of the distinguishing attack for the same success probability. In this paper, we perform a rigorous analysis to understand this gap between the data complexities of distinguishing attack and message recovery attack.

1.1 Our Contributions

We observe that there exist different approaches to estimate the data complexity of a distinguisher, yielding different expressions, albeit sometimes one may be a crude approximation of the other. Moreover, the data complexity of a message recovery attack based on the distinguisher is usually taken to be the same (or of the same order) as the distinguisher itself, though in practice it is not necessarily true in all scenarios.

Our current work has the following contributions.

1. In Section 2, we review the different approaches used in the literature to estimate the data complexity and point out their connections and applicable

scenarios. To our knowledge, such an expository coverage of the different approaches has not been done so far. Wherever possible, we provide short proofs of the results to make this exposition self-sufficient. For longer proofs, we cite appropriate references.

2. In Section 3 and 4, we perform a rigorous statistical analysis of the message recovery attack and show that in practice there is a significant gap between the data complexities of a message recovery attack and the underlying distinguishing attack. This gap is not necessarily determined by a constant factor as a function of the false positive and negative rate, as one would expect. Rather, this gap is also a function of the number of samples of the distinguishing attack. Note that all the results (lemmas and theorems) except Lemma 7 in Section 3 and 4 are our original contributions.

   Though we have focused on one biased keystream byte in our analysis, the result is directly applicable to biased sum of keystream bits from which biased sum of message bits can be recovered, or more generally, it is also applicable to biased vector of keystream bits from which the corresponding biased vector of message bits can be recovered.

3. In Section 5, we perform a case study on RC4 stream cipher to demonstrate that the typical message complexities inferred in the literature are but under-estimates and the actual estimates are quite larger. We choose RC4 as our case study, as it has several well-known biases in the keystream and serves as a good model to illustrate our theoretical analysis.

1.2 Notations

Before going into technical discussion, we list down some notations frequently used in this article below.

$\mathcal{M}$:	The message space (the set of all possible bytes)
$\mathcal{P}$:	The distribution of the keystream bytes over $\mathcal{M}$
$\mathcal{P} \oplus m$:	Distribution of the random variable $X \oplus m$, where $X \sim \mathcal{P}$ and $m \in \mathcal{M}$
p_z :	Probability of the byte z in the distribution $\mathcal{P}$
$\mathcal{P}^{(k)}$:	The distribution of vector of k keystream bytes over $\mathcal{M}^k$
$p_{\boldsymbol{z}}^{(k)}$:	Probability of the k-byte vector $\boldsymbol{z}$ in the distribution $\mathcal{P}^{(k)}$
$\mathcal{Q}$:	The prior distribution of the plaintext bytes over $\mathcal{M}$
q_z :	Probability of the byte z in the distribution $\mathcal{Q}$
$\mathcal{Q}^{(k)}$:	The prior distribution of vector of k plaintext bytes over $\mathcal{M}^k$
$q_{\boldsymbol{z}}^{(k)}$:	Probability of the k-byte vector $\boldsymbol{z}$ in the distribution $\mathcal{Q}^{(k)}$
D_n :	n-dimensional discrete distribution over some countable set (same is P_n, Q_n)
$E_P[X]$:	Expectation of the random variable X under distribution P
$V_P(X)$:	Variance of the random variable X under distribution P
$\sigma_P(X)$:	Standard deviation of the random variable X under distribution P
$\mathcal{R}^c$:	Complement of a set $\mathcal{R}$
$\mathcal{B}er(p)$:	Bernoulli distribution with success probability p
$\mathcal{B}(n,p)$:	Binomial distribution with n trials and success rate p
$\mathcal{N}(\mu, \sigma^2)$:	Normal distribution with mean μ and variance σ^2
Φ :	Distribution function of standard normal distribution
ϕ :	Density of standard normal distribution
$A\mathcal{N}$:	Asymptotic Normal distribution
$\xrightarrow{P}$:	Convergence in probability [15]
$\xrightarrow{D}$:	Convergence in distribution [15]
$\xrightarrow{a.s.}$:	Almost sure Convergence [15]
$diag(\boldsymbol{v})$:	Diagonal matrix with diagonal being the vector $\boldsymbol{v}$

2 Revisiting Data Complexity of Distinguishing Attacks

In this section, we revisit the existing techniques for estimating the data complexity of a distinguisher and point out their relations and subtleties.

2.1 Distance between Expectations

This approach has been used in [20]. We revisit their main result below.

Theorem 1 *Suppose the event e happens in distribution $\mathcal{P}_0$ with probability p and in distribution $\mathcal{P}_1$ with probability $p(1+q)$. Then for small p and q, $O(\frac{1}{pq^2})$ samples suffice to distinguish $\mathcal{P}_0$ from $\mathcal{P}_1$ with a constant probability of success.*

Proof Suppose we observe a random variable X_e specifying the number of occurrences of e in n samples. We are to make a decision whether the samples come from $\mathcal{P}_0$ or $\mathcal{P}_1$. Then, X_e have binomial distributions with parameters (n, p) and $(n, p(1+q))$ under $\mathcal{P}_0$ and $\mathcal{P}_1$ respectively. Its expectations, variances and standard deviations are (assuming both $p, q \ll 1$) as follows.

$$E_{\mathcal{P}_0}[X_e] = np, E_{\mathcal{P}_1}[X_e] = np(1+q),$$

$$\begin{aligned}
V_{\mathcal{P}_0}(X_e) &= np(1-p) \approx np, \\
V_{\mathcal{P}_1}(X_e) &= np(1+q)(1-p(1+q)) \approx np(1+q), \\
\sigma_{\mathcal{P}_0}(X_e) &= \sqrt{V_{\mathcal{P}_0}(X_e)} \approx \sqrt{np}, \\
\sigma_{\mathcal{P}_1}(X_e) &= \sqrt{V_{\mathcal{P}_1}(X_e)} \approx \sqrt{np(1+q)} \approx \sqrt{np}.
\end{aligned}$$

The authors of [20] consider the size of n that implies a difference of at least one standard deviation between the expectations of the two distributions. So, we shall infer that the underlying distribution of X_e is $\mathcal{P}_1$, if the difference between X_e and np is at least one standard deviation, i.e.

$$X_e - np \geq \sqrt{np}.$$

Hence, if the true sample generating distribution is $\mathcal{P}_1$, and α denotes the failure probability, then our probability of success is given by

$$\begin{aligned}
1-\alpha &= \Pr(X_e - np \geq \sqrt{np} | \mathcal{P}_1) \\
&= \Pr\left(\frac{X_e - np(1+q)}{\sqrt{np}} \geq -q\sqrt{np} + 1 | \mathcal{P}_1\right) \\
&\approx 1 - \Phi(-q\sqrt{np} + 1) = \Phi(q\sqrt{np} - 1).
\end{aligned}$$

And therefore to get a success probability $1-\alpha$, we have to get number of samples

$$n \geq \frac{\left(\Phi^{-1}(1-\alpha)+1\right)^2}{pq^2}.$$

□

For the sake of completeness, we present the algorithm for the above distinguisher below:

```
Algorithm Distinguisher 1
    $X_e$ ← Number of occurrences of the event $e$ in $n$ trials
    if $X_e \geq np + \sqrt{np}$
        Infer the distribution is $\mathcal{P}_1$;
    else
        Infer the distribution is $\mathcal{P}_0$;
```

Another important thing to note is the advantage of the above distinguisher:

$$\begin{aligned}
\text{Advantage} &= \Pr(\text{Inferred to be } \mathcal{P}_1 | \mathcal{P}_1) - \Pr(\text{Inferred to be } \mathcal{P}_1 | \mathcal{P}_0) \\
&= \Pr(X_e - np \geq \sqrt{np} | \mathcal{P}_1) - \Pr\left(X_e - np \geq \sqrt{np} | \mathcal{P}_0\right) \\
&= 1 - \alpha - \Pr\left(\frac{X_e - np}{\sqrt{np}} \geq 1 | \mathcal{P}_0\right) \\
&\approx 1 - \alpha - (1 - \Phi(1)) = \Phi(1) - \alpha.
\end{aligned}$$

Note that when at least one of $p \ll 1$ and $q \ll 1$ does not hold, the above approach does not work.

2.2 Simple Hypothesis Testing

A more rigorous analysis appeared in [6] that gets rid of the restriction $p \ll 1$ and $q \ll 1$. We revisit this technique here.

Theorem 2 *Suppose the event e happens in uniform random bitstream with probability p and in keystream of a stream cipher with probability $p(1+q)$. Then the data complexity of the distinguisher with false positive and false negative rates α and β is given by*

$$n > \frac{\left(\kappa_1\sqrt{1-p} + \kappa_2\sqrt{(1+q)\,(1-p(1+q))}\right)^2}{pq^2},$$

where $\Phi(-\kappa_1) = \alpha$ and $\Phi(\kappa_2) = 1-\beta$.

Proof Consider an event e with $\Pr(e) = p^*$, while observing samples of keystream words of a stream cipher. Let $X_r = 1$, if the event e occurs in the r-th sample; $X_r = 0$, otherwise. In other words, $\Pr(X_r = 1) = p^*$ for all r. Thus,

$$X_r \sim \mathcal{B}er(p^*).$$

If we observe n many samples, then

$$\sum_{r=1}^{n} X_r \sim \mathcal{B}(n, p^*).$$

When X_r's are independent and identically distributed (i.i.d.) random variables and n is large enough,

$$\sum_{r=1}^{n} X_r \sim \mathcal{N}\left(np^*, np^*(1-p^*)\right).$$

We are interested in testing the null hypothesis

$$H_0 : p^* = p(1+q), \qquad q > 0,$$

against the alternative hypothesis

$$H_1 : p^* = p.$$

The objective is to find a threshold c in $[np, np(1+q)]$ such that

$$\Pr\left(\sum_{r=1}^{n} X_r \leq c \mid H_0\right) \leq \alpha,$$

i.e.,

$$\Pr\left(\frac{\sum_{r=1}^{n} X_r - np(1+q)}{\sigma_1} \leq \frac{c - np(1+q)}{\sigma_1} \mid H_0\right) \leq \alpha \iff c \leq np(1+q) - \kappa_1\sigma_1$$

and

$$\Pr\left(\sum_{r=1}^{n} X_r > c \mid H_1\right) \leq \beta,$$

i.e.,

$$\Pr\left(\frac{\sum_{r=1}^{n} X_r - np}{\sigma_2} \geq \frac{c - np}{\sigma_2} \mid H_1\right) \leq \beta \iff c \geq np + \kappa_2\sigma_2,$$

where

$$\begin{aligned} \sigma_1^2 &= np(1+q)\left(1 - p(1+q)\right), \\ \sigma_2^2 &= np(1-p), \\ \varPhi(-\kappa_1) &= \alpha, \\ \text{and} \qquad \varPhi(\kappa_2) &= 1 - \beta. \end{aligned}$$

For such a c to exist, we need

$$np(1+q) - \kappa_1\sigma_1 \geq np + \kappa_2\sigma_2, \tag{1}$$

i.e.,

$$np(1+q) - np > \kappa_1\sigma_1 + \kappa_2\sigma_2, \tag{2}$$

This gives,

$$n > \frac{\left(\kappa_1\sqrt{1-p} + \kappa_2\sqrt{(1+q)\left(1 - p(1+q)\right)}\right)^2}{pq^2}. \tag{3}$$

□

In the special case, when both $p, q \ll 1$, the numerator of Equation (3) is approximately equal to $(\kappa_1 + \kappa_2)^2$, and one needs at least $\frac{(\kappa_1+\kappa_2)^2}{pq^2}$ many samples to perform the test.

Table 1 gives the sample complexity, false positive and negative rates and the success probability for some selected values of k_1 and k_2.

Since $0.6915 > 0.5$ is a reasonably good success probability, $O(\frac{1}{pq^2})$ many samples are enough to mount a distinguisher and this threshold is indeed used as a benchmark to compare the data complexities of different distinguishing attacks in practice.

A question now remains in the above distinguisher about how to choose the cut-off point c. Theoretically, any value between the limits given in (1) will

Table 1 Sample complexity and success probability for distinguishers.

κ_1	κ_2	Number of samples	α	β	Success probability$(1-\alpha)$ %
0.5	0.5	$1/pq^2$	0.3085	0.3085	69.15%
1	0.5	$2.25/pq^2$	0.1587	0.3085	84.13%
2	0.5	$6.25/pq^2$	0.0228	0.3085	97.72%
0.5	1	$2.25/pq^2$	0.3085	0.1587	69.15%
1	1	$4/pq^2$	0.1587	0.1587	84.13%
2	1	$9/pq^2$	0.0228	0.1587	97.72%
0.5	2	$6.25/pq^2$	0.3085	0.0228	69.15%
1	2	$9/pq^2$	0.1587	0.0228	84.13%
2	2	$16/pq^2$	0.0228	0.0228	97.72%

work. As we move along from the left hand upper bound, the false positive rate decreases and the false negative rate increases. A popular idea in statistical literature is to prefix the maximum value of the false positive rate, which is called the *level* and consider the tests with a certain level. If we adopt the same idea here, we should choose $c = np(1+q) - \kappa_1\sigma_1$. Then the false positive error will be exactly equal to α and the false negative error will be

$$1 - \Phi\left(\frac{npq - \kappa_1\sigma_1}{\sigma_2}\right) \leq \beta.$$

We would also like to mention that the above test procedure is asymptotically equivalent to the most powerful test with level α [9] (asymptotically equivalent because it uses normal approximation to determine the critical values), which can be constructed with a sample of n iid Bernoulli observations. For the sake of completeness, the algorithm of this distinguisher is given below:

```
Algorithm Distinguisher 2
        $X_r \leftarrow$ Indicator of occurrences of event $e$ in the $r$-th sample, $r = 1, \ldots, n$;
        $X = \sum_{r=1}^{n} X_r$;
        if $X < np(1+q) - \kappa_1\sigma_1$
            Infer $H_0$ is false;
        else
            Infer $H_0$ is true;
```

2.3 Relative Entropy Between Distributions

This analysis appeared in [19, Appendix A]. The relative entropy between two discrete probability distributions $P(\cdot)$ and $Q(\cdot)$ is given by the Kullback-Leibler divergence [17]

$$D_{KL}(P||Q) := \sum_x P(x) \log_2 \frac{P(x)}{Q(x)}, \tag{4}$$

where x runs over all the sample points. Note that this can also be written as

$$D_{KL}(P||Q) = E_P \left[\log_2 \frac{P(X)}{Q(X)} \right],$$

where $X \sim P$. We have the following straight-forward result.

Proposition 1 *For the above-mentioned single event e with probabilities p and $p(1+q)$ in two different distributions $P(\cdot)$ and $Q(\cdot)$, the relative entropy is approximately equal to pq^2, for small p, q.*

Proof We have

$$\begin{aligned} D_{KL}(P||Q) &= p \log_2 \left[\frac{p}{p(1+q)} \right] \\ &+ (1-p) \log_2 \left[\frac{1-p}{1-p(1+q)} \right] \\ &= p \log_2 \left[1 - \frac{q}{1+q} \right] \\ &+ (1-p) \log_2 \left[1 + \frac{pq}{1-p(1+q)} \right] \\ &\approx -p \left(\frac{q}{1+q} \right) + (1-p) \left(\frac{pq}{1-p(1+q)} \right) \\ &\approx pq^2. \end{aligned}$$

Similarly, for small p, q, we also have,

$$\begin{aligned} D_{KL}(Q||P) &= p(1+q) \log_2 \left[\frac{p(1+q)}{p} \right] \\ &+ (1-p(1+q)) \log_2 \left[\frac{1-p(1+q)}{1-p} \right] \\ &= p(1+q) \log_2(1+q) - (1-p(1+q)) \log_2 \left[1 + \frac{pq}{1-p(1+q)} \right] \\ &\approx p(1+q)q - (1-p(1+q)) \left(\frac{pq}{1-p(1+q)} \right) \\ &\approx pq^2. \end{aligned}$$

□

Also, the following small technical result directly follows from the definition in Equation (4).

Proposition 2 *If P, Q are two distributions defined over the domain A and P', Q' are two other distributions defined over the domain B, then it can be shown that the overall relative entropy of the joint distributions (considering independence of the corresponding random variables over the two domains) PP' and QQ' is given by $D_{KL}(PP'||QQ') = D_{KL}(P||Q) + D_{KL}(P'||Q')$.*

Now we can state the following result.

Lemma 1 *For n independent occurrences of the event e with probabilities p and $p(1+q)$ in two different distributions $P(\cdot)$ and $Q(\cdot)$, the relative entropy is approximately equal to npq^2, for small p, q.*

Proof Applying Proposition 2 to n samples from the same distribution as in Proposition 1, we get the result. □

Now, according to [12,7], we have the following result connecting the relative entropy to the false positive and negative rates.

Lemma 2 *Suppose D_n is an unknown discrete distribution and P_n and Q_n are two known distributions. Suppose we have a test (may be randomized) for*

$$H_0 : D_n = P_n \quad vs \;\; H_1 : D_n = Q_n \,,$$

based on $\boldsymbol{X} := (X_1, X_2, \ldots, X_n)$, a sample from the distribution D_n, with false positive rate (α) and false negative rate (β). Then we have the following bound

$$D_{KL}(Q_n||P_n) \geq \beta \log_2 \frac{\beta}{1-\alpha} + (1-\beta)\log_2 \frac{1-\beta}{\alpha}. \tag{5}$$

Proof Proof is given at the Appendix A.

Now, combining Lemma 1 and Lemma 2, we have the following result on the data complexity.

Theorem 3 *For n independent occurrences of the event e with probabilities p and $p(1+q)$ in two different distributions, the sample complexity of a distinguisher with false positive and negative rates α and β is given by*

$$n \geq \frac{1}{pq^2}\left(\beta \log_2 \frac{\beta}{1-\alpha} + (1-\beta)\log_2 \frac{1-\beta}{\alpha}\right),$$

for small p, q.

The equality may hold true only for the *Neymann-Pearson Test* [21], which is the optimal test, i.e., given a fixed level this test maximizes the power. This test is described by the *Fundamental Neymann-Pearson Lemma* [21].

Lemma 3 (Neymann-Pearson Lemma) *Suppose we have $\boldsymbol{X} := (X_1, \ldots, X_n) \sim D_n$, where D_n is an unknown discrete distribution. We are to test the hypothesis $H_0 : D_n = P_n$ versus the alternative $H_1 : D_n = Q_n$. Suppose $\mathcal{S}$ be the set of all possible values that X_i's can take. Then take any arbitrary constant k and consider any test function $\phi : \mathcal{S} \longrightarrow [0, 1]$ satisfying the following conditions:*

$$\phi(\boldsymbol{x}) = 0, if \ \frac{P_n[(x_1, \ldots, x_n)]}{Q_n[(x_1, \ldots, x_n)]} > k,$$

$$= 1, if \ \frac{P_n[(x_1, \ldots, x_n)]}{Q_n[(x_1, \ldots, x_n)]} < k.$$

Define $\alpha = E_{H_0}[\phi(\boldsymbol{X})]$, and $\beta = 1 - E_{H_1}[\phi(\boldsymbol{X})]$. Then for any other test (may be randomized) for the above hypothesis with error probabilities α' and β', we have

$$\alpha' \leq \alpha \Rightarrow \beta \leq \beta'.$$

In other words the test satisfying the conditions stated is the most powerful level α test.

In our context, P_n is the distribution of n *i.i.d.* Bernoulli trials with success probability $p(1 + q)$, and Q_n is the same with success probability p. Here *false positive* means that the test sequence is actually from the stream cipher, but we decide it to be random and *false negative* means that the test sequence is actually random, but we decide it to be from the stream cipher. Here, the *Neymann-Pearson Test* reduces to be the test discussed in Theorem 2 for large sample (the test statistic is the same for both the tests but the cut-off points differ, as we have used the normal approximation to find the critical values in Theorem 2). Though *Fundamental Neymann-Pearson Lemma* gives us the optimum test for level α [9], the exact values of the optimum error probabilities for this test is difficult to find in general case. So, in that case we use some approximation techniques, two of which are discussed in Theorem 2 and Theorem 3. If we allow both the error rates taken to be equal as in Theorem 3, i.e. $\alpha = \beta$, the distinguishing complexity bound reduces to

$$n \geq \left(\frac{1}{pq^2} \right) \cdot (1 - 2\alpha) \log_2 \frac{1 - \alpha}{\alpha}.$$

Thus, for a given false positive or negative rate α $(= \beta)$, one needs roughly $O(1/pq^2)$ many samples to perform the distinguishing test. In particular, $n \geq 1/pq^2$ signifies $\alpha \approx 0.2227$, i.e., a success probability of approximately 0.7773. Since $0.7773 > 0.5$ is a reasonably good success probability, $O(1/pq^2)$ many samples are considered enough to reliably apply the distinguisher.

For a discussion on the distinguisher algorithm, see Remark 2.

2.4 Asymptotic Approach I: Chernoff-Stein Lemma

Another method to find the expression for the error probabilities for the optimum test is to use the asymptotic analysis given by *Chernoff-Stein Lemma* [10]. This approach has been used by [24] to mount distinguishing attack on the stream-cipher HC-128 [25].

Lemma 4 (Chernoff-Stein Lemma) *Suppose we have* $X_1, \ldots, X_n \overset{i.i.d.}{\sim} D$*, where* D *is unknown. We are to test the hypothesis* $H_0 : D = P$ *versus the alternative* $H_1 : D = Q$*, where* P *and* Q *are two known distributions. Suppose* χ *be the set of all possible values that* X_i*'s can take. Suppose,*P_n *and* Q_n *are the joint distributions of* $(X_1, \ldots, X_n)$ *under the null and the alternative respectively. Let us fix* $0 < \alpha < 0.5$*. Define,*

$$\beta_{n,\alpha} := \min \left\{ \beta | \mathcal{R} \subset \chi^n, P_n[\mathcal{R}] < \alpha, \beta = 1 - Q_n[\mathcal{R}] \right\},$$

In other words, $\beta_{n,\alpha}$ *is the least false negative error probability attainable for level* α *non-randomized tests. Then*

$$\lim_{n \to \infty} \frac{\log_2 \beta_{n,\alpha}}{n} = -D_{KL}(P||Q).$$

Proof Proof is given at the Appendix B.

The above lemma states that, whatever be the pre-specified false positive error, asymptotically the best possible false negative error is $2^{-nD_{KL}(P||Q)}$. Suppose now that we fix the false positive error at α and want false negative error to be β. Then the approximate sample size we need is $n \approx -\log_2(\beta)/D_{KL}(P||Q)$. Therefore, combining Lemma 1 and Lemma 4, we have the following result on the data complexity.

Theorem 4 *For* n *independent occurrences of the event* e *with probabilities* p *and* $p(1 + q)$ *in two different distributions, the sample complexity of a distinguisher with false positive and negative rates* α *and* β *is given by*

$$n > -\frac{1}{pq^2} \log_2(\beta),$$

for small $p, q,$ *and small* β*.*

Remark 1 Note that, in Theorem 4, β is required to be small, as Lemma 4 is an asymptotic result. So the best possible false negative rate is well approximated for large n and consequently for small β.

For a discussion on the distinguisher algorithm, see Remark 2.

2.5 Asymptotic Approach II: Chernoff Information

Apart from *Chernoff-Stein Lemma*, another way to approximate the error probabilities asymptotically is the *Chernoff Information* [10]. This approach has been used in [11] to mount distinguishing attacks on SOBER-t16 and t32 stream ciphers. The method is developed from a Bayesian perspective rather than the classical hypothesis testing perspective which gives rise to the *Chernoff-Stein Lemma.*

Recall the set-up in Lemma 4. Consider a rejection region $\mathcal{R} \subset \chi^n$ such that

$$\Pr(\mathcal{R}|H_0) = \alpha_n, \;\; \Pr(\mathcal{R}^c|H_1) = \beta_n,$$

i.e, α_n, β_n are Type-1 and Type-2 error probabilities respectively. Suppose now that the hypotheses have some prior probabilities attached on them, H_0 has prior probability π_0 and H_1 has π_1. So, the overall probability of error becomes,

$$P_n^e := \pi_0 \alpha_n + \pi_1 \beta_n.$$

Our target is to choose $\mathcal{R}$ such that P_n^e is minimized. The following result quantifies this best achievable error rate in terms of the sample size.

Lemma 5 *Suppose we have $X_1, \ldots, X_n \overset{i.i.d.}{\sim} D$, where D is unknown. We are to test the hypothesis $H_0 : D = P_0$ versus the alternative $H_1 : D = P_1$, where P_0 and P_1 are two known distributions. Suppose χ be the set of all possible values that X_i's can take. Suppose,$P_{0,n}$ and $P_{1,n}$ are the joint distributions of $(X_1, \ldots, X_n)$ under the null and the alternative respectively. π_0 and π_1 be prior probabilities on P_0 and P_1 respectively. For any $\mathcal{R} \subset \chi^n$, define*

$$\alpha_n := \Pr(\mathcal{R}|H_0) = P_{0,n}(\mathcal{R}), \;\; \beta_n := \Pr(\mathcal{R}^c|H_1) = P_{1,n}(\mathcal{R}^c),$$

and

$$P_n^e := \pi_0 \alpha_n + \pi_1 \beta_n.$$

Then,

$$\lim_{n \longrightarrow \infty} \frac{1}{n} \log_2 (\inf_{\mathcal{R} \subset \chi^n} P_n^e) = -D^*,$$

where $D^* := D_{KL}(P_{\lambda^*} || P_0)$, with

$$P_\lambda(x) = \frac{P_0^{1-\lambda}(x) P_1^\lambda(x)}{\sum_{x \in \chi} P_0^{1-\lambda}(x) P_1^\lambda(x)}, \; \forall \, 0 \leq \lambda \leq 1.$$

and D^* is such that

$$D^* = D_{KL}(P_{\lambda^*} || P_0) = D_{KL}(P_{\lambda^*} || P_1).$$

Moreover, in this case, we have

$$D^* = -\inf_{0<\lambda<1} \log_2(\sum_{x\in\chi} P_0^{1-\lambda}(x)P_1^{\lambda}(x)).$$

This quantity D^* is defined as the *Chernoff Information* between P_0 and P_1 and is denoted by $C(P_0, P_1)$, i.e.,

$$C(P_0, P_1) := -\inf_{0<\lambda<1} \log_2(\sum_{x\in\chi} P_0^{1-\lambda}(x)P_1^{\lambda}(x)).$$

Proof of this lemma is given in [10]. Note that, the result in Lemma 5 is independent of the prior probabilities and therefore by taking $\pi_0 = \pi_1 = \frac{1}{2}$, we get the following result.

Corollary 1 *With the same set-up as described in Theorem 5, we have*

$$\lim_{n\longrightarrow\infty} \frac{1}{n} \log_2(\inf_{\mathcal{R}\subset\chi^n} (\alpha_n + \beta_n)) = C(P_0, P_1),$$

i.e.,

$$\lim_{n\longrightarrow\infty} \frac{1}{n} \log_2(1 - \sup_{\mathcal{R}\subset\chi^n} \mathit{Advantage}) = C(P_0, P_1).$$

Corollary 1 says that if we try to distinguish between distributions P and Q, then asymptotically the best possible advantage is $1 - 2^{-nC(P,Q)}$. Therefore, we can write the following result.

Theorem 5 *For n independent occurrences of the event e with probabilities p and $p(1+q)$ in two different distributions, the asymptotic sample complexity of a distinguisher with false positive and negative rates α and β is given by*

$$n > -\frac{1}{C(\mathcal{B}er(p(1+q)), \mathcal{B}er(p))} \log_2(\alpha + \beta),$$

for small α, β.

In general, finding an explicit analytic expression for $C(P,Q)$ is difficult. So, numerical methods are to be employed to find the *Chernoff Information* for specific problems. Approximate algebraic expressions are available [4], when the distributions P and Q are very close, which is as follows.

Proposition 3 *If the distributions P and Q are very close, then*

$$C(P,Q) \approx \frac{1}{8\ln 2} \sum_{x\in\chi} \frac{(P(x) - Q(x))^2}{P(x)} \approx \frac{1}{8\ln 2} \sum_{x\in\chi} \frac{(P(x) - Q(x))^2}{Q(x)}$$

Using Proposition 3, we can say that, if $q \ll 1$, then

$$\begin{aligned} C(\mathcal{B}er(p(1+q)), \mathcal{B}er(p)) &\approx \frac{1}{8 \ln 2}(\frac{(p(1+q)-p)^2}{p} + \frac{(1-p(1+q)-1+p)^2}{1-p} \\ &= \frac{1}{8 \ln 2} p^2 q^2 (\frac{1}{p} + \frac{1}{1-p}) \\ &= \frac{1}{8 \ln 2} \frac{p}{1-p} q^2. \end{aligned}$$

Thus, for small q, α, β, number samples needed to distinguish with error rates α and β, is

$$n > -\frac{8(1-p)\ln 2}{pq^2} \log_2(\alpha + \beta).$$

Now we make the following remark about the distinguisher algorithms corresponding to the approaches of Section 2.3, 2.4 and 2.5.

Remark 2 It is to be noted that the optimal test based on the Neymann-Pearson Lemma for our case is of the form:

$$\text{If } X < c, \text{ Infer } H_0 \text{ to be false.}$$

$$\text{If } X > c, \text{ Infer } H_0 \text{ to be true.}$$

$$\text{If } X = c, \text{ Infer } H_0 \text{ to be false with prob. } p \text{ and true with prob. } 1-p,$$

where p, c is chosen such that

$$\Pr(X < c | H_0) + p \Pr(X = c | H_0) = \alpha,$$

and X is as usual total number of occurrences of event e in the total sample. Computation of these constants involves computing a large number of very small binomial probabilities, which in most of the cases is very difficult to perform, even for moderately large value of n. Therefore, for determining the threshold for the decisions we actually use the normal approximations which take us back to the distinguishing attack discussed in section 2.2. Thus algorithm-wise both attacks are same, but we get two different bounds to measure the complexity. The approaches with Stein's Lemma and Chernoff Information also use the same algorithm, but just give different bounds on the data complexity by considering the asymptotics.

2.6 Comparison amongst the Above Approaches

So far we have discussed four bounds on data complexities obtained from Theorem 1, Theorem 2, Theorem 3, Theorem 4 and Theorem 5. We now compare these bounds for small values of p, q and small failure rate α (i.e., both the error rate is α). We first exclude the bound from Theorem 1 from this comparison, as it doesn't consider the optimum test and therefore expected to give much larger estimate of data complexity.

We observe that the bound obtained from Theorem 2 is relatively larger than other two bounds and the "actual complexity"(which is defined as the minimum over the number of samples needed to distinguish between two distributions by all possible test procedures with given false positive and false negative rates). This phenomenon occurs as Theorem 2 considers only non-randomized tests to distinguish between the two distributions and the critical values are derived under normal approximations.

On the other hand, the bound given by Theorem 3 is the smallest as it considers all the randomized tests. Moreover, equality holds in this bound only under some strict conditions which will imply the equality condition in *Jensen's inequality*. So, the actual complexity is always somewhat bigger than this bound.

Theorem 4 and Theorem 5 also consider all non-randomized tests. However, these bounds are derived from an asymptotic result of minimum possible false negative error and maximum possible advantage, and hence will be greater than the bound from Theorem 3. Theorem 5 gives a crude bound unless the errors are very small. It is also easy to prove that the bound obtained from Theorem 4 is larger than that from Theorem 3 for fixed false negative and false positive error rate (both equal to α), as

$$\begin{aligned}\alpha \in \left(0, \frac{1}{2}\right) &\Rightarrow (1-\alpha)^{1-2\alpha} < \alpha^{-2\alpha} \\ &\Rightarrow (1-2\alpha)(\log_2(1-\alpha) - \log_2 \alpha) < -\log_2 \alpha \\ &\Rightarrow (1-2\alpha)\log_2\left(\frac{1-\alpha}{\alpha}\right) < -\log_2 \alpha.\end{aligned}$$

Thus, the bound obtained from Theorem 4 lies in between those given by Theorem 2 and Theorem 3.

The actual complexity and the bounds obtain from Theorem 4 and Theorem 5 lie between the other two bounds, and they become very close to each other as the error rate becomes small. Now, it is natural to ask: *which bound to use for estimating the data complexity?* From the context, it is clear that for small error rates, the bound from Theorem 4 or Theorem 5 (if the errors are very small) should be preferred. Otherwise, it is better to use the bound from Theorem 2, as this bound is greater than the actual complexity.

Table 2 Sample complexity and success probability for the distinguisher of RC4 based on the bias of the second keystream byte to the value 0, with $p = \frac{1}{256}, q \approx 1$, $C \approx 0.0005$.

		Sample Complexity (as exponents of 2) using			
$\alpha = \beta$	Success Prob. $(1-\alpha)\%$	Theorem 2	Theorem 3	Theorem 4	Theorem 5
0.3	70	8.67	6.97	8.79	10.52
0.15	85	10.63	8.80	9.45	11.76
0.1	90	11.25	9.34	9.73	12.18
0.05	95	11.97	9.93	10.11	12.70
0.01	99	12.97	10.70	10.73	13.46

Table 3 Sample complexity and success probability for the distinguisher of RC4 based on the bias of the fourth keystream byte to the value 0, with $p = \frac{1}{256}, q \approx 0.005106232$, $C \approx 2 \times 10^{-8}$.

		Sample Complexity (as exponents of 2) using			
$\alpha = \beta$	Success Prob. $(1-\alpha)\%$	Theorem 2	Theorem 3	Theorem 4	Theorem 5
0.3	70	23.36	22.19	24.02	25.14
0.15	85	25.33	24.04	24.68	26.37
0.1	90	25.94	24.57	24.96	26.79
0.05	95	26.66	25.16	25.34	27.31
0.01	99	27.66	25.93	25.96	28.07

In Table 2 and 3, we present estimated data complexities (as exponents of 2, i.e., the presented values are the logarithms of the data complexities w.r.t. base 2) of two different distinguishers for RC4, using all the methods discussed above. For Table 2, the value of Chernoff Information C is numerically computed to be approx. 0.0005, as demonstrated in Fig. 1.

A related question is to ask what is the advantage of this distinguisher. It is clear from the context that

$$\begin{aligned}\text{Advantage} &= \Pr(\text{Inferred } H_0 \text{is true}|H_0) - \Pr(\text{Inferred} H_0 \text{is true}|H_1)\\ &= 1 - \text{false positive error} - \text{false negative error}\\ &= 1 - \alpha - \beta.\end{aligned}$$

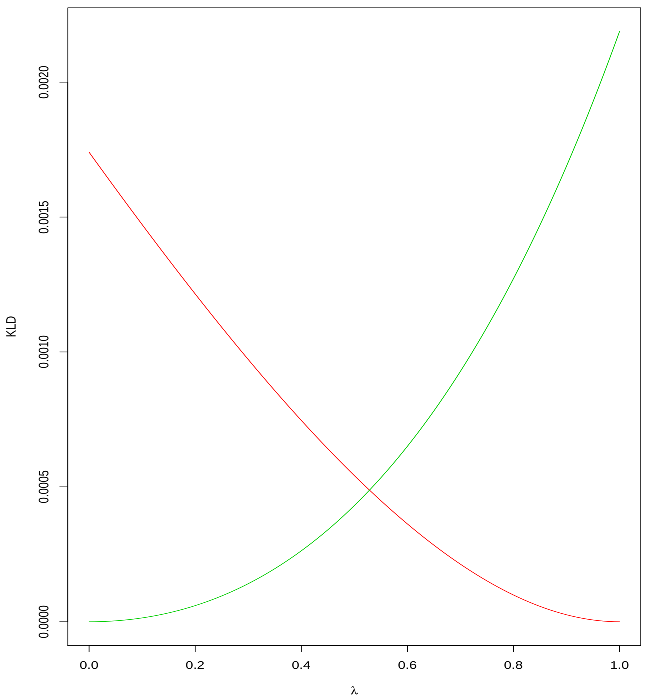

Fig. 1 Chernoff Information, $D_{KL}(P_\lambda||P_0)$(bottom-left to top-right), $D_{KL}(P_\lambda||P_1)$(top-left to bottom-right) for the distribution of the second keystream byte of RC4.

2.7 Other Related Works

In the above discussion, we have considered distinguishing tests which minimize the false negative error for a given false positive error level. The work [3] considered another paradigm considering the tests minimizing the average of both kind of error rates, and derived the data complexity of the optimal distinguishing test in that scenario, which is also of $O(1/pq^2)$, for small p, q. However, we note that joint minimization of both types of errors is an unusual approach in hypothesis testing framework. In practical scenario, often one would like to strictly bound a particular type of error. Thus, it is more pragmatic to fix one error and minimize the other. Moreover, the main focus of [3] is block cipher cryptanalysis and here we concentrate on stream ciphers.

The work [2] considers data complexity of a particular differential cryptanalysis with a set of 2^m sequences, where only one of them verifies the alternative hypothesis and all others verify the null hypothesis. The scenario we consider is completely different and hence we do not discuss the work [2] here.

Another work [8] attempted to give accurate estimates of the data complexity and success probability of differential and linear cryptanalysis of block ciphers under two different scenarios, namely, (i) when the probability of acceptance of a wrong key is fixed, and (ii) when the number of candidate solutions is fixed.

A recent work [22] that has been carried out simultaneously with and independently of our current work, takes a detailed look at the error in normal approximations and points out several limitations in applying these approximations to block cipher cryptanalysis. A more recent work [23] (by the same authors as [22]) derives rigorous upper bounds on the data complexity (i.e., the no. of plaintext-ciphertext pairs) required to achieve at least a pre-specified success probability (for key recovery of a block cipher) and at least a pre-specified advantage (if the advantage is a, then the number of false alarms is a fraction 2^{-a} of the number of possible values of the sub-key which is the target of the attack). We emphasize again that our motivation is completely different, i.e., to analyze the gap between the data complexity estimates of distinguishing attacks and message recovery attacks in the context of stream ciphers.

3 Data Complexity of Message Recovery Attacks using Sample Mode Approach

Now we turn to Message Recovery Attack under the *Broadcast* scenario for stream ciphers. We shall consider two approaches: one is a simple message recovery attack exploiting the largest bias in the keystream byte distribution and the other is maximum likelihood estimation. The first one, called Sample Mode approach, is discussed in this section and the second one, called Bayesian approach, will be discussed in the next section.

We consider again the single byte-bias attack. Suppose m be the mode of the keystream distribution $\mathcal{P}$. We shall assume that the distribution $\mathcal{P}$ is unimodal, since the approach will fail if the distribution is multimodal. Suppose M is the secret message and $Z_1, \ldots, Z_n \overset{i.i.d}{\sim} \mathcal{P}$ be the keys for n broadcasts. We observe the ciphertexts

$$C_i = (M + Z_i) \bmod N, \; \forall\, i = 1, \ldots, N.$$

Since m is the mode, we expect it to occur more frequently in the i.i.d. sample $Z_1, \ldots, Z_n$ and hence we expect $(M + m) \bmod N$ to be most frequent in the ciphertext sample $C_1, \ldots, C_n$. Therefore, we estimate M by $\hat{M} = (Mode(C_1, \ldots, C_n) - m) \bmod N$. Below we mention the algorithm for the message recovery attack using this approach.

Algorithm: Message recovery using sample mode

```
    $N_i \leftarrow$ Number of occurrences of byte $i$ in the ciphertext samples, $i = 0, \ldots, N-1$;
    $Mo \leftarrow \arg\max_{0 \le i \le N-1} N_i$;
    $m \leftarrow$ Mode of the keystream distribution;
        Estimate of unknown plain text byte $\leftarrow Mo - m$;
```

In this case the probability of success is

$$\begin{aligned}
&\Pr(\hat{M} = M)\\
&= \Pr[(Mode(C_1, \ldots, C_n) - m) \bmod N = M]\\
&= \Pr(Mode(Z_1, \ldots, Z_n) = m).
\end{aligned}$$

So, computing the probability of success for this attack boils down to the problem of finding the probability of sample mode being equal to the population mode for an *i.i.d.* sample of size n from distribution $\mathcal{P}$. Suppose Y_k be the frequency of k in the sample $\mathbf{Z} = (Z_1, \ldots, Z_n)$, $\forall\, k = 0, \ldots, N-1$. Then

$$\mathbf{Y} := (Y_0, \ldots, Y_{N-1}) \sim Multinomial(n; p_0, \ldots, p_{N-1});$$

where $p_i = \Pr[Z_1 = i]$, $\forall\ i = 0, \ldots, N-1$. For simplicity of notation, we assume $m = 0$ (however, the result holds for any mode). So,

$$\Pr(Mode(Z_1, \ldots, Z_n) = 0) = \Pr(Y_0 > Y_k,\ \forall\, k = 0, \ldots, N-1). \tag{6}$$

By *Law of Large Numbers* [15], we have,

$$\frac{1}{n}(Y_0, \ldots, Y_{N-1}) \xrightarrow{P} (p_0, \ldots, p_{N-1});$$

i.e.

$$\frac{1}{n}(Y_0 - Y_1, \ldots, Y_0 - Y_{N-1}) \xrightarrow{P} (p_0 - p_1, \ldots, p_0 - p_{N-1}). \tag{7}$$

the distribution $\mathcal{P}$ is unimodal, we have $p_0 > p_k$, $\forall\ k = 1, \ldots, N-1$. Therefore, using (7) we get,

$$\frac{Y_0 - Y_k}{n} \xrightarrow{P} p_0 - p_k > 0 \ \Rightarrow\ \Pr(Y_0 > Y_k) \longrightarrow 1,\ \ \forall\ k = 1, \ldots, N-1. \tag{8}$$

Using (6) and (8), we can write,

$$\begin{aligned}\Pr(Mode(Z_1,\ldots,Z_n)\neq 0) &= \Pr(\exists\ 0\leq k\leq N-1\ s.t.\ Y_0\leq Y_k)\\ &\leq \sum_{k=1}^{N-1}\Pr(Y_0\leq Y_k)\longrightarrow 0.\end{aligned}$$

Therefore, $\Pr(\hat{M}=M)\longrightarrow 1$, which implies $\hat{M}\xrightarrow{P} M$, as $n\to\infty$, i.e., the estimator is at least consistent [9].

Again, using *Central Limit Theorem* [15] for multinomial distribution, we can write

$$\mathbf{Y}\sim AN(n\mathbf{p}, n(diag(\mathbf{p})-\mathbf{p}\mathbf{p}^T));$$

i.e.,

$$n^{-1}\mathbf{Y}\sim AN(\mathbf{p}, n^{-1}(diag(\mathbf{p})-\mathbf{p}\mathbf{p}^T)),$$

where $\mathbf{p}:=(p_0,\ldots,p_{N-1})^T$. So, for large n we have by normal approximation

$$\Pr(Y_0>Y_k,\ \forall\ k=0,\ldots,N-1)\approx\Pr(U_0>U_k,\ \forall\ k=0,\ldots,N-1); \quad (9)$$

where $\mathbf{U}:=(U_0,\ldots,U_{N-1})^T\sim\mathcal{N}_N(\mathbf{p},\frac{1}{n}(diag(\mathbf{p})-\mathbf{p}\mathbf{p}^T))$.

But evaluation of this probability will lead us to an expression containing multiple integrals, and getting a closed form expression is almost impossible in general. Therefore, we consider two different special cases in the next two subsections and go through some further approximations.

3.1 Second Highest Probability far from Lower Ones

In this case, we consider the situation where the second highest probability in the distribution $\mathcal{P}$ is distinguishably apart from the other lower probabilities in that distribution, i.e., if $p_{(0)}\leq p_{(1)}\leq\cdots\leq p_{(N-1)}$ be the probabilities $p_0,\ldots,p_{N-1}$ in increasing order, then $p_{(N-2)}-p_{(N-3)}\neq 0$. In this case we can have a simplified approximation for the data complexity using the following lemma.

Lemma 6 *Let $\boldsymbol{\mu}:=(\mu_0,\ldots,\mu_{N-1})^T\in\mathbb{R}^{N-1}$ with $\mu_s\geq\mu_i,\forall\ i=0,\ldots,N-1$, and $\Sigma=((\sigma_{ij}))_{i,j=0}^{N-1}$ be a positive semi-definite matrix. Suppose,*

$$\boldsymbol{W}:=(W_0,\ldots,W_{N-1})^T\sim\mathcal{N}_N(\mu,\frac{1}{n}\Sigma).$$

Then,

$$P\big(\max_{0\leq i\leq(N-1)}W_i\neq W_s\big)\approx\Phi(-\sqrt{n}\delta),$$

as $n \longrightarrow \infty$, *where* $\delta := \delta_j = \min_{i \neq s} \delta_i$, *and*

$$\delta_i := \frac{\mu_s - \mu_i}{\sqrt{\sigma_{ss} + \sigma_{ii} - 2\sigma_{si}}}, \ \forall \, i \neq s,$$

provided

$$\sigma_{ss} + \sigma_{ii} - 2\sigma_{si} > 0, \ \ \forall \, i \neq s,$$

and $\delta_j < \delta_i$, $\forall i \neq j$.

Proof Proof is given at the Appendix C.

Using Lemma 6 we can arrive at the following theorem.

Theorem 6 *If* $\mathcal{P}$ *is the distribution of the keystream bytes on the space* $\mathcal{M}$*, with population mode* 0*, and the second and third highest probabilities of* $\mathcal{P}$*,i.e.* $p_{(N-2)}$ *and* $p_{(N-3)}$ *are distinguishably apart, then the data complexity of the message recovery attack with failure probability at most* α*, using sample mode approach, is given by*

$$n > \left(\frac{\Phi^{-1}(1-\alpha)}{\delta} \right)^2, \tag{10}$$

where

$$\delta_k := \frac{p_0 - p_k}{\sqrt{p_0 + p_k - (p_0 - p_k)^2}} > 0, \ \forall \, k = 1, \ldots, N-1.$$

and $\delta := \min\{\delta_k : k = 1, \ldots, N-1\}$.

Proof With the notations in (9), we shall use Lemma 6 with $\boldsymbol{W} = \boldsymbol{U}, \boldsymbol{\mu} = \boldsymbol{p}$ and $\Sigma = diag(\mathbf{p}) - \mathbf{p}\mathbf{p}^T$. Then δ_i becomes as defined in the statement of Theorem 6. Suppose, $p_{(N-2)} = p_j$, where $1 \leq j \leq N-1$. It is easy to notice that, $p_k \leq p_l$ implies $\delta_k \geq \delta_l$, $\forall \ 1 \leq k, l \leq N-1$. Therefore, δ_k is minimum for $k = j$. Also, note that the condition $p_{(N-2)}$ and $p_{(n-3)}$ are distinguishably apart guarantees that $\delta = \delta_j < \delta_k$, $\forall k \neq j$. Therefore, using Lemma 6 we have, So, $\Pr(\max_{0 \leq k \leq N-1} U_k \neq U_0) \approx \Phi(-\sqrt{n}\delta)$. Hence to get success probability at least $1 - \alpha$, we should have $\alpha \geq \Phi(-\sqrt{n}\delta)$. Hence, the sample size we need is

$$n \geq \left(\frac{\Phi^{-1}(1-\alpha)}{\delta} \right)^2.$$

But the above approximations performs miserably if all the probabilities in the p.m.f. $\mathcal{P}$, except the highest one, are very close to each other since then the limit above converges very slowly. We shall consider this situation in the next case.

3.2 Almost-uniform Except the Highest Probability-point

Here, suppose all the probabilities in the p.m.f. $\mathcal{P}$, except the highest one i.e. p_0, are equal to r. Hence, $p_0+(N-1)r=1$. Continuing with the notation in (9) let us define $\mathbf{V} = (V_1, \ldots, V_{N-1})^T := (U_0 - U_1, \ldots, U_0 - U_{N-1})^T$. Therefore, we have

$$\mathbf{V} \sim \mathcal{N}_{N-1}((p_0 - r)\mathbf{1}, \mathbf{\Sigma})$$

where $\mathbf{1}$ is the $(N-1)$-dimensional column vector with all entries equal to 1, and $\mathbf{\Sigma}$ is a positive-definite matrix of dimension $(N-1) \times (N-1)$ with all diagonal entries equal to σ^2, where

$$\begin{aligned}\sigma^2 &:= Var(U_0 - U_1)\\ &= Var(U_0) + Var(U_1) - 2Cov(U_0, U_1)\\ &= n^{-1}(p_0(1-p_0) + r(1-r) + 2p_0 r)\\ &= n^{-1}(p_0 + r - (p_0 - r)^2)\end{aligned}$$

and all non-diagonal entries equal to $\rho\sigma^2$, where

$$\begin{aligned}\rho\sigma^2 &:= Cov(U_0 - U_1, U_0 - U_2)\\ &= Var(U_0) - Cov(U_0, U_1) - Cov(U_0, U_2) + Cov(U_1, U_2)\\ &= n^{-1}(p_0(1-p_0) + 2p_0 r - r^2)\\ &= n^{-1}(p_0 - (p_0 - r)^2) > 0.\end{aligned}$$

Here, σ^2 and ρ are the common variance and common correlation coefficient of the elements in $\mathbf{V}$. Hence, $\mathbf{V}$ has the equicorrelation structure and the correlation coefficient $\rho > 0$. Now consider

$$W_0, \ldots, W_{N-1} \overset{i.i.d.}{\sim} \mathcal{N}(0,1);$$

and define

$$S_k := (p_0 - r) + \sigma(\sqrt{\rho}W_0 + \sqrt{1-\rho}W_k), \ \ \forall\, k = 1, \ldots, N-1.$$

Then, clearly $S_k \sim \mathcal{N}(p_0 - r, \sigma^2)$, $\forall\ k = 1, \ldots, N-1$ and $Cov(S_i, S_k) = \rho\sigma^2$, $\forall\, 1 \leq i \neq k \leq (N-1)$ which implies

$$\mathbf{S} := (S_1, \ldots, S_{N-1})^T \overset{\mathcal{D}}{=} \mathbf{V}.$$

Hence, we have

$$
\begin{aligned}
&\Pr(U_0 > U_k,\ \forall\, k = 0, \ldots, N-1)\\
&= \Pr(V_k > 0,\ \forall\, k = 1, \ldots, N-1)\\
&= \Pr(S_k > 0,\ \forall\, k = 1, \ldots, N-1)\\
&= \Pr(\sqrt{\rho}W_0 + \sqrt{1-\rho}W_k > -\frac{(p_0 - r)}{\sigma},\ \forall\, k = 1, \ldots, N-1)\\
&= \Pr(W_k > -\frac{(p_0 - r)}{\sigma\sqrt{1-\rho}} - \frac{\sqrt{\rho}}{\sqrt{1-\rho}}W_0,\ \forall\, k = 1, \ldots, N-1)\\
\\
&= \Pr(W_{(1)} > T_0)
\end{aligned}
$$

where we define $W_{(1)} := min\,\{W_k : 1 \leq k \leq N-1\}$ and

$$T_0 := \frac{(p_0 - r)}{\sigma\sqrt{1-\rho}} - \frac{\sqrt{\rho}}{\sqrt{1-\rho}}W_0 \sim \mathcal{N}(-\sqrt{n}\gamma, \sigma_0^2)$$

where

$$\gamma := \frac{p_0 - r}{\sqrt{n}\sigma\sqrt{1-\rho}} = \frac{p_0 - r}{\sqrt{1-\rho}\sqrt{p_0 + r - (p_0 - r)^2}}, \tag{11}$$

$$\sigma_0^2 := \frac{\rho}{1-\rho}. \tag{12}$$

Note that, T_0, being a function of W_0, is independent of $(W_1, \ldots, W_{N-1})$, hence of $W_{(1)}$. Therefore,$\forall\, t \in \mathbb{R}$,

$$
\begin{aligned}
\Pr(W_{(1)} > T_0 | T_0 = t) &= \Pr(W_{(1)} > t)\\
&= \Pr(W_1, \ldots, W_{N-1} > t)\\
&= (1 - \Phi(t))^{N-1}.
\end{aligned}
$$

This lead us to

$$
\begin{aligned}
\Pr(W_{(1)} \geq T_0) &= E[\Pr(W_{(1)} > T_0 | T_0)]\\
&= E((1 - \Phi(T_0))^{N-1})\\
&= \int_{\mathbb{R}} (1 - \Phi(x))^{N-1} \frac{1}{\sigma_0} \phi\left(\frac{x + \sqrt{n}\gamma}{\sigma_0}\right) dx.
\end{aligned}
$$

Hence, for the data complexity of the message recovery attack in this case using sample mode approach we have the following result.

Theorem 7 *If the distribution $\mathcal{P}$ of the keystream bytes on the space $\mathcal{M}$, has only one mode at* 0 *and all remaining probabilities are equal, then the success probability of the message recovery attack using the sample mode approach is $E((1 - \Phi(T_0))^{N-1})$, where $T_0 \sim \mathcal{N}(-\sqrt{n}\gamma, \sigma_0^2)$ and γ and σ_0^2 are as defined as in Equation* (11) *and* (12)*.*

Remark 3 For general $p.m.f.'s$ on the message space, the above integral is very difficult to work out analytically. Hence, to proceed towards further analysis we must pass through numerical methods to approximate the above probability. Two possible ways are approximating the above integral by different available numerical integration method or simulating large number of times independently from the distribution of T_0 and take the sample mean of the function $(1-\Phi(\cdot))^{N-1}$ to obtain an approximation of the above expectation [15].

4 Data Complexity of Message Recovery Attacks using Bayesian Approach

The Bayesian method was first discussed in [1, Section 4.1], and it finally boils down to maximum likelihood estimation. Here we shall only explore the method for some general version of single-byte bias attack. The Bayesian approach for multiple-byte bias attacks are defined similarly. The set up is as follows.

Suppose, $\mathcal{M} = \{0, 1, \dots, N-1\}$, where N is the size of the message space. For simplicity of language we shall call the elements of the message space as *bytes*. $\mathcal{P}$ is the distribution of the keystream bytes of the concerned stream cipher. Suppose M is the secret message-byte (or, plaintext) and $Z_1, \dots, Z_n \overset{i.i.d}{\sim} \mathcal{P}$ be the keys for n broadcasts. We observe the ciphertext bytes $C_i = (M + Z_i) \bmod N,\ \forall i = 1, \dots, n$. The observed ciphertexts are $c_1, \dots, c_n$. To make the notations clear, C_is are random variables and c_is are their particular realizations.

We also assume a prior distribution [9] on the message space $\mathcal{M}$, say $\mathcal{Q}$. Then q_x denotes the relative frequency of the message-byte x in a large message. If enough prior information is not available, this prior distribution is taken as uniform (i.e., a non-informative prior [9]). In Bayesian approach, we want to maximize the posterior probability [9] of the message-byte given the ciphertext bytes, i.e., we want to maximize $\Pr(M = m | C_i = c_i,\ \forall\, i = 1, \dots, n)$, over $m \in \mathcal{M}$.

Now, before going into the maximization problem, we introduce a notation

$$N_{m,z} := |\{i : c_i = (z + m) \bmod N\}| = \sum_{i=1}^{n} I(c_i = (z + m) \bmod N), \qquad (13)$$

$\forall\, z \in \mathcal{M}$, where I denotes the indicator function. Notice that $N_{m,z}$ denotes the number of occurrences of the byte z in the n keystream bytes, if the plaintext is m. It is easy to see that

$$\sum_{z\in\mathcal{M}} I(c = (z+m) \bmod N) = 1, \ \ \forall c, z, m \in \mathcal{M},$$

which gives

$$\sum_{z\in\mathcal{M}} N_{m,z} = \sum_{z\in\mathcal{M}} \sum_{i=1}^{n} I(c_i = (z{+}m) \bmod N) = \sum_{i=1}^{n} \sum_{z\in\mathcal{M}} I(c_i = (z{+}m) \bmod N) = n.$$

Now we have the following result according to [13].

Lemma 7 *Maximization of* $\Pr(M = m|C_i = c_i, \ \forall\, i = 1, \dots, n)$*, over* $m \in \mathcal{M}$ *is equivalent to maximizing*

$$h(m) := \log(q_m) + \sum_{z\in\mathcal{M}} N_{m,z} \log(p_z),$$

over $m \in \mathcal{M}$*.*

Proof Note that if $c_1, \dots, c_n$ are the observed ciphertexts then

$$\begin{aligned} &\Pr(M = m|C_i = c_i, \ \forall\, i = 1, \dots, n) \\ &= \Pr(C_i = c_i, \ \forall\, i = 1, \dots, n|M = m) \frac{\Pr(M = m)}{\Pr(C_i = c_i, \ \forall\, i = 1, \dots, n)} \\ &= \Pr(Z_i = z_i, \ \forall\, i = 1, \dots, n) \frac{\Pr(M = m)}{\Pr(C_i = c_i, \ \forall\, i = 1, \dots, n)}, \end{aligned}$$

where $z_i := (c_i - m) \bmod N; \forall\, i = 1, \dots, n$. Now the denominator in the above expression, $\Pr(C_i = c_i, \ \forall\, i = 1, \dots, n)$ does not involve m. Therefore, we are to maximize only

$$\Pr(Z_i = z_i, \ \forall\, i = 1, \dots, n) \Pr(M = m),$$

over $m \in \mathcal{M}$. As the broadcasts are independent, we have

$$\begin{aligned} \Pr(Z_i = z_i, \forall\, i = 1, \dots, n) &= \prod_{i=1}^{n} \Pr(Z_i = z_i) \\ &= \prod_{i=1}^{n} p_{z_i} \\ &= \prod_{z\in\mathcal{M}} p_z^{|\{i|z_i = z, 1\le i\le n\}|} = \prod_{z\in\mathcal{M}} p_z^{N_{m,z}}, \end{aligned}$$

as $|\,\{i|z_i = z, 1 \le i \le n\}\,| = |\,\{i|c_i = (z+m) \bmod N, 1 \le i \le n\}\,| = N_{m,z}$. Therefore, we are to maximize

$$g(m) := \Pr(M = m) \prod_{z\in\mathcal{M}} p_z^{N_{m,z}} = q_m \prod_{z\in\mathcal{M}} p_z^{N_{m,z}}, \tag{14}$$

over $m \in \mathcal{M}$. Taking log on both sides in (14) (as computationally it is easy to work with the function after logarithm), we get

$$h(m) := \log g(m) = \log(q_m) + \sum_{z \in \mathcal{M}} N_{m,z} \log(p_z). \tag{15}$$

□

So, we get the estimator for the unknown plaintext byte as

$$\hat{M} := \arg \max_{m \in \mathcal{M}} h(m).$$

For the sake of completeness, below we mention the algorithm for the message recovery attack using this approach.

Algorithm: Message recovery using Bayesian Approach and single byte bias

```
    $N_{m,z} \leftarrow$ Number of occurrences of byte $(m+z)$ in the ciphertext samples, $m, z \in \mathcal{M}$;
    $h(m) \leftarrow \arg \sum_{z \in \mathcal{M}} N_{m,z} \log(p_z) + \log(q_m)$;
    $M \leftarrow \arg\max_{m \in \mathcal{M}} h(m)$;
        Estimate of unknown plain text byte $\leftarrow M$;
```

If prior information is not available and we take $\mathcal{Q}$ to be uniform over the message space, then our objective boils down to maximizing

$$h_0(m) := \sum_{z \in \mathcal{M}} N_{m,z} \log(p_z); \tag{16}$$

over $m \in \mathcal{M}$, and this objective function is nothing but the constant times log-likelihood for the data $C_1, \dots, C_n \overset{i.i.d}{\sim} (\mathcal{P} + m) \bmod N$, where $m \in \mathcal{M}$ acts as the unknown parameter. So, in this case, the Bayesian estimator is also the maximum likelihood estimator.

If we use the above idea for multiple-byte bias attack, the message space $\mathcal{M}$ will be substituted by $\mathcal{M}^k$, for some $k \in \mathbb{N}$. The basic methodology remains same. We need a prior distribution $\mathcal{Q}^{(k)}$ on $\mathcal{M}^k$, as the prior distribution of the plaintexts, and the distribution $\mathcal{P}^{(k)}$ of vector of k keystream-bytes, where we shall exploit the biases in the later distribution to mount the message recovery attack. Let $p_{\boldsymbol{z}}^{(k)}$ and $q_{\boldsymbol{z}}^{(k)}$ denote the probabilities of a k-byte vector $\boldsymbol{z}$ for the distributions $\mathcal{P}^{(k)}$ and $\mathcal{Q}^{(k)}$ respectively. The secret message is $\boldsymbol{M}$ and $\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_n \overset{i.i.d.}{\sim} \mathcal{P}^{(k)}$ be the keys for n broadcasts. We observe the ciphertexts, $\boldsymbol{C}_i = (\boldsymbol{M} + \boldsymbol{Z}_i) \bmod N, \forall\, i = 1, \dots, n$, where the operation $\bmod N$ on a vector means mod N at each coordinate. $\boldsymbol{c}_1, \dots, \boldsymbol{c}_n$ be the observed ciphertexts. By similar arguments used in Lemma 7, our objective is to maximize

$$h(\boldsymbol{m}) = \log(q_{\boldsymbol{m}}^{(k)}) + \sum_{\boldsymbol{z} \in \mathcal{M}^k} N_{\boldsymbol{m},\boldsymbol{z}} \log(p_{\boldsymbol{z}}^{(k)}),$$

over $\boldsymbol{m} \in \mathcal{M}^k$, where

$$N_{\boldsymbol{m},\boldsymbol{z}} := |\{i | \boldsymbol{c}_i = (\boldsymbol{m} + \boldsymbol{z}) \bmod N, 1 \leq i \leq n\}| = \sum_{i=1}^{n} I(\boldsymbol{c}_i = (\boldsymbol{m} + \boldsymbol{z}) \bmod N),$$

The above maximization may be computationally difficult, as $\mathcal{M}^k$ may contain a very large number of elements. As for example, in the case of RC4, even if k=3 or 4, it becomes too large. So, we go for some approximation techniques.

One of these approximation techniques is based on the assumption that if $\boldsymbol{Z}_i = (Z_{i1}, \ldots, Z_{ik})$, then $Z_{i1}, \ldots, Z_{ik}$'s are independent. This approximation is very good for the cases where the single-byte biases are dominant. Under this assumption we have

$$p_{\boldsymbol{z}}^{(k)} = \prod_{i=1}^{k} p_{z_i},$$

for all $\boldsymbol{z} = (z_1, \ldots, z_k) \in \mathcal{M}^k$. Our new objective function becomes

$$h(\boldsymbol{m}) = \log(q_{\boldsymbol{m}}^{(k)}) + \sum_{\boldsymbol{z} \in \mathcal{M}^k} \sum_{i=1}^{k} N_{\boldsymbol{m},\boldsymbol{z}} \log(p_{z_i}). \tag{17}$$

To simplify (17), let us denote, $\boldsymbol{c}_j = (c_{j1}, \ldots, c_{jk})$, $\forall\, j = 1, \ldots, n$. Define,

$$N_{m,z}^{(i)} := |\{j : c_{ji} = (z + m) \bmod N, 1 \leq j \leq n\}|, \ \forall\, m, z \in \mathcal{M}, 1 \leq i \leq k.$$

Notice that,

$$N_{m,z}^{(i)} := \sum_{j=1}^{n} I(c_{ji} = (z + m) \bmod N).$$

$N_{m,z}^{(i)}$ actually denotes the number of occurrences of z in the i-th byte of the keystream, if the i-th byte of the plaintext was m Then,$\forall$ $\boldsymbol{m} = (m_1, \ldots, m_k) \in \mathcal{M}^k$, we have

$$N_{m_i,z}^{(i)} = \sum_{\boldsymbol{z} \in \mathcal{M}^k, z_i = z} N_{\boldsymbol{m},\boldsymbol{z}} \ ,$$

because,

$$\begin{aligned}
\sum_{\boldsymbol{z}\in\mathcal{M}^k, z_i=z} N_{\boldsymbol{m},\boldsymbol{z}} &= \sum_{\boldsymbol{z}\in\mathcal{M}^k, z_i=z} \sum_{j=1}^{n} I(\boldsymbol{c}_j = (\boldsymbol{z}+\boldsymbol{m}) \bmod N) \\
&= \sum_{\boldsymbol{z}\in\mathcal{M}^k, z_i=z} \sum_{j=1}^{n} \prod_{l=1}^{k} I(c_{jl} = (z_l + m_l) \bmod N) \\
&= \sum_{j=1}^{n} \sum_{\boldsymbol{z}\in\mathcal{M}^k, z_i=z} \prod_{l=1}^{k} I(c_{jl} = (z_l + m_l) \bmod N) \\
&= \sum_{j=1}^{n} I(c_{ji} = (z + m_i) \bmod N) = N^{(i)}_{m_i,z}.
\end{aligned}$$

which gives,

$$\begin{aligned}
\sum_{z\in\mathcal{M}} N^{(i)}_{m_i,z} \log(p_z) &= \sum_{z\in\mathcal{M}} \sum_{\boldsymbol{z}\in\mathcal{M}^k, z_i=z} N_{\boldsymbol{m},\boldsymbol{z}} \log(p_z) \\
&= \sum_{z\in\mathcal{M}} \sum_{\boldsymbol{z}\in\mathcal{M}^k, z_i=z} N_{\boldsymbol{m},\boldsymbol{z}} \log(p_{z_i}) \\
&= \sum_{\boldsymbol{z}\in\mathcal{M}^k} N_{\boldsymbol{m},\boldsymbol{z}} \log(p_{z_i})
\end{aligned}$$

Therefore,

$$h(\boldsymbol{m}) = \log(q^{(k)}_{\boldsymbol{m}}) + \sum_{i=1}^{k} \sum_{z\in\mathcal{M}} N^{(i)}_{m_i,z} \log(p_z). \tag{18}$$

Furthermore, if $q^{(k)}_{(m_1,\ldots,m_k)} = q_{m_1} \ldots q_{m_k}$ (note that this is a weaker assumption than uniformity of $q^{(k)}$), then this multiple-byte bias attack reduces to nothing but single-byte bias attack at k plaintext points.

Another approximation reduces multiple-byte bias attacks to double byte bias attacks. For this, we make the Markovian assumption, i.e., for the i-th key $\boldsymbol{Z}_i$, the random variables $Z_{i1}, \ldots, Z_{ik}$ satisfies the Markov property. Therefore,

$$\begin{aligned}
p^{(k)}_{\boldsymbol{z}} &= P[\boldsymbol{Z}_j = \boldsymbol{z}] \\
&= P[Z_{j1} = z_1] \prod_{i=2}^{k} P[Z_{ji} = z_i | Z_{j,i-1} = z_{i-1}] \\
&= \frac{\prod_{i=2}^{k} p^{(2)}_{(z_i,z_{i-1})}}{\prod_{i=2}^{k-1} p_{z_i}}.
\end{aligned}$$

So, the objective function turns out to be

$$\begin{aligned} h(\boldsymbol{m}) &= \log(q_{\boldsymbol{m}}^{(k)}) + \sum_{i=2}^{k} \sum_{\boldsymbol{z}\in\mathcal{M}^k} N_{\boldsymbol{m},\boldsymbol{z}} \log(p^{(2)}_{(z_i,z_{i-1})}) \\ &\qquad - \sum_{\boldsymbol{z}\in\mathcal{M}^k} \sum_{i=2}^{k-1} N_{\boldsymbol{m},\boldsymbol{z}} \log(p_{z_i}) \\ &= \log(q_{\boldsymbol{m}}^{(k)}) + \sum_{i=2}^{k} \sum_{z,y\in\mathcal{M}} N^{(i)}_{m_i,m_{i-1},z,y} \log(p^{(2)}_{(z,y)}) \\ &\qquad - \sum_{i=2}^{k-1} \sum_{z\in\mathcal{M}} N^{(i)}_{m_i,z} \log(p_z), \end{aligned}$$

where

$$\begin{aligned} N^{(i)}_{m,m^*,z,y} &:= |\{j : c_{ji} = (z+m) \bmod N,\ c_{j,i-1} = (y+m^*) \bmod N\}| \\ &= \sum_{j=1}^{n} I(c_{ji} = (z+m) \bmod N) I(c_{j,i-1} = (y+m^*) \bmod N); \end{aligned}$$

where $2 \leq i \leq k$, and we have used the identity which can be checked as previous.

$$\sum_{\boldsymbol{z}\in\mathcal{M}^k} N_{\boldsymbol{m},\boldsymbol{z}} \log(p^{(2)}_{(z_j,z_{j-1})}) = \sum_{z,y\in\mathcal{M}} N^{(i)}_{m_i,m_{i-1},z,y} \log(p^{(2)}_{(z,y)}).$$

We shall now discuss about the performance of this Bayesian estimation technique. We shall concentrate only on the case where the prior distribution is taken to be uniform and hence the procedure boils down to ML estimation. It is to be noted that the exact calculation of the success probabilities is very difficult in this set up. So we shall continue by doing some approximate calculations under some conditions.

4.1 Approximate Calculation of success probability

Now we shall concentrate on both unimodal and multi-modal distribution (by multi-modal, we actually intend to mean that the highest probabilities have negligible difference) for the keystream bytes, simultaneously. We may have more than one sample modes for the ciphertexts (or the sample mode of the ciphertext may not be equal to the modulo sum of actual plaintext and the population mode of the keystream distribution, i.e., the population and sample mode for the Z's may differ even in the long run). But if the distributions of

$(\mathcal{P}+m) \bmod N$, where $m \in \mathcal{M}$, are different for different m's, then also the ML estimation succeeds in long run as it not only take into account the sample mode of the ciphertexts but also other points, and hence able to distinguish between the distributions $(\mathcal{P}+m) \bmod N$, for all possible m's.

To estimate the data complexity of the ML method based attack, we shall use Theorem 6 and Theorem 7. Recall that our ML estimate was

$$\hat{M} = \arg\max_{m \in \mathcal{M}} \sum_{z \in \mathcal{M}} N_{m,z} \log(p_z), \tag{19}$$

Then, define

$$\boldsymbol{t}_k := (\log(p_{(0-k) \bmod N}), \ldots, \log(p_{((N-1)-k) \bmod N}))^T, \forall\, k = 0, \ldots N-1,$$

$$G := [\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{N-1}]^T,$$

and

$$\boldsymbol{N} := (N_{0,0}, \ldots, N_{0,N-1})^T.$$

Now, as $N_{(m+k) \bmod N, z} = N_{m,(z+k) \bmod N}$, it is easy to see that Equation 19 reduces to

$$\hat{M} = \arg\max_{m \in \mathcal{M}} \boldsymbol{t}_m{}^T \boldsymbol{N}, \tag{20}$$

as we have

$$\begin{aligned}
\boldsymbol{t}_m^T \boldsymbol{N} &= \sum_{z=0}^{N-1} N_{0,z} \log(p_{(z-m) \bmod N}) \\
&= \sum_{z=0}^{N-1} N_{m,(z-m) \bmod N} \log(p_{(z-m) \bmod N}) \\
&= \sum_{z=0}^{N-1} N_{m,z} \log(p_z).
\end{aligned}$$

Note that, $N_{0,z}$ is equal to the number of occurrences of the byte z in the n obtained ciphertext bytes. Suppose the unknown plaintext is m^*. Define, $\boldsymbol{q}^T := (p_{(0-m^*) \bmod N}, \ldots, p_{(N-1-m^*) \bmod N})$. Then, we have

$$\boldsymbol{N} \sim Multinomial(n; \boldsymbol{q}),$$

and hence, by *Central Limit Theorem* [15],

$$n^{-1}\boldsymbol{N} \sim A\mathcal{N}(\boldsymbol{q}, n^{-1}(diag(\boldsymbol{q}) - \boldsymbol{q}\boldsymbol{q}^T)). \tag{21}$$

Now, define

$$\boldsymbol{N}_0 := (\boldsymbol{t}_0^T \boldsymbol{N}, \ldots, \boldsymbol{t}_{N-1}^T \boldsymbol{N})^T = G\boldsymbol{N} = (R_0, \ldots, R_{N-1})^T.$$

and let $\boldsymbol{t}_i^T\boldsymbol{q} = r_i = \sum_{z=0}^{N-1} p_{(z-m^*) \bmod N} \log(p_{(z-i) \bmod N})$, $\forall\, i = 0, \ldots, N-1$. Then by (21) we have,

$$n^{-1}\boldsymbol{N}_0 = Gn^{-1}\boldsymbol{N} \sim A\mathcal{N}((r_0, \ldots, r_{N-1})^T, n^{-1}\boldsymbol{\Sigma}'),$$

where $\boldsymbol{\Sigma}' := G(diag(\boldsymbol{q}) - \boldsymbol{q}\boldsymbol{q}^T)G^T = ((\sigma_{ij}))_{i,j}$. Note that, by rearrangement inequality [16], r_i is maximum if $i = m^*$, and

$$r_{m^*} = \boldsymbol{t}_{m^*}^T\boldsymbol{q} = \sum_{k=0}^{N-1} p_{(k-m^*) \bmod N} \log(p_{(k-m^*) \bmod N}) = \sum_{k=0}^{N-1} p_k \log(p_k),$$

i.e., doesn't depend on the value of m^*. Now, we can readily recognize the set-up perfect for applying Lemma 6, as here success probability,$\Pr(\hat{M} = m^*)$, is nothing but the probability of the maximum of $\boldsymbol{N}_0$ co-ordinates occurring in the co-ordinate with highest mean. So, keeping the idea from Lemma 6 in mind, we define

$$\eta_k := \frac{r_{m^*} - r_k}{\sqrt{\sigma_{m^*m^*} + \sigma_{kk} - 2\sigma_{m*k}}}; \forall\, k \neq m^*. \tag{22}$$

We would like to simplify the above expressions to get an idea what these η_k actually signify. It is immediate that,

$$r_k = \sum_{l=0}^{N-1} p_{(l-k) \bmod N} \log(p_{(l-m^*) \bmod N}) = \sum_{l=0}^{N-1} p_{(l+m^*-k) \bmod N} \log(p_l),$$

and so

$$r_{m^*} - r_k = \boldsymbol{t}_{m^*}^T\boldsymbol{q} - \boldsymbol{t}_k^T\boldsymbol{q} = \sum_{l=1}^{N-1} p_l \log(p_l) - \sum_{l=0}^{N-1} p_{(l+m^*-k) \bmod N} \log(p_l). \tag{23}$$

From the definition of Σ', it is also immediate that

$$\begin{aligned}\sigma_{kl} &= \boldsymbol{t}_k^T diag(\boldsymbol{q})\boldsymbol{t}_l - \boldsymbol{t}_k^T\boldsymbol{q}\boldsymbol{q}^T\boldsymbol{t}_l \\ &= \sum_{z=0}^{N-1} p_{(z-m^*) \bmod N} \log(p_{(z-k) \bmod N}) \log(p_{(z-l) \bmod N}) - r_k r_l,\end{aligned}$$

and therefore, putting these expressions together we get,

$$\sigma_{m^*m^*} + \sigma_{kk} - 2\sigma_{m*k} = \sum_{i=0}^{N-1} p_i(\log(p_i) - \log(p_{(i+m^*-k) \bmod N}))^2 - (r_m^* - r_k)^2. \tag{24}$$

A close inspection of the expressions in (23) and (24) shows that both the expressions depend only on the difference between m^* and k, and therefore if we replace m^* by 0, the set of values of η will be unchanged. Hence, their ordered sequence is fixed and known, call it $\eta = \eta_{(1)} \leq \eta_{(2)} \leq \cdots \leq \eta_{(N-1)}$. Then using Lemma 6, we have the following result.

Corollary 2 *If all the η's are defined (i.e. finite) and $\eta_{(1)}$ and $\eta_{(2)}$ are distinguishably apart, then the data complexity of the message recovery attack with failure probability at most α using ML approach is given by*

$$n > \left(\frac{\Phi^{-1}(1-\alpha)}{\eta}\right)^2, \tag{25}$$

where $\eta_{(1)}, \eta_{(2)}, \eta$ are as defined earlier.

Proof By taking $\boldsymbol{W}$ in Lemma 6 to be $\frac{1}{n}\boldsymbol{N}_0$, we get that

$$\Pr(\hat{M} = m^*) = \Pr(\max_{0 \le i \le N-1} \frac{1}{n} R_i = \frac{1}{n} R_{m^*}) \approx 1 - \Phi(-\sqrt{n}\eta).$$

Hence, to get $\Pr(\hat{M} = m^*)$ at least $(1-\alpha)$, we must have,

$$n > \left(\frac{\Phi^{-1}(1-\alpha)}{\eta}\right)^2.$$

It is interesting to note that

$$\begin{aligned} r_{m^*} - r_k &= \sum_{l=1}^{N-1} p_l \log(p_l) - \sum_{l=0}^{N-1} p_{(l+m^*-k) \bmod N} \log(p_l) \\ &= \sum_{i=1}^{N-1} p_i \log\left(\frac{p_i}{p_{(i-m^*+k) \bmod N}}\right), \end{aligned}$$

and therefore, $r_{m^*} - r_k$ is the KL distance between $\mathcal{P}$ and $(\mathcal{P} - k + m^*) \bmod N$. Hence, the η s are like normalized KL distances between $\mathcal{P}$ and different $(\mathcal{P} + l) \bmod N$ distributions.

The condition needed in the above corollary is that $\eta_{(1)}$ and $\eta_{(2)}$ are distinguishably apart. If this condition doesn't hold true the ML method even works, but the above approximation for message recovery attack complexity won't work.

However, if there are two or more $m \in \mathcal{M}$ (not necessarily modes) such that $(\mathcal{P}+m) \bmod N$ have same distributions, then ML method fail. We take an example for this. Consider $\mathcal{M} = \{0, \ldots, 255\}$ and suppose that the distribution $\mathcal{P}$ is such that 0 and 128 are two modes and all other probabilities are equal. Then if the objective function is maximized in m then will also be maximized in $(m + 128) \bmod 256$ as in this case $h_0(m) = h_0((m + 128) \bmod 256)$. So, ML method would fail in all cases in this example. Not only ML method, any estimator fails here miserably as the parameter m is not *identifiable*[1] in this case.

4.2 Comparison between Bayesian and Sample Mode Approach

So far we have discussed two ways for message recovery attack, one by *Sample Mode Approach* and another by *Bayesian Approach.*

In this context we would like to point out the following result for unimodal case of Bayesian approach.

Lemma 8 *If the distribution $\mathcal{P}$ of the keystream bytes on the space $\mathcal{M}$, has only one mode at* 0 *and all remaining probabilities are equal, then the message recovery attack using ML estimation and sample mode approach give the same result.*

Proof Here ML method maximizes

$$h_0(m) = \sum_{k=1}^{N-1} N_{m,k} \log r + N_{m,0} \log p_0 = N_{m,0} \log p_0 + (n - N_{m,0}) \log r,$$

i.e., practically we are to maximize $N_{m,0}(\log p_0 - \log r)$ therefore only $N_{m,0}$ over m. Clearly, it is maximized if we take m to be equal to sample mode of the ciphertexts and thus the two estimates coincide. □

Note that if the mode is different from 0, then also the same result holds.

In general, as the Bayesian approach uses more information about the keystream distribution (as it considers biases in all positions) than in the later approach, it has always greater success probability. But on the other hand, in the first approach, we have to maximize a complex function over a huge set, which may turn out to be computationally inefficient sometime. Therefore, we may follow the following rules while deciding which method to apply:

1. As discussed earlier, for unimodal keystream byte distribution, both methods gives same estimate for large sample size, and hence in that we should go for computationally more efficient sample mode approach.
2. For small sample sizes or multi-modal distributions (with identifiable plaintext parameter), we should go for Bayesian approach.

[1] A family of probability distributions $\{P_\theta\}$ indexed by the parameter θ is said to be *identifiable* w.r.t. θ, if

$$\theta_1 \neq \theta_2 \Rightarrow P_{\theta_1} \neq P_{\theta_2}.$$

Otherwise the family is said to be *non-identifiable*

3. For, non-identifiable plaintext case, both methods fail miserably, and hence none is preferred.

Finally, we want to discuss about the adversarial advantage of the message recovery attack. It is defined as

$$\begin{aligned} Adv = P(\hat{M} = M|\text{keystream generated from the cipher}) \\ -P(\hat{M} = M|\text{keystream is uniform}). \end{aligned}$$

The first term in the above expression is the success probability, say α, and the second term is essentially the probability of a random guess to be correct, which is equal to $\dfrac{1}{N}$ for single-byte bias attack and $\dfrac{1}{N^k}$ for multiple-byte bias attack, where k is the message size. So, if the success probability can be estimated, methods of which is discussed so far, advantage can be computed easily.

5 Connecting the Complexities of Distinguisher and Message Recovery: a Case Study

We are interested in the relation between the data complexities of *Distinguishing* and *Message Recovery Attack*. We define the function *Distinguish-equivalent* on the set of natural numbers as follows.

Definition 1 *Distinguish-equivalent*(n) is the number of samples needed in the distinguishing attack to have the same success probability as that in the message recovery attack for sample size equal to n.

To compare the data complexities in the two cases, we define another function *Multiplier* on the set of natural number as follows.

Definition 2 *Multiplier*(n) is the ratio $\frac{n}{\mathit{Distinguish\text{-}equivalent}(n)}$, which indicates how many times more sample is needed to recover the message than to only distinguish from uniform distribution with same probability of success.

Due to the remark after Theorem 7, derivation of a closed-form expression of this quantity is not possible.

Now, we illustrate our previously derived results by pursuing the attack on RC4 stream cipher based on its second byte bias. Here $N = 256$ and the distributional node is 0 with $p_0 \approx \frac{2}{256}$. In the other sample points, biases are very small which makes the other probabilities (i.e., $p_1, \ldots, p_{255}$) almost equal. Therefore, according to Lemma 8, one may use either ML or Sample Mode approach.

In a broadcast attack scenario, we have same message encrypted by different RC4 keystreams (say n times). We collect the second byte of the ciphertexts $C_1, \dots, C_n$ and guess the secret message by $Mode(C_1, \dots, C_n)$. Our probability of success is given by the integral or expectation stated in Theorem 7. In this particular context

$$\gamma = 0.06286849 \ , \ \sigma_0^2 = 2.003922.$$

The change in success probability for different sample size is given in *Fig. 2*.

The behaviour of the *Multiplier* function with n is shown in *Fig. 3*, where the distinguishing data complexity is calculated using the result stated in *Theorem 2* and taking the both way success probabilities equal (i.e., equal false positive and negative errors).

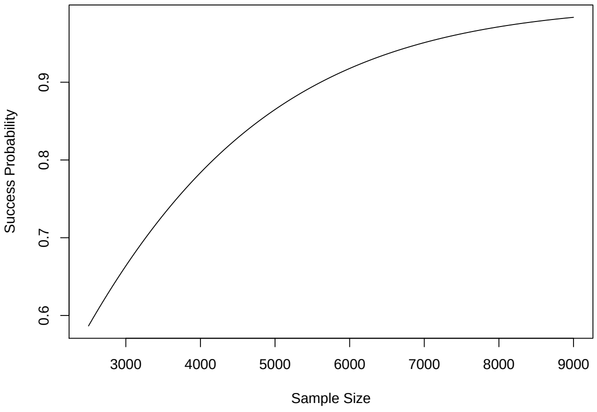

Fig. 2 Success Rate for Different Sample Sizes for RC4 Second Byte

From *Fig.3* we see that the *Multiplier* function is continuously decreasing and decreasing very rapidly at the recovery attack sample size 2400 to 3000 (i.e., at success rate in $[0.55, 0.66]$. At success rate 0.7, i.e., near sample size 3250 for recovery attack, we note that we need almost 8 times more samples in the recovery attack than that in the distinguishing attack, whereas near success rate $[0.97, 0.98]$, it becomes almost 1.3 to 1.35.

We would now be interested in estimating the *Multiplier* function in terms of some known handy function. It is very interesting to note that we find empirically $\ln(n)$ and $\ln(\ln(Multiplier(n)))$ to be highly linearly related. We

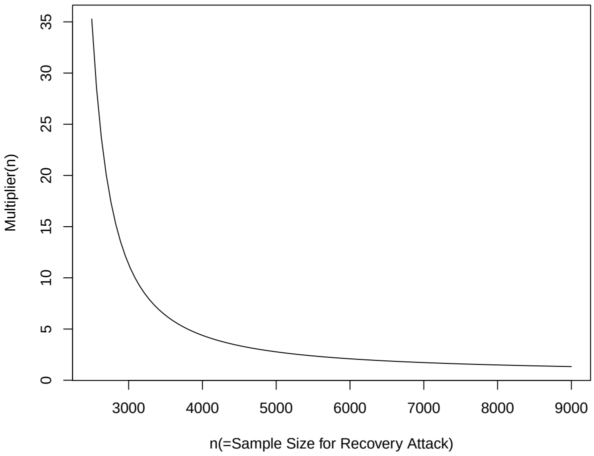

Fig. 3 Multiplier for Different Sample size in Recovery Attack

define two functions f and g on the set of natural numbers as follows:

$$f(n) := \ln\left(\frac{n}{1000}\right); \quad g(n) := \ln(\ln(Multiplier(n))).$$

The graph of $f(n)$ vs $g(n)$ in *Fig.4* shows empirically high linear relationship.

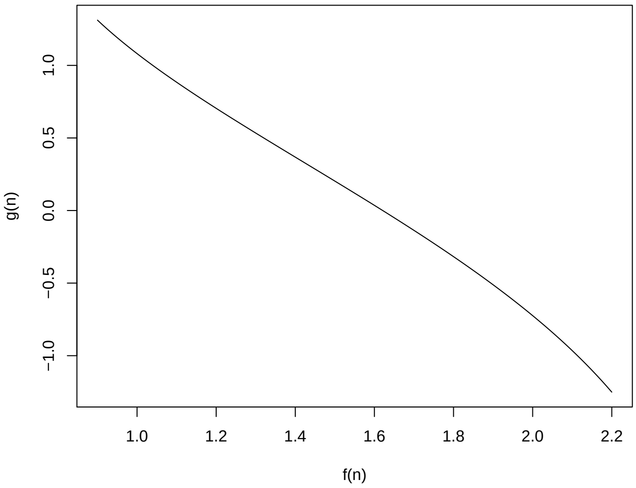

Fig. 4 Graph of $f(n)$ vs. $g(n)$

Fig.4. shows the function $g(n)$ is slightly convex in the region $(0.9, 1.4)$ and slightly concave in the region $(1.8, 2.2)$ with respect to $f(n)$. In the middle region it is almost linear. So, we try to approximate the above relationship i.e. the function $g(n)$ in these three regions separately.

For the region $(1.8, 2.2)$ we empirically find that $h_1(n) := exp\left(\frac{g(n)}{2}\right)$ is almost linear w.r.t. $f(n)$. We try to estimate $h_1(n)$ by a linear function of $f(n)$ by minimizing the distance over the range $(1.8, 2.2)$(where the distance between two integrable functions f and g over the range (a, b) is defined as $\int_a^b (f(x) - g(x))^2 \, dx$.) The function $h_1(n)$ and its estimate looks like in *Fig.5* where the estimating linear function $h_1'(n) := 2.249515 - 0.7759676f(n)$ is in dotted line.

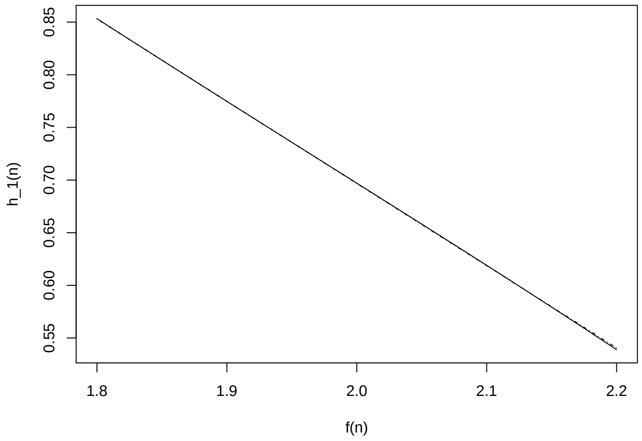

Fig. 5 Graph of $h_1(n)$ and its estimator linear function

So, we get

$$g(n) \approx 2(\ln(2.249515 - 0.7759676f(n));$$

$$for \;\; 1.8 \le f(n) \le 2.2 \;\; i.e. \; 6000 \le n \le 9000.$$

For the region $(1.4, 1.8)$ we estimate the function $g(n)$ itself by linear functions by same method as described above.

The function and its estimating line $g'(n) := 2.758714 - 1.703975f(n)$ looks like *Fig.6.* So, we get

$$g(n) \approx 2.758714 - 1.703975f(n));$$

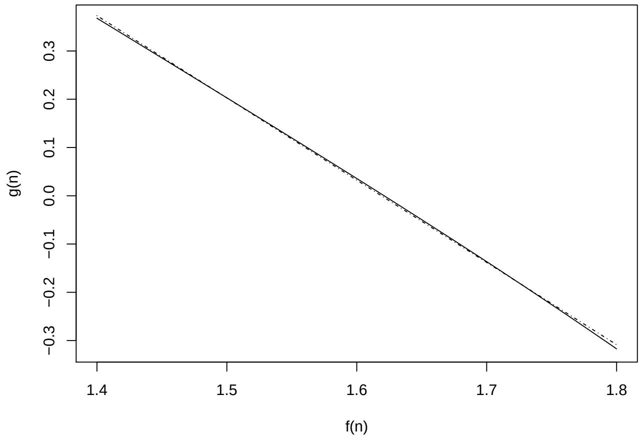

Fig. 6 Graph of $g(n)$ and its estimator linear function (in dotted line)

$$for \;\; 1.4 \leq f(n) \leq 1.8 \;\; i.e. \; 4000 \leq n \leq 6000.$$

For the region $(0.9, 1.4)$ we find empirically the function $h_3(n) := (g(n))^{\frac{3}{4}}$ to be highly linearly related with $f(n)$. As previous we estimate by linear function by minimizing the distance and the estimating line $h_3'(n) := 2.393338 - 1.336852 f(n)$ looks like in *Fig.7*.

So, we get

$$g(n) \approx (2.393338 - 1.336852 f(n))^{\frac{4}{3}};$$

$$for \;\; 0.9 \leq f(n) \leq 1.4 \;\; i.e. \; 2450 \leq n \leq 4000.$$

6 Conclusion

In this paper, we review different approaches towards estimating the data complexity of distinguishing attacks on stream ciphers and analyze their interrelationships and applicable scenarios. We also formally analyze the data complexity of message recovery attack that exploits a distinguisher and show that in practice there is a significant gap between the two complexities. This gap turns out to be a function of the number of samples of the distinguishing attack. We perform a case study on RC4 stream cipher to demonstrate how these two complexities are related.

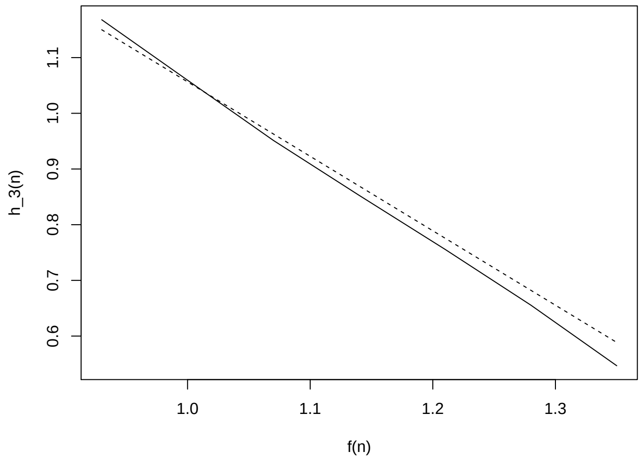

Fig. 7 Graph of $h_3(n)$ and its estimator linear function (in dotted line)

A Proof of Lemma 2

Proof Suppose $\mathcal{S}$ is the sample space and $\phi : \mathcal{S} \longrightarrow [0,1]$ be the *test function* [9] for the concerned test with false positive rate (α) and false negative rate (β), i.e., we reject H_0 with probability $\phi(\boldsymbol{x})$, when $\boldsymbol{X} = \boldsymbol{x}$ is observed. Then we have by definition

$$E_{H_0}[\phi(\boldsymbol{X})] = \sum_{\boldsymbol{x}\in\mathcal{S}} \phi(\boldsymbol{x}) P_n(\boldsymbol{x}) = \alpha,$$

$$E_{H_1}[(1-\phi)(\boldsymbol{X})] = \sum_{\boldsymbol{x}\in\mathcal{S}} (1-\phi(\boldsymbol{x})) Q_n(\boldsymbol{x}) = \beta.$$

Note that,

$$\begin{aligned} D_{KL}(Q_n||P_n) &= \sum_{\boldsymbol{x}\in\mathcal{S}} Q_n(\boldsymbol{x}) \log_2 \frac{Q_n(\boldsymbol{x})}{P_n(\boldsymbol{x})} \\ &= \sum_{\boldsymbol{x}\in\mathcal{S}} P_n(\boldsymbol{x}) \frac{Q_n(\boldsymbol{x})}{P_n(\boldsymbol{x})} \log_2 \frac{Q_n(\boldsymbol{x})}{P_n(\boldsymbol{x})} \\ &= \sum_{\boldsymbol{x}\in\mathcal{S}} P_n(\boldsymbol{x}) f\left(\frac{Q_n(\boldsymbol{x})}{P_n(\boldsymbol{x})}\right) \end{aligned}$$

where $f : \mathbb{R}^+ \to \mathbb{R}$ defined as $f(z) = z\log_2(z), \forall\, z > 0$. Then, $\dfrac{d^2 f(z)}{dz^2} = (z \ln 2)^{-1} > 0, \forall\, z > 0$; which implies f is *convex* and *continuous* also. Hence, using *Jensen's Inequality*, we have

$$\begin{aligned}
&\sum_{\boldsymbol{x}\in\mathcal{S}} \frac{\phi(\boldsymbol{x})P_n(\boldsymbol{x})}{\sum_{\boldsymbol{x}\in\mathcal{S}}\phi(\boldsymbol{x})P_n(\boldsymbol{x})} f\left(\frac{Q_n(\boldsymbol{x})}{P_n(\boldsymbol{x})}\right)\\
&\geq f\left(\sum_{\boldsymbol{x}\in\mathcal{S}} \frac{\phi(\boldsymbol{x})P_n(\boldsymbol{x})}{\sum_{\boldsymbol{x}\in\mathcal{S}}\phi(\boldsymbol{x})P_n(\boldsymbol{x})} \frac{Q_n(\boldsymbol{x})}{P_n(\boldsymbol{x})}\right)\\
&= f\left(\frac{\sum_{\boldsymbol{x}\in\mathcal{S}}\phi(\boldsymbol{x})Q_n(\boldsymbol{x})}{\sum_{\boldsymbol{x}\in\mathcal{S}}\phi(\boldsymbol{x})P_n(\boldsymbol{x})}\right)\\
&= f\left(\frac{1-\beta}{\alpha}\right).
\end{aligned}$$

Hence,

$$\begin{aligned}
\sum_{\boldsymbol{x}\in\mathcal{S}} \phi(\boldsymbol{x})P_n(\boldsymbol{x}) f\left(\frac{Q_n(\boldsymbol{x})}{P_n(\boldsymbol{x})}\right) &\geq \sum_{\boldsymbol{x}\in\mathcal{S}} \phi(\boldsymbol{x})P_n(\boldsymbol{x}) f\left(\frac{1-\beta}{\alpha}\right)\\
&= \alpha f\left(\frac{1-\beta}{\alpha}\right)\\
&= (1-\beta)\log_2\left(\frac{1-\beta}{\alpha}\right).
\end{aligned}$$

Replacing ϕ by $1-\phi$ and taking similar sums we get,

$$\sum_{\boldsymbol{x}\in\mathcal{S}} (1-\phi(\boldsymbol{x}))P_n(\boldsymbol{x}) f\left(\frac{Q_n(\boldsymbol{x})}{P_n(\boldsymbol{x})}\right) \geq \beta\log_2\frac{\beta}{1-\alpha}.$$

Summing the above two inequalities we get

$$\begin{aligned}
&\sum_{\boldsymbol{x}\in\mathcal{S}} P_n(\boldsymbol{x}) f\left(\frac{Q_n(\boldsymbol{x})}{P_n(\boldsymbol{x})}\right)\\
&\geq \beta\log_2\frac{\beta}{1-\alpha} + (1-\beta)\log_2\frac{1-\beta}{\alpha},
\end{aligned}$$

and hence the desired result. □

B Proof of Lemma 4

Proof This proof of *Chernoff-Stein Lemma* occurs in [10]. First note that,

$$\log_2\left[\frac{P_n(X_1,\ldots,X_n)}{Q(X_1,\ldots,X_n)}\right] = \sum_{k=1}^{n}\log_2\left[\frac{P(X_k)}{Q(X_k)}\right],$$

and by *Law of Large numbers*

$$\begin{aligned}
\frac{1}{n}\sum_{k=1}^{n}\log_2\left[\frac{P(X_k)}{Q(X_k)}\right] &\xrightarrow{p} E_P\left[\log_2\left(\frac{P(X_1)}{Q(X_1)}\right)\right]\\
&= D_{KL}(P||Q),
\end{aligned}$$

under the null. Hence,

$$\frac{1}{n}\log_2\left[\frac{P_n(X_1,\ldots,X_n)}{Q(X_1,\ldots,X_n)}\right] \xrightarrow{p} D_{KL}(P||Q),$$

which by definition gives that $\forall \epsilon, \alpha > 0$, $\exists\ N_{\epsilon,\alpha} \in \mathbb{N}$ such that, $\forall\, n \geq N_{\epsilon,\alpha}$, we have

$$P_n\left[|\frac{1}{n}\log_2\left[\frac{P_n(\boldsymbol{X})}{Q_n(\boldsymbol{X})}\right] - D_{KL}(P||Q)| < \epsilon\right] \geq 1-\alpha, \tag{26}$$

where $D = D_{KL}(P||Q)$ and $\boldsymbol{X} = (X_1,\ldots,X_n)$. Now, define A_n^ϵ be the subset of χ^n consisting of all $\boldsymbol{x} = (x_1,\ldots,x_n)$ such that

$$P_n(\boldsymbol{x})2^{-n(D+\epsilon)} < Q_n(\boldsymbol{x}) < P_n(\boldsymbol{x})2^{-n(D-\epsilon)},$$

i.e.,

$$|\frac{1}{n}\log_2\left[\frac{P_n(\boldsymbol{x})}{Q_n(\boldsymbol{x})}\right] - D| < \epsilon.$$

Then, Equation (26) gives,

$$P_n(A_n^\epsilon) \geq 1-\alpha,$$

$\forall n \geq N_{\epsilon,\alpha}$. Also note that

$$\begin{aligned} Q_n(A_n^\epsilon) &= \sum_{\boldsymbol{x}\in A_n^\epsilon} Q_n(\boldsymbol{x}) & (27)\\ &< \sum_{\boldsymbol{x}\in A_n^\epsilon} P_n(\boldsymbol{x})2^{-n(D-\epsilon)} < 2^{-n(D-\epsilon)}, & (28)\end{aligned}$$

and

$$\begin{aligned} Q_n(A_n^\epsilon) &= \sum_{\boldsymbol{x}\in A_n^\epsilon} Q_n(\boldsymbol{x}) & (29)\\ &> \sum_{\boldsymbol{x}\in A_n^\epsilon} P_n(\boldsymbol{x})2^{-n(D+\epsilon)} & (30)\\ &= 2^{-n(D+\epsilon)}P_n(A_n^\epsilon) \geq (1-\alpha)2^{-n(D+\epsilon)}, & (31)\end{aligned}$$

$\forall n \geq N_{\epsilon,\delta}$. Now consider the test which rejects the null if and only if $\boldsymbol{x} \notin A_n^\epsilon$. Then, by equation (27) $\forall\, n \geq N_{\epsilon,\alpha}$,

$$1 - P_n(A_n^\epsilon) < \alpha \text{ and } Q_n(A_n^\epsilon) < 2^{-n(D-\epsilon)},$$

which says that the non-randomized test with acceptance region A_n^ϵ has size less than α and has false negative error less than $2^{-n(D-\epsilon)}$. So, by definition of $\beta_{n,\alpha}$, which is the least attainable false negative error for level α non-randomized tests, we have

$$\beta_{n,\alpha} < 2^{-n(D-\epsilon)}.$$

Thus we have, $\forall\, n \geq N_{\epsilon,\alpha}, \epsilon > 0$,

$$\frac{\log_2 \beta_{n,\alpha}}{n} < -D + \epsilon \Longrightarrow \limsup_{n\to\infty} \frac{\log_2 \beta_{n,\alpha}}{n} \leq -D. \tag{32}$$

On the other hand, consider any other test with rejection region $\mathcal{R}$, such that $P_n(\mathcal{R}) < \alpha$. Then we have,$\forall\, n \geq N_{\epsilon,\alpha}$,

$$\begin{aligned}
Q_n(\mathcal{R}^c) &\geq Q_n(\mathcal{R}^c \cap A_n^\epsilon) \\
&= \sum_{\boldsymbol{x} \in \mathcal{R}^c \cap A_n^\epsilon} Q_n(\boldsymbol{x}) \\
&> \sum_{\boldsymbol{x} \in \mathcal{R}^c \cap A_n^\epsilon} 2^{-n(D+\epsilon)} P_n(\boldsymbol{x}) \\
&= 2^{-n(D+\epsilon)} P_n(\mathcal{R}^c \cap A_n^\epsilon) \\
&\geq 2^{-n(D+\epsilon)} (P_n(A_n^\epsilon) - P_n(\mathcal{R})) \\
&\geq 2^{-n(D+\epsilon)} (1 - 2\alpha)
\end{aligned}$$

Hence,$\forall\, n \geq N_{\epsilon,\alpha}$,

$$\beta_{n,\alpha} = \min_{\mathcal{R}, P_n(\mathcal{R}) < \alpha} Q_n(\mathcal{R}^c) > 2^{-n(D+\epsilon)}(1 - 2\alpha),$$

which in turn gives,

$$\liminf_{n \to \infty} \frac{\log_2 \beta_{n,\alpha}}{n} \geq -D - \epsilon, \ \forall\, \epsilon > 0.$$

Therefore,

$$\liminf_{n \to \infty} \frac{\log_2 \beta_{n,\alpha}}{n} \geq -D. \tag{33}$$

Combining Equation (32) and Equation (33) we get the desired result. □

C Proof of Lemma 6

Proof Without loss of generality assume that $s = 0$. Note that

$$\begin{aligned}
\sum_{r=1}^{N-1} \Pr(W_0 < W_r) &\geq \Pr(\exists\, 1 \leq k \leq (N-1) \ s.t. \ W_0 < W_k) \\
&= \Pr(\max_{0 \leq i \leq N-1} W_i \neq W_0) \\
&\geq \Pr(W_0 \leq W_l), \ \ \forall\, l = 1, \ldots, N-1.
\end{aligned}$$

$\forall\, k = 1, \ldots, N-1$, we have, $\Pr(W_0 < W_k) = \Pr(W_0 - W_k < 0)$ and

$$W_0 - W_k \sim \mathcal{N}(\mu_0 - \mu_k, \frac{1}{n}(\sigma_{00} + \sigma_{kk} - 2\sigma_{0k})),$$

i.e.,

$$R_k := \frac{\sqrt{n}}{\sqrt{\sigma_{00} + \sigma_{kk} - 2\sigma_{0k}}}(W_0 - W_k) \sim \mathcal{N}(\sqrt{n}\delta_k, 1).$$

Hence,

$$\Pr(W_0 - W_k < 0) = \Pr(R_k < 0) = \Phi(-\sqrt{n}\delta_k). \tag{34}$$

Thus,

$$\begin{aligned}
\sum_{r=1}^{N-1} \Phi(-\sqrt{n}\delta_r) &\geq \Pr(\exists\, 1 \leq k \leq (N-1) \ s.t. \ W_0 < W_k) \\
&\geq \Phi(-\sqrt{n}\delta_l), \ \ \forall\, l = 1, \ldots, N-1.
\end{aligned}$$

which gives

$$\begin{aligned}\sum_{r=1}^{N-1} \Phi(-\sqrt{n}\delta_r) &\geq \Pr(\exists\ 1 \leq k \leq (N-1)\ s.t.\ W_0 < W_k) \\ &\geq \max_{1 \leq l \leq N-1} \Phi(-\sqrt{n}\delta_l) = \Phi(-\sqrt{n}\delta)\end{aligned}$$

Let, Now, we shall show that the ratio of the two extremes in the inequality stated above goes to 1 as n goes to infinity, i.e., for large n they are quite close and then we can approximate the middle term by the right-hand extreme. The limit we get by using *L'Hospital's Rule* is as follows,

$$\begin{aligned}\lim_{n\to\infty} \frac{\Phi(-\sqrt{n}\delta)}{\sum_{r=1}^{N-1} \Phi(-\sqrt{n}\delta_r)} &= \lim_{n\to\infty} \frac{\delta e^{-\frac{n\delta^2}{2}}}{\sum_{r=1}^{N-1} \delta_r e^{-\frac{n\delta_r^2}{2}}} \\ &= 1\end{aligned}$$

as $\delta < \delta_k$, $\forall k \neq 0, j$. So,

$$\Pr(\max_{0 \leq i \leq N-1} W_i \neq W_0) = \Pr(\exists\ 1 \leq k \leq (N-1)\ s.t.\ W_0 < W_k) \approx \Phi(-\sqrt{n}\delta).$$

References

1. Nadhem J. AlFardan, Daniel J. Bernstein, Kenneth G. Paterson, Bertram Poettering, and Jacob C. N. Schuldt. On the security of RC4 in TLS. In Samuel T. King, editor, *Proceedings of the 22th USENIX Security Symposium, Washington, DC, USA, August 14-16, 2013*, pages 305–320. USENIX Association, 2013.
2. Jean-Philippe Aumasson, Simon Fischer, Shahram Khazaei, Willi Meier, and Christian Rechberger. New features of latin dances: Analysis of salsa, chacha, and rumba. In Kaisa Nyberg, editor, *Fast Software Encryption, 15th International Workshop, FSE 2008, Lausanne, Switzerland, February 10-13, 2008, Revised Selected Papers*, volume 5086 of *Lecture Notes in Computer Science*, pages 470–488. Springer, 2008.
3. Thomas Baignères, Pascal Junod, and Serge Vaudenay. How far can we go beyond linear cryptanalysis? In Pil Joong Lee, editor, *Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004, Proceedings*, volume 3329 of *Lecture Notes in Computer Science*, pages 432–450. Springer, 2004.
4. Thomas Baignères, Pouyan Sepehrdad, and Serge Vaudenay. Distinguishing distributions using chernoff information. In Swee-Huay Heng and Kaoru Kurosawa, editors, *Provable Security - 4th International Conference, ProvSec 2010, Malacca, Malaysia, October 13-15, 2010. Proceedings*, volume 6402 of *Lecture Notes in Computer Science*, pages 144–165. Springer, 2010.
5. Subhadeep Banik and Takanori Isobe. Cryptanalysis of the full spritz stream cipher. In Thomas Peyrin, editor, *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, volume 9783 of *Lecture Notes in Computer Science*, pages 63–77. Springer, 2016.
6. Riddhipratim Basu, Shirshendu Ganguly, Subhamoy Maitra, and Goutam Paul. A complete characterization of the evolution of RC4 pseudo random generation algorithm. *J. Mathematical Cryptology*, 2(3):257–289, 2008.

7. Richard E. Blahut. *Principles and Practice of Information Theory.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1987.
8. Céline Blondeau, Benoît Gérard, and Jean-Pierre Tillich. Accurate estimates of the data complexity and success probability for various cryptanalyses. *Des. Codes Cryptography*, 59(1-3):3–34, 2011.
9. George Casella and Roger Berger. *Statistical Inference.* Duxbury Resource Center, June 2001.
10. Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing).* Wiley-Interscience, 2006.
11. Patrik Ekdahl and Thomas Johansson. Distinguishing attacks on sober-t16 and t32. In Joan Daemen and Vincent Rijmen, editors, *Fast Software Encryption, 9th International Workshop, FSE 2002, Leuven, Belgium, February 4-6, 2002, Revised Papers*, volume 2365 of *Lecture Notes in Computer Science*, pages 210–224. Springer, 2002.
12. Scott R. Fluhrer and David A. McGrew. Statistical analysis of the alleged RC4 keystream generator. In Bruce Schneier, editor, *Fast Software Encryption, 7th International Workshop, FSE 2000, New York, NY, USA, April 10-12, 2000, Proceedings*, volume 1978 of *Lecture Notes in Computer Science*, pages 19–30. Springer, 2000.
13. Christina Garman, Kenneth G. Paterson, and Thyla Van der Merwe. Attacks only get better: Password recovery attacks against RC4 in TLS. In Jaeyeon Jung and Thorsten Holz, editors, *24th USENIX Security Symposium, USENIX Security 15, Washington, D.C., USA, August 12-14, 2015.*, pages 113–128. USENIX Association, 2015.
14. Sourav Sen Gupta, Subhamoy Maitra, Goutam Paul, and Santanu Sarkar. (non-)random sequences from (non-)random permutations - analysis of RC4 stream cipher. *J. Cryptology*, 27(1):67–108, 2014.
15. Allan Gut. *Probability: a graduate course. 2nd ed.* New York, NY: Springer, 2nd ed. edition, 2013.
16. G.H. Hardy, J.E. Littlewood, and G. Pólya. *Inequalities.* Cambridge Mathematical Library. Cambridge University Press, 1952.
17. S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:49–86, 1951.
18. Subhamoy Maitra, Goutam Paul, Shashwat Raizada, Subhabrata Sen, and Rudradev Sengupta. Some observations on HC-128. *Des. Codes Cryptography*, 59(1-3):231–245, 2011.
19. Itsik Mantin. Predicting and distinguishing attacks on RC4 keystream generator. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 491–506. Springer, 2005.
20. Itsik Mantin and Adi Shamir. A practical attack on broadcast RC4. In Mitsuru Matsui, editor, *Fast Software Encryption, 8th International Workshop, FSE 2001 Yokohama, Japan, April 2-4, 2001, Revised Papers*, volume 2355 of *Lecture Notes in Computer Science*, pages 152–164. Springer, 2001.
21. J. Neyman and E. S. Pearson. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231:pp. 289–337, 1933.
22. Subhabrata Samajder and Palash Sarkar. Another look at normal approximations in cryptanalysis. *IACR Cryptology ePrint Archive*, 2015:679, 2015.
23. Subhabrata Samajder and Palash Sarkar. Rigorous upper bounds on data complexities of block cipher cryptanalysis. *IACR Cryptology ePrint Archive*, 2015:916, 2015.

24. Paul Stankovski, Sushmita Ruj, Martin Hell, and Thomas Johansson. Improved distinguishers for HC-128. *Des. Codes Cryptography*, 63(2):225–240, 2012.
25. Hongjun Wu. The stream cipher HC-128. In Matthew J. B. Robshaw and Olivier Billet, editors, *New Stream Cipher Designs - The eSTREAM Finalists*, volume 4986 of *Lecture Notes in Computer Science*, pages 39–47. Springer, 2008.