Automatic solution in depth of one time pads

Automatisk parallellforcering av blankettchiffer

Martin Ekerå ekera@kth.se
Björn Terelius terelius@kth.se

March 14, 2008

The Royal Institute of Technology
Engineering Physics
Stockholm
Sweden

Abstract

We review, implement and evaluate a recent method for breaking one time pad ciphers when the same pad is reused. The attack can also be applied to a number of other ciphers when they are used incorrectly. Some possible enhancements to the algorithm are discussed, including how to exploit multiple reuses of the same pad, what happens when the the encryption function is not a group operation, how to parallelize the algorithm and how to deal with gaps in the ciphertexts.

The method works by considering the plaintexts $\bar{p}_1$ and $\bar{p}_2$ as generated by n-gram Markov models L_1 and L_2 chosen in such a way so as to resemble the real languages in which the texts were written. Given the models, the method implicitly builds a hidden Markov model of the cross product $L_1 \times L_2$. The most likely recovery of the plaintexts given the ciphertexts $\bar{c}_1, \bar{c}_2$ corresponds to the most likely path through the HMM subject to the constraints that $\bar{p}_1 \oplus \bar{k} = \bar{c}_1$ and $\bar{p}_2 \oplus \bar{k} = \bar{c}_2$ for some choice of $\bar{k}$.

In principle, the problem can be solved by the Viterbi algorithm, but in order to avoid the prohibitive time and memory requirements of exploring the complete state graph, we prune the graph by keeping only the N most probable states in each iteration. Although this sacrifices the optimality of the Viterbi algorithm, it works quite well in practice, successfully recovering about $97-99\%$ of all characters depending on the texts. It also reduces the complexity to $\mathcal{O}(\log(|A|)N\min(|\bar{c}_1|, |\bar{c}_2|))$ memory and $\mathcal{O}(|A|N\min(|\bar{c}_1|, |\bar{c}_2|))$ time, where A is the set of all possible message characters.

Contents

1 Introduction

In the next section, we describe the Vernam cipher, its invention and its entry into the world of cryptography. After a formal definition, we also show that the *one time pad* is provably secure if a random, uniformly distributed key is selected, but weak if the key is reused. After that, we discuss some notable general *stream ciphers* and how implementation flaws can render them insecure. Readers already familiar with these concepts can skip the first sections and start directly at section four, without missing anything.

In sections four to six, we describe a method, originally suggested by Mason et al. [7], which may be used to break one time pads and related ciphers when the key is reused. We show how the method can be implemented efficiently in practice, and also that it applies to binary operations other than exclusive-or, and languages other than English.

2 The Vernam cipher

We begin with an elementary presentation of the Vernam cipher and its properties, since it is the model for most modern stream ciphers. The Vernam cipher, patented by Gilbert Vernam in 1919 [10], encrypts a plaintext character selected from some alphabet by combining it under an invertible operation with a key character selected from the same alphabet.

2.1 A brief historical review

In his patent [10], Vernam describes an ciphering apparatus designed primarily for protecting the confidentiality of messages transmitted over telegraph wires. The 5-bit Baudot code is used to code characters as a sequence of electric pulses, suitable for transmission over the wires.

Vernam suggested that the key and message to be encrypted or decrypted be fed into the apparatus using perforated paper tapes. The message and key characters would be combined under the exclusive-or operation, which apart from being bitwise and simple to implement in electric circuits, had the added advantage of being its own inverse, allowing the exact same apparatus to be used for both encryption and decryption.

2.1.1 Generating key sequences and using keys

Little is said in the patent about how the key sequence should be generated. Vernam states that the key characters should preferably be selected at random but that if desired any series of letters or words may be selected. Nothing is said on whether or not the same key may be used to encrypt multiple messages.

As the cipher is described in the patent, it is assumed that the key tape is of equal length to the message tape, which is often impractical from a logistical perspective. However, Vernam initially believed that repeating a short key over and over again, simply by gluing the two ends of the key tape to each other, would solve the problem and provide sufficient cryptographic security. [2] Needless to say, Vernam soon realized that using a short repeating key is equivalent to using a Vigenère cipher, which had already been broken independently by Kasiski and Babbage in the late 19th century.

To provide a longer key, Vernam then suggested that two key sequence tapes of co-prime lengths n and m should be used in parallel. The actual key character is the exclusive-or of the current character on each tape. If both tapes are stepped in sequence with the message tape, the key repeats [2] [11] after nm steps, forming a sufficiently long key sequence for most messages.

Such a machine was fabricated and entered into service of the United States Army, until Major Joseph Mauborgne showed [2] that this kind of cipher was susceptible to the same kind of cryptanalytic method usually employed to break running-key ciphers.

2.1.2 The invention of the one time pad

Instead, Joseph Mauborgne and William Friedman suggested that a unique random key be used for each message transmitted, in which each character is selected independently from a random uniform distribution over the whole alphabet. In so doing, they invented the *one time pad cipher*, also known [3] [2] as the Mauborgne-Vernam cipher. Furthermore, Mauborgne and Friedman conjectured that a cipher which uses a random one time key is unconditionally secure; a conjecture first proved [9] by Shannon in 1949.

2.2 Formal description

Formally, let the alphabet A be a group under the commutative [1] operation $\oplus$. Then, we define encryption $E : A \times A \to A$ and decryption $D : A \times A \to A$ as

$$E(x, k) = x \oplus k \quad D(c, k) = c \ominus k$$

where k is a key character, x is the plaintext character, c is the ciphertext character, and $\ominus$ is the inverse of the $\oplus$ operation.

The definition may be extended to sequences of plaintext characters $\bar{x}$ or ciphertext characters $\bar{c}$ and key characters $\bar{k}$, by applying the operations to each pair of characters in the two input sequences. Then, we have

$$E(\bar{x}, \bar{k}) = \bar{x} \oplus \bar{k} \quad D(\bar{c}, \bar{k}) = \bar{c} \ominus \bar{k}$$

2.3 Security and key reuse

Lemma 2.1 *If the key k is uniformly distributed, then the cipher text c of any message m encrypted under k is uniformly distributed as well.*

Proof

$$\begin{aligned} P(c) &= \sum_{x \in A} P(c \,|\, x) P(x) = \sum_{x \in A} P(k = c \ominus x) P(x) \\ &= \sum_{x \in A} \frac{1}{|A|} P(x) = \frac{1}{|A|} \sum_{x \in A} P(x) = \frac{1}{|A|} \end{aligned}$$

since $P(c \,|\, x)$ is precisely the probability that $k = c \ominus x$. □

[1] Although commutativity is not strictly needed in this case, we require it to avoid some otherwise cumbersome expressions.

Theorem 2.2 *If the key sequence k is chosen from a uniform distribution and used only once, then the Vernam cipher does not leak any information about the contents of the plaintext. Stated differently, if x is the message and c the cipher text, then $P(x \mid c) = P(x)$.*

Proof

$$P(x \mid c) = \frac{P(x \wedge c)}{P(c)} = \frac{P(c \mid x)P(x)}{P(c)} = \frac{P(c \mid x)}{P(c)}P(x)$$

Using lemma 2.1, we see that $P(c) = P(c \mid x) = 1/|A|$, so $P(x \mid c)$ is indeed equal to $P(x)$. □

Theorem 2.3 *If the same key sequence is used to encrypt multiple plaintexts, the Vernam cipher leaks the difference $x \ominus y$ between the plaintexts.*

Proof
If x and y are encrypted using the same key k, then

$$E(x,k) \ominus E(y,k) = (x \oplus k) \ominus (y \oplus k) = x \oplus k \ominus k \oplus y = x \ominus y$$

□

The theorem above shows that if one of the plaintexts is recovered, the other can immediately be determined from $E(x,k) \ominus E(y,k)$.

Theorem 2.4 *If the same key sequence k is used to encrypt the plaintexts x and y, the difference $x \ominus y$ contain the same information about the plaintexts as the ciphertexts $E(x,k)$ and $E(y,k)$ do.*

Proof
Given $x \ominus y$ we can generate all possible ciphertexts having this difference by choosing the first ciphertext C_x as $x \ominus y \oplus C_y$ where the second ciphertext C_y is arbitrary. Clearly

$$C_x \ominus C_y = (x \ominus y \oplus C_y) \ominus C_y = x \ominus y$$

so the ciphertext pairs indeed have the correct difference. Let $C_y = y \oplus k'$ for some unknown k'. Then

$$C_x = x \ominus y \oplus C_y = x \ominus y \oplus y \oplus k' = x \oplus k'$$

so the different ciphertext pairs (C_x, C_y) we generate simply correspond to encryptions of x and y with different keys. While we can not determine the actual ciphertexts $E(x,k)$ and $E(y,k)$, knowing them in addition to the difference $x \ominus y$ does not give more information about the plaintext, but instead determine the key sequence k. □

3 Stream ciphers

The Vernam-Mauborgne cipher is often hard to implement in practice, due to the difficulties associated with generating, distributing and managing large random key sequences which need to be kept secret. Instead, *stream ciphers* may be used to mimic the Vernam-Mauborgne cipher.

A synchronous stream cipher expands a short random key into a large pseudo-random key sequence which is then combined character-wise with the plaintext under some invertible operation to form the ciphertext.

Although there are other classes of stream ciphers, we will only consider synchronous stream ciphers in this report and will therefore not write out the word synchronous henceforth.

3.1 The notion of internal and external keys

The key is usually split into two parts; the internal and external key. The internal key is kept secret and used for a certain time period, whilst the external key is public and selected at random for each message to be encrypted.

3.2 Constructing stream ciphers from block ciphers

Block ciphers constitute a class of simple cryptographic primitives, which may be used to encrypt fixed-length input plaintext message into fixed-length ciphertext messages and vice versa.

Definition 3.1 *A block cipher is a keyed substitution cipher, defined over the elements in* $X = \{0,1\}^l$ *for some* $l > 0$ *and for keys* $k \in K = \{0,1\}^m$*. Let*

$$E(x,k) : X \times K \to X$$

denote the encryption function of the block cipher. Since $\{0,1\}^l$ *is a block of* l *bits,* l *is called the* block length*. Similarly,* m *is called the key length.*

A *mode of operation* describes how a block cipher may by used to encrypt or decrypt arbitrary length messages. Thus, a common method of synchronous stream cipher construction is to run a block cipher in a mode of operation which depends only on the encryption key, and not on the plaintext or ciphertext.

Then, provided the output of the block cipher is pseudo-random, it will generate a pseudo-random sequence of l bit blocks $(k_0, k_1, \ldots, k_{n-1}) \in X^n$, which may be input as the key stream into a Vernam cipher setup. Two such modes of operation are described below.

3.2.1 Block ciphers in output feedback mode

Let $\kappa_I \in K$ be the internal key and let $\kappa_E \in X$ be the external key. In output feedback mode, the key sequence is then given as $(k_0, .., k_{n-1}) \in X^n$, where

$$k_i = \begin{cases} E(\kappa_E, \kappa_I) & i = 0 \\ E(k_{i-1}, \kappa_I) & i > 0 \end{cases}$$

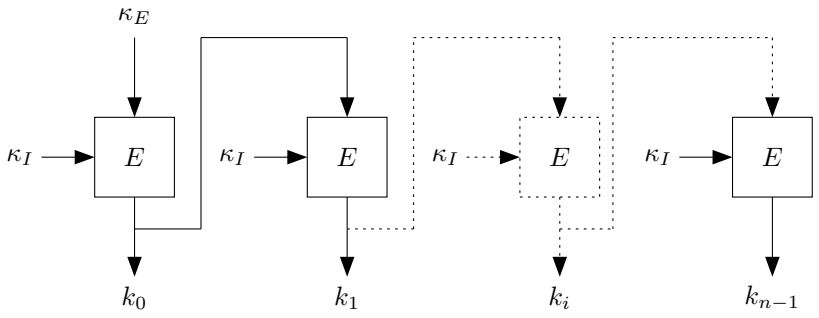

Figure 1: A block cipher in OFB mode.

3.2.2 Block ciphers in counter mode

Let $\kappa_I \in K$ be the internal key and let $\kappa_E \in X$ be the external key. In counter mode, the key sequence is given as $(k_0, .., k_{n-1}) \in X^n$, where

$$k_i = E(\kappa_E + i, \kappa_I) \qquad i \geq 0$$

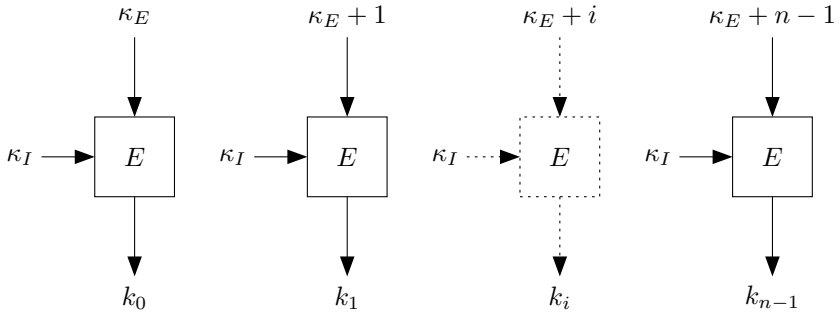

Figure 2: A block cipher in CTR mode.

3.3 Weaknesses of stream ciphers

Stream ciphers with pseudo-random key stream are secure if and only if the pseudo-random key is indistinguishable from a truly random key stream, and key streams are never reused.

If the same internal and external key combination is used to encrypt multiple plaintexts, then the stream ciphers do of course exhibit the same weaknesses as the Vernam-Mauborgne cipher when the key is reused, allowing the immediate recovery of $x \ominus y$.

3.4 Recent weaknesses due to key reuse

There have been vulnerabilities due to incorrect usage of external keys in a number of software programs and standards including e.g. WinZip [6], Microsoft Office [4] and the wired equivalent privacy (WEP) algorithm [8] specified in the IEEE 802.11 wireless network standard [5].

3.4.1 WinZip AE-2

The WinZip AE-2 encryption scheme is based on AES in CTR mode. The key is derived from a user password by combining it with a random salt. Since the salt is only 64 bits and the counter always start at 0, one can expect a key reuse after about 2^{32} files. Since it encrypts each file in an archive independently, it is not entirely impossible to find 2^{32} files encrypted with the same password. Although this is a flaw, it is not very easy to exploit since the files are compressed prior to encryption, removing much of the redundancy needed for cryptanalysis.

3.4.2 Microsoft Office

In our opinion, the vulnerability in Microsoft Office is more severe. Microsoft Office 2002 uses the RC4 stream cipher for encrypting documents. The problem arises when an encrypted document is edited and saved using the same password. When a file is edited, Word uses the same IV resulting in the two versions being combined under the exclusive-or operation with the same key stream. If the changes to the file involved an insertion near its beginning, it is probable that a large portion of the remainder of the file may be recovered.

3.4.3 Wireless equivalent privacy

The wired equivalent privacy also uses the RC4 cipher with a 24 bit IV to encrypt traffic. To prevent key reuse attacks, it is suggested in the IEEE standard that the IV be changed between every packet. However, if the IV changes randomly one can expect a key reuse after about 2^{12} packets, and in any case all possible IVs will have been exhausted after 2^{24} packets. Since user passwords changes only infrequently, it is likely that an IV reuse will result in a full key stream reuse.

4 Language models

As has already been stated, the objective of this paper is to describe an algorithm for the recovery of $\bar{x}$ and $\bar{y}$, when $\bar{z} = \bar{x} \ominus \bar{y}$ is known.

In general, this is not possible, since a message of length $2 \cdot \min(|\bar{x}|, |\bar{y}|)$ is compressed to a message of length $\min(|\bar{x}|, |\bar{y}|)$, and hence information is lost. If x and y are messages in some redundant language, however, enough information may still be preserved in $\bar{z}$ for $\bar{x}$ and $\bar{y}$ to be recovered.

If the characters in $\bar{x}$, $\bar{y}$ belong to some alphabet A, there are a total of $|A|^{|\bar{z}|}$ possible plaintext pairs $\bar{x}$, $\bar{y}$ which have the correct difference $\bar{z}$. Since there is no unique solution, the best we can do is to find the most "probable" plaintexts, given knowledge of the language they were written in. Hence, we require some method of associating a probability to each text in the language.

As it is difficult to estimate or even define these probabilities, we will instead assume that the texts belong to a language model which is selected to mimic the natural language.

Definition 4.1 *A language model is an assignment of probabilities $P(\bar{x})$ to sequences of characters $\bar{x}$.*

From the probability $P(\bar{x})$, we can compute other probabilities.

Definition 4.2 *Let x_i denote the ith element of $\bar{x}$ and x_i^j denote the substring $(x_i x_{i+1} \dots x_j)$. Then we define the probability $P(x_i|x_j^{i-1})$ of x_i following x_j^{i-1} as*

$$P(x_i|x_j^{i-1}) = \frac{P(x_j^i)}{P(x_j^{i-1})}$$

4.1 Collecting statistical information

We would of course like the language model to resemble the real language as closely as possible. Obviously, one method of computing $P(\bar{x})$ would then be to collect occurrence frequencies for all texts ever written in the language, and derive a probability distribution from them. This is of course impossible for a number of reasons. Instead we choose a *corpus* of texts and collect occurrence frequencies for short substrings of length n, commonly called n-grams.

Definition 4.3 *An n-gram is an ordered sequence of n characters $x_i^{i+n-1} = (x_i, x_{i+1}, \dots, x_{i+n-1})$. We will use x^n to denote an n-gram when the position of the n-gram in a larger context is irrelevant.*

Also, the word monogram is used to denote 1-grams, bigram to denote 2-grams and trigrams to denote 3-grams.

Unfortunately, using the frequency directly as an estimate of the probability does not give a realistic model of the real language. This is because in our selection of a corpus, we select only a tiny sample of the possible texts meaning that many n-grams will have frequency 0 in our corpus even though they are perfectly valid as parts of the real language. To avoid this it is customary to reserve a small fraction of the total probability for unseen n-grams; a technique known as *smoothing*. For the time being, we shall not concern ourselves with how the smoothing is done, postponing a discussion of methods for estimating the probabilities to section 6.4.

At this point we have only estimated the probabilities for short substrings. Of course, we will need a method to extend our probabilities estimates to longer strings, which we chose to do by means of a Markov model.

4.2 The Markov assumption

In an n-Markov model, the probability of a character x_i depends only on the n previous characters $x_{i-n-1}, \dots, x_{i-1}$.

The Markov assumption is fairly intuitive, since the most recent few characters will greatly affect the probability of a certain continuation of the current text fragment. After a few characters, however, the effect is no longer visible, and so it is reasonable to set some limit n at which it is considered negligible.

In a Markov language model, there are $|A|^n$ nodes representing all possible n-grams in the message $\bar{x}$. A total of $|A|$ arcs leave each node x_0^{n-1}, leading to $|A|$ nodes x_1^n. The transition probability $P(x_n|x_0^{n-1})$ is associated with each arc.

Two models may be created; one for the transitions in $\bar{x}$ and one for the transitions in $\bar{y}$, with transition probabilities given by the P_x and P_y functions.

Another, and perhaps better solution, is to create a single model with $|A|^{2n}$ nodes, where each node (x^n, y^n) contains both an n-gram in $\bar{x}$ and an n-gram in

$\bar{y}$. Each node has $|A|$ outgoing arcs leading to $|A|$ nodes, each with the associated transitional probability given by the joint probability function $P_x \cdot P_y$.

4.2.1 Hidden Markov models

The problem with using an ordinary Markov model is that it is difficult to include the constraint $x_{n-1} \ominus y_{n-1} = z_{n-1}$ in the model. To remedy this we use a hidden Markov model.

Definition 4.4 *A hidden Markov model or HMM is a Markov model in which the internal state is invisible to an observer. An observer will only see certain output symbols with probabilities that depend on the internal state. In general, a HMM is fully specified by five components; the set of internal states, the set of observable symbols, the transition probabilities p_{ij} of going from state i to state j, the emission probabilities $e_i(a)$ of outputting the symbol a from the state i, and finally the initial distribution π_i or probability of starting the HMM in state i.*

We define a HMM that associates a single output character $x_{n-1} \ominus y_{n-1}$ with probability one to each state (x^n, y^n) in the model. The transition probabilities are estimated in the same way as in the previous section.

We can now rephrase our original problem of separating the text $\bar{z} = \bar{x} \ominus \bar{y}$ as seeking the path of n-grams in the HMM which maximizes the probability of our observed sequence $\bar{z} = \bar{x} \ominus \bar{y}$.

5 Plaintext recovery

Let L_x and L_y be language models for the texts $\bar{x}$ and $\bar{y}$ respectively. (We do not necessarily assume that they are written in the same language, but we do require that they use the same alphabet A and were encrypted with the same group operation). Given $\bar{z}$ we seek the most likely plaintexts $\bar{x}$ and $\bar{y}$ whilst respecting the constraint $\bar{z} = \bar{x} \ominus \bar{y}$.

5.1 A naive algorithm

Had we not made a Markov assumption, assuming instead that the probability of the next character being a depended on all the previous characters in the text, we would have obtained the following algorithm:

For the first character, we can estimate the probabilities $P_x(x_0)$ and $P_y(y_0)$, from the monogram counts in L_x and L_y. Specifying the value of x_0 fixes the value of y_0, since $y_0 = x_0 \ominus z_0$, and thus there are only $|A|$ possible choices for the character pair (x_0, y_0). The joint probability of each such pair is given by the expression

$$P(x_0, y_0) = P_x(x_0) \cdot P_y(y_0)$$

For each character pair (x_0, y_0) there are again only $|A|$ possible choices for the next pair (x_1, y_1), giving a total of $|A|^2$ possible message pairs of two characters. The probability for each such message (x_0^1, y_0^1) is given by

$$P(x_0^1, y_0^1) = P(x_0, y_0) \cdot P_x(x_1 \,|\, x_0) \cdot P_y(y_1 \,|\, y_0)$$

In general, there are $|A|^l$ possible message pairs (x_0^{l-1}, y_0^{l-1}) of length $l = |\bar{x}| = |\bar{y}|$ characters. The probability of each such pair is computed recursively

$$P(x_0^{l-1}, y_0^{l-1}) = P(x_0^{l-2}, y_0^{l-2}) \cdot P_x(x_{l-1} \,|\, x_0^{l-2}) \cdot P_y(y_{l-1} \,|\, y_0^{l-2})$$

with monograms as a base case. The message pair (x_0^{l-1}, y_0^{l-1}), which maximizes $P(x_0^{l-1}, y_0^{l-1})$ is the most likely $\bar{x}, \bar{y}$.

The number of states in the algorithm grows exponentially, effectively trying all $|A|^l$ possible message pairs of length l. Furthermore, the algorithm needs an extremely large language model, to be able to compute accurate estimates of the probabilities. It is obviously not feasible to use this method even for short messages. This provides further motivation for the Markov assumption that the next character only depends on some limited number of previous characters.

5.2 The Viterbi algorithm

The Viterbi algorithm was originally introduced by A. Viterbi [12] as a decoding algorithm for a type of error-correcting codes. It has since been applied to many other problems like DNA analysis, target tracking, speech recognition and optical character recognition.

In fact, the Viterbi algorithm is a very general algorithm for computing the sequence of hidden states in a HMM most likely to generate a particular sequence of outputs. Stated formally, in a HMM with known initial distribution π_i, transition probabilities p_{ij} and emission probabilities $e_i(a)$, and a given sequence of outputs $(a_0, a_1, \ldots, a_T)$, the Viterbi algorithm computes a sequence of hidden states $(q_0, q_1, \ldots, q_T)$ that maximize the conditional probability

$$P(\bar{q} \,|\, \bar{a}) = P((q_0, q_1, \ldots, q_T) \,|\, (a_0, a_1, \ldots, a_T))$$

Notice that

$$\arg\max_{\bar{q}} P(\bar{q} \,|\, \bar{a}) = \arg\max_{\bar{q}} \frac{P(\bar{q} \wedge \bar{a})}{P(\bar{a})} = \arg\max_{\bar{q}} P(\bar{q} \wedge \bar{a})$$

i.e. the argument which maximizes $P(\bar{q} \wedge \bar{a})$, since $P(\bar{a})$ is independent of $\bar{q}$. Hence it suffices to maximize $P(\bar{q} \wedge \bar{a})$, and to do this we compute

$$\delta_t(j) = \begin{cases} \pi_j e_j(a_0) & t = 0 \\ \max_i(\delta_{t-1}(i) p_{ij} e_j(a_t)) & t > 0 \end{cases}$$

and

$$\psi_t(j) = \begin{cases} \texttt{NULL} & t = 0 \\ \arg\max_i(\delta_{t-1}(i) p_{ij} e_j(a_t)) & t > 0 \end{cases}$$

Lemma 5.1 *$\delta_t(i)$ is the probability of the most likely path of t states ending with i to generate the output $(a_0, a_1, \ldots, a_{t-1}, a_t)$. More precisely*

$$\delta_t(i) = \max_{q_0, q_1, \ldots, q_{t-1}} P((q_0, q_1, \ldots, q_{t-1}, q_t = i) \wedge (a_0, a_1, \ldots, a_{t-1}, a_t))$$

Proof
For $t = 0$, the statement is true by definition. Assume it is valid for some t. Then it is also valid for $t + 1$ since

$$\begin{aligned}\delta_t(i) &= \max_j \delta_{t-1}(j) p_{ji} e_i(a_t) \\ &= \max_j \max_{q_0,q_1,\ldots,q_{t-2}} P((q_0, q_1, \ldots, q_{t-1} = j) \wedge (a_0, a_1, \ldots, a_{t-1})) p_{ji} e_i(a_t) \\ &= \max_{q_0,q_1,\ldots,q_{t-1}} P((q_0, q_1, \ldots, q_{t-1}, q_t = i) \wedge (a_0, a_1, \ldots, a_{t-1}, a_t))\end{aligned}$$

so the statement is valid for all t by induction. □

Using this lemma it is obvious that the sought sequence $\bar{q}$ of hidden states is one that satisfies $P(\bar{q} \wedge \bar{a}) = \max_i \delta_T(i)$. We set the last element q_T in the sequence $\bar{q}$ to $\arg\max_i \delta_T(i)$ and use the function $\psi_t(j)$ to backtrack by setting $q_{t-1} = \psi_t(q_t)$.

Theorem 5.2 *The sequence $\bar{q}$ obtained in this way indeed satisfies $P(\bar{q} \wedge \bar{a}) = \delta_T(q_T)$, and more generally.*

$$P((q_0, q_1, \ldots, q_{t-1}, q_t) \wedge (a_0, a_1, \ldots, a_{t-1}, a_t)) = \delta_t(q_t)$$

Proof
For $t = 0$, we have $P(q_0 \wedge a_0) = \pi_{q_0} e_{q_0}(a_0) = \delta_0(q_0)$. Assume that the statement is valid for some t. It follows that the formula is valid for $t + 1$

$$\begin{aligned}&P((q_0, q_1, \ldots, q_{t-1}, q_t) \wedge (a_0, a_1, \ldots, a_{t-1}, a_t)) = \\ &P((q_0, q_1, \ldots, q_{t-1}) \wedge (a_0, a_1, \ldots, a_{t-1})) p_{q_{t-1}q_t} e_{q_t}(a_t) = \\ &\delta_{t-1}(q_{t-1}) p_{q_{t-1}q_t} e_{q_t}(a_t) = \delta_t(q_t)\end{aligned}$$

By induction it holds for all t. The particular case $t = T$ gives $P(\bar{q} \wedge \bar{a}) = \delta_T(q_T)$, proving that the algorithm is correct. □

5.3 Applying the Viterbi algorithm

In our case, the initial distribution is estimated by using states of lengths 1 through $n-1$, the transition probabilities are given by the language model and the emission probability from the state (x^n, y^n) is fixed to 1 for the n-gram $z^n = x^n \ominus y^n$ and 0 for all other. For each n-gram pair, we consider all $|A|$ possible transitions leading to new n-gram pairs while obeying the constraint $x_{i+n} \ominus y_{i+n} = z_{i+n}$. The probability of the new state then becomes

$$\max_{(x_i,y_i)} \quad P(x_i^{i+n-1}, y_i^{i+n-1}) \cdot P_x(x_{i+n} \,|\, x_i^{i+n-1}) \cdot P_y(y_{i+n} \,|\, y_i^{i+n-1})$$

where $x_i \oplus y_i = z_i$. In each state, we store a pointer to the state most likely to have generated it, that is to

$$\arg\max_{(x_i,y_i)} \quad P(x_i^{i+n-1}, y_i^{i+n-1}) \cdot P_x(x_{i+n} \,|\, x_i^{i+n-1}) \cdot P_y(y_{i+n} \,|\, y_i^{i+n-1})$$

When the entire message has been processed, we will have $|A|^n$ candidates for the final n-gram. We choose the most likely of them, and follow the pointers

to find the most likely previous n-gram. We continue this backtracking until we have reached the beginning of the message.

Thus far, we have only shown how to extend the path from a given n-gram to $|A|$ following n-gram. When commencing execution however, we have no n-grams to extend, so we must generate an initial set. This may be done by iterating over all n-grams x_0^{n-1}, and associating with each n-gram the probability

$$P(x_0^i, y_0^i) = P(x_0^{i-1}, y_0^{i-1}) \cdot P_x(x_i \,|\, x_0^{i-1}) \cdot P_y(y_i \,|\, y_0^{i-1}) \qquad 0 < i < n$$

as we do in the naive algorithm described in section 5.1. This will produce a set of $|A|^n$ possible initial n-grams.

The operation of the Viterbi algorithm is illustrated in figure 3, where a complete set of $|A|^n = 2^3 = 8$ n-grams is constructed over the initial $n = 3$ steps, after which extension occurs in each step.

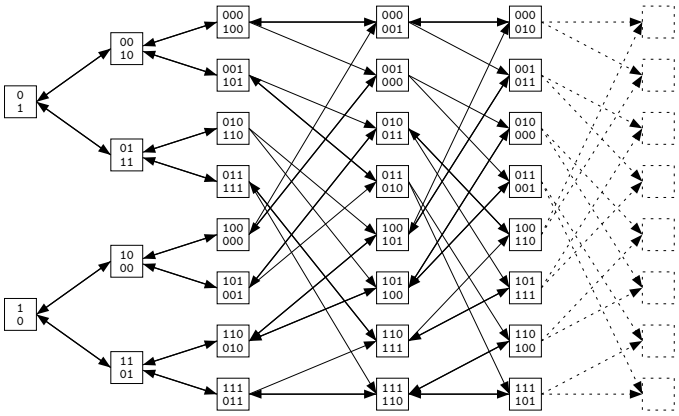

Figure 3: Viterbi graph for $A = \{0, 1\}$, $\bar{z} = \texttt{10010}$ and $n = 3$.

5.4 The Viterbi algorithm with pruning

For each character in the input, the Viterbi algorithm above considers $|A|$ transitions from each of the $|A|^n$ nodes, giving a total complexity of $\mathcal{O}(|A|^{n+1})$ per character. This complexity is prohibitively large, since we may wish to run the algorithm with $n = 7$ and $A = \mathbb{Z}_{256}$. To improve the running time, one might even be prepared to forsake the optimality of the Viterbi algorithm.

Observe that many of the $|A|^n$ states will have very low probability. Since it is unlikely that a state with low probability will be in the most probable path found by the Viterbi algorithm, one can limit the number of states in each iteration by pruning all but the N most probable nodes.

It may be easier to understand how the algorithm works in detail by consulting its pseudo-code, see figure 5. See also figure 4, which depicts how the pruned graph is implicitly computed by the algorithm. It should be compared to the implicit graph produced by the full Viterbi algorithm, shown in figure 3.

5.4.1 Complexity analysis

As may be seen in the pseudo-code, the algorithm iterates over $z_i = x_i \ominus y_i$, the difference between the two plaintexts. The code maintains a list of the N most

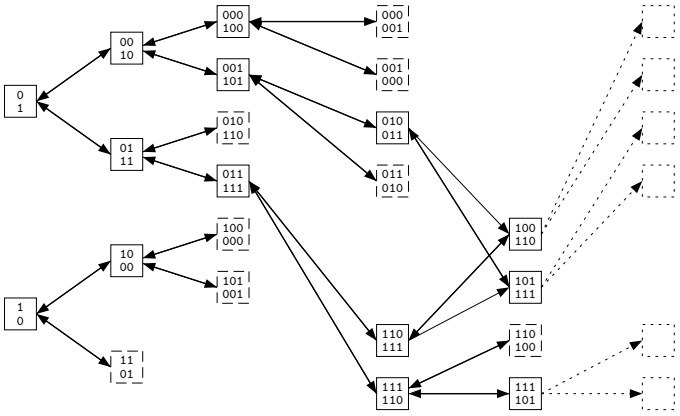

Figure 4: Pruned Viterbi graph for $A = \{0, 1\}$, $\bar{z} = \texttt{10010}$ and $n = N = 3$.

likely n-grams in each position i. To generate the most likely n-gram in position $i + 1$, the code generates the list of all n-grams that can be created from the N n-grams in position i by appending some character in the alphabet.

Clearly the generation of this table takes $\mathcal{O}(N|A|)$ operations, assuming that elementary operations on the n-grams can be done in constant time. Section 6 describes suitable data structures and efficient algorithms which may be used to implement the algorithm.

We must also compute the probabilities of these new elements, which in practise seems to limit the speed of the algorithm. On the other hand, obtaining good estimates of the probabilities is of the utmost importance to get accurate decryptions, so it is probably not worthwhile to use faster but less exact smoothing algorithms in this step.

In the pruning step, we select the N most probable elements from the entire list of length at most $N\,|A|$. This can be done in $\mathcal{O}(N\,|A|)$ expected (or worst-case) time using one of the selection algorithms described in 6.3.

The total time complexity becomes

$$\mathcal{O}(N\,|A|\,|\bar{z}|)$$

It should be noted, however, that the reduction in search complexity obtained by pruning the tree comes at the cost of not exploring all possible solutions. Thus, there is no guarantee that the original plaintext messages are recovered correctly.

6 Implementation

6.1 Representing n-grams

A natural way of representing the n-grams in a `C/C++` implementation of the Viterbi algorithm is to use a `char` array. For such an array, the complexity of prepending a character is $\mathcal{O}(n)$ where n is the length of array, since each element in the array must be shifted one step before the new character may be inserted.

The complexity of removing the first character is $\mathcal{O}(n)$ by a similar argument, as are the complexities for addition, subtraction or computing the exclusive-or.

```
// difference buffer z̄, n-gram length n, queue size N, language models L1, L2
viterbi(z̄, n, N, L1, L2)
  q̄ ← [("","",1,"")] // we begin in the empty state with probability 1
  for i = 1 to |z̄| // for each character in z̄
    q̄′ ← [ ] // clear q̄′
    // w1, w2 are n-grams, p partial path probability, b previous character
    for each state s = (w1, w2, p, b) in q̄
      for each character a in A
        w1′ ← append_character(w1, zi ⊕ a)
        w2′ ← append_character(w2, a)
        p1 ← transition_probability(w1, w1′, L1)
        p2 ← transition_probability(w2, w2′, L2)
        p′ ← p · p1 · p2 // the probability of the new state
        if (|w1′| > n)
          w1′ ← remove_first_character(w1′)
          w2′ ← remove_first_character(w2′)
          b′ ← get_first(w1) // the first character of w1
        else
          b′ ← NULL
        end
        if (q̄′[(w1′, w2′)] = NULL  or  q̄′[(w1′, w2′)].p < p′)
          q̄′[(w1′, w2′)] ← (w1′, w2′, p′, b′)
        end
      end
    end
    q̄ ← max(q̄′, N) // Keep only the N most probable states in q̄′
    M̄[i] ← q̄ // Save the state in the backtracking matrix M
  end
  return M
```

Figure 5: Pseudo-code for the Viterbi algorithm with pruning.

To improve the performance of these operations, we may encode the n-grams in integers, thereby eliminating the need of using for-loops and memory access instructions, as will be described below.

6.1.1 Using unsigned integers to represent strings

If we restrict the n-gram length to a maximum of 8 characters, n-grams may be represented as unsigned 64-bit integers, by encoding the first character in the first 8 bits of the integer, the second character in bits 8 through 15, and so forth for the remaining characters as illustrated below.

```
inline UI64 stringToInt(UI8 * str, int n) {
    UI64 rep = 0;
    for (unsigned i = 0; i < n; i++)
        rep ^= UI64(str[i]) << (8 * i);
    return rep;
}
```

Also note that if $n < 8$, the bits $8n + 1$ to 64 are all set to zero.

6.1.2 Removing and appending characters

To remove the first character in the n-gram, we may now simply shift its integer representation 8 steps to the right. Appending a character to the n-gram is equivalent to typecasting the character to a 64-bit integer, shifting it $8n$ positions to the left, and adding it to the integer representation of the n-gram.

```
inline UI64 appendCharacter(UI64 str, UI8 c, unsigned int n) {
    return str ^ (UI64(c) << (8 * n));
}

inline UI64 removeFirstCharacter(UI64 str) {
    return str >> 8;
}
```

6.1.3 The exclusive-or operation

Since the exclusive-or operation is bitwise, taking the exclusive-or of two n-grams is equivalent to taking the exclusive-or of their integer representations.

If the n-grams have length less than 8 characters, the last $8 - n$ position will be zero in both n-grams and so will the exclusive-or of these position, which is what we would expect.

```
#define XOR_STRINGS(x,y)   (x ^ y)
```

6.1.4 The byte-wise subtraction and addition operations

Performing byte-wise addition and subtraction is slightly trickier, since we need to prevent any carry effects from propagating past the 8th, 16th, .., 56th bit.

To accomplish this, we mask out the odd and even characters form the operand integers, add the odd and even characters separately, mask out any carry bits and add the results. In `C` the addition may be written as

```
#define MASK_EVEN   0xFF00FF00FF00FF00ULL
#define MASK_ODD    0x00FF00FF00FF00FFULL

#define MOD_ADD_STRINGS(x,y) \
    ((((x & MASK_EVEN) + (y & MASK_EVEN)) & MASK_EVEN) | \
     (((x &  MASK_ODD) + (y &  MASK_ODD)) &  MASK_ODD))
```

and the subtraction in the same manner, except that instead of masking out the odd characters in `x`, we set them to `0xFF` to absorb carries.

```
#define MOD_SUBTRACT_STRINGS(x,y) \
    ((((x |  MASK_ODD) - (y & MASK_EVEN)) & MASK_EVEN) | \
     (((x | MASK_EVEN) - (y &  MASK_ODD)) &  MASK_ODD))
```

Note that if we settle with being able to represent n-grams of maximum length 7 characters, then we may encode the characters with a single separator bit after each byte. Then, we may cancel carry effects by setting this single bit to either `1` or `0` depending on the operation we are performing.

Also note that if being restricted to a single processor architecture is acceptable, then it may be possible to gain significant performance by using an extended extension set capable of performing byte-wise addition of larger data types. An example of such an instruction set is SSE/SSE2.

6.2 Data structures for the language model

The language model consists of a set of eight hash tables, where each table maps the seen n-grams for $n = 1$ up to 8 to a 32-bit integer counter indicating the number of occurrences of the n-gram in the corpus. If an n-gram has not been seen, it will not be in the table.

In the previous section, we saw how n-grams could be represented as 64-bit unsigned integers to optimize performance. Representing n-grams in this manner has the added advantage of speeding up the computation of the hash function used in the hash table.

6.3 Pruning the states

To reduce the time complexity of the Viterbi-algorithm, we need to prune the states at each iteration of the algorithm by retaining only the N most probable states. A natural way to perform this pruning, is to sort the list of states with respect to their probabilities, retaining only the first N elements.

6.3.1 Using heaps to extract the N largest elements

If the heapsort algorithm is used to sort the n elements in the list, then clearly the sorting takes $\mathcal{O}(n \log n)$ operations, as it requires n insertions in the heap and each insertion takes $\mathcal{O}(\log n)$ time.

We can improve the performance of the heapsort algorithm by limiting the heap size to the N elements we are trying to find, since there is no reason to keep elements that are smaller than the Nth largest. We put the first N elements directly into a min-heap. For each subsequent element we insert it, and then remove the smallest element from the heap.

Since the heap will never have more than $N+1$ elements, each of the n insertions and removals takes only $\mathcal{O}(\log N)$ time. Thus the total time is $\mathcal{O}(n \log N)$, which is an improvement since N is much smaller than n.

Another improvement may be made by noticing that the element we insert will be removed immediately if it turns out to be smaller than all the other elements already in the heap. We can avoid these unnecessary insertions/removals by checking that the new element is larger than the smallest element in the heap prior to the insertion. Performing this check does not change the worst case behavior of the algorithm.

6.3.2 Using partitioning instead of heaps

Instead of modifying the heapsort algorithm, we can start out with the *quicksort* algorithm. The quicksort algorithm sorts the input list by partitioning it around a pivot element, and sorting the two partitions recursively.

The partitioning should move all larger elements to the first partition, and can be performed in linear time. Hopefully, the partitions will be of approximately equal size, giving an expected running time of $\mathcal{O}(n \log n)$.[2]

If we only want to extract the N largest elements, we may improve the algorithm's performance by processing only one of the two partitions at each iteration. Assume that the first partition contains k elements. If k is greater than the sought N elements, it will suffice to re-run the algorithm extracting the N largest from the first partition only. If $k = N$ then we are done, since we have succeeded in moving the N largest elements to the first partition. Otherwise, if $k < N$, we know that all elements in the first partition are amongst the N largest, so it remains to select the largest $N - k$ from the second partition.

The expected time complexity is thus $\mathcal{O}(n)$ operations.

6.4 Transitional probabilities and smoothing

In the description of the Viterbi algorithm in section 5.3, we assumed that given an n-gram x_m^{n-1}, we knew how to estimate the transitional probability

$$P(x_n \,|\, x_m^{n-1})$$

i.e. the probability that x_n follows the sequence x_m^{n-1} for $m < n$. One naive approximation of P is given by the expression

$$P(x_n \,|\, x_m^{n-1}) = \begin{cases} |A|^{-1} & \text{if } c(x_m^{n-1}) = 0 \\ \\ c(x_m^n)/c(x_m^{n-1}) & \text{otherwise} \end{cases}$$

where $c(x_m^n)$ is the number of *observed* occurrences of the sequence x_m^n in the corpus, and $|A|$ is the number of possible characters in the alphabet.

There are two obvious flaws in the naive model. First and foremost, all character sequences which were not in the training data will be considered equally likely. For instance, the sequence "quandary" will be considered equally probable as "wxyzwxyz", provided none of the two sequences were observed.

Secondly, a disproportionate amount of probability mass is associated with observed character sequences. It would be prudent to shift some probability mass from the observed sequences to the unseen sequences.

[2] For a more complete complexity analysis of quicksort, see [1].

6.4.1 An improved model

Let $0 \leq a, \lambda < 1$ be two model parameters. Define P as

$$P(x_n \,|\, x_m^{n-1}) = \begin{cases} (1-a) \cdot \hat{p}(x_n \,|\, x_m^{n-1}) + a \cdot P(x_n \,|\, x_{m+1}^{n-1}) & \text{if } m < n \\ (c(x_n) + \lambda)/(\sum_{y \in A} c(y) + \lambda \cdot |A|) & \text{if } m = n \end{cases}$$

where $\hat{p}(x_n \,|\, x_m^{n-1})$ for $m < n$ is defined as

$$\hat{p}(x_n \,|\, x_m^{n-1}) = \begin{cases} |A|^{-1} & \text{if } c(x_m^{n-1}) = 0 \\ c(x_m^n)/c(x_m^{n-1}) & \text{otherwise} \end{cases}$$

The parameter a indicates how much probability mass should be shifted to character sequences which have not been observed, whilst the parameter λ indicates how much probability mass to shift to unseen monograms. When $c(x_m^{n-1}) = 0$ we return $|A|^{-1}$ to normalize $\hat{p}$. If it is not required that $\hat{p}$ be normalized, some other constant may be returned, including zero.

The expression may be easily rewritten as a sum on the form

$$P(x_n \,|\, x_m^{n-1}) = \sum_{j=m}^{n-1} a^{j-m} \cdot (1-a) \cdot \hat{p}(x_n \,|\, x_j^{n-1}) + a^{n-m} \frac{c(x_n) + \lambda}{\sum_{y \in A} c(y) + \lambda \cdot |A|}$$

which may be efficiently computed without resorting to recursion.

7 Results and observations

7.1 English telegrams

We choose A to be characters represented as 8-bit integers using the ISO-8859-1 character encoding, and let $\oplus$ be the exclusive-or operation, which is its own inverse, so $\ominus \equiv \oplus$.

A language model for 1-grams up to 7-grams was created from Charles Dickens' famous books A Christmas Carol, The Pickwick Papers, Battle for Life, Some Christmas Stories and A Tale of Two Cities, giving approximately 3 MB of plain text material.

As input, we selected, two famous telegrams. The first, sent by the confederate general in Charleston to his adversary following the bombardment of Fort Sumter triggering the American civil war, reads as follows

```
47 69 76 65 20 69 6e 20  6c 69 6b 65 20 61 20 67  |Give in like a g|
6f 6f 64 20 66 65 6c 6c  6f 77 2c 20 61 6e 64 20  |ood fellow, and |
62 72 69 6e 67 20 79 6f  75 72 20 67 61 72 72 69  |bring your garri|
73 6f 6e 20 74 6f 20 64  69 6e 6e 65 72 2c 20 61  |son to dinner, a|
6e 64 20 62 65 64 73 20  61 66 74 65 72 77 61 72  |nd beds afterwar|
64 73 2e 20 4e 6f 62 6f  64 79 20 69 6e 6a 75 72  |ds. Nobody injur|
65 64 2c 20 49 20 68 6f  70 65 3f                 |ed, I hope?|
```

The second telegram was transmitted in April 1912 by the SS Carpathia, to the offices of the White Star Line, following the demise of the HMS Titanic.

```
44 65 65 70 6c 79 20 72  65 67 72 65 74 20 61 64  |Deeply regret ad|
76 69 73 65 20 79 6f 75  72 20 54 69 74 61 6e 69  |vise your Titani|
63 20 73 75 6e 6b 20 74  68 69 73 20 6d 6f 72 6e  |c sunk this morn|
69 6e 67 20 66 69 66 74  65 65 6e 74 68 20 61 66  |ing fifteenth af|
74 65 72 20 63 6f 6c 6c  69 73 69 6f 6e 20 69 63  |ter collision ic|
65 62 65 72 67 20 72 65  73 75 6c 74 69 6e 67 20  |eberg resulting |
73 65 72 69 6f 75 73 20  6c 6f 73 73 20 6c 69 66  |serious loss lif|
65 20 66 75 72 74 68 65  72 20 70 61 72 74 69 63  |e further partic|
75 6c 61 72 73 20 6c 61  74 65 72 2e              |ulars later.|
```

When the plaintexts are combined under the $\oplus$ operation, the following character sequence is obtained, which will be given as input to the solver.

```
03 0c 13 15 4c 10 4e 52  09 0e 19 00 54 41 41 03  |....L.NR....TAA.|
19 06 17 45 46 1c 03 19  1d 57 78 49 15 0f 0a 49  |...EF....WxI...I|
01 52 1a 1b 09 4b 59 1b  1d 1b 53 47 0c 1d 00 07  |.R...KY...SG....|
1a 01 09 00 12 06 46 10  0c 0b 00 11 1a 0c 41 07  |......F.......A.|
1a 01 52 42 06 0b 1f 4c  08 15 1d 0a 1c 57 08 11  |..RB...L.....W..|
01 11 4b 52 29 4f 10 0a  17 0c 4c 1d 07 04 12 52  |..KR)O....L....R|
16 01 5e 49 26 55 1b 4f  1c 0a 4c                 |..^I&U.O..L|
```

Running the solver using 6-gram nodes, and retaining only $N = 256$ nodes in each iteration, results in the following two most likely messages, where `1` and `2` are labels, and erroneous characters are printed in red and underlined.

```
1:  Deeply regret advise your citanic sunk this morning fifteenth
1:  after collision iceberg resulting serious los

2:  Give in like a good fellow; and bring your garrison to dinner,
2:   and beds afterwards. Nobody injured, I hope?
```

7.1.1 Analysis of the result

The telegrams are quite different from the textual material of the books which were used to build the language model. For instance, the telegrams are very concise − to the point of being gramatically incorrect − and very seldom use conjunctions.

Still, the original messages are promptly extracted. The `T` in `Titanic` is not found, but then the word `Titanic` is nowhere to be found in the books either. Since `c` is selected instead of `T`, `;` is selected in the second text instead of `,`.

In total, 106 out of the 107 characters were correctly recovered. When taking into account the small size of the language model, and the discrepancies between the nature of the plaintext and that of the language model, this is a highly satisfactory result.

7.2 English novels

To measure the performance of the algorithm with respect to the error and the time required to separate a set of combined plaintexts, as well as its dependence upon the choice of parameters n, N and the binary operation, we computed the most likely separation of sixteen combined plaintext for n, N such that

$$n \in \{3, 5, 7\} \quad \text{and} \quad N \in \{8, 16, 32, 64, 128, 256, 512, 1024\}$$

and measured the number of CPU cycles required to complete each computation.

To create the plaintext differences p_1 thru p_{16}, eight random chunks of one thousand characters each were first selected from the four novels Alice in Wonderland, Crime and Punishment, Pride and Prejudice and The Lost World.

The chunks, which will be denoted $a_1, .., a_8$ for chunks from Alice in Wonderland, b_1, .., b_8 for chunks from Pride and Prejudice, c_1, .., c_8 for chunks from Crime and Punishment, and d_1, .., d_8 for chunks from The Lost World, where combined in the following arrangement to form the combined plaintexts $p_1, .., p_{16}$. We considered both addition and exclusive-or as binary operations.

$$\begin{array}{llll} p_1 = a_1 \ominus a_5 & p_5 = b_1 \ominus a_6 & p_9 = c_1 \ominus a_7 & p_{13} = d_1 \ominus a_8 \\ p_2 = a_2 \ominus b_5 & p_6 = b_2 \ominus b_6 & p_{10} = c_2 \ominus b_7 & p_{14} = d_2 \ominus b_8 \\ p_3 = a_3 \ominus c_5 & p_7 = b_3 \ominus c_6 & p_{11} = c_3 \ominus c_7 & p_{15} = d_3 \ominus c_8 \\ p_4 = a_4 \ominus d_5 & p_8 = b_4 \ominus d_6 & p_{12} = c_4 \ominus d_7 & p_{16} = d_4 \ominus c_9 \end{array}$$

The language model, which is approximately 65 MB in size, was constructed from the famous novels A Christmas Carol, A Tale of Two Cities, The Battle of Life, The Mystery of Edwin Drood, The Old Curiosity Shop, Gulliver's Travels, In Midsummer Days and Other Tales, Married, Political Ideals, Proposed Roads to Freedom, The Problems of Philosophy, Robinson Crusoe, The Adventures of Huckleberry Finn, The Adventures of Tom Sawyer, The Jungle Book, The Second Jungle Book, The Man in the Iron Mask, The Three Musketeers, The Treasure and The Wonderful Adventures of Nils. All texts were encoded using the ISO-8859-1 character set.

Once the most likely separation of each combined plaintext pair $p_1, .., p_{16}$ has been computed, we compared the separated plaintexts to the known plaintexts, and computed the number of erroneous characters $e_1, .., e_{16}$ for each choice of n, N and the binary operation.

Also, given $e_1, .., e_{16}$, we computed the mean error $\mu(e)$ and the standard deviation of the error $\sigma(e)$. Finally, we computed the average CPU time $\mu(T)$ required to process each character.

The values $e_1, .., e_{16}$, $\mu(e)$, $\sigma(e)$ and $\mu(T)$ are tabulated in table 3 for addition in $\mathbb{Z}_{256}$, and in table 4 for the exclusive-or operation in $\{0,1\}^8$. All computations were carried out on a 64-bit UltraSPARC IIIi processor running at 1593 MHz.

7.2.1 Analysis of the running time and the time complexity

As may be seen in tables 3 and 4, the average computation time $\mu(T)$ increases linearly in N, which is in accordance with our assumptions; when N is doubled, so is $\mu(T)$. Also, we see that $\mu(T)$ increases linearly in n. Of course, there are some fluctuations in $\mu(T)$ due to the fairly small number of samples used to compute it, but the overall behaviour seems clear.

It seems as if the exclusive-or operation is slightly more costly to separate than the addition operation, with respect to the time required for each character. At first, this may seem surprising, since exclusive or is faster to compute than byte-wise addition, but the behaviour is fairly natural; there is a higher error rate when using exclusive-or and the texts may become interchanged, giving more plausible messages than is the case for addition and requiring a larger effort to separate.

For a description of why the texts become interchanged when using exclusive-or, see section 7.4.1.

7.2.2 Analysis of the error frequencies

The average error $\mu(e)$ is strictly decreasing in N for both operations, and almost strictly decreasing in n for addition, which is as expected. For small n, N the error is very large, then it decreases rapidly as n, N increases and finally the error levels out at some point where the language model is simply not able to provide a better separation of the texts.

Some texts appear to be much more difficult to separate than others. All plaintext differences involving excerpts from Crime and Punishment, that is $p_3, p_7, p_9, .., p_{12}$ and p_{15} seem to have higher error frequencies than the other plaintext differences.

For example, the error e_{12} is very large. This is explained by the fact that the text chunk c_4 selected from Crime and Punishment contains a lot of Russian proper names, such as Porfiry (2 times), Razumihin (3 times), Sofya Ivanovna (1 time), Sofya Semyonovna (3 times) and Raskolnikov (1 time).

In total, this gives 214 characters of Russian proper names in a 1,000 character text. Clearly, the algorithm still does a fairly good job separating the texts with an error of 62 characters for addition, and 82 characters for exclusive-or.

If the plaintext differences $p_3, p_7, p_9, .., p_{12}, p_{15}$ are excluded, the average error drops from $\mu(e) = 16.1$ to $\mu(e) = 8.8$. This underlines the importance of selecting a language model which is representative of the language it is supposed model.

7.3 Motions written by Swedish members' of Parliament

To demonstrate the algorithms ability to separate contents in languages different from English, we chose to consider a collection of motions written by Swedish members' of Parliament instead of English novels. We computed the most likely separations of four combined plaintext for n, N such that

$$n \in \{3, 5, 7\} \quad \text{and} \quad N \in \{8, 16, 32, 64, 128, 256, 512, 1024\}$$

and measured the number of CPU cycles required to complete each computation. Since the contents of the motions is more homogeneous than that of the novels, we decided to consider a small set of plaintexts in this section.

The language model, which is approximately 58 MB in size, was constructed from all motions submitted to the Swedish Parliament in the parliamentary year 2006/07. To create the plaintext differences p_1 thru p_4, eight random chunks of one thousand characters each were first selected from motions written in the parliamentary year of 2004/05. All motions were encoded using the ISO-8859-1 character set.

The chunks, which will be denoted $a_1, .., a_4, b_1, .., b_4$, were combined in the following arrangement to form the combined plaintexts $p_1, .., p_4$. We considered both addition and exclusive-or as binary operations.

$$p_1 = a_1 \ominus b_1 \qquad p_2 = a_2 \ominus b_2$$
$$p_3 = a_3 \ominus b_3 \qquad p_2 = a_4 \ominus b_4$$

Once the most likely separation of each combined plaintext pair $p_1, .., p_4$ had been computed, we proceeded as in section 7.2, and compared the separated plaintexts to the known plaintexts by computing $e_1, .., e_4$ and $\mu(e)$, $\sigma(e)$ and $\mu(T)$ for each choice of n, N and the binary operation.

The results are tabulated in table 1 for addition in $\mathbb{Z}_{256}$, and in table 2 for the exclusive-or operation in $\{0,1\}^8$. All computations were carried out on a 64-bit UltraSPARC IIIi processor running at 1593 MHz.

7.3.1 Analysis of the error frequencies

The average error $\mu(e)$ is strictly decreasing in N and almost strictly decreasing in n for addition, which is as expected. For small n, N the error is very large, then it decreases rapidly as n, N increases and finally the error levels out at some point where the language model is simply not able to provide a better separation of the texts.

The average errors $\mu(e) = 4.0$ for addition and $\mu(e) = 5.0$ for exclusive-or are considerably smaller than we dared hope. It appears as if exclusive-or is a bit more difficult than addition, which is explained by errors being introduced when the texts are interchanged. Finally, we note that the standard deviation $\mu(e) \approx \sigma(e)$, which is fairly high.

7.3.2 Analysis of the running time

The running times evidenced for exclusive-or and addition are comparable to those seen for English novels in section 7.2.

7.4 Observations

7.4.1 Interchanged plaintexts

We observe that if the difference between the two plaintexts is computed using a commutative operation, such that $x \ominus y = y \ominus x$, and the same language model is used to model both texts, then the Viterbi algorithm may switch between the texts during plaintext recovery. Examples of commuting operations are exclusive-or and modular addition.

For each state in the state vector, the algorithm selects x, and computes $y = y \ominus x \oplus x$. Then, the transition probability is computed as $P_c \cdot P_x(x \,|\, \bar{x}) \cdot P_y(y \,|\, \bar{y})$, where P_c is the probability of the path up to and including the current state $(\bar{x}, \bar{y})$. However, if

$$x \ominus y = y \ominus x \quad \forall\, x, y$$

then there is no way of knowing if x was selected and y computed, or vice versa, and thus to which text x and y belong, respectively. Most of the time, this information will be provided by the language model; more specifically whenever x is a probable continuation of either $\bar{x}$ or $\bar{y}$, but not both. If it is likely that x follows both $\bar{x}$ and $\bar{y}$, however, there is a risk that the sequences will be interchanged.

Note that if the texts being separated belong to very different distributions, for example if two texts written in different languages are separated, then the texts are very unlikely to be interchanged, since it is then unlikely for both

$P(x \mid \bar{x})$ and $P(y \mid \bar{x})$ to be great at the same time, unless a common name, geographic name, etc. occurs in both texts.

8 Enhancements and generalizations

8.1 More than two parallel messages may be used

We have so far only explored cryptanalysis of the Vernam cipher when the same key sequence is used to encrypt $m = 2$ messages. This problem is harder to solve than the case where $m > 2$ messages have been encrypted under the same key, since the m texts may in the latter case be combined pairwise in different constellations to create a number of problems of the type we have explored. These problems may then be solved independently, making it easy to guess $\bar{k}$ and once we know $\bar{k}$, we know all the plaintexts.

It is however trivial to generalize the algorithm itself so that it may process $m > 2$ messages in parallel, rendering any manual post-processing unnecessary. Consider the system of equations for the relation between the plaintexts $p_1, \ldots, p_m$ and the ciphertexts $\bar{c}_1, .., \bar{c}_m$.

$$\begin{aligned}
\bar{c}_1 &= \bar{p}_1 \oplus \bar{k} && \Rightarrow \quad \bar{p}_1 = \bar{c}_1 \ominus \bar{k} \\
\bar{c}_2 &= \bar{p}_1 \oplus \bar{k} && \Rightarrow \quad \bar{p}_3 = \bar{c}_2 \ominus \bar{k} \\
&\vdots && \vdots \\
\bar{c}_m &= \bar{p}_m \oplus \bar{k} && \Rightarrow \quad \bar{p}_m = \bar{c}_m \ominus \bar{k}
\end{aligned}$$

If we iterate over all possible characters k in each step of the Viterbi algorithm, and not over one of the plaintext characters as we did previously, then we will find all possible sets $(p_1, \ldots, p_m)$ at each position in the text by subtraction.

Provided the previous n-gram for our candidate messages $p_1, \ldots, p_m$ have been stored in the Viterbi graph, then we may compute the transition probabilities in each of the texts, and multiply it with the probability of being in the previous state, to form the probability of the next state. Hence, the algorithm works for any $m > 2$.

Making this modification to the algorithm increases the time complexity to $\mathcal{O}(m\,|K|\,N\,|\bar{z}|)$. Since we now have more information, however, it may be possible to pick a smaller value for N and n without incurring any penalty with respect to the error rate. If this is done, it is quite possible that the algorithm will run faster when $m > 2$, than when $m = 2$.

8.2 Any invertible binary operation may be used

In our previous description of the algorithm, we assumed that the characters in the messages and the key sequence all belonged to some group A, for the sake of simplifying the presentation of the algorithm.

This is however not necessary. As may be seen in the equation system above, all we require is that there exists an *invertible* binary operation $\odot$ such that for

two *sets* A, K, it holds that

$$c = p \odot k$$
$$p = c \odot k^{-1} = p \odot k \odot k^{-1} = p$$

for all $p, c \in A$ and all k K.

Then, we may use the generalized algorithm described in section 8.1 to solve the problem with time complexity $\mathcal{O}(|K|\, N\, |z|)$. The complexity involves a factor $|K|$ and not a factor $|A|$ since we are now iterating over all possible characters in K.

8.3 Parallelization

There are at least two natural ways of parallelizing the algorithm.

8.3.1 Parallelization of the state expansion

If the key alphabet is large, and the machine on which the algorithm is run has multiple CPUs and/or cores, it may be advantageous to parallelize the operation of generating $|K| \cdot N$ continuations from the N retained states.

This may be done by simply splitting the Q vector into t partitions and running the algorithm on each partition; an operation which may be executed in parallel. The expected complexity would then become

$$O\left(\frac{N\,|K|}{t}|\bar{z}|\right)$$

Unless $|K|$ is very large, this parallelization does not improve the performance of the algorithm when it is run a cluster of t machines, since the I/O penalty incurred when sending data back and forth would greatly outweigh the benefits in terms of the slightly reduced time complexity.

The result obtained when running the algorithm with this form of parallelization is equivalent to that obtained when running the original algorithm; the algorithm's behaviour is unaffected.

8.3.2 Parallelization of the whole algorithm

If a cluster of T machines is available, then one possible way of parallelizing the algorithm is to split the input message into T partitions, and then process each partition on a separate machine. This will reduce the time complexity by a factor of T, but may instead lead to an increased error rate, should the partial plaintexts recovered not fit together.

A better way may be to split the input into $2T$ partitions to process partitions $0, 2, .., 2T$ first. Then, process partitions $1, 3, .., 2T - 1$ would be processed. Let us consider the case of partition i for i odd. We start with the last n-grams generated in partition $i - 1$, and iterate until the beginning of partition $i + 1$ is reached. Then we continue computing even further, until at least n characters of our computed plaintext overlap in partition $i + 1$. We will then have bridged the two partitions, so we may backtrack and superimpose the new plaintext over partitions $i - 1, i$ and $i + 1$.

It may of course be the case that it is not possible to bridge the partitions, so we will need to set some upper bound on how far into the next plaintext we will seek until giving up. In general, however, it will be possible to find a common segment of plaintext.

Note that this parallelization will affect the overall behaviour of the algorithm, and possibly increase the error rate. In return, however, it will provide a substantial reduction in time complexity. If a large cluster is available, it will be possible to process long messages within reasonable time, and short messages very swiftly.

9 Conclusion

We have described an algorithm for the automatic cryptanalysis of one-time pads, and similar ciphers, when the key sequence is used multiple times. Although the idea is not new, we hope that this presentation will be more accessible than previous descriptions.

The method described may be efficiently implemented and has time complexity $\mathcal{O}(|K|\, m\, N\, |\bar{z}|)$, giving it several practical applications in the fields of cryptanalysis and data recovery. It makes it possible to automatically recover the plaintext whenever a one time pad, stream cipher or simple book cipher is used incorrectly. This may be done regardless of which invertible operation was used to combine the plaintext and key stream characters.

Although the performance of the algorithm depends heavily on the language model's correspondence to the texts being separated, the values of the parameters n and N may often be selected so as to give a reasonable compromise between the average running time and the average error.

It is not at all unusual to see an average error frequency below 1%. If the inverse binary operation is commutative, the texts tend to be interchanged, giving a slightly larger average error frequency than what is otherwise the case.

On a 1593 MHz 64-bit UltraSPARC IIIi processor, it takes approximately 10 ms to 1.4 s per character to recover the plaintexts, depending on the values selected for n and N, the binary operation and the language model used.

Acknowledgments

The authors would like to extend their gratitude to professor Johan Håstad for his supervision and advice, and to the Swedish Parliament for its assistance in procuring ample amounts of Swedish text material for the language model.

References

[1] Cormen, T., Leiserson, C., Rivest, R., Stein, C. *Introduction to Algorithms*. MIT Press, September 2001.

[2] Encyclopaedia Britannica. *Cryptology*. Retreived on January 31 2008, January 2008. `http://www.britannica.com/article-233471`.

[3] Friedman, W. F. *The Index of Coincidence and Its Applications in Cryptography. The Riverbank Publications*, 22, 1922.

[4] Hongjun, W. *The Misuse of RC4 in Microsoft Word and Excel*. Cryptology ePrint Archive, Report 2005/007, January 2005. `http://eprint.iacr.org/2005/007.pdf`.

[5] IEEE Computer Society. *IEEE 802.11 Wireless Network Standard*, 1999. `http://standards.ieee.org/getieee802/download/802.11-1999.pdf`.

[6] Kohno, T. *Attacking and repairing the WinZip encryption scheme*. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 72–81. ACM, New York, NY, USA, October 2004.

[7] Mason, J., Watkins, K., Eisner, J., Stubblefield, A. *A natural language approach to automated cryptanalysis of two-time pad*. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pages 235–244. ACM, New York, NY, USA, 2006.

[8] N, Borisov, Goldberg, I., Wagner, D. *Intercepting mobile communications: the insecurity of 802.11*. In *Proceedings of the seventh Annual International Conference on Mobile Computing and Networking*, pages 180–189. ACM, July 2001.

[9] Shannon, C. *The Communication Theory of Secrecy Systems*. Bell System Technical Journal, October 1949.

[10] Vernam, G. *Secret signalling system*. U.S. Patent No. 1,310,719, July 1919. `http://patft.uspto.gov/netacgi/nph-Parser?patentnumber=1310719`.

[11] Vernam, G. *Ciphering device*. U.S. Patent No. 1,416,765, May 1922. `http://patft.uspto.gov/netacgi/nph-Parser?patentnumber=1416765`.

[12] Viterbi, A. *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory*, 13(2):260–269, April 1967.

A Tables

A.1 Error frequency tables

In the following tables, n denotes the n-gram length, N the number of retained states after each iteration and e_1 through e_m the error frequencies in parts per thousands for the combined plaintexts p_1 through p_m.

The average error is given by $\mu(e)$ and the standard deviation by $\sigma(e)$, both expressed in parts per thousand. The average run time in milliseconds per character by $\mu(T)$.

n	N	e_1	e_2	e_3	e_4	$\mu(e)$	$\sigma(e)$	$\mu(T)$
3	8	17	75	58	51	50.3	24.3	7 ms
3	16	19	61	33	29	35.5	18.0	14 ms
3	32	19	54	27	21	30.3	16.2	29 ms
3	64	15	51	27	20	28.3	15.9	60 ms
3	128	15	45	27	20	26.8	13.1	119 ms
3	256	15	45	25	20	26.3	13.1	237 ms
3	512	15	45	25	20	26.3	13.1	450 ms
3	1024	15	45	25	20	26.3	13.1	861 ms
5	8	21	47	19	33	30.0	12.9	11 ms
5	16	14	41	8	7	17.5	16.0	21 ms
5	32	12	29	8	2	12.8	11.6	41 ms
5	64	7	17	9	3	9.0	5.9	85 ms
5	128	7	19	9	3	9.5	6.8	161 ms
5	256	7	19	7	3	9.0	6.9	321 ms
5	512	7	19	7	3	9.0	6.9	628 ms
5	1024	7	19	7	3	9.0	6.9	1218 ms
7	8	16	50	26	28	30.0	14.3	13 ms
7	16	15	32	7	2	14.0	13.1	24 ms
7	32	8	19	8	2	9.2	7.1	48 ms
7	64	4	15	8	1	7.0	6.1	93 ms
7	128	4	11	5	1	5.3	4.2	181 ms
7	256	4	9	2	1	4.0	3.6	349 ms
7	512	4	9	2	1	4.0	3.6	675 ms
7	1024	4	9	2	1	4.0	3.6	1303 ms

Table 1: Error frequencies for various n and N under addition in $\mathbb{Z}_{256}$. The corpus and the textual data separated were built from motions written by members of the Swedish Parliament. From left to right, n is the n-gram length, N is the number of retained states, e_i is the error frequency in parts per thousand for the plaintext difference p_i, $\mu(e)$ is the average error frequency, $\sigma(e)$ the standard deviation of the error frequency, and $\mu(T)$ is the average running time in milliseconds per character.

n	N	e_1	e_2	e_3	e_4	$\mu(e)$	$\sigma(e)$	$\mu(T)$
3	8	42	118	92	116	92.0	35.4	6 ms
3	16	21	79	64	49	53.3	24.7	12 ms
3	32	19	58	47	44	42.0	16.5	26 ms
3	64	18	54	43	37	38.0	15.1	58 ms
3	128	15	48	38	36	34.3	13.9	118 ms
3	256	15	49	38	36	34.5	14.2	231 ms
3	512	15	49	38	36	34.5	14.2	449 ms
3	1024	15	49	38	36	34.5	14.2	869 ms
5	8	16	93	63	57	57.3	31.7	10 ms
5	16	11	61	30	25	31.8	21.1	21 ms
5	32	13	36	13	4	16.5	13.7	43 ms
5	64	9	29	7	1	11.5	12.2	85 ms
5	128	7	20	7	1	8.8	8.0	171 ms
5	256	7	17	7	1	8.0	6.6	337 ms
5	512	7	17	7	1	8.0	6.6	660 ms
5	1024	7	12	7	1	6.8	4.5	1289 ms
7	8	21	85	46	47	49.8	26.4	12 ms
7	16	13	45	40	6	26.0	19.4	25 ms
7	32	15	43	14	3	18.8	17.1	50 ms
7	64	10	38	8	3	14.8	15.8	98 ms
7	128	9	12	3	3	6.8	4.5	191 ms
7	256	7	9	3	3	5.5	3.0	371 ms
7	512	7	9	1	3	5.0	3.7	727 ms
7	1024	7	9	1	3	5.0	3.7	1397 ms

Table 2: Error frequencies for various n and N exclusive-or in $\{0, 1\}^8$. The corpus and the textual data separated were built from motions written by members of the Swedish Parliament. From left to right, n is the n-gram length, N is the number of retained states, e_i is the error frequency in parts per thousand for the plaintext difference p_i, $\mu(e)$ is the average error frequency, $\sigma(e)$ the standard deviation of the error frequency, and $\mu(T)$ is the average running time in milliseconds per character.

n	N	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}	e_{12}	e_{13}	e_{14}	e_{15}	e_{16}	$\mu(e)$	$\sigma(e)$	$\mu(T)$
3	8	72	61	71	42	60	36	79	24	94	57	99	106	67	28	47	92	64.7	25.2	6 ms
3	16	55	46	39	23	46	21	65	20	75	32	82	85	59	24	32	80	49.0	23.1	13 ms
3	32	28	46	26	21	29	21	55	14	72	28	69	89	53	20	28	70	41.8	23.2	28 ms
3	64	26	38	26	20	26	15	51	14	72	25	63	89	41	20	26	66	38.6	22.9	57 ms
3	128	27	38	25	19	26	15	53	14	67	28	64	86	39	20	25	66	38.3	22.1	115 ms
3	256	25	38	25	19	26	15	54	14	67	23	64	86	39	20	24	66	37.8	22.4	223 ms
3	512	25	38	25	19	26	15	54	14	63	23	64	86	39	20	24	66	37.6	22.1	430 ms
3	1024	25	38	25	19	26	15	54	12	63	23	64	86	39	20	24	66	37.4	22.3	827 ms
5	8	51	47	44	26	48	29	35	10	101	50	68	93	53	24	17	78	48.4	26.1	10 ms
5	16	36	38	23	27	31	26	32	4	73	36	50	74	26	12	13	41	33.9	19.3	20 ms
5	32	16	32	12	10	15	22	39	4	57	32	44	68	18	13	10	32	26.5	18.2	42 ms
5	64	9	14	12	9	15	15	34	6	54	17	34	64	12	13	6	23	21.1	17.1	83 ms
5	128	9	14	12	9	15	15	33	6	50	14	31	64	13	13	6	23	20.4	16.4	164 ms
5	256	9	14	12	9	15	14	33	6	46	14	31	67	11	13	6	23	20.2	16.6	319 ms
5	512	7	14	12	9	13	14	33	6	46	14	31	66	11	13	6	23	19.9	16.6	620 ms
5	1024	7	14	12	9	13	14	33	4	46	17	31	66	11	13	6	23	19.9	16.6	1203 ms
7	8	58	62	40	44	72	34	64	17	145	56	79	91	39	39	45	84	60.6	30.1	12 ms
7	16	36	33	27	14	17	19	28	3	99	35	48	75	22	15	13	64	34.3	25.7	24 ms
7	32	30	28	21	8	10	16	22	3	76	21	35	64	9	10	7	37	24.8	20.5	48 ms
7	64	20	18	9	8	8	10	15	3	58	26	28	66	7	7	6	20	19.3	18.3	95 ms
7	128	17	16	8	8	10	10	14	2	45	21	24	66	11	7	6	17	17.6	16.3	183 ms
7	256	17	14	8	8	10	10	14	2	45	19	22	65	9	7	6	15	16.9	16.1	353 ms
7	512	12	10	8	8	9	10	14	2	44	19	25	64	9	7	6	15	16.4	16.1	681 ms
7	1024	12	10	8	8	9	10	14	0	43	19	25	62	9	7	6	15	16.1	15.7	1315 ms

Table 3: Error frequencies for various n and N under addition in $\mathbb{Z}_{256}$. The corpus and the textual data separated were built from a collection of English novels written by famous authors. From left to right, n is the n-gram length, N is the number of retained states, e_i is the error frequency in parts per thousand for the plaintext difference p_i, $\mu(e)$ is the average error frequency, $\sigma(e)$ the standard deviation of the error frequency, and $\mu(T)$ is the average running time in milliseconds per character.

n	N	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}	e_{12}	e_{13}	e_{14}	e_{15}	e_{16}	$\mu(e)$	$\sigma(e)$	$\mu(T)$
3	8	158	133	105	106	125	86	141	101	167	99	169	155	152	97	112	132	127.4	27.3	6 ms
3	16	100	84	71	66	88	45	82	57	119	55	122	137	111	61	65	99	85.1	27.3	12 ms
3	32	62	66	41	40	63	24	63	33	95	31	106	125	68	33	59	84	62.1	28.9	27 ms
3	64	42	63	34	24	44	22	64	29	87	27	101	124	57	29	46	84	54.8	30.4	58 ms
3	128	43	60	32	28	41	22	60	28	84	27	98	113	52	28	46	83	52.8	28.1	120 ms
3	256	42	60	32	27	41	22	60	28	83	27	98	113	50	28	43	80	52.1	27.9	236 ms
3	512	42	56	30	27	41	22	60	28	81	27	98	113	50	28	43	80	51.6	27.8	446 ms
3	1024	42	56	30	27	41	22	60	26	80	29	98	113	50	28	43	80	51.6	27.8	858 ms
5	8	143	71	93	87	56	87	73	32	131	66	147	118	88	49	49	89	86.2	34.1	11 ms
5	16	68	52	79	44	33	31	44	10	109	42	91	88	48	48	31	32	53.1	26.7	21 ms
5	32	40	47	27	21	28	32	32	10	86	30	69	84	39	28	18	37	39.3	22.2	43 ms
5	64	28	39	21	13	23	29	34	10	68	21	54	83	28	22	13	26	32.0	20.2	87 ms
5	128	29	30	15	17	24	22	33	11	63	21	50	83	19	19	10	23	29.3	19.9	173 ms
5	256	27	26	15	17	27	20	37	11	63	18	54	82	17	19	10	22	29.1	20.3	336 ms
5	512	20	23	15	17	18	20	33	11	58	12	46	84	17	19	10	23	26.6	20.0	661 ms
5	1024	18	23	15	17	18	20	36	9	58	12	46	81	17	19	10	23	26.4	19.7	1288 ms
7	8	132	208	91	119	84	48	78	32	134	42	178	117	105	61	57	94	98.8	48.6	13 ms
7	16	77	72	78	67	39	31	51	11	105	51	80	103	52	30	30	44	57.6	27.0	25 ms
7	32	44	35	24	28	27	24	30	10	85	25	79	96	36	16	15	39	38.3	25.7	51 ms
7	64	24	33	18	18	21	19	28	5	81	25	57	89	21	11	13	23	30.4	24.2	100 ms
7	128	23	21	16	11	19	11	18	6	75	18	44	73	22	6	12	19	24.6	21.2	198 ms
7	256	23	17	14	8	14	9	16	6	61	14	44	79	19	6	7	18	22.2	21.0	383 ms
7	512	20	18	14	8	17	11	23	6	60	13	38	81	13	6	7	19	22.1	20.8	746 ms
7	1024	18	14	14	7	17	11	25	4	55	11	39	82	13	6	7	19	21.4	20.8	1436 ms

Table 4: Error frequencies for various n and N under exclusive-or in $\{0, 1\}^8$. The corpus and the textual data separated were built from a collection of English novels written by famous authors. From left to right, n is the n-gram length, N is the number of retained states, e_i is the error frequency in parts per thousand for the plaintext difference p_i, $\mu(e)$ is the average error frequency, $\sigma(e)$ the standard deviation of the error frequency, and $\mu(T)$ is the average running time in milliseconds per character.

B Excerpts from English novels

B.1 The plaintext difference p_1

The text a_1

made a memorandum of the fact.

'I keep them to sell,' the Hatter added as an explanation; 'I've none of my own. I'm a hatter.'

Here the Queen put on her spectacles, and began staring at the Hatter, who turned pale and fidgeted.

'Give your evidence,' said the King; 'and don't be nervous, or I'll have you executed on the spot.'

This did not seem to encourage the witness at all: he kept shifting from one foot to the other, looking uneasily at the Queen, and in his confusion he bit a large piece out of his teacup instead of the bread-and-butter.

Just at this moment Alice felt a very curious sensation, which puzzled her a good deal until she made out what it was: she was beginning to grow larger again, and she thought at first she would get up and leave the court; but on second thoughts she decided to remain where she was as long as there was room for her.

'I wish you wouldn't squeeze so.' said the Dormouse, who was sitting next to her. 'I can

The text a_5

cucumber-frames there must be!' thought Alice. 'I wonder what they'll do next! As for pulling me out of the window, I only wish they COULD! I'm sure I don't want to stay in here any longer!'

She waited for some time without hearing anything more: at last came a rumbling of little cartwheels, and the sound of a good many voices all talking together: she made out the words: 'Where's the other ladder?–Why, I hadn't to bring but one; Bill's got the other–Bill! fetch it here, lad!–Here, put 'em up at this corner–No, tie 'em together first–they don't reach half high enough yet–Oh! they'll do well enough; don't be particular– Here, Bill! catch hold of this rope–Will the roof bear?–Mind that loose slate–Oh, it's coming down! Heads below!' (a loud crash)–'Now, who did that?–It was Bill, I fancy–Who's to go down the chimney?–Nay, I shan't! YOU do it!–That I won't, then!–Bill's to go down–Here, Bill! the master says you're to go down the chimney!'

B.2 The plaintext difference p_2

The text a_2

o stay in here any longer!'

She waited for some time without hearing anything more: at last came a rumbling of little cartwheels, and the sound of a good many voices all talking together: she made out the words: 'Where's the other ladder?–Why, I hadn't to bring but one; Bill's got the other–Bill! fetch it here, lad!–Here, put 'em up at this corner–No, tie 'em together first–they don't reach half high enough yet–Oh! they'll do well enough; don't be particular– Here, Bill! catch hold of this

rope–Will the roof bear?–Mind that loose slate–Oh, it's coming down! Heads below!' (a loud crash)–'Now, who did that?–It was Bill, I fancy–Who's to go down the chimney?–Nay, I shan't! YOU do it!–That I won't, then!–Bill's to go down–Here, Bill! the master says you're to go down the chimney!'

'Oh! So Bill's got to come down the chimney, has he?' said Alice to herself. 'Shy, they seem to put everything upon Bill! I wouldn't be in Bill's place for a good deal

The text b_5

actuated by one spirit, everything relating to their journey was speedily settled. They were to be off as soon as possible. "But what is to be done about Pemberley?" cried Mrs. Gardiner. "John told us Mr. Darcy was here when you sent for us; was it so?"

"Yes; and I told him we should not be able to keep our engagement. _That_ is all settled."

"What is all settled?" repeated the other, as she ran into her room to prepare. "And are they upon such terms as for her to disclose the real truth? Oh, that I knew how it was!"

But wishes were vain, or at least could only serve to amuse her in the hurry and confusion of the following hour. Had Elizabeth been at leisure to be idle, she would have remained certain that all employment was impossible to one so wretched as herself; but she had her share of business as well as her aunt, and amongst the rest there were notes to be written to all their friends at Lambton, with false excuses for their sudden departure. An h

B.3 The plaintext difference p_3

The text a_3

o was reading the list of singers.

'You may go,' said the King, and the Hatter hurriedly left the court, without even waiting to put his shoes on.

'–and just take his head off outside,' the Queen added to one of the officers: but the Hatter was out of sight before the officer could get to the door.

'Call the next witness!' said the King.

The next witness was the Duchess's cook. She carried the pepper-box in her hand, and Alice guessed who it was, even before she got into the court, by the way the people near the door began sneezing all at once.

'Give your evidence,' said the King.

'Shan't,' said the cook.

The King looked anxiously at the White Rabbit, who said in a low voice, 'Your Majesty must cross-examine THIS witness.'

'Well, if I must, I must,' the King said, with a melancholy air, and, after folding his arms and frowning at the cook till his eyes were nearly out of sight, he said in a deep voice, 'What are tarts made of?'

The text c_5

dark brown hair and with a hectic flush in her cheeks. She was pacing up and down in her little room, pressing her hands against her chest; her lips were

parched and her breathing came in nervous broken gasps. Her eyes glittered as in fever and looked about with a harsh immovable stare. And that consumptive and excited face with the last flickering light of the candle-end playing upon it made a sickening impression. She seemed to Raskolnikov about thirty years old and was certainly a strange wife for Marmeladov.... She had not heard them and did not notice them coming in. She seemed to be lost in thought, hearing and seeing nothing. The room was close, but she had not opened the window; a stench rose from the staircase, but the door on to the stairs was not closed. From the inner rooms clouds of tobacco smoke floated in, she kept coughing, but did not close the door. The youngest child, a girl of six, was asleep, sitting curled up on the floor with her head on the sofa. A

B.4 The plaintext difference p_4

The text a_4

easant state of mind, she turned away.

'Come back!' the Caterpillar called after her. 'I've something important to say!'

This sounded promising, certainly: Alice turned and came back again.

'Keep your temper,' said the Caterpillar.

'Is that all?' said Alice, swallowing down her anger as well as she could.

'No,' said the Caterpillar.

Alice thought she might as well wait, as she had nothing else to do, and perhaps after all it might tell her something worth hearing. For some minutes it puffed away without speaking, but at last it unfolded its arms, took the hookah out of its mouth again, and said, 'So you think you're changed, do you?'

'I'm afraid I am, sir,' said Alice; 'I can't remember things as I used–and I don't keep the same size for ten minutes together!'

'Can't remember WHAT things?' said the Caterpillar.

'Well, I've tried to say "HOW DOTH THE LITTLE BUSY BEE," but it all came different!' Alice replied in a very melancholy

The text d_5

have got our chart, our one and only immediate duty is to get ourselves safe and sound out of this awful place."

"The flesh-pots of civilization," groaned Challenger.

"The ink-pots of civilization, sir. It is our task to put on record what we have seen, and to leave the further exploration to others. You all agreed as much before Mr. Malone got us the chart."

"Well," said Challenger, "I admit that my mind will be more at ease when I am assured that the result of our expedition has been conveyed to our friends. How we are to get down from this place I have not as yet an idea. I have never yet encountered any problem, however, which my inventive brain was unable to solve, and I promise you that to-morrow I will turn my attention to the question of our descent." And so the matter was allowed to rest.

But that evening, by the light of the fire and of a single candle, the first map of the lost world was elaborated. Every detail which I had roughly noted from m

B.5 The plaintext difference p_5

The text b_1

took from me my faculties."

"Your attendance upon her has been too much for you. You do not look well. Oh that I had been with you! you have had every care and anxiety upon yourself alone."

"Mary and Kitty have been very kind, and would have shared in every fatigue, I am sure; but I did not think it right for either of them. Kitty is slight and delicate; and Mary studies so much, that her hours of repose should not be broken in on. My aunt Phillips came to Longbourn on Tuesday, after my father went away; and was so good as to stay till Thursday with me. She was of great use and comfort to us all. And Lady Lucas has been very kind; she walked here on Wednesday morning to condole with us, and offered her services, or any of her daughters', if they should be of use to us."

"She had better have stayed at home," cried Elizabeth; "perhaps she _meant_ well, but, under such a misfortune as this, one cannot see too little of one's neighbours. Assistance is impos

The text a_6

?' Alice whispered to the Gryphon. 'They can't have anything to put down yet, before the trial's begun.'

'They're putting down their names,' the Gryphon whispered in reply, 'for fear they should forget them before the end of the trial.'

'Stupid things!' Alice began in a loud, indignant voice, but she stopped hastily, for the White Rabbit cried out, 'Silence in the court!' and the King put on his spectacles and looked anxiously round, to make out who was talking.

Alice could see, as well as if she were looking over their shoulders, that all the jurors were writing down 'stupid things!' on their slates, and she could even make out that one of them didn't know how to spell 'stupid,' and that he had to ask his neighbour to tell him. 'A nice muddle their slates'll be in before the trial's over!' thought Alice.

One of the jurors had a pencil that squeaked. This of course, Alice could not stand, and she went round the court and got behind him, and very soo

B.6 The plaintext difference p_6

The text b_2

pected. Her uncle and aunt were all amazement; and the embarrassment of her manner as she spoke, joined to the circumstance itself, and many of the circumstances of the preceding day, opened to them a new idea on the business. Nothing had ever suggested it before, but they felt that there was no other way of accounting for such attentions from such a quarter than by supposing a partiality for their niece. While these newly-born notions were passing in their heads, the perturbation of Elizabeth's feelings was at every moment increasing. She was quite amazed at her own discomposure; but amongst other causes of disquiet, she dreaded lest the partiality of the brother should have said too

much in her favour; and, more than commonly anxious to please, she naturally suspected that every power of pleasing would fail her.

She retreated from the window, fearful of being seen; and as she walked up and down the room, endeavouring to compose herself, saw such looks of inquiri

The text b_6

I certainly _have_ had my share of beauty, but I do not pretend to be anything extraordinary now. When a woman has five grown-up daughters, she ought to give over thinking of her own beauty."

"In such cases, a woman has not often much beauty to think of."

"But, my dear, you must indeed go and see Mr. Bingley when he comes into the neighbourhood."

"It is more than I engage for, I assure you."

"But consider your daughters. Only think what an establishment it would be for one of them. Sir William and Lady Lucas are determined to go, merely on that account, for in general, you know, they visit no newcomers. Indeed you must go, for it will be impossible for _us_ to visit him if you do not."

"You are over-scrupulous, surely. I dare say Mr. Bingley will be very glad to see you; and I will send a few lines by you to assure him of my hearty consent to his marrying whichever he chooses of the girls; though I must throw in a good word for my little Lizzy."

"I de

B.7 The plaintext difference p_7

The text b_3

, was nothing in comparison to that total want of propriety so frequently, so almost uniformly betrayed by herself, by your three younger sisters, and occasionally even by your father. Pardon me. It pains me to offend you. But amidst your concern for the defects of your nearest relations, and your displeasure at this representation of them, let it give you consolation to consider that, to have conducted yourselves so as to avoid any share of the like censure, is praise no less generally bestowed on you and your elder sister, than it is honourable to the sense and disposition of both. I will only say farther that from what passed that evening, my opinion of all parties was confirmed, and every inducement heightened which could have led me before, to preserve my friend from what I esteemed a most unhappy connection. He left Netherfield for London, on the day following, as you, I am certain, remember, with the design of soon returning.

"The part which I acted is n

The text c_6

s past he had feared more than anything was _being shown up_ and this was the chief ground for his continual uneasiness at the thought of transferring his business to Petersburg. He was afraid of this as little children are sometimes panic-stricken. Some years before, when he was just entering on his own career, he had come upon two cases in which rather important personages in the province, patrons of his, had been cruelly shown up. One instance had ended in great scandal for the person attacked and the other had very nearly ended in

serious trouble. For this reason Pyotr Petrovitch intended to go into the subject as soon as he reached Petersburg and, if necessary, to anticipate contingencies by seeking the favour of "our younger generation." He relied on Andrey Semyonovitch for this and before his visit to Raskolnikov he had succeeded in picking up some current phrases. He soon discovered that Andrey Semyonovitch was a commonplace simpleton, but that by no means reassur

B.8 The plaintext difference p_8

The text b_4

ubject of your reverie."

"I should imagine not."

"You are considering how insupportable it would be to pass many evenings in this manner–in such society; and indeed I am quite of your opinion. I was never more annoyed! The insipidity, and yet the noise–the nothingness, and yet the self-importance of all those people! What would I give to hear your strictures on them!"

"You conjecture is totally wrong, I assure you. My mind was more agreeably engaged. I have been meditating on the very great pleasure which a pair of fine eyes in the face of a pretty woman can bestow."

Miss Bingley immediately fixed her eyes on his face, and desired he would tell her what lady had the credit of inspiring such reflections. Mr. Darcy replied with great intrepidity:

"Miss Elizabeth Bennet."

"Miss Elizabeth Bennet!" repeated Miss Bingley. "I am all astonishment. How long has she been such a favourite?–and pray, when am I to wish you joy?"

"That is exactly the questi

The text d_6

nk of, for an active drama was in progress. Two of the ape-men had seized one of the Indians out of the group and dragged him forward to the edge of the cliff. The king raised his hand as a signal. They caught the man by his leg and arm, and swung him three times backwards and forwards with tremendous violence. Then, with a frightful heave they shot the poor wretch over the precipice. With such force did they throw him that he curved high in the air before beginning to drop. As he vanished from sight, the whole assembly, except the guards, rushed forward to the edge of the precipice, and there was a long pause of absolute silence, broken by a mad yell of delight. They sprang about, tossing their long, hairy arms in the air and howling with exultation. Then they fell back from the edge, formed themselves again into line, and waited for the next victim.

This time it was Summerlee. Two of his guards caught him by the wrists and pulled him brutally to the front. H

B.9 The plaintext difference p_9

The text c_1

ave been meaning to buy a lock for these two years. People are happy who have no need of locks," he said, laughing, to Sonia. They stood still in the gateway.

"Do you go to the right, Sofya Semyonovna? How did you find me, by the way?" he added, as though he wanted to say something quite different. He wanted to look at her soft clear eyes, but this was not easy.

"Why, you gave your address to Polenka yesterday."

"Polenka? Oh, yes; Polenka, that is the little girl. She is your sister? Did I give her the address?"

"Why, had you forgotten?"

"No, I remember."

"I had heard my father speak of you... only I did not know your name, and he did not know it. And now I came... and as I had learnt your name, I asked to-day, 'Where does Mr. Raskolnikov live?' I did not know you had only a room too.... Good-bye, I will tell Katerina Ivanovna."

She was extremely glad to escape at last; she went away looking down, hurrying to get out of sight as soon as possible, to walk t

The text a_7

ways of living would be like, but it puzzled her too much, so she went on: 'But why did they live at the bottom of a well?'

'Take some more tea,' the March Hare said to Alice, very earnestly.

'I've had nothing yet,' Alice replied in an offended tone, 'so I can't take more.'

'You mean you can't take LESS,' said the Hatter: 'it's very easy to take MORE than nothing.'

'Nobody asked YOUR opinion,' said Alice.

'Who's making personal remarks now?' the Hatter asked triumphantly.

Alice did not quite know what to say to this: so she helped herself to some tea and bread-and-butter, and then turned to the Dormouse, and repeated her question. 'Why did they live at the bottom of a well?'

The Dormouse again took a minute or two to think about it, and then said, 'It was a treacle-well.'

'There's no such thing!' Alice was beginning very angrily, but the Hatter and the March Hare went 'Sh! sh!' and the Dormouse sulkily remarked, 'If you can't be

B.10 The plaintext difference p_{10}

The text c_2

ience is at rest. Of course, it was a legal crime, of course, the letter of the law was broken and blood was shed. Well, punish me for the letter of the law... and that's enough. Of course, in that case many of the benefactors of mankind who snatched power for themselves instead of inheriting it ought to have been punished at their first steps. But those men succeeded and so _they were right_, and I didn't, and so I had no right to have taken that step."

It was only in that that he recognised his criminality, only in the fact that he had been unsuccessful and had confessed it.

He suffered too from the question: why had he not killed himself? Why had he stood looking at the river and preferred to confess? Was the desire to live

so strong and was it so hard to overcome it? Had not Svidrigailov overcome it, although he was afraid of death?

In misery he asked himself this question, and could not understand that, at the very time he had been standing looking into the ri

The text b_7

begin. In anticipating the happiness of Bingley, which of course was to be inferior only to his own, he continued the conversation till they reached the house. In the hall they parted.

Chapter 59

"My dear Lizzy, where can you have been walking to?" was a question which Elizabeth received from Jane as soon as she entered their room, and from all the others when they sat down to table. She had only to say in reply, that they had wandered about, till she was beyond her own knowledge. She coloured as she spoke; but neither that, nor anything else, awakened a suspicion of the truth.

The evening passed quietly, unmarked by anything extraordinary. The acknowledged lovers talked and laughed, the unacknowledged were silent. Darcy was not of a disposition in which happiness overflows in mirth; and Elizabeth, agitated and confused, rather _knew_ that she was happy than _felt_ herself to be so; for, besides the immediate embarrassment, there were other evils befor

B.11 The plaintext difference p_{11}

The text c_3

so _all_ is permitted. No, such people, it seems, are not of flesh but
of bronze!"
One sudden irrelevant idea almost made him laugh. Napoleon, the
pyramids, Waterloo, and a wretched skinny old woman, a pawnbroker with
a red trunk under her bed–it's a nice hash for Porfiry Petrovitch to
digest! How can they digest it! It's too inartistic. "A Napoleon creep
under an old woman's bed! Ugh, how loathsome!"
At moments he felt he was raving. He sank into a state of feverish
excitement. "The old woman is of no consequence," he thought, hotly and
incoherently. "The old woman was a mistake perhaps, but she is not
what matters! The old woman was only an illness.... I was in a hurry to
overstep.... I didn't kill a human being, but a principle! I killed the
principle, but I didn't overstep, I stopped on this side.... I was
only capable of killing. And it seems I wasn't even capable of that...
Principle? Why was that fool Razumihin abusing the socialists? They are
industrious, c

The text c_7

are my first patient–well–we fellows just beginning to practise love our first patients as if they were our children, and some almost fall in love with them. And, of course, I am not rich in patients."

"I say nothing about him," added Raskolnikov, pointing to Razumihin, "though he has had nothing from me either but insult and trouble."

"What nonsense he is talking! Why, you are in a sentimental mood to-day, are you?" shouted Razumihin.

If he had had more penetration he would have seen that there was no trace of sentimentality in him, but something indeed quite the opposite. But Avdotya Romanovna noticed it. She was intently and uneasily watching her brother.

"As for you, mother, I don't dare to speak," he went on, as though repeating a lesson learned by heart. "It is only to-day that I have been able to realise a little how distressed you must have been here yesterday, waiting for me to come back."

When he had said this, he suddenly held out his hand to his

B.12 The plaintext difference p_{12}

The text c_4

raight to Porfiry? Eh? What do you think? The matter might be settled more quickly. You see, mother may ask for it before dinner."

"Certainly not to the police station. Certainly to Porfiry," Razumihin shouted in extraordinary excitement. "Well, how glad I am. Let us go at once. It is a couple of steps. We shall be sure to find him."

"Very well, let us go."

"And he will be very, very glad to make your acquaintance. I have often talked to him of you at different times. I was speaking of you yesterday. Let us go. So you knew the old woman? So that's it! It is all turning out splendidly.... Oh, yes, Sofya Ivanovna..."

"Sofya Semyonovna," corrected Raskolnikov. "Sofya Semyonovna, this is my friend Razumihin, and he is a good man."

"If you have to go now," Sonia was beginning, not looking at Razumihin at all, and still more embarrassed.

"Let us go," decided Raskolnikov. "I will come to you to-day, Sofya Semyonovna. Only tell me where you live."

He was not exac

The text d_7

some queer things before we get back. What gun have you?"

He crossed to an oaken cupboard, and as he threw it open I caught a glimpse of glistening rows of parallel barrels, like the pipes of an organ.

"I'll see what I can spare you out of my own battery," said he.

One by one he took out a succession of beautiful rifles, opening and shutting them with a snap and a clang, and then patting them as he put them back into the rack as tenderly as a mother would fondle her children.

"This is a Bland's .577 axite express," said he. "I got that big fellow with it." He glanced up at the white rhinoceros. "Ten more yards, and he'd would have added me to HIS collection.

'On that conical bullet his one chance hangs, 'Tis the weak one's advantage fair.'

Hope you know your Gordon, for he's the poet of the horse and the gun and the man that handles both. Now, here's a useful tool–.470, telescopic sight, double ejector, point-blank up to three-fift

B.13 The plaintext difference p_{13}

The text d_1

n, even with his primitive weapons, had established his ascendancy upon the plateau. We were soon to discover that it was not so, and that he was still there upon tolerance.

It was on the third day after our forming our camp near the Indian caves that the tragedy occurred. Challenger and Summerlee had gone off together that day to the lake where some of the natives, under their direction, were engaged in harpooning specimens of the great lizards. Lord John and I had remained in our camp, while a number of the Indians were scattered about upon the grassy slope in front of the caves engaged in different ways. Suddenly there was a shrill cry of alarm, with the word "Stoa" resounding from a hundred tongues. From every side men, women, and children were rushing wildly for shelter, swarming up the staircases and into the caves in a mad stampede.

Looking up, we could see them waving their arms from the rocks above and beckoning to us to join them in their refuge. We ha

The text a_8

f anything to say, she simply bowed, and took the thimble, looking as solemn as she could.

The next thing was to eat the comfits: this caused some noise and confusion, as the large birds complained that they could not taste theirs, and the small ones choked and had to be patted on the back. However, it was over at last, and they sat down again in a ring, and begged the Mouse to tell them something more.

'You promised to tell me your history, you know,' said Alice, 'and why it is you hate–C and D,' she added in a whisper, half afraid that it would be offended again.

'Mine is a long and a sad tale!' said the Mouse, turning to Alice, and sighing.

'It IS a long tail, certainly,' said Alice, looking down with wonder at the Mouse's tail; 'but why do you call it sad?' And she kept on puzzling about it while the Mouse was speaking, so that her idea of the tale was something like this:–

'Fury said to a mouse, That he

B.14 The plaintext difference p_{14}

The text d_2

or the camp. There I got you and the guns, and here we are."

"But the professors!" I cried, in consternation.

"Well, we must just go back and fetch 'em. I couldn't bring 'em with me. Challenger was up the tree, and Summerlee was not fit for the effort. The only chance was to get the guns and try a rescue. Of course they may scupper them at once in revenge. I don't think they would touch Challenger, but I wouldn't answer for Summerlee. But they would have had him in any case. Of that I am certain. So I haven't made matters any worse by boltin'. But we are honor bound to go back and have them out or see it through with them. So you can

make up your soul, young fellah my lad, for it will be one way or the other before evenin'."

I have tried to imitate here Lord Roxton's jerky talk, his short, strong sentences, the half-humorous, half-reckless tone that ran through it all. But he was a born leader. As danger thickened his jaunty manner would increase, his s

The text b_8

s other feelings, which will probably soon drive away his regard for me. You do not blame me, however, for refusing him?"

"Blame you! Oh, no."

"But you blame me for having spoken so warmly of Wickham?"

"No–I do not know that you were wrong in saying what you did."

"But you _will_ know it, when I tell you what happened the very next day."

She then spoke of the letter, repeating the whole of its contents as far as they concerned George Wickham. What a stroke was this for poor Jane! who would willingly have gone through the world without believing that so much wickedness existed in the whole race of mankind, as was here collected in one individual. Nor was Darcy's vindication, though grateful to her feelings, capable of consoling her for such discovery. Most earnestly did she labour to prove the probability of error, and seek to clear the one without involving the other.

"This will not do," said Elizabeth; "you never will be able to make both of them goo

B.15 The plaintext difference p_{15}

The text d_3

tortured iguanodon–that dreadful cry which had echoed through the woods. I thought, too, of the glimpse I had in the light of Lord John's torch of that bloated, warty, blood-slavering muzzle. Even now I was on its hunting-ground. At any instant it might spring upon me from the shadows–this nameless and horrible monster. I stopped, and, picking a cartridge from my pocket, I opened the breech of my gun. As I touched the lever my heart leaped within me. It was the shot-gun, not the rifle, which I had taken!

Again the impulse to return swept over me. Here, surely, was a most excellent reason for my failure–one for which no one would think the less of me. But again the foolish pride fought against that very word. I could not–must not–fail. After all, my rifle would probably have been as useless as a shot-gun against such dangers as I might meet. If I were to go back to camp to change my weapon I could hardly expect to enter and to leave again without being seen

The text c_8

f you! You see what rich men we are!"

"What profit could you make?"

"How can I tell you? How do I know? You see in what a tavern I spend all my time and it's my enjoyment, that's to say it's no great enjoyment, but one must sit somewhere; that poor Katia now–you saw her?... If only I had been a glutton now, a club gourmand, but you see I can eat this."

He pointed to a little table in the corner where the remnants of a terrible-looking beef-steak and potatoes lay on a tin dish.

"Have you dined, by the way? I've had something and want nothing more. I don't drink, for instance, at all. Except for champagne I never touch anything, and not more than a glass of that all the evening, and even that is enough to make my head ache. I ordered it just now to wind myself up, for I am just going off somewhere and you see me in a peculiar state of mind. That was why I hid myself just now like a schoolboy, for I was afraid you would hinder me. But I believe," he pulled out his

B.16 The plaintext difference p_{16}

The text d_4

ll. It was at least seven feet high, and so thin that she could hardly balance upon it. A more absurd object than she presented cocked up there with her face convulsed with anger, her feet dangling, and her body rigid for fear of an upset, I could not imagine.

"Let me down!" she wailed.

"Say 'please.'"

"You brute, George! Let me down this instant!"

"Come into the study, Mr. Malone."

"Really, sir—-!" said I, looking at the lady.

"Here's Mr. Malone pleading for you, Jessie. Say 'please,' and down you come."

"Oh, you brute! Please! please!"

"You must behave yourself, dear. Mr. Malone is a Pressman. He will have it all in his rag to-morrow, and sell an extra dozen among our neighbors. 'Strange story of high life'–you felt fairly high on that pedestal, did you not? Then a sub-title, 'Glimpse of a singular menage.' He's a foul feeder, is Mr. Malone, a carrion eater, like all of his kind–porcus ex grege diaboli– a swine from the devil's herd. Th

The text d_8

first day of our circumnavigation of the plateau–a great experience awaited us, and one which for ever set at rest any doubt which we could have had as to the wonders so near us.

You will realize as you read it, my dear Mr. McArdle, and possibly for the first time that the paper has not sent me on a wild-goose chase, and that there is inconceivably fine copy waiting for the world whenever we have the Professor's leave to make use of it. I shall not dare to publish these articles unless I can bring back my proofs to England, or I shall be hailed as the journalistic Munchausen of all time. I have no doubt that you feel the same way yourself, and that you would not care to stake the whole credit of the Gazette upon this adventure until we can meet the chorus of criticism and scepticism which such articles must of necessity elicit. So this wonderful incident, which would make such a headline for the old paper, must still wait its turn in the editorial drawer.

And y

C Excerpts from motions to the Swedish Parliament

C.1 The plaintext difference p_1

The text a_1

önebidragsnivåerna

Förslag till riksdagsbeslut

Riksdagen tillkännager för regeringen som sin mening vad i motionen anförs om översyn av lönebidragsnivåerna.

Motivering

Möjligheten till lönebidragsanställning ger ökat tillträde till arbetsmarknaden för stora grupper som annars hänvisas till utanförskap. Det handlar i hög grad om livskvalitet för personer med nedsatt arbetsförmåga av skilda orsaker. Tillgången till lönebidragsanställd personal har också haft stor betydelse för folkrörelse- och föreningslivet.

Behovet av att höja den högsta bidragsgrundande lönenivån har påtalats under lång tid från såväl arbetstagare som arbetsgivare. I betänkandet av Lönebidragsutredningen "ArbetsKraft" (SOU 2003:95) påpekar utredaren att arbetsmarknadsutskottet redan hösten 2002 framförde att frågan var angelägen. Utskottet ansåg också att det var rimligt att snarast frigöra ekonomiskt utrymme för att möjliggöra en höjning.

En översyn av lönebidragsnivåerna bör därför ske snarast.

Stockholm den 24 se

The text b_1

utom i långa stycken vara samverkande.

Kristdemokraterna har under åren motionerat om en mera sammanhållen skärgårdspolitik och i några avseenden har förslagen lett till riksdagsbeslut och vidtagna åtgärder. Den utredning som sett över jordförvärvslagen gällande såväl boende- som sysselsättningsfrågor vid fastighetsöverlåtelser har dock inte presenterat några förslag som gynnar en levande skärgård. För stora delar av skärgårdsbefolkningen är fastighetsbeskattningens villkor och dess koppling till förmögenhetsvärdet helt avgörande frågor. De regionala miljö- och hushållningsprogrammen pekar på att för de människor som har sin försörjning i de traditionella skärgårdsnäringarna är taxeringsvärdena och fastighetsskatten ett betydande problem.

Orimligt höga taxeringsvärden leder i många fall till att en sedan generationer bofast befolkning tvingas bort till förmån för penningstarka fastighetsspekulanter. Det är oacceptabelt att ta ut en skatt på imaginära värden där fastighetens pris best

C.2 The plaintext difference p_2

The text a_2

ammanhang under längre tid. Diskrimineringsutredningen har också föreslagit att man skall låta dessa tjänster omfatta diskrimineringslagstiftningen men regeringen vågar inte ta steget fullt ut. Det område som här har störst betydelse är pensionsförsäkringar där kvinnor och män betalar samma avgift men där sedan kvinnorna får en betydligt lägre pensionsutbetalning varje månad på

grund av att kvinnor förväntas leva längre än män. Argumenteringen utgår från en studie av dödligheten mellan åren 1951 och 1985. Sedan det sista året i mätperioden har det gått 20 år. Medellivslängden mellan män och kvinnor har under denna tid närmat sig varandra. Det är dessutom svårt att veta hur det kommer att se ut om 25 till 30 år när dagens förvärvsaktiva går i pension. Att då förutsätta att inget har förändrats är att dra stora växlar på osäkra prognoser.

Kvinnor tecknar ofta kompletterande pensionsförsäkringar för att kompensera den låga lön de haft och att de därmed också får en låg pension. Det är in

The text b_2

bildning sker via lärarledda lektioner och studiecirklar, men även viss typ av utbildning via Internet och CD-rom förekommer.

Prov för Förarintyget, den lägsta formen av nautisk kompetens, avläggs av mellan 9 000 - 11 000 personer per år. Kursen består i allmänhet av 10 sammankomster · 3 timmar/sammankomst, d.v.s. totalt 30 studietimmar.

fven intensivkurser med båtpraktik omfattande två hela helger förekommer.

Kostnaden för en 30-timmarskurs ligger på i snitt 1 500 - 1 900 kronor.

Nära 3 000 personer tar Kustskeppar-intyget och mellan 25 - 40 personer tar Manöverintyg för högfartsbåtar. 400 personer tar Båtmekanikerintyget.

Cirka 85

Nämnden För Båtlivsutbildning (NFB) bildades på regeringens uppdrag under 80-talet av Sjöfartsverket och båtorganisationerna. NFB tar fram de krav som ska gälla för olika intyg, utser förhörsförrättare och utfärdar intyg efter avlagda prov. Detta gjordes tidigare av sjöfartsverket. När

C.3 The plaintext difference p_3

The text a_3

ovan anförts om processen kring ansökningstider och överklagandeprocedurer bör ges regeringen till känna.

6.15

Bostadsbidrag

Bostadsbidrag är ett inkomstrelaterat bidrag som ges till barnfamiljer samt till personer mellan 18 och 29 år utan barn. Just det faktum att bidraget är relaterat till inkomst gör att bidraget genererar ogynnsamma marginaleffekter. En höjd inkomst gör att bidraget trappas ned. För dem som uppbär bostadsbidrag kommer således en stor del av en löneökning att ätas upp av skatter och sänkta bidrag. Detta är en typisk fattigdomsfälla som samhället måste motverka. Med fattigdomsfälla menas ett läge där en person inte på egen hand kan arbeta sig ur sin fattigdom eftersom en löneökning äts upp av de ovan nämnda faktorerna. Oavsett vad personen gör stannar han eller hon kvar i samma löneläge. Bostadsbidraget är utformat på så sätt att ett preliminärt bostadsbidrag betalas ut först. I efterhand görs därefter en slutlig justering, och om det visar sig att bidragstagaren

The text b_3

05:157 Antagande av rambeslut om straffrättsliga regler vid föroreningar från fartyg.

Förslag till riksdagsbeslut

Riksdagen tillkännager för regeringen som sin mening vad i motionen anförs om återvinning av uttjänta plastbåtar.

Motivering

Uttjänta och skrotfärdiga plastbåtar är ett växande problem. Det finns inte idag något system för att tillvarata och återvinna materialet i båtarna. En skrotningspremie vore en lösning. En annan möjlighet är att införa ett producentansvar.

Den statliga fritidsbåtsutredningen från 1974 hade ett kapitel om destruktion av uttjänta båtar. Ingen generell metod eller återvinning kunde rekommenderas. Möjligen kunde plastbåten eldas upp men då uppstod ett nytt problem i form av föroreningar i luften.

Kommunala avfallsbränningsanläggningar med filter har svårt att använda båtar som bränsle.

År 1982 presenterade båtindustrins branschorganisation Sweboat forskningsrapporten ” Återvinning av restprodukter vid tillverkning samt skrotning av plastbåtar”. Sedan dess

C.4 The plaintext difference p_4

The text a_4

ör både synliga och dolda handikapp. Med dolda handikapp menas till exempel sjukdomar som allergi, epilepsi och stomi. Kultur har en viktig betydelse för en god hälsa. Att tillgången till kultur ökar är ett viktigt mål för kulturpolitiken, men även för folkhälsopolitiken och inte minst för handikappolitiken. Inom dessa politikområden arbetas det flitigt med att försöka göra samhället mer tillgängligt för alla. Idag begränsas människor med dolda handikapp i sin kulturkonsumtion bland annat på grund av dåligt anpassade lokaler. För att en kulturlokal ska vara tillgänglig för alla krävs att ventilationssystem och städning utförs på ett passande sätt så att personer med allergibesvär fritt kan vistas i lokalen. Det kan också handla om vilken scenografi, ljussättning och musikinslag som används i kulturproduktion. Det krävs att personal i kulturlokaler utbildas i vad epilepsi, diabetes och andra sjukdomar innebär så att de kan hjälpa till om personer med dessa handikapp skulle behöva akut h

The text b_4

de asylsökande bostad på egen hand. Samtidigt som EBO-systemet successivt har byggts ut har Migrationsverket genomfört omfattande nedskärningar vad gäller antalet mottagningsplatser på förläggningarna.

Regeringen har nu överlämnat en lagrådsremiss om att avveckla stödet för eget boende och hänvisa asylsökande till de bostäder staten erbjuder. Därmed går regeringen de kommunpolitiker, bl.a. i Malmö, Södertälje och Göteborg, som vill begränsa flyktinginflyttningen till mötes.

EBO ger den asylsökande rätt till ersättning för att bo var denne vill i väntan på uppehållstillstånd. Ensamstående kan, efter det att ersättningen sänkts med

30 procent 2003, få 350 kronor i månaden och barnfamiljer 850 kronor - en summa som definitivt inte räcker till att betala hyran med.

Att avveckla EBO löser inga problem med segregation, trångboddhet m.m. Ungefär hälften av de som söker asyl ordnar bostad själva och många har självklart sökt sig till bostadsorter där det finns en rimlig chans att få jobb oc