Decoding Running Key Ciphers

Victor Redko
260659220

Marie Payne
260686859

Abstract

Artificial intelligence and natural language processing techniques can be used to decode substitution ciphers given the plaintext and keytext follow conventions of the English language. Groupings of n letters for n as large as 6 can be used in decoding methods comprising the Viterbi algorithm and in classifiers such as Multinomial Naive Bayes, logistic regression and support vector machines. We evaluate these methods with a baseline comparison model.

1 Introduction

In cryptography, a letter substitution cipher is constructed by inputting a plaintext and a key into a substitution function to produce a ciphertext. The original plaintext can be recovered by inputting the ciphertext and the key into the inverse of the function. A common choice for the substitution scheme is the *tabula recta,* where c = (p + r) mod 26 for same-case letters in the English alphabet, where c is the ciphertext, p is the plaintext and r is the keystream. The function is iterated over every letter in the plaintext, presuming the key is truncated if longer than the plaintext or wrapped around if shorter. If the plaintext or key contain patterns that can be deduced, then they can be decoded through the ciphertext by the means of statistical attacks. In the case that the key is perfectly random, never reused, and kept secret, this results in an unbreakable one-time pad. A variation of the substitution cipher, the running key cipher, uses a stream of characters from, for example, a book to create a long non-repeating key. The plaintext is typically preprocessed to remove occurences of whitespace and punctuation.

Since we know the plaintext and keystream are both obtained from the English language, our goal is to determine the most likely plaintext and keystream belonging to a subset of English that produced the given ciphertext.

2 Related Work

3 Methods

4 Results

5 Future Work

6 Conclusion