Information security
LIS4061

HW No.4: RC4 with Vigenere cipher

Luis Agustín Hernández Juárez - 147841
Carlos Octavio Estrada Pérez - 149171

31/03/2017

Abstract

For this homework assignment, a new methodology for encryption proposed in a research paper will be implemented, namely, VRC4. At its core, VR4 is a combination of two different Cyphers, the ARC4 cypher and the Vignere Cypher.

The author of this technique mentions that the main weak point of RC4 lies in the relative weakness of its Secure Pseudo Random Number Generator, and proposed appending a Vignere cypher to the encryption process to further obfuscate this property and render the cyphertext more resistant to attacks. This, the author claims, will make it more in the likelihood of DES, which by merging different cryptographic operations boasts significant security.

Introduction

ARC4:

ARC4: ARC4 stands for “Allegedly RC4), it is a free derivation of a proprietary technology held by the RSA Security. RSA Security is an American company which specializes in computer and network security.

ARC4 was created after an email containing leaked descriptions of the inner mechanics of the true RC4 was sent to a group of privacy and cryptography enthusiasts called the Cypher-punks. This group later reverse engineered the original data and built their own public domain ARC4.

Despite the fact that ARC4 is technically free, it is still dubious to which degree does the RSA have power to attack it on grounds of intellectual property. Any corporation that aims to use RC4 encryption for profit is probably safer by purchasing the services through the RSA, or requesting them for permission to use the algorithm.

Vignere Cypher:

The Vignere Cypher is also frequently called the Alphabet Cypher. It was develop by various parties independently throughout history, but most of the attribution is given to its latest inventor Blaise de Vignere, who worked upon the legacy of Giovan Battista Bellaso, who invented a simple version roughly three centuries earlier.

As is frequently the case in cryptography, this Cypher found its way to the military and the governments of the early modern era. Furthermore, this Cypher very strongly appealed to the military organizations of its time because of its relative simplicity, which meant that soldiers could be taught to utilize it without disrupting their other duties significantly, which made communications much safer. In the end, in an epoch without computers, this cypher was far more formidable than it is today (also when considering that Cryptoanalitics were still relatively young, meaning that attackers were much less sophisticated).

In general terms, this Cypher is a Caesar Cypher with a variable sized key, this key determines the amount of times a given symbol will be shifted in the ciphertext. The need for

speed in encryption and decryption meant that a lot of very interesting tools were developed to assist in the quick computation of the Vignere algorithm.

Description

Our implementation is build around a custom built, special purpose class called Easy Bit Object, or EBObj. By developing this first, we abstract all bitwise logic from the application and made it much quicker to develop the cyphers on top of the functionalities we designed for this object. Aside from this, any operations which we determined were likely to get called repeatedly, were built into this class as well.

EBObj borrows from Java's String design philosophy, it is immutable. All EBObj are unchangeable once they are built. This allowed for the safer and systematic handling of dynamic memory, since all objects would behave the same, it became possible to quickly analyze all possible use cases and develop to cover memory allocation safely for all of these.

Following an Object Oriented Programming paradigm, all the cyphers are implemented as objects, and in the case of RC4, two objects were implemented to separate two seemingly weakly coupled operations: key scheduling and random number generation.

In compliance with Object Oriented Programming, all attributes are private to the class, and minimal public methods have been implemented. However, we made the decision to minimize the load on constructors and have all parameters be loaded by public methods instead, which makes the code much simpler to manipulate as a user, as constructor calls assume some degree of familiarization with the workings of a given software's internals. Thus, it can be said that our goal was to make utilization of this method as independent from internal specifics as possible. Some improvement is still possible, specially it might be good to eliminate complicated array initializations which are required to make the software work, in future homeworks this functionality might be added to our already existing reusable classes and architecture.

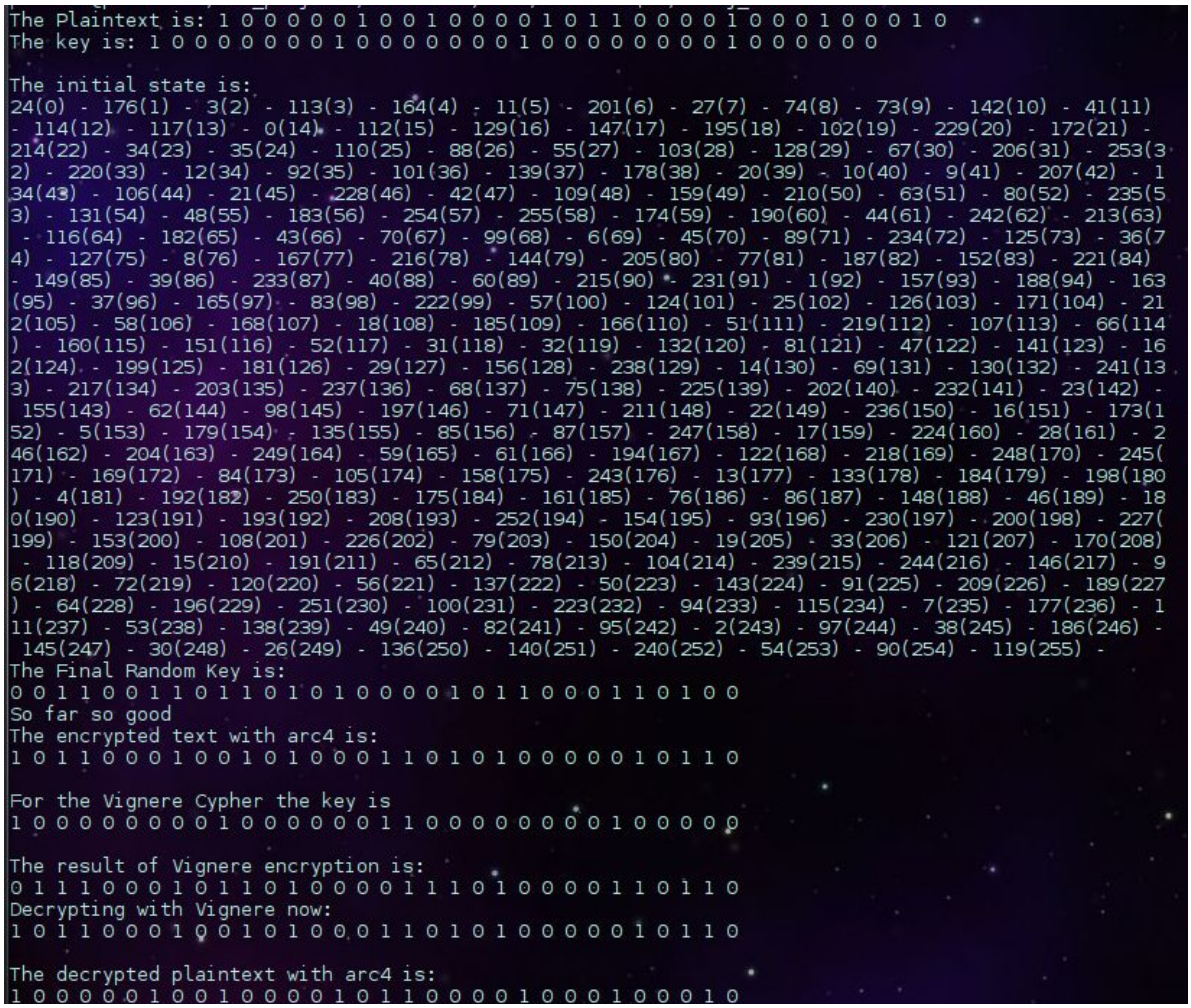
The Cyphers work like state machines, they are given a state by the user, and then told to perform some task; the state in this case is the key and plaintext, and the task is either encryption or decryption. We took special care to make the public methods of the Cyphers as simple as possible, and although further refinement is still possible, for most purposes the current architecture is sufficiently robust and orderly to serve our needs in the future and in the present.

Results

The result had us apply our implementation of RC4 and Vignere in succession for the purposes of encryption.

For decryption we applied Vignere and then RC4, since both of these algorithms are symmetric cyphers, the same key was used.

The full result is as follows:



Conclusions

Overall, Stream Cyphers were made obsolete by the availability of more powerful hardware, and in some recent time in the past, is was believed that they were no longer practical. However, the introduction of the Internet of Things and Wearable Computing has made Stream Cyphers come to the forefront of cryptography once again. This is because devices with limited power are once again becoming prevalent, but their security requirements have only gotten harsher as time progresses, thus thrusting Stream Cyphers as a lively field of research once again.

The current algorithm VRC4 is a take on this problem, by merging two different Cyphers, the researchers claim to have hardened the security of ARC4, thus opening up the path for more innovation towards a renewed implementation of RC4.

As a proof of concept this algorithm may be quite functional, but further research is probably necessary to refine the idea.

References

[1] Vignere Cypher, available in: https://www.youtube.com/watch?v=9zASwVoshiM

[2] RC4, available in: https://www.youtube.com/watch?v=9zASwVoshiM

[3] VRC4, available in: https://www.researchgate.net/publication/51958956_An_Improvement_of_RC4_Cipher_Using_Vigenere_Cipher