# Key-Amplified Cipher

Ahhyun Ahn
Hankuk Academy of Foreign Studies/International, Seoul, Korea, Republic of
Email: ahhyunahn@gmail.com

Dooyoung Kim and Taeseon Yoon
Hankuk Academy of Foreign Studies /Natural Science, Yong-in, Korea, Republic of
Email: leokim1022@naver.com, tsyoon@hafs.hs.kr

***Abstract*—The Vigenère encryption is a method that enciphers original text by adding a continuously changing correspondence of keys into plain text by means of one to one correspondence. A simple example would be to match A with 0, B with 1, and so on until Z is matched with 25. The weakness of the Vigenère encryption is that the code can be broken without the key by guessing its length. We attempted to strengthen the Vigenère encryption by multiplying the length of "key groups" so that the length of the final key would be amplified. This study reveals that the security of the Vigenère cipher, once thought to be low, can be supplemented through the use of an effective key.**

***Index Terms*—vigenère cipher, divided keys, ASCII, key length, least common multiple, kasiski, friedman, key-amplified**

## I. Introduction

Widespread development of digital communications and growing need for information security aroused the interest in the field of cryptography. The purpose of cryptography as a tool for communication is to secure data transmitted in the likely presence of an adversary, satisfying four represented requirements: Confidentiality, Authentication, Non-repudiation, and Data Integrity.

As one of primary methods of securing information, the Vigenère Encryption was utilized since it was first introduced in 1586 by French diplomat Blaise de Vigenère. The Vigenère cipher is a basic polyalphabetic cipher based on 'Tabula Recta', a square table of letters with each row shifted by one letter to the left from the upper row. Vigenère was not the first to use 'Tabula Recta', but his algorithm is meaningful since he applied concept of the "key". Although Vigenère cipher seemed to be unbreakable, it was soon proved to be vulnerable by the Kasiski method supported by the Friedman test.

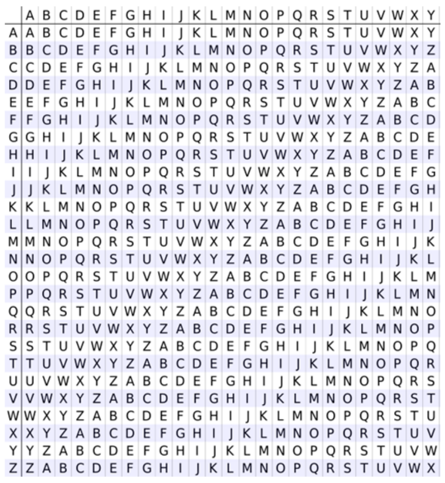
To supplement the weakness of Vigenère Cipher, we attempted to extend the limit of the original Vigenère Cipher so that a third party may not decipher the code by guessing the key length. With an improved randomizing system and encryption with plain text containing spaces, tabs, and special characters, we continued to enhance security and efficiency of Vigenère Cipher.

## II. Vigenère Algorithm and Existing Application

Manuscript received July 12, 2013; revised September 11, 2013.

### A. The Vigenère Encryption

*Vigenère Encryption:* Vigenère Encryption is a renowned encryption that utilizes more than two keys to cipher by concealing alphabet frequency in plain text. The encryption is done by substituting the plain text with the key one by one according to Fig. 1, the Tabula Recta, a 26 by 26 sized table of alphabets that consists of rows shifted one space from the upper row. The substitution continues until the end of the plain text.



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y |
| B | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| C | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A |
| D | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B |
| E | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |
| F | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |
| G | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |
| H | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |
| I | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G |
| J | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H |
| K | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I |
| L | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J |
| M | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K |
| N | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L |
| O | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M |
| P | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
| Q | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| R | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
| S | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
| T | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
| U | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
| V | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
| W | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |
| X | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V |
| Y | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W |
| Z | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X |

Figure 1. Tabula recta

TABLE I. An Example of Encryption with Vigenère Cipher

| Plain Text | T | H | E | W | A | Y | T | O | E | N | C | O | D | E |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Key | C | A | N | D | Y | C | A | N | D | Y | C | A | N | D |
| Cipher Text | V | H | R | Z | Y | A | T | B | H | L | E | O | Q | H |

For example, if the key is "candy", the first letter of the plain text is substituted with the 3-shifted-key, 'c'. The second is substituted with the 1-shifted key, and the third with 14-shifted-key. Table I specifies an example of the encryption process with plain text "The way to encode."

### B. The Weakness of Vigenère Encryption

Although Vigenère Cipher is able to supplement the weakness of direct exposure of frequency in plain text, it fails to perfectly eliminate the possibility of key breaking. If the length of the key is 5, then the $i^{th}$ ($i \equiv 1 \pmod 5$) letters in the cipher text are substituted with the same letter in the key. This process can be applied to the same remainder of 5. Therefore the Vigenère encryption is also vulnerable to “breaking” by the analysis of the frequency of letters in plain text. The key to breaking the Vigenère Cipher is to know the length of the key.

*Kasiski Method*: The Kasiski method, a renowned way to decipher a Vigenère cipher, uses the repetition of certain letters in cipher text. [1] For instance, ‘the’ is one of the most used words in English. If the key is ‘CANDY’, ‘the’ will be encrypted with the parts of the key, ‘CAN’, ‘AND’, ‘NDY’, ‘DYC’, ‘YCA’. If we find three sequences of letters repeated in the cipher text, there is a high possibility that those letters correspond to the word ‘the’ in the plain text. From this process, we can guess that those letters are substituted with one the keys: ‘CAN’, ‘AND’, ‘NDY’, ‘DYC’, ‘YCA’.

By analyzing certain repetitions in cipher code, one can easily calculate the greatest common divisor between such repetition intervals. This calculated number is highly likely to represent a multiple of the length of the key, because the key will correspond to the same letter in plain text after it has finished one loop of Tabula Recta. By tracking down the repetitions of specific elements in cipher text, Kasiski initiated the breaking of the Vigenère encryption.

*Friedman test:* The Freidman test enables intruders to estimate the length of the key used in the Vigenère Encryption with a statistical method. In specific, the Freidman test utilizes the index of coincidence, estimating 1) the possibility that randomly selected 2 letters in plain text are the same letters, and 2) the possibility of uniformly selecting (selecting through a calculated pattern) 2 letters in plain text which are same letters.[2] This test becomes more accurate as the cipher text becomes longer.

## C. *Another Application of Vigenère Encryption*

‘Alpha-Qwerty Cipher’ [3]: One of the recently introduced new applications of Vigenère Cipher, the journal ‘Alpha–Qwerty Cipher’ introduces an improved version of Vigenère Cipher. It aimed to extend the Vigenère Cipher by adding several digits and symbols, thus extending the limit of characters from 26 to 92; the new cipher utilizes q-z, Q-Z, ` ~! @ # $ % ^ & * ( ) _ - = + { } [ ] | ; : " <> , . ? /, 0-9 in both plain text and cipher text.

The Alpha–Qwerty algorithm process can be specified with experimental implementation.

Plain text: transfer10, 100 to swiss account

Key: Hell12*

Cipher text: `csn ] ^"N$)igg Y` lRI*|7Iufm;+8

Step 1: Input the key in Array ‘ke []’, and copy the determined key length to int len.

Ke [] : Hell12*

len : 7

Step 2: Using the for loop, find the corresponding cipher text based upon the key declared formerly.
For each element in the key, corresponding element from Qwerty will be saved according to e[i] [n]=Qwerty[f] to e[i][n]=Qwerty[k].

Step 3: copy message to Array txt [ ] and put length of the message to int charlen.

Step 4: Encrypt by finding the corresponding cipher text in extended Alpha–Qwerty table: if the txt[m] is equal to alpha[r], print txt[m]. Else, print en[s] [r].

Alpha–Qwerty Cipher uses Tabula Recta arranged by the order of letters on the qwerty keyboard. Table II specifies the simplified version of tabula recta used in ‘Alpha-Qwerty’, in which special characters are ignored; Table II, initially 92by92, is reduced to 26by26 in size.

TABLE II. TABULA RECTA OF ‘ALPHA-QWERTY CIPHER

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | Q | W | E | R | T | Y | U | I | O | P | A | S | D | F | G | H | J | K | L | Z | X | C | V | B | N | M |
| B | W | E | R | T | Y | U | I | O | P | A | S | D | F | G | H | J | K | L | Z | X | C | V | B | N | M | Q |
| C | E | R | T | Y | U | I | O | P | A | S | D | F | G | H | J | K | L | Z | X | C | V | B | N | M | Q | W |
| D | R | T | Y | U | I | O | P | A | S | D | F | G | H | J | K | L | Z | X | C | V | B | N | M | Q | W | E |
| E | T | Y | U | I | O | P | A | S | D | F | G | H | J | K | L | Z | X | C | V | B | N | M | Q | W | E | R |
| F | Y | U | I | O | P | A | S | D | F | G | H | J | K | L | Z | X | C | V | B | N | M | Q | W | E | R | T |
| G | U | I | O | P | A | S | D | F | G | H | J | K | L | Z | X | C | V | B | N | M | Q | W | E | R | T | Y |
| H | I | O | P | A | S | D | F | G | H | J | K | L | Z | X | C | V | B | N | M | Q | W | E | R | T | Y | U |
| I | O | P | A | S | D | F | G | H | J | K | L | Z | X | C | V | B | N | M | Q | W | E | R | T | Y | U | I |
| J | P | A | S | D | F | G | H | J | K | L | Z | X | C | V | B | N | M | Q | W | E | R | T | Y | U | I | O |
| K | A | S | D | F | G | H | J | K | L | Z | X | C | V | B | N | M | Q | W | E | R | T | Y | U | I | O | P |
| L | S | D | F | G | H | J | K | L | Z | X | C | V | B | N | M | Q | W | E | R | T | Y | U | I | O | P | A |
| M | D | F | G | H | J | K | L | Z | X | C | V | B | N | M | Q | W | E | R | T | Y | U | I | O | P | A | S |
| N | F | G | H | J | K | L | Z | X | C | V | B | N | M | Q | W | E | R | T | Y | U | I | O | P | A | S | D |
| O | G | H | J | K | L | Z | X | C | V | B | N | M | Q | W | E | R | T | Y | U | I | O | P | A | S | D | F |
| P | H | J | K | L | Z | X | C | V | B | N | M | Q | W | E | R | T | Y | U | I | O | P | A | S | D | F | G |
| Q | J | K | L | Z | X | C | V | B | N | M | Q | W | E | R | T | Y | U | I | O | P | A | S | D | F | G | H |
| R | K | L | Z | X | C | V | B | N | M | Q | W | E | R | T | Y | U | I | O | P | A | S | D | F | G | H | J |
| S | L | Z | X | C | V | B | N | M | Q | W | E | R | T | Y | U | I | O | P | A | S | D | F | G | H | J | K |
| T | Z | X | C | V | B | N | M | Q | W | E | R | T | Y | U | I | O | P | A | S | D | F | G | H | J | K | L |
| U | X | C | V | B | N | M | Q | W | E | R | T | Y | U | I | O | P | A | S | D | F | G | H | J | K | L | Z |
| V | C | V | B | N | M | Q | W | E | R | T | Y | U | I | O | P | A | S | D | F | G | H | J | K | L | Z | X |
| W | V | B | N | M | Q | W | E | R | T | Y | U | I | O | P | A | S | D | F | G | H | J | K | L | Z | X | C |
| X | B | N | M | Q | W | E | R | T | Y | U | I | O | P | A | S | D | F | G | H | J | K | L | Z | X | C | V |
| Y | N | M | Q | W | E | R | T | Y | U | I | O | P | A | S | D | F | G | H | J | K | L | Z | X | C | V | B |
| Z | M | Q | W | E | R | T | Y | U | I | O | P | A | S | D | F | G | H | J | K | L | Z | X | C | V | B | N |

*Flaws not subsidized in 'Alpha–QwertyCipher':* One of the main functions of encryption is securing the initial information in the most original way. Despite its extended limit of character and digits, the 'Alpha–Qwerty cipher' seriously lacks in eligibility due to the absence of character 'space'. In this case, the longer the text is, the harder it is for the reader. The juxtaposition based on Qwerty order in Alpha–Qwerty Cipher is inefficient regarding the unnecessary space used in storing additional arrays. The original Vigenère Cipher, however, does not require storing such pointless arrays because it enciphers the plain text according to the ASCII code order. Moreover, the gratuitous array of Alpha–Qwerty does not supplement the fundamental flaw of Vigenère Encryption, the exposure of key length via knowing the key length. Encrypting with the Tabula Recta in the Alpha–Qwerty Cipher may take some more time to break than the original Vigenère Cipher, but both of them are vulnerable when the third party knows the key length. Given the key length, breaking 'Alpha–Qwerty Cipher' is just a matter of time.

## III. Key-Amplified Encryption

### A. Motivation

The repetition of certain letters in cipher text has long been a major problem for substitution encryption, especially for monoalphabetic cipher. The simple substitution cipher also suffered from key length exposure through analyzing frequency of certain letter in the cipher text. This problem led people to elongate the key length, but the inefficient storage and application made it difficult to secure long texts. Enigma, for example, started to lengthen the key by strapping the round cylinder: the elongated key length would be the number of alphabets powered by the number of wheels [4]. We attempted to find an efficient way to encrypt the plain text without simply elongating the key length. We invented key-amplified-encryption based on the idea that enciphering several times with different length of keys generates the same effect of using the key, whose length is the least common multiplier of those key lengths. With a key length longer than $10^{10}$, the repetition of certain letters will supplement the weakness of original Vigenère Encryption.

### B. Procedure

*Key length & Key by Randomizing Program:* The Key-Amplified Cipher randomizes the key through selecting several prime numbers and multiplying them. In order to guarantee that the length of the key is long enough to hide its repetition, we limited the selection of prime numbers to including at least different 15 prime numbers (selecting a same prime number twice does not matter). 15 prime numbers listed in size are 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47. For example, selecting {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47} or {2, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47}(2 is repeated but the number of different prime number is 15) is accepted, but {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43} (the number of prime number is 14, which is smaller than 15) or {2, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43} is not accepted. The reason we chose to select numbers in prime numbers is because we wanted the least common multiplier (LCM) to be the largest, because LCM of prime numbers is just the product of each number. Otherwise, we should divide the common divisor of the numbers when we are finding their LCM. Moreover, the fifteen prime numbers make it difficult for others to estimate the right key length, because the repetition of a certain letter corresponds to the factor of the key length.

For instance, a possible key is 2*3*5*….*47, which is the product of selected 15 prime numbers that leads to 615,889,782,588,491,410. This number is longer than normal plain text length in English, thus no longer vulnerable of repetition pattern.

This implies that the least number in the possible key set is long enough not to expose the key length. However, actual memory needed to store the key is 2+3+5+ … +43+47=328. This means that we can make a key of which the length is 615,889,782,588,491,410, just with a memory to store 328 numbers. We do not need to store the whole 615,889,782,588,491,410 numbers as a key. We can pick other different 15 prime numbers or more than this, but whatever it is, the length of the final key is bigger than 615,889,782,588,491,410, because {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47} is the smallest case with 15 different prime numbers.

The key randomization is vital in encryption. The main 'security' of the key depends on how hard the third person feels is to guess the right key. In order to prevent possible guessing of the key, we utilized randomization three times in our entire Key-Amplified Encryption.

As stated above, we randomize the prime numbers to generate a key. Because we use different kind of prime numbers as factors, the Key-Amplified Cipher guarantees the possible final key length to be longer than $61*10^{15}$. There is no limit in selecting the prime number. The only limit we set is not to count any reused prime numbers (reused prime number does not consist 15 prime numbers that make the key), which is to ensure that the least common multiplier is long.

The danger of exposing the length of the key is also solved by using the key length that has many factors. The repetitions of certain letters represent all the factors used to generate the key, which means that the possibility of others to guess the right key length through calculating the least common multiplier is strongly low. In other words, the repetition in the cipher text does not represent the key length.

Not only randomizing the key factors but also randomizing the number of fractions that split the key into groups strengthens the security of the Key-Amplified Cipher. After selecting fifteen prime numbers as factors for generating the key, we divide the key into random numbers of pieces. This process further improves the frequency problem of the Vigenère Cipher.

The possible number of cases for splitting the key is

$$ {}_{4}H_{15}+{}_{5}H_{15}+\ldots+{}_{15}H_{15}={}_{30}C_{15}\approx10^{20.} \quad (1)$$

Equation (1) indicates that by splitting the key, the possibility of correctly guessing the key is successfully deducted.

The last randomization deals with rudimentary randomization. Every time the encryption is utilized, the key of the Encryption is randomized. Because the use of same key several times can expose the user in the danger of a key leak, we attempted to revitalize the security by randomizing the key every time the cipher is activated.

Through these randomizations, the Key-Amplified Encryption supplemented the correspondence between the frequency of plain English text and the cipher text.

*Encryption of Plain Text:* Through the whole process of randomizing the key, we stepped toward the enciphering process of the plaintext. The main enciphering process is done through several steps:

Step 1: Scan the code that is generated through randomizing process.

Step 2: Read each letter from the plain text and apply the Key-Amplified Cipher code in order to supplement it with cipher text. This step is just the way we do with Vigenère Cipher. Scan the plain text letter by letter until the plain text is over.

Step3: Until the plain text is over, substitute each plain text again with corresponding cipher text according to the next key. In this case, the tab key, enter, and the end of the plain text are enciphered to specific code and printed out in the cipher text. Adding the specific key code and regulating the numbers that has gone over limit, save the enciphered plain text in another temporary file and go on to the next letter in the plain text.

Step 4: Reset the data of former letters and loop until the plaintext ends.

Step 5: Save the output in the “output.txt” file, and close every file and temporary files used in encipher.

Through these multiple steps, the plain text, which should be saved in the input file, is stored in the output file. All the characters including tabs, enters and spaces are saved in the form of cipher text.

*Decryption of Cryptogram:* Contrary to the encryption process, the decryption process deciphers the cipher text into the plain text. This process also utilizes deciphering each letter by letter in the cipher text. Starting from scanning the output file, which is the file that saved the information of the cipher text, the detailed procedure of the decrypting for Key-Amplified Cipher is:

Step1: Scan the key from the key file, and save it in a temporary file.

Step 2: After the opening the cipher text file and opening the reoutput file, which the deciphered text will be saved, read the element one by one from the cipher text.

Step 3: Loop until the last element from the cipher text is scanned and supplement each element in the cipher text ends. The elements should be rightly transformed into original plain text form, including enters, tabs and spaces. The over risen number should be regulated according to the Equation (2),

C (element in cipher text)=a(value of the cipher text element)+126-29+30-b(key)[(count-1)%98]. (2)

Step 4: Loop this process until the cipher text is over, and reset the values of the variants each time the loop goes on.

Step5: Close the entire temporary and cipher text files and save the deciphered cipher text in the re-output file.

These processes of decryption successfully decipher the cipher text into the original form of the plain text. Including all the spaces and characters such as “and ‘, the original form of plain text is maintained in the re-output of cipher text. This leads to higher eligibility and efficiency for people to successfully secure their information using efficient space with split keys.

### *C. Advantages and Comparative benefits of Key-Amplified Encryption*

Contrary to other applications of Vigenère Cipher, we enhanced the eligibility of plain text. In contrast to Alpha–Qwerty, the plain text of our enhanced Vigenère Encryption does not require the user to modify the plain text through eliminating all the spaces and tabs. This advantage allows the user to secure the original form of the plain text and increases the eligibility of plain text. Encrypting the spaces and enters is especially important for long texts. Not only was the advantage of maintaining the original form of the plain text but also the efficiency regarding the memory takeup was enhanced in the Key-Amplified Encryption. The juxtaposition based on Qwerty order in Alpha–Qwerty Encryption is inefficient regarding the unnecessary space used for storing additional arrays. The original Vigenère Encryption, however, does not require storing such pointless arrays because it enciphers the plain text according to the ASCII code order. Moreover, the gratuitous array of Alpha–Qwerty does not supplement the fundamental disadvantage of Vigenère Encryption, the exposure of key length via frequency.

The efficiency regarding the key length was also improved through using Key-Amplified Encryption. Although there has been a suggestion to use the ‘Running Key Cipher’ – which uses certain books or phrases as a key – to decipher the cipher text without any relevant information, the inefficiency of this long key made this method inapplicable. The Key-Amplified Encryption, however, has a shorter key with equal or more efficiency compared to such lengthy key from the books. By splitting the key into random numbers, this Key-Amplified Encryption guarantees intensive security considering its length. This renewed encryption will allow users to utilize stronger and more effective Vigenère Encryption.

With advanced efficiency, the new key from the generated dataset that consist of numbers larger than $10^{10}$ elevates the level of security present in the original Vigenere encryption. Because the key is larger than $10^{10}$, it is difficult to apply the Kasiski method to break the code without any related information. With such long keys, the Kasiski method will be incapable of attaining enough examples of divided keys to gather the remaining cipher texts and to decrypt them as monoalphabetic encryptions. Furthermore, the Friedman test would be unable to break the code because there would be too many cases to divide the prime numbers.

With advanced efficiency, the new key from the generated dataset that consist of numbers larger than $10^{10}$ elevates the level of security present in the original Vigenere encryption. Because the key is larger than $10^{10}$, it is hard to apply the Kasiski method to break the code without any related information. With such long keys, the Kasiski method will be incapable of attaining enough examples of divided keys to gather the remaining cipher texts and to decrypt them as monoalphabetic encryptions. Furthermore, the Friedman test would be unable to break the code because there would be too many cases of dividing the prime numbers.

The Key-Amplified Encryption cannot be deciphered through the method of defining frequency. Unlike the 'Running Key Cipher', randomized number keys do not correspond to frequency of plain text in English in this case. As randomizing the splices and generating the length of the key is done without a pattern by computer programs, the key is safe from being recognized by its frequency of reoccurring letters. Furthermore, because we directly randomize ASCII code, the generated key is not a word but a collection of random letters. This method lowers the possibility of guessing the key to decipher the cipher text.

Formerly, the key for the Vigenère Encryption was prone to be a phrase or a juxtaposition of words in common English. This pattern lead the breakers to be able to correctly 'guess' the key. We aimed to get rid of even the lowest possibility of 'guessing' the key based on the 1,022,000 words in English

## IV. Result and Analysis

We attempted to supplement the flaw of the traditional Vigenère Cipher by enhancing the algorithm to cover other characters in plain text and improving the randomization to lower the correlation between the frequency of letters in the plain text and that of the cipher text. As stated in other researches, the main weakness of the Vigenère cipher is the use of periodic key streams made by repeating a chosen keyword. The tightness of the Vigenère encryption depends on the randomness of the key-stream used [5]. Thus, by splitting the key and choosing numerous prime numbers to generate the impenetrably random length of the key, we successfully improved the Vigenère Encryption to the Key-Amplified Encryption.

The experimental process of this new algorithm tested on plain text – made to contain special characters and spaces with tabs – is further illustrated below:

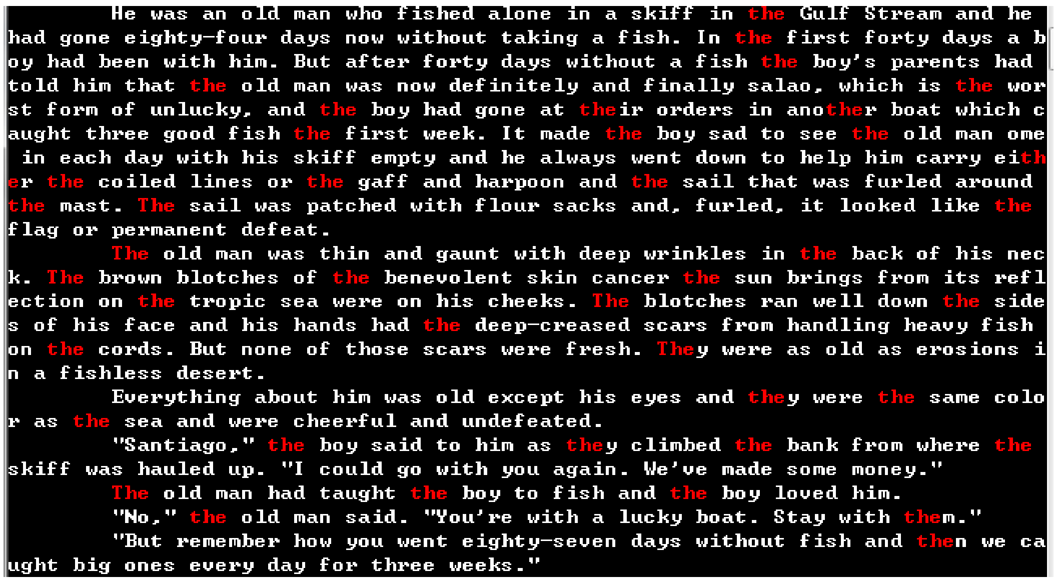
The result of our team's attempt to supplement the weakness of the Vigenère Encryption has succeeded. All of the original forms including the spaces and tabs were supplemented into the cipher code, elevating the eligibility of the Key-Amplified Encryption compared to other encryptions using applications of the Vigenère Encryption, such as the Alpha-QWERTY Cipher.

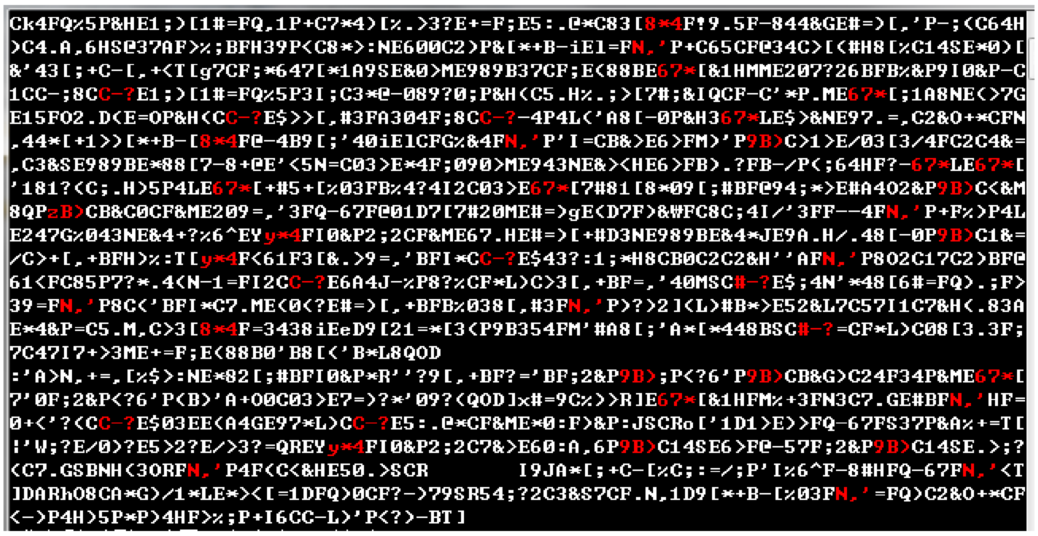
To analyze the frequency of the cipher text, we converted the ASCII code of output into letter form.

To further compare the frequency of letters present the plaintext (input), the cipher text (output) and the normal English text, we graphed the frequency rate of each letter in all texts. By illustrating the correspondence between the texts, we realized that the process of revitalizing the frequency of the normal text was solved through our invention: the Key-Amplified Encryption.

Comparing the letter frequency of the plain text, cipher text and common English writing, our cipher text frequency did not correspond to that of the plain text. Thus, one of the major problems of security in the Vigenère cipher has been solved.

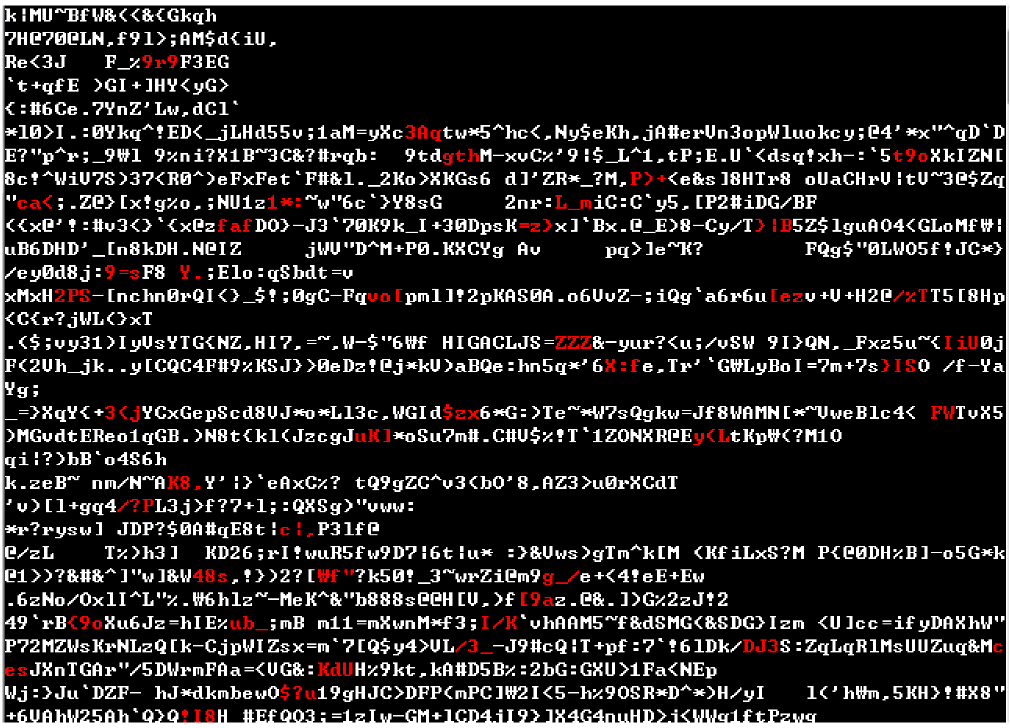
Furthermore, the efficiency and the frequency problems of the original Vigenère cipher have been eradicated regarding the length of the key and text frequency.


```
        He was an old man who fished alone in a skiff in the Gulf Stream and he
had gone eighty-four days now without taking a fish. In the first forty days a b
oy had been with him. But after forty days without a fish the boy's parents had
told him that the old man was now definitely and finally salao, which is the wor
st form of unlucky, and the boy had gone at their orders in another boat which c
aught three good fish the first week. It made the boy sad to see the old man ome
 in each day with his skiff empty and he always went down to help him carry eith
er the coiled lines or the gaff and harpoon and the sail that was furled around
the mast. The sail was patched with flour sacks and, furled, it looked like the
flag or permanent defeat.
        The old man was thin and gaunt with deep wrinkles in the back of his nec
k. The brown blotches of the benevolent skin cancer the sun brings from its refl
ection on the tropic sea were on his cheeks. The blotches ran well down the side
s of his face and his hands had the deep-creased scars from handling heavy fish
on the cords. But none of those scars were fresh. They were as old as erosions i
n a fishless desert.
        Everything about him was old except his eyes and they were the same colo
r as the sea and were cheerful and undefeated.
        "Santiago," the boy said to him as they climbed the bank from where the
skiff was hauled up. "I could go with you again. We've made some money."
        The old man had taught the boy to fish and the boy loved him.
        "No," the old man said. "You're with a lucky boat. Stay with them."
        "But remember how you went eighty-seven days without fish and then we ca
ught big ones every day for three weeks."
```
Figure 2. Input



```
Ck4FQ%5P&HE1;)[1#=FQ,1P+C7*4)[%.)3?E+=F;E5:.@*C83[8*4F!9.5F-844&GE#=)[,'P-;<C64H
)C4.A,6HS@37AF)%;BFH39P<C8*):NE600C2)P&[*+B-iE1=FN,'P+C65CF@34C)[<#H8[%C14SE*0)[
&'43[;+C-[,+<T[g7CF;*647[*1A9SE&0)ME989B37CF;E<88BE67*[&1HMME207?26BFB%&P9I0&P-C
1CC-;8CC-?E1;)[1#=FQ%5P3I;C3*@-089?0;P&H<C5.H%.;)[7#;&IQCF-C'*P.ME67*[;1A8NE<)7G
E15FO2.D<E=OP&H<CC-?E$))[,#3FA304F;8CC-?-4P4L<'A8[-0P&H367*LE$)&NE97.=,C2&O+*CFN
,44*[+1))[*+B-[8*4F@-4B9[;'40iE1CFG%&4FN,'P'I=CB&)E6)FM)'P9B)C)1)E/03[3/4FC2C4&=
,C3&SE989BE*88[7-8+@E'<5N=C03)E*4F;090)ME943NE&)<HE6)FB).?FB-/P<;64HF?-67*LE67*[
'181?<C;.H)5P4LE67*[+#5+[%03FB%4?4I2C03)E67*[7#81[8*09[;#BF@94;*)E#A402&P9B)C<&M
8QPzB)CB&C0CF&ME209=,'3FQ-67F@01D7[7#20ME#=)gE<D7F)&WFC8C;4I/'3FF--4FN,'P+F%)P4L
E247G%043NE&4+?%6^EYy*4FI0&P2;2CF&ME67.HE#=)[+#D3NE989BE&4*JE9A.H/.48[-0P9B)C1&=
/C)+[,+BFH)%:T[y*4F<61F3[&.)9=,'BFI*CC-?E$43?:1;*H8CB0C2C2&H''AFN,'P802C17C2)BF@
61<FC85P7?*.4<N-1=FI2CC-?E6A4J-%P8?%CF*L)C)3[,+BF=,'40MSC#-?E$;4N'*48[6#=FQ).;F)
39=FN,'P8C<'BFI*C7.ME<0<?E#=)[,+BFB%038[,#3FN,'P)?)2]<L)#B*)E52&L7C57I1C7&H<.83A
E*4&P=C5.M,C)3[8*4F=3438iEeD9[21=*[3<P9B354FM'#A8[;'A*[*448BSC#-?=CF*L)C08[3.3F;
7C47I7+)3ME+=F;E<88B0'B8[<'B*L8QOD
:'A)N,+=,[%$):NE*82[;#BFI0&P*R''?9[,+BF?='BF;2&P9B);P<?6'P9B)CB&G)C24F34P&ME67*[
7'0F;2&P<?6'P<B)'A+00C03)E7=)?*'09?<QOD]x#=9C%))R]E67*[&1HFM%+3FN3C7.GE#BFN,'HF=
0+<'?<CC-?E$03EE<A4GE97*L)CC-?E5:.@*CF&ME*0:F)&P:JSCRo['1D1)E))FQ-67FS37P&A%+=T[
!'W;?E/0)?E5)2?E/)3?=QREYy*4FI0&P2;2C7&)E60:A,6P9B)C14SE6)F@-57F;2&P9B)C14SE.);?
<C7.GSBNH<3ORFN,'P4F<C<&HE50.)SCR       I9JA*[;+C-[%C;:=/;P'I%6^F-8#HFQ-67FN,'<T
]DARhO8CA*G)/1*LE*)<[=1DFQ)0CF?-)79SR54;?2C3&S7CF.N,1D9[*+B-[%03FN,'=FQ)C2&O+*CF
<-)P4H)5P*P)4HF)%;P+I6CC-L)'P<?)-BT]
```

Figure 3. Short



```
k!MU~BfW&<<&<Gkqh
7H@70@LN,f91);AM$d<iU,
Re<3J    F_%9r9F3EG
`t+qfE )GI+]HY<yG)
<:#6Ce.7YnZ'Lw,dCl`
*10)I.:0Ykq^!ED<_jLHd55v;1aM=yXc3Aqtw*5^hc<,Ny$eKh,jA#erVn3opWluokcy;@4'*x"^qD`D
E?"p^r;_9W1 9%ni?X1B~3C&?#rqb:  9tdgthM-xvC%'9!$_L^1,tP;E.U`<dsq!xh-:`5t9oXkIZN[
8c!^WiV7S)37<R0^)eFxFet`F#&1._2Ko)XKGs6 d]'ZR*_?M,P)+<e&s]8HTr8 oUaCHrV!tV~3@$Zq
"ca<;.Z@)[x!g%o,;NU1z1*:~w"6c`)Y8sG      2nr:L_miC:C`y5,[P2#iDG/BF
<<x@'!:#v3<)`<x@zfafDO)-J3`70K9k_I+30DpsK=z)x]`Bx.@_E)8-Cy/T)!B5Z$1guA04<GLoMfW!
uB6DHD'_[n8kDH.N@IZ     jWU"D^M+P0.KXCYg Av     pq)]e~K?        FQg$"0LWO5f!JC*)
/ey0d8j:9=sF8 Y.;Elo:qSbdt=v
xMxH2PS-[nchn0rQI<)_$!;0gC-Fqvo[pml]!2pKAS0A.o6VvZ-;iQg`a6r6u[ezv+U+H2@/%TT5[8Hp
<C<r?jWL<)xT
.<$;vy31)IyUsYTG<NZ,HI7,=~,W-$"6Wf HIGACLJS=ZZZ&-yur?<u;/vSW 9I)QN,_Fxz5u~<IiU0j
F<2Vh_jk..y[CQC4F#9%KSJ))0eDz!@j*kU)aBQe:hn5q*'6X:fe,Tr'`GWLyBoI=7m+7s)IS0 /f-Ya
Yg;
_=)XqY<+3<jYCxGepScd8VJ*o*Ll3c,WGId$zx6*G:)Te~*W7sQgkw=Jf8WAMN[*~VweBlc4< FWTvX5
)MGvdtERe o1qGB.)N8t<kl<JzcgJuK]*oSu7m#.C#U$%!T`1ZONXR@Ey<LtKpW<?M1O
qi!?)bB`o4S6h
k.zeB~ nm/N~AK8,Y'!)`eAxC%? tQ9gZC^v3<bO'8,AZ3)u0rXCdT
'v)[l+gq4/?PL3j)f?7+l;:QXSg)"vww:
*r?rysw] JDP?$0A#qE8t!c!,P31f@
@/zL    T%)h3]  KD26;rI!wuR5fw9D7!6t!u* :)&Vws)gTm^k[M <KfiLxS?M P<@0DH%B]-o5G*k
@1))?&#&^]"w]&W48s,!))2?[Wf"?k50!_3~wrZi@m9g_/e+<4!eE+Ew
.6zNo/OxlI^L"%.W6hlz~-MeK^&"b888s@@H[V,)f[9az.@&.])G%2zJ!2
49`rB<9oXu6Jz=hIE%ub_;mB m11=mXwnM*f3;I/K`vhAAM5~f&dSMG<&SDG)Izm <U]cc=ifyDAXhW"
P72MZWsKrNLzQ[k-CjpWIZsx=m`7[Q$y4)VL/3_-J9#cQ!T+pf:7`!61Dk/DJ3S:ZqLqRlMsUUZuq&Mc
esJXnTGAr"/5DWrmFAa=<VG&:KdUH%9kt,kA#D5B%:2bG:GXU)1Fa<NEp
Wj:)Ju`DZF- hJ*dkmbewO$?u19gHJC)DFP<mPC]W2I<5-h%9OSR*D^*)H/yI    l<'hWm,5KH)!#X8"
+6UAhW25Ah`Q)Q!I8H #EfQQ3;=1zIw-GM+1CD4jI9)]X4G4nuHD)j<WWg1ftPzwg
```

Figure 4. Long

Fig. 2, Fig. 3, and Fig. 4 describe the ways to crack the codes with Kasiski method. One of the most frequently used string in English is "The". There are 42 "the" in original text, including "either" or "their". When you try to crack the ciphered text, you would look for repeated string, and as "the" is one of the most frequently used words, it would be easy to find repeating 3-alphabet-words to crack it. If certain string which is consisted of 3 words is repeated, then it would most likely be changed with word, "the". In Fig. 3, the code is ciphered with the key, "CANDY", which is only five-alphabet long. We can see "8*4", "67*", "y*4", "9B>", and "C-?" repeated. Those are possible words that can be replaced with the word "the". However, in Fig. 4, the code is enciphered with the "Key-Amplified Cipher", and none of each "the" is replaced with same cipher.

## V. Additional Researches to be Done

### A. *Optimal Final Key Length*

Choice of the prime numbers decides the length of the final key. The number of prime numbers that make up the key and the size of LCM, which is the product of these prime numbers, are related. To make the final key a bigger size, we should calculate a large LCM. There are two ways to make the LCM bigger. One is choosing more prime numbers, and the other is choosing large prime numbers. Deciding on any of them, all or even both, does not matter, but we should consider the memory that we need to store the actual key. We should balance between the length of final key and the available storage.

### *B. Optimal Ordered Pairs of the Number of the 'Key Group'and the Length of Each 'Key Group'*

We also need to balance the number of 'Key Group' and length of each key group. If we split the key into many groups, then the size of each group diminishes, and the sum of the length of actual keys is small. However we need to store bigger information of key split more than less split case. If we split the key into fewer groups, the size of each group and the sum of the length of actual keys increase. Moreover, prime numbers in one key group are multiplied, and they make up new keys. If a large prime number belongs to a group, the key made from that group can be especially longer than those by other groups and this make the sum of actual key length bigger. This brings inefficiency, so balancing the size of each prime number is also an important problem.

## REFERENCES

[1] H, C. A. Van Tilborg, "Fundamentals of cryptology: A professional reference and interactive tutorial," in *Fundamentals of Cryptology*, New York: Springer, 2000. pp. 16.

[2] H. C. A. Van Tilborg, *Encyclopedia of Cryptography and Security*, New York: Springer, 2005. pp. 115.

[3] M. K. I. Rahmani, N. Wadhwa, and V. Malhotra, "Alpha-Qwerty cipher: An extended vigenère cipher," *Advanced Computing: An International Journal*, pp. 107-18, 2012.

[4] R. F. Churchhouse, "Codes and ciphers: Julius caesar, the enigma, and the internet," Cambridge University Press, 2002, pp. 110.

[5] Singh, and Y. Kirani. "Generalization of vigenère cipher," *Center for Development of Advanced Computing*, Saltlake Sector-V, Kolkata, India, Jan. 2012. Web.

**Ahhyun Ahn** was born in 1995. She is currently a student in Hankuk Academy of Foreign Studies, Korea.
She published Fair Trade: Understanding Its Significance (Seoul, Korea: Ahhyun Ahn, 2012).
She is interested in classifying and analyzing the impact of intellectual property theft on customer's response in specific markets.
Ms. Ahn also participated in IP Panorama Course in Seoul, 2012.

**Dooyoung Kim** was born in 1995. He is currently a student in Science Major of Hankuk Academy of Foreign Studies, Korea. He is mostly interested Artificial Intelligence and has been studying support vector machine and its application to DNA.

**Taeseon Yoon** was born in Seoul, Korea, in 1972. He received the B.Eng. degree in Computer engineering from the University of INHA (IIT), Incheon, Korea, in 1998, and Ph.D. Candidate degree in Computer education from the Korea University, Seoul, Korea, in 2003.
From 1998 to 2003, he was with EJB analyst and JAVA Programmer (SCJP) in Y&J Electronics. From 2003 to 2004, he joined the Department of Computer Education, University of Korea, as a Lecturer and Ansan University, as an Adjunct professor. Since December 2004, he has been with the Hankuk Academy of Foreign Studies, where he was a Computer Science and Statistics Teacher. His current research interests include Education, Algorithm, Artificial Intelligence, Bioinformatics, and Statistics.
He was the recipient of the Best Teacher Award of the Science Conference, Gyeonggi-do, Korea, 2013.