A Modified Version of the Vigenère Algorithm

Phillip I Wilson,† and Mario Garcia†,

† Computing and mathematical Sciences, Texas A&M University-Corpus Christi, 78412 USA

Summary
Stream ciphers in general tend to be weak since they lack the benefit of diffusion. It is for this reason alone that the Vigenère cipher is able to be cracked. Without the ability to detect cycles, and intern, derive the key length, the Vigenère cipher would be highly secure. By adding a few bits of random padding to each byte, one can diffuse the statistical retentiveness found within most messages. The exact quantity of pad will be determined by a one way function in an effort to eliminate the distinguishably of the message bits from the padded random bits. This methodology moderately increases the size of the cipher text, but greatly increases the security of the cipher

Key words:
Cryptography, Vigenere, Cypher.

Introduction

With the growth of internet services requiring sensitive, such as banking, the risk to such data has also increased [7]. The only sure way to protect the data from being stolen or being accessed without authorization is to encrypt it at all times. Currently there are several good methods for encryption like AES and DES. However, the limited key space of 56 bits for DES is rapidly making it not secure [9]. AES, on the other hand, has a short history or only a few years. This means its security has not yet been proven by time as DES has.

Both of these algorithms require several rounds to encrypt a relatively small block of data. Stream ciphers, like Vigenère and Caesar in particular, only require one round. The Vigenère and Caesar cipher, however, provide little to no security. DES and AES utilize the two factors needed for a cryptographically secure algorithm, confusion and diffusion [9]. Confusion means making the correlation between the cipher text and plain text as complex as possible. Diffusion is used to mask the statistical properties of the data by spreading it through out the cipher text. Stream ciphers in general only rely on the principle on confusion [1]. This, intern makes them more susceptible to being cracked.

This paper will present an algorithm that allows diffusion that can be easily incorporated to the Vigenère stream cipher, strengthening is considerably. While the focus was on the Vigenère cipher, the principle can also be applied to other stream ciphers, and with some modification to a block cipher.

2. The Vigenère Algorithm

In the 16th century a French diplomat named Blaise de Vigenère developed a new substitution cipher. This cipher relied on a using multiple Caesar ciphers based on a key. This polyalphabetic cipher used each letter of the key to determine which Caesar cipher shift to use. Once all letters of the key had been used the cycle begins again by using the first letter of the key. This is illustrated as follows with the key "KEY":

Key:	KEYKEYK
Message:	MESSAGE
CipherTxt:	WIRCEEO

This cipher solely relies on the confusion methodology for creating cipher text [1]. The repetitive nature of message is not diffused, only camouflaged by the series of Caesar shifts.

The Vigenère cipher was considered unbreakable for nearly 300 years. However, a method to crack it was discovered by Kasiski and Kerckhoff. Both of the methods rely on the fact that the key is repeated and languages in general are relatively repetitive. Given a message is much longer than the key, the key will eventually encrypting the same set of letters previously encrypted by the key [9]. This creates a small pattern of repeating groups of letter. By finding the frequency between the repeating groups and factoring them it is possible to derive the key length [1]. Once the length of the key is know, the key is easily derived by using letter

Manuscript received February 25, 2006.

frequency analysis on each group of Caesar ciphers. The longer the key length is, the more arduous the task of breaking the code. In fact, if the key is at least as long as the message, the cipher text is immune from a cipher text only attack. When this occurs it is known as a one time pad.

With the advent of computer the Vigenère cipher has become even easier to break. Most cipher texts can be cracked within a few seconds even with long keys. This cipher is now considered trivial to break and provides no security by today's standards. However, it is used in many stronger encryption algorithms like the Advance Encryption Standard (AES) [8]. This is because when used the exclusive or (XOR) operation is performed on with a binary key and message, a type of Vigenère cipher is performed, all be it with only an alphabet of size two.

3. Description of the New Algorithm

The new algorithm boasts one major advantage over the classical Vigenère cipher; it has the added benefit of diffusion. The diffusion is provided by adding random bits to each byte before the message is encrypted using Vigenère. The amount of random bits is determined by a one way function, F(x), consisting of a prime, p, a generator less than p, g , and a positive constant less than eight, c. The message length should be less than p to prevent the possible detection of cycles. The equation 1 shows F(x) where x represents the n^{th} character of a message. To reduce the size of the pad to a reasonable number, F(x) is reduced by performing (F(x) mod 8) + 1. This allows the pad to range from 1 to 8 bits.

$$F(x) = (g^{x} + c) \bmod p \qquad (1)$$

Table 1: Distribution of padded bits, x is a random bit, t is the bit value of the message

Number of padded bits	Byte 1	Byte 2
1	Xttttttt	t0000000
2	Xttttxtt	tt000000
3	Txtttxtt	txt00000
4	Xttxttxt	txtt0000
5	Ttxtxtxt	xtxtt000
6	Xttxtxtx	txtxtt00
7	Txtxtxtx	txtxtxt0
8	Xtxtxtxt	xtxtxtxt

The average pad per byte will depend of the variables chosen for F(x), but ranges from 4 to 5 bits. For the ease of computation 4.5 will be used in the future to represent the average bits of pad per byte. Each of the eight possible values for the pad has a unique distribution for the padded bits as shown in table 1. Once each byte has been padded it is concatenated to the last bit of the previously padded bytes. The result is that the start of the next padded byte may be in the middle of the byte as shown below where F(1) = 5 and F(2)=4:

ttxtxtxt xtxtt**x**tt xttxttxt t0000000

These random bits provide two important facets to the encrypted message. The natural repetitiveness of a language is obscured and diffused by the random padded bits. With the exception of the first byte, the most significant bit (MSB) of each plain text byte has an equal probability of being at any other bit position once the padding is added. This diffuses the byte characteristics, and in turn, the language characteristics, throughout the padded message. The remainder of last byte is filled with random padding. Since random padding is used, the message itself is obscured since one cannot distinguish which bits belong to the message and which bits are random. It has the added benefit that the same message padded with the same F(x) function twice will yield to different padded messages.

Distinguishing which bits belong to the message and which are random solely from the padded text is an NP-hard problem [4]. However computing the values for F(x), where x ranges from 1 to the length of the message, takes O(n) time. This makes the padding very efficient and effective. Since the Vigenère cipher is very good at the confusion aspect of the cryptography, the padded message is then encrypted using the 256 character ASCII alphabet. By using the ASCII alphabet instead of AND an XOR, more of the bits have a potential for change. For example if an 'A' is XOR with a 'b', the MSB will never change. But if the ASCII alphabet is used, the MSB changes from a 0 to a 1. The key needed to perform encryption and decryption using this methodology is as follows:

Key: (p, g, c , Vigenère key)

The characteristics of the enciphered padded text greatly differ from the enciphered text generated using only the Vigenère cipher. The Kasiski/Kerckhoff method for discovering the key used to perform the Vigenère cipher no longer is effective when the message is padded with random bits [5]. Thus the length of the key cannot be

determined, nor can the value for the key. This can be illustrated by encrypting a highly repetitive text, a message consisting of 240 'a' characters and a short Vigenère key. The key used to perform the encrypt was (60217, 9472, 4, 'key'). Under normal Vigenère encryption the pattern would emerge after six characters since the key was only three characters long. However using the modified algorithm, between 0 and 2, two character group repetitions will appear depending on the random bits used. There will not be any repetitions of three character groups, as it would happen with the classical Vigenère cipher.
It may be possible to determine the length of the Vigenère key with such a repetitive message, if the repeating character groups appear near each other, but determining the key will not be so easy. Since the byte characteristic no longer exhibit the frequency characteristics of the language, determining the value of the Caesar shift must be done by brute force [2]. This problem has a complexity of 256^n, where n is the key length. A key of eight or more characters would provide at least the same resistance to a brute force attack as 128-bit AES. Even if a probable Vigenère key is discovered, it is required to distinguish the random bits from the message bits without computing F(x). As previously mentioned, this is an NP-hard problem.

With only the cipher text, deriving a decent size message would be computationally infeasible, if not impossible [3]. With a key however, it takes the same amount of time to decrypt the message as it does to encrypt it. One simply decrypts the message using the Vigenère key. The F(x) function is then computed for each byte, removing and discarding the pad after each F(x) is computed. This produces the original plain text in its original size.

This methodology does have some potential drawbacks. The main drawback is that the size of the encrypted message will be increased by around 56%. For areas with low bandwidth or limited storage capacity this cipher cannot be used. However for most communication channels where encryption is required, a moderate increase in message size will not have a significant impact. The other major drawback is that a good random number generator is required to create an effective cipher. Since every message is essentially padded with a random number, any cycles or tendencies of a random number generator would make deriving the pad length easier. This cipher is also not parallelizable in its current state. However, if a fixed plain text block length is used, the algorithm can be converted to work as a block cipher similar to [6]. It is also important to realize that if one bit is lost; the remainder of the message will be useless.

4. Other Potential Applications

While the Vigenère cipher was used to illustrate the effectiveness of the padding at diffusing the message, it can easily be adapted to any steam ciphers. This algorithm will allow the diffusion to be added to any stream cipher, increasing the security of the cipher. It can also be modified to work with a block cipher by fixing the block size. Since both parties are privy to the F(x) function, a prearranged plain text block size can result in a fixed cipher block size. F(x) would need to be computed for 1 to the length of the plain text to determine the proper cipher text block size.
As previously mentioned, a one-time pad (OTP) is currently the only provably secure encryption algorithm. The cravat is that the key has to be random and at least as long as the message. If a key of length m can be secretly exchange, it is possible to exchange a message of length m without using the OTP. In other words, with a one time pad, the problem lies in key exchange. With the padding algorithm, around one third of the message is random. If one uses a method to create a key as long as the message using the random bits, an OTP key could be used to encrypt the message. To retrieve the OTP key it is required to use the Vigenère key to get the first few bytes of the random data, which is used to derive the OTP. As more of the message is decrypted more random data is exposed, allowing more of the OTP key to be derived. The OTP key is encrypted and transmitted within the message. The Vigenère key only exposes the first few bytes of the OTP key while each byte of the OTP key exposes more of the key.

The proposed method for creating and using an OTP key would be just as secure as any other OTP since the key is encrypted with an OTP. The only difference would be that the key exchange problem has been resolved by increasing the length of the encrypted message by around 56%. With most environments this should not be a problem since most traffic is not encrypted. It should also be mentioned that the Vigenère key needs to be at least sixteen bytes, or 96 bits long to ensure that the there at least two byts of random data to derive the six byte of OTP needed to decrypted the next six bytes of the message.

5. Conclusion

Although the Vigenère cipher is not secure when used to encrypt unpadded messages, it and other stream ciphers can be made as secure as most block ciphers by adding random bits of padding to each byte to diffuse the language characteristics. This padding methodology will

be secure as long as computing discrete logarithm is difficult [3] and distinguishing random bits from non-random bits is a hard problem. This padding can also be used to enhance the security of a block cipher or create a one time pad to encrypt the message. The OTP would be embedded and encrypted in the messaged allowing for secure transference of the key.

References

[1] Bishop, M. *Computer Security Art and Security*. Person Education Inc. New York, New York, 2003.

[2] Dachselt, F., Kelber, K., Schwarz, W., Vandewalle, J. "Chaotic versus classical stream ciphers-a comparative study**,"** Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium on Volume 4, 31 May-3 June 1998 Page(s):518 - 521 vol.4

[3] Du, W., Atallah, M. "Privacy-Preserving Cooperative Statistical Analysis," ACSAC, p. 102, 17th Annual Computer Security Applications Conference (ACSAC'01), 2001.

[4] Impagliazzo, R. "No Better Ways to Generate Hard NP Instances than Picking Uniformly at Random," Foundations of Computer Science, 1990. Proceedings., 31st Annual Symposium on 22-24 Oct. 1990 Page(s):812 - 821 vol.2

[5] Jones, C.F., III, Christman, M. "Genetic algorithm solution of Vigenere alphabetic codes," Soft Computing in Industrial Applications, 2001. SMCia/01. Proceedings of the 2001 IEEE Mountain Workshop on 25-27 June 2001 Page(s):59 – 63

[6] Nacira, G., Abdelaziz, A. "Secured net-banking by θ-Vigenere in Syverson's protocol," Computer Systems and Applications, 2005. The 3rd ACS/IEEE International Conference on 2005 Page(s):67

[7] Nacira, G., Abdelaziz, A. "The θ-vigenere cipher extended to numerical data," Information and Communication Technologies: From Theory to Applications, 2004. Proceedings. 2004 International Conference on 19-23 April 2004 Page(s):413 – 414

[8] Sanchez-Avila, C., Sanchez-Reillol, R. "The Rijndael block cipher (AES proposal): a comparison with DES," Security Technology, 2001 IEEE 35th International Carnahan Conference on 16-19 Oct. 2001 Page(s):229 – 234

[9] Schineier, B. *Applied Cryptography*. John Wiley and Sons Inc. New York, New York, 1995.

Mario Garcia Mario A Garcia is an Associate Professor at Texas A&M University-Corpus Christi. Dr. Garcia received a B.S. degree in Electrical Engineering from Tecnologico de Saltillo, Mexico, He received a M.S. in Electrical Engineering from Tecnologico de la Laguna, Mexico, He received a M.S. in Artificial Intelligence from ITESM, Mexico, and he received his Ph.D. in Computer Science from Texas A&M University. garciam@falcon.tamucc.edu

Phillip Wilson Phillip I Wilson is a graduate student in Computer Science at Texas A&M University-Corpus Christi. He received a B.S. degree in Computer Science and Mathematics from Texas A&M University – Corpus Christi. He will be graduating in May 2005 with his Master's degree in Computer Science. He has been a USAA intern for three summers and has accepted an offer to work as an IT analyst/programmer for USAA upon graduation.