Deterministic random number generation for one time pads

Creating a Whitenoise super key

André Brisson
Whitenoise Laboratories Canada Inc.
#701 – 1736 W. 10th Ave.
Vancouver, British Columbia Canada V6J 2A6
abrisson@wnlabs.com

***Abstract*—In this paper we examine the creation of Whitenoise super keys. A Whitenoise key is an exponential length key that can be used as a one-time-pad and that can in turn create an unlimited number of unique keys. Because its structure is more akin to the mechanical Enigma it is not vulnerable to arithmetic attacks. Because they are random they are resistant to brute force and quantum computing attacks.**

Keywords—Whitenoise; one time pad; encryption; authentication

I. INTRODUCTION

The two primary underlying frameworks to provide encryption, authentication and other key based security controls for electronic communications are asymmetric (PKI) and symmetric systems. Combining both in a hybrid system provides a two-channel, multi-factor challenge to any bad actors. In symmetric systems, the sender and the recipient use the same code or key to encrypt and decrypt the message or authenticate the person or device.

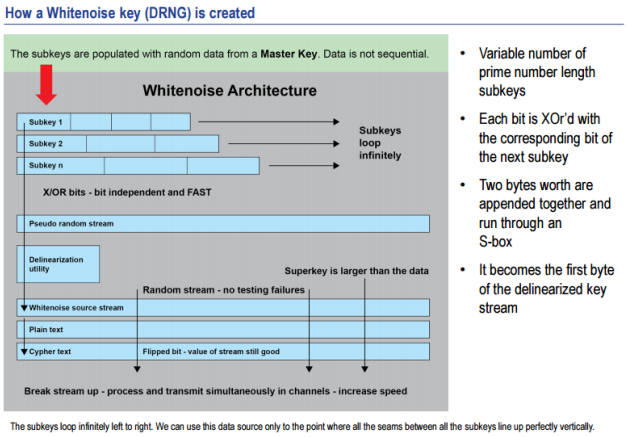
The only completely secure cipher that creates these keys and which cannot possibly be broken or deciphered is the One-Time Pad (OTP). An OTP operates on a stream of bits that contains the plaintext message. A secret, random bit-stream of the same length as the plaintext (the key) is configured to operate as a one time pad.

A one time pad has the following characteristics:

- The key or token from a key stream is random.
- The key or token from a key stream is used only once.
- The key use is at least twice the size of the plaintext to be encrypted.

To encrypt the plaintext with the key, each pair of bits from the key and plaintext is sequentially acted on by the exclusive-or function to obtain the ciphertext bit. The ciphertext cannot be deciphered if the key is truly random and the key is kept secret from an unauthorized party. The problem with this method is that the key should be at least the same length as the message. If a shorter key is used and repeated then the cipher can be broken. In some cases the data which needs to be encrypted is extremely large.

The approach we study herein is a method of generating a random key, or OTP, which is of variable length and that allows for encryption of very large amounts of data.



II. KEY CREATION

The process of generating a Whitenoise super key having length x is comprised of the following steps:

i) selecting a number n of sub-keys each having a unique non-repeating length m;

ii) generating n random numbers, one for each sub-key, each having length m;

iii) generating a n+1st random number R;

iv) for each bit whose position in said nth random number is calculated as Modm(R) applying a function to all n bits to generate a binary value;

v) concatenating said binary value to the end of the encryption key; and

vi) repeating step iv) until the key is x bits in length.

Preferably the selected length m of each sub-key is a prime number.

Each of the n random numbers is generated by:

i) generating a first random number which is not a perfect square;

ii) calculating the square root of the first random number;

iii) generating a second random number;

iv) commencing with a digit whose position in the first random number is calculated based on the second random number, taking finite strings of digits sequentially and converting each finite string into a hexadecimal byte;

v) concatenating each hexadecimal byte sequentially to the random number until the selected length m of the random number has been reached.

Creating a non-repeating key of indefinite length, a Super Key, is formed by combining sub-keys. Any number n of sub keys Kj, can be specified depending on the application, security need and processing resources available. Because of multiplicity more sub-keys create longer non-repeating Super Keys. The length of each sub key is a prime number of bytes (preferably with prime numbers larger than 10). The length of a key is determined by multiplying the lengths of the subkeys. A huge advantage is that only the key structure needs to be stored (i.e. the length of the subkeys) in order to identically recreate the super key. This allows the creation of exponentially length keys with absolutely minimal amount of storage.

The first step in the process is to determine how large a Super Key, or cipher, to deploy. The number of sub-keys and the non-repeating length of each sub-key, in bytes, is selected. The sub-keys each have a unique non-repeating length. No two sub-keys are of the same non-repeating length. Preferably the sub-key non-repeating lengths are prime numbers of bytes. The selection may be done by manually entering the number of sub-keys and their prime number non-repeating lengths. Alternatively, the number of keys and their prime number non-repeating lengths is programmed into an application, or a program randomly selects the number of sub-keys and their non-repeating length. For n sub-keys K^ the non-repeating length of the Super Key will be Size (K2) X Size (K2) X Size (K3) ... X Size (KJ. For example, assume 10 sub-keys of the following prime number non-repeating lengths are used:

Sub Key 1=13 bytes= K1

Sub Key 2=17 bytes= K2

Sub Key 3=19 bytes= K3

Sub Key 4=23 bytes= K4

Sub Key 5=29 bytes= K5

Sub Key 6=31 bytes= K6

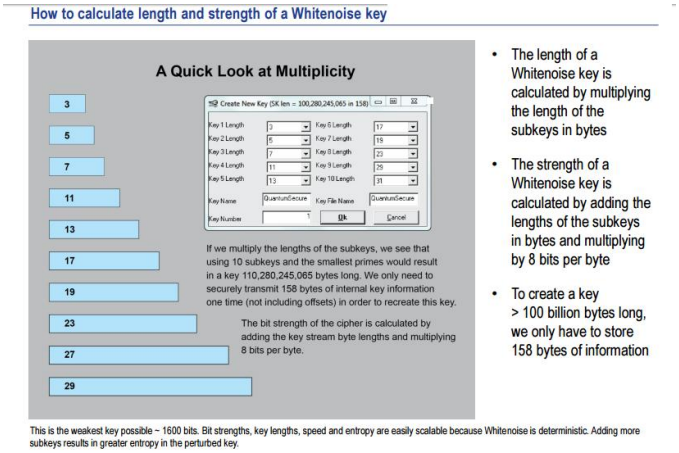
Sub Key 7=37 bytes= K7

Sub Key 8=41 bytes= K8

Sub Key 9=43 bytes= K9

Sub Key 10=47 bytes= K10

The resulting non-repeating Super Key length is 13x17x19x23x29x31x37x41x43x47=266,186,053,068,611 bytes.



Thus, using a small number of sub-keys, each of small prime number non-repeating length results in an extremely long non-repeating Super Key. The total definition for the size for the multi-key above is contained in 300 bytes and the header.

While preferably the non-repeating length of each sub-key is a prime number of bytes, to improve the randomness of the resulting cipher, the method will also work if non-prime number lengths are used, as long as the resulting cipher is very large.

Each sub-key of the multi-key process may be created as follows. First a random number which is not a perfect square is generated, preferably by a computer random number generator. This serves as a "first seed value" O. Random number generators that are included in the operating systems of most computers are pseudo-random and not very robust.

These values, however, are sufficient as a starting point. It is verified that the selected value O is not a perfect square. If it is, then additional random values will be generated until one meets this criterion. A second random number P ("second seed value") is also generated by the computer's random number generator to serve as an offset to be utilized in this process. The square root Q of this first seed value O is calculated, resulting in an irrational number Q (one that extends infinitely after the decimal point since it is not evenly divisible). The resultant string of digits after the decimal point is potentially infinite in length and is highly random. The computer discards the digits in front of the decimal and computes the number Q up to P digits after the decimal. Then, starting at the Pth digit of Q after the decimal point, the computer sequentially selects 4 digits at a time, and calculates the Mod 256 value of the 4 digits. The single resultant random 8-bit byte may be represented in hexadecimal notation. This value is used as the first byte of the sub-key. This process is repeated 4 digits at a time, continuing with the next digits in sequence, until a string of random data equal to the prime number non-repeating length of the sub-key being created is completed. This process is repeated for all the sub keys until the non-repeating length for all the sub keys are created. Each sub-key then is formed by taking the non-repeating string of bytes thus created, and repeating it as often as necessary in combination with the other sub-keys to create the Super Key.

Once all the sub-keys are created as above, the Super Key (cipher) is created to the length required. This means the Super Key will continue to be created to encrypt the associated data to be encrypted, and continues to be created only until all the data is encrypted. First a random number R ("third seed value", or the starting offset for the Super Key, as opposed to the starting offset P for the number Q) is generated. Starting with any one of the n sub-keys, having length m, the Modm of R is calculated and the Modm(R)th byte of each sub-key is consecutively exclusive-or'd (X/OR'd) with the corresponding Modm(R)th byte of every other sub-key. For example, if R=100, and the length of the first sub-key is 97 bytes, then the 3rd byte of sub-key 1 is selected and X|OR'd with the corresponding bytes of the other remaining sub-keys based on R selected in the same way. The process is repeated until all the selected bytes from each sub-key have been X/OR'd. The resultant binary value is then added to the Super Key. The next, subsequent bytes of sub-key 1 are then X|OR'd with the next byte of Sub key 3 and so on. Again the process is repeated until all the selected bytes from each sub-key have been X/OR'd. The resulting binary value of each function is again added to the Super Key. While the X/OR function is preferred, it is apparent that other functions can be applied. For example, mathematical functions of addition or subtraction can be used. As each byte of the Super Key is generated, the corresponding byte of the plaintext message is

then encrypted with the corresponding byte of the Super Key by the exclusive-or function or some other mathematical function. Once all the bytes of the plaintext message have been encrypted the generation of the Super Key terminates. The encrypted message can then be decrypted applying the inverse of the encrypting function to it and the Super Key.

While preferably the random non-repeating string which forms each sub-key is generated as described above, the method will also work if the non-repeating string of each sub-key is simply generated by a random number generator to form each sub-key, as long as the overall resultant length of the Super key is sufficiently large so that the resultant Super Key is at least double the size of the data to be encrypted.

III. Key management

The following cloud paradigm is one that benefits citizens, consumers and service providers. It is a technological approach that can be used to hold back encroachment on personal rights and create a manageable environment for service providers. The question that must be answered democratically and in public view is "How do we balance security and privacy?"

There are challenges faced by carriers and service providers in complying with government surveillance mandates.

One implementation uses the distribution of a generic key schedule by carriers and cloud service providers to their clients. The endpoint client perturbs this generic key schedule with their own secret pass phrases that serve as secret subkeys to make a unique key secret to the endpoint client. When the client sends encrypted data through communications and uploads it into the cloud for storage the carrier or service provider does not have a copy of the key. As such, there is little other than the capture and retention of encrypted data that a service provider can be compelled to do.

When the upload or transmission is further encrypted (double encrypted) over TLS or SSL the carriers have no copies of the first key and cannot compromise client data themselves. Neither are they able to provide all the necessary key material to outside agencies pushing responsibility to the endpoint originator of communications.

This technique was originally developed as a method of allowing manufacture of electronic components requiring cryptography in non-friendly trading partner countries without compromising final security. This was because the US DoD had so many components which were manufactured in China.

This is globally patented technology and any commercial use of this technology requires a license. Any physical product or source code is available with permit for research.

IV. Conclusion

Recent events highlight the need for a renaissance in cryptography and trusted computing. NSA and NIST announced in 2015 that they are moving away from the current cipher suite because RSA, AES and ECC are all easily broken with quantum computing attacks.

Google recently demonstrated a collision attack and broke the SHA-1 hash function which is a workhorse in computer security.

The latest Wiki Leaks dump, Vault 7, has highlighted the pervasive insecurity of current cryptographic standards and trusted computing frameworks.

An easy-to-implement one-time-pad key is critical to rapidly harden our computing systems, telecommunications and critical infrastructures.

A one-time-pad is the only technology that can be proven mathematically to be unbreakable.

V. Acknowledgement

Stephen Lawrence Boren, Vancouver, British Columbia, Canada

VI. REFERENCES

[1] D. Wagner, "A Security Evaluation of Whitenoise," University of California, Berkeley, Oct. 2003. [Online]. Available: https://eprint.iacr.org/2003/218.pdf

[2] I. Traore and M.Y. Liu, "Evaluation of Whitenoise Cryptosystem. Part 1: Encryption Algorithm," Technical Report ECE03-3, University of Victoria, British Columbia, Canada, Feb. 2003. http://www.wnlabs.com/downloads/UVIC_Performance_Analysis.pdf

[3] A.J. Brisson,"Cyber Belt presentation for NSA/NIST 2016 – Quantum computing attack resistant security and the cyber belt." http://www.wnlabs.com/pdf/Cyber_Belt_Presentation.pdf

[4] S.L. Boren and DW (CSE), Mathematical rebuttal refuting false break technique, http://www.wnlabs.com/pdf/Response.pdf

[5] Laurie Perrin and A.J. Brisson, TLS Whitenoise DIVA extension, http://www.wnlabs.com/technology/WNL_TLS_extension.php

[6] André Brisson,"Key distribution method to shield service providers from key management - https://www.linkedin.com/pulse/key-distribution-paradigm-removes-management-from-carriers-brisson

[7] A.J. Brisson, "Factorization of a prime number composite" https://www.youtube.com/watch?v=GwkwgR_78dQ&feature=youtu.be

[8] A.J. Brisson, "Rapid Factorization of Semi Primes," http://www.wnlabs.com/pdf/Rapid_Factorization_of_semiprimes.pdf

[9] A.J. Brisson, "The Whitenoise Algorithm – A Visual Look," Technical Report, Whitenoise Laboratories, Vancouver, British Columbia, Canada, Nov. 2011. [Online]. Available: http://www.wnlabs.com/pdf/WhitenoiseAlgorithmVisualLook.pdf

[10] C. Coram-Mekkey, "Whitenoise Laboratories – An Overview," White Paper, Whitenoise Laboratories, Vancouver, British Columbia, Canada, June 2015. [Online]: http://www.wnlabs.com/papers/Whitenoise Overview Short.pdf

[11] S.L. Boren and A.J. Brisson, "Dynamic Distributed Key System and Method for Identity Management, Authentication Servers, Data Security and Preventing Man-in-the-Middle Attacks," U.S. Patent Application 2009/0106551 A1, Apr. 2009.

[12] A.J. Brisson, "Dynamic Identity Verification and Authentication, Dynamic Distributed Key Infrastructures, Dynamic Distributed Key Systems and Method for Identity Management, Authentication Servers, Data Security and Preventing Man-in-the-Middle Attacks, Side Channel Attacks, Botnet Attacks, and Credit Card and Financial Transaction Fraud, Mitigating Biometric False Positives and False Negatives, and Controlling Life of Accessible Data in the Cloud," U.S. Patent Application 2013/0227286 A1, Aug. 2013.