A 0ne-pass Key Distribution Protocol Based on the Hamming Code

Magdy Saeb,

Arab Academy of Science, Technology and Maritime Transport, Alexandria, Egypt

Great Wall Information Security (GWIS), Kuala Lumpur, Malaysia

mail@magdysaeb.net

Abstract: Encryption key distribution is a fundamental cryptographic research area. In this work, we present a one-pass key distribution protocol utilizing the Hamming Error Detection and Correction Code. The shared secret is a random string of bits. For high-security applications, this shared secret can be changed for generating a new key. For other types of applications, we show that this step is nonobligatory since the bit string is partially updated every time a new key is generated. Additionally, the Strict Avalanche Criteria (SAC) of any hash function design results in a change of 50% of the output bits for a one-bit change in the input to the hash function. Therefore, the shared secret is acquired only one time as an initial value (IV). It also serves as an authentication vehicle. The proposed technique uses simple arithmetic and logic operations that provide uncomplicated and efficient software and hardware implementations.

Keywords: Key Distribution, Hamming Code, Error Detection, Error Correction, Cryptography, Authentication, Encryption.

I. Introduction

Cryptography aim is to provide security and authentication between communicating entities. To achieve this two-fold aim, in the presence of an active adversary, the two communicating parties must have a shared secret. This shared secret is traditionally used to generate the encryption key. Key distribution is one of the most addressed problems in information security. The problem was first studied by Needham and Schroeder [1]. Other works followed such as Baur et al. [2] who presented a key distribution protocol using event markers where communicants do not have to keep absolute sense of time. Bellare and Rogaway [3, 4] provided a discussion on the session key distribution in the three-party setting of Needham and Schroeder. They presented a definition, a protocol and a proof that the protocol satisfies the definition assuming minimal assumption of a pseudorandom function. In their second paper they discussed the problems of two-party mutual authentication and key distribution in the complexity-theoretic framework of modern cryptography, Huang [5] presented a key pre-distribution scheme for secure wireless sensor networks. The paper provides an approach that any pair of sensor nodes can find a common pairwise secret key between them with a simple calculation. Chang et.al [6] presented a conference key distribution based on interpolation polynomials. A sealed lock is used to lock the conference key in such a way that only the private keys of the invited members are matched. The sealed lock is then made public or distributed to all. Only legitimate users can disclose it and obtain the conference key. Liu et al. [7] proposed a practical deployment model, where sensor nodes are deployed in groups, and the nodes in the same group are close to each other after the deployment. Based on this model, the paper develops a novel group-based key pre-distribution framework, which can be combined with any of existing key redistribution techniques. Wahiddin, et. al., [8] used satellite random transmissions, Von Neumann corrector and a hash function to generate the key between communicating entities. Cervesato et al. [9] discussed a method that assembles the security properties of a protocol by composing the guarantees offered by embedded fragments and patterns. It sheds light on fundamental notions such as challenge-response and fed a growing taxonomy of protocols. Sun et al. [10] proposed three secure authentication and key distribution protocols to provide perfect forward secrecy of these three classes. All these protocols are used in protecting poorly-chosen passwords chosen by users from guessing attacks and replay attacks. Jingbo, et. al, [11] presented a model of multi-party secret key agreement, in which one terminal called the communicator can transmit public messages to other terminals before all terminals agree on a secret key. A single-letter characterization of the achievable region is derived in the stationary memory less case. The model generalizes some other models of key agreement. Particularly, key generation with an all-knowing helper where the communicator knows all sources, for which they can derive a one-shot zero-rate converse for the secret key per bit of communication.

In the present work, we focus on the two communicating parties' symmetric encryption case. We propose a key distribution protocol based on the Hamming error detection and correction code. The relative simplicity of the method contributes to uncomplicated software and hardware implementations. In the following sections, we discuss the methodology of the proposed technique and furnish an example of using this protocol. Finally, we provide a summary and our conclusions.

II. The Protocol

In the next few lines, we discuss the protocol steps using a 16-bit random string S and four check bits $\mathbf{C_i} = \{C_1, C_2, C_4, C_8\}^{i}$ $\forall$ i = 1, 2, .., n where i is sequence number of the generated key. For a 32-bit random string, we have to use five check bits $\mathbf{C_i} = \{C_1, C_2, C_4, C_8, C_{16}\}^{i}$ and so forth. As observed before, n sets of check bits can be transmitted to change n bits of the shared secret bit string.

The proposed protocol is summarized as follows:

Protocol KeyDist

The agreed-upon parameters:

- *The number of bits of the shared secret random bit string S (say,16 or 32 bits),*
- *The number of transmitted sets of check bits n,*
- *The key generation hash function h (SHA-1, SHA-256, SHA-512).*

1. Initialization:
   Sender A and receiver B share a secret, say, 16-bit string S; \\This random string S is used only one time. It is the initial value (IV).
2. A $\{C_1, C_2, C_4, C_8\}^{n} \rightarrow$ B;
   \\ A sends to B n sets of randomly generated check bits C_i to change n bits in the string S
3. A, B use $\{C_1, C_2, C_4, C_8\}^{n} \rightarrow S_c$;
   \\ A, B use the check bits to correct n bits of S to get S_c
4. A, B acquire S = S_c;
   \\ Replace S by S_c.
5. S_k = Von Neumann Corrector (S_c);

\\ Apply Von Neumann Corrector on S_c to get S_k.

6. A, B generate K using: K = h (S_k); \\ Generate key (K) using the hash function (h)
7. Next Key: Go To 2;
   Or for added security, Go TO 1 using a new random bit string S. This step is optional.
8. If all keys are generated then HALT.

III. An implementation example

The time sequence diagram of this protocol is shown in Figure 1.

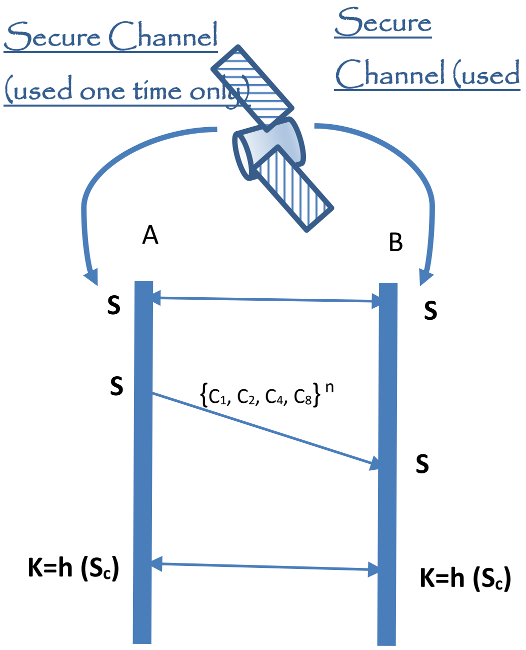

Figure 1. *The time sequence diagram of the proposed protocol demonstrating its one-shot structure.*

The shared secret is a random string of bits. This shared secret can be updated whenever a new key is generated for a high-security application. However, for other types of security applications, this step is not necessary since the random string of bits is partially updated every time a new key is generated. Recalling that the Strict Avalanche Criteria (SAC) of any hash function design requires a change of 50% of the output bits for a one-bit change in the input to the hash function. Therefore, the shared secret is acquired only one time as an initial value (IV). It also serves as an authentication vehicle. Each time a new key is generated, a new string is updated using the Hamming code check bits.

A detailed example of the proposed procedure is shown next.

1. S = 1001 0101 1000 0110
2. C_1 = {1, 1, 0, 1}
3. S_c = 1001 0111 1000 0110
4. S = S_c
5. S_K = 100101 = 91_h
6. K = SHA-1(S_K) = e9f987c3ab268ba6cf1c2ca075d6d26b01791214

   K = SHA-256 (S_K) = 7da59d0dfbe21f43e842e8afb43e12a6445bbac07c2fc26984c71d0de3f99c9c

   K = SHA-512(S_K) = bd91c84932ea4e848cde061e0a6a271ed3d381289fcf68e487119c4ac898f0919db0a287db1dec5a3870021670273268132b6445034b060ee36180b8955ddcc7

7. Next key

Summary and Conclusions

We have presented a one-shot (one-pass) key distribution protocol. The thought behind this approach is based on error detection and correction Hamming Code. The protocol starts with an agreed upon random bit string and updates it using the transmitted check bits. Von Neumann Corrector is then applied and a hash function is used to locally generate the encryption key. The adversary accesses and accumulates the check bits. However, he or she cannot figure out the resulting key since the initial value (IV) is kept secret. This initial value is left to the user to decide when to change it depending on the required degree of security and the convenience of usage. It serves three functions; as a starting point of the protocol, as an authentication vehicle and also as a group key.

We conjecture that the proposed methodology is secure against known attacks.

References:

[1] R. M. Needham, M. D. Schroeder, "Using Encryption in Large Networks of Computers," Communication ACM21, pp.993-999, 1978.

[2] R. K. Bauer, T. A. Berson, R. J. Freietag, " A Key Distribution Protocol Using Event Markers," ACM Transactions on Computer Systems, Vol. 1 Number 3, pp. 249-255, August 1983.

[3] M. Bellare, P. Rogaway, "Entity Authentication and key Distribution," Advances in Cryptology, Crypto93, Proceedings, Springer-Verlag, 1993.

[4] M. Bellare, R. Canetti, H. Krawczyk, "A Modular Approach for the Design and Analysis of Authentication and Key Exchange protocols," Proceedings of the 30 th Annual Symposium On the Theory of Computing, ACM, 1998.

[5] H. Huang, "A Pairwise key Pre- distribution for Wireless Sensor networks," ISI 2008 Workshops, LNCS 5075, pp. 77–82, 2008.

[6] C. Chang, C. Lin, C. Chen, "A Conference Key Distribution Scheme Using Interpolating Polynomials," International Journal of Security and its Applications, Vol. 1, No. 2, October, 2007.

[7] D. Liu, P. Neng, W. Du, "GroupBased Key PreDistribution in Wireless Sensor Networks," WiSE'05, Cologne, Germany. September 2, 2005.

[8] M. R. Wahiddin, N. S. Noor Sham, M. Saeb, M. Hamdan, "A Protocol for Secret Key Infusion from Satellite Transmissions," International Journal of Computer and Network Security(IJCNS), Vol.2, No.7, July 2010.

[9] I. Cervesato, C. Meadows, and D. Pavlovic, "An Encapsulated Authentication Logic for Reasoning About Key Distribution Protocol," Eighteenth Computer Security Foundations Workshop, IEEE Computer Society, Press, CSFW-18, pages 48–61, Aix-en-Provence, France, 2005.

[10] H. Sun, H. Yeh, "Password-based authentication and key distribution protocols with perfect forward secrecy," Journal of Computer and System Sciences 72, 1002–1011, 2006.

[11] Liu, Jingbo, Paul Cuff, and Sergio Verdu. "Secret Key Generation with One Communicator and a Zero-Rate One-Shot via Hyper contractivity." *arXiv preprint arXiv: 1504.05526* (2015).

Magdy Saeb received the BSEE, School of Engineering, Cairo University, in 1974, the MSEE, and Ph.D. degrees in Electrical & Computer Engineering, University of California, Irvine, in 1981 and 1985, respectively. He was with Kaiser Aerospace and Electronics, Irvine California, and The Atomic Energy Establishment, Anshas, Egypt. Currently, he is a professor in the Department of Computer Engineering, Arab Academy for Science, Technology & Maritime Transport, Alexandria, Egypt; He was on-leave working as a principal researcher in the Malaysian Institute of Microelectronic Systems (MIMOS). While in MIMOS, he obtained five US/International Patents in Cryptography. He is the CTO, Great Wall Information Security. His current research interests include Cryptography, FPGA Implementations of Cryptography and Steganography Data Security Techniques, Encryption Processors, Mobile Agent Security.

mail@magdysaeb.net

www.magdysaeb.net