CUCS-136-84

A Secure and Useful 'Keyless Cryptosystem'.

Mordechai M. Yung[1]

Department of Computer Science
Columbia University
New York, N.Y. 10027

Abstract

Keyless cryptography is a technique in which the basis of security is the anonymity of the sender. We describe a protocol which fits the technique to realistic communication environments, and extend the security and the range of applications of the technique.

keywords: cryptography, cryptosystems, cryptographic protocols, communication channels, communication security, key distribution, encryption, conferencing, authentication

[1]Research supported in part by NSF grant MCS-8303139

1. Introduction

Alpern and Schneider [1] proposed a new cryptographic technique in which the security lies in hiding the identity of the message's originator. This technique, *keyless cryptography*, is similar to public-key cryptosystems [5] in that it transmits keys using the data channels. This points to its suitability for use in computer networks.

The security is a function of the nature of the computer system and its capability to hide the sender's identity. The application given in [1] is a protocol which allows two users to create a random key and prevent other users or eavesdroppers from getting its value assuming we have an ***anonymous channel***, that is, a channel which perfectly hides the message originator.

We suggest some extensions to the original scheme. In section 2 we summarize the Alpern-Schneider protocol and in section 3 we show how to implement the key distribution in a more realistic environment which is a ***semi-anonymous channel***, a channel which is not totally secure and the message origin is hidden only with a given probability. In section 4 we suggest some extensions to the original scheme which solve problems proposed by its inventors.

2. A Key Generation Protocol Over an Anonymous Channel

The following protocol and its implementations were given in [1].

The Key Generation Protocol:

step 1: User A chooses a secret random bit string $K_a(0),...,K_a(2n-1)$.

User B chooses a secret random bit string $K_b(0),...,K_b(2n-1)$.

step 2: For i:= 0 to 2n-1 do simultaneously:

A transmits 'Concerning K_{ab} my i^{th} bit is: $K_a(i)$';

B transmits 'Concerning K_{ab} my i^{th} bit is: $K_b(i)$';

step 3: For i:= 0 to 2n-1 do:

if $K_a(i)= K_b(i)$ then delete the i-th bit from the string;

By convention the common key K_{ab} is the remaining bits in K_a. The bits are chosen at random. Therefore, on the average there are n bits in K_{ab}. The process can be repeated if a larger key is desired.

end {protocol}

Since any eavesdropper gets the synchronized transmissions in step 2 simultaneously over an anonymous channel, he can not detect which bit is A's and which is B's. This implies the perfect secrecy of K_{ab}. The users change their key from time to time to maintain the communication security.

Three implementations of the protocol were suggested in [1]: a centralized version consists of a central key distribution facility (a central trusted process with a blackboard); and two distributed versions: a broadcast network (e.g. a ring, an Ethernet or a satellite network) and a special communication channel consisting of two wires connecting any couple of users.

3. A Secure Key Generation Over a Semi-Anonymous Channel

In an environment where we have semi-anonymous channels, the origin of some of the bits may be detected (for example by wiretappers who may occasionally notice the difference in signal propagation). Assume that the probability of a bit being detected is ϵ. Then on the average, $n\epsilon$ of the key's bits are not hidden, and as a consequence large portions of the messages are not secure. In order to get a better hidden key we propose the following protocol:

The Secure Key Generation Protocol:

step 1: For i:=1 to r do the **key generation protocol** resulting in a random key K_{ab}^i;

step 2: $K_{ab} = K_{ab}^1 \oplus K_{ab}^2 \oplus \ldots\ldots \oplus K_{ab}^r$; (The $\oplus$ is the 'exclusive-or' operation.)

end {protocol}

We observe that if the j-th bit in any of the r strings ($K_{ab}^i(j)$ i=1,r) is secure, then $K_{ab}(j)$ is secure. As a result the fraction of bits which are not hidden, on the average, is ϵ^r. This means that choosing fairly small r gives a highly secure key (choose r such that the expected number of insecure bits $n\epsilon^r < 1/2$). For example, if $\epsilon = 1/2$ and n=100, choosing r=8 gives a key in which all bits are likely to be secure.

4. Extending the 'Keyless Protocol'

In this section we show how small modifications of the original system can solve some problems presented in the original paper [1] making the keyless cryptography system more secure and more useful.

4.1. A Fixed Key Distribution And Its Applications

In [1] Alpern and Schneider say that while the keyless cryptography system can disguise data during transmission, they have been unable to devise a scheme using keyless cryptography to encrypt data for secure storage. Analysing the problem, it seems that the difficulty arises because the key generated by the protocol is a random one (a function of the random choices of both users), while data is stored using predetermined keys. The following simple observation enables a user A to transfer a secure fixed key K_D to B: First A and B generate a random key K_R, then A transmits $K_D \oplus K_R$. K_D can be any block cipher or DES key (see [6] chapter 6). If A's data files are encrypted by K_D then the key transfer is a way to implement a granting of a read capability to these files (see [2] chapter 4).

Another important property of a key distribution system is its ability to support secure computer conferencing, that is, the distribution of a key to a group of users. For example it was shown in [3] and [4] how to generate keys for all subsets of users from a linear number of keys, and in [9] it was shown that a public key based on the factorization problem (like RSA [8]) can be distributed to a group. In the keyless cryptosystem a user can distribute a fixed group key K_G to a selective group of users by repeating the fixed key distribution protocol independently with each of its members: The user generates together with the i^{th} member a random key K_{R_i} and then sends him $K_G \oplus K_{R_i}$.

4.2. Preventing Active Wiretapper Attacks Using 'Mutual Authentication'

In [1] the authors say that all the keyless cryptography protocols that they have been able to devise appear to be vulnerable to an active wiretapper attack. Only one attack was described with the original protocol: the wiretapper can not delete messages, and all users are active all the time. This attack can be frustrated since when A and B exchange keys, A will detect more than one message transmitted concurrently if C pretends to be B. User C can, however, pretend to be B while B

is temporarily detached from the network (or forced off by C), in which case C would get the key and the secure information that A wants to transmit to B. We notice that since the protocol is performed in synchronized time slots, it should be easy to protect by adopting the authentication tag technique.

If both users share a secure, very long, one time authentication string of bits (S(i),i=0,1,..), when they simultaneously transmit the i^{th} bit of their keys, they also exchange an authentication pair (S(2i),S(2i+1)). This gives both users a ***mutual authentication***, k bit transmission is authenticated with probability $1-(1/2^k)$. This probability indicates perfect authentication (analogous to the perfect secrecy of one time pad). This looks unrealistic since S has to be transmitted over the network first. We can, however, approximate this authentication in the keyless cryptosystem by trading authentication quality for feasibility, and communication bandwidth for transmission of authentication bits. Users can generate authentication bits the same way that key bits are generated. This enables the design of a system which is immune to active wiretapper attacks.

The first short segment of the authentication string is distributed to A and B by a trusted authentication server [7], implemented as the central version of the keyless cryptosystem described in [1] (section 3.1). (Its job is analogous to that of the trusted public directory in a public-key system, since after the initiation the system becomes totally distributed.) We cannot provide authentication for each newly generated bit since each new bit requires 4 authentication bits on the average and the generated string is substantially shorter than the authentication string used. Therefore the users transmit packets of a fixed length, each of which consists of an authentication bit followed by some random bits. Since the first bit of each packet is known a priori only to both users, they achieve mutual authentication, and the probability of a successful attack is 1/2 per packet. Letting the number of packets be large enough makes the probability of a successful attack on the protocol negligible. Whenever A and B generate a new key some of the generated bits do not serve as key bits. Rather, they become the authentication strings for the next key exchange. When the key length is k and p packets are used (for example: k=100, p=20), (2k/p)+4 random bits are included in a packet. 2(k+2p) random bits are exchanged in total, out of which k+2p on the average are secure, k are used as the new key and 2p as new authentication bits. The probability of a

successful attack is $1/2^p$.

5. Conclusions

We identified some problems in the keyless cryptography system. In practical systems (for example a broadcasting network) it is very hard to get an anonymous channel environment. The importance of our protocol is that we can amplify the security of the key in more realistic semi-anonymous channel environments. The problems of the randomness of the system's key and the system's vulnerability to an active wiretapper attack limit its security and applicability as well. Our solutions to these problems make the system more attractive.

Acknowledgement

I would like to thank an anonymous referee for the terms 'anonymous' and 'semi-anonymous' channels.

References

1. Alpern B. and F.B. Schneider. "Key Exchange Using 'Keyless Cryptography'." *Information Processing Letters 16*, 2 (February 1983), 79-81.

2. Denning D.E.. *Cryptography and Data Security.* Addison-Wesley, Reading, Massachusetts, 1982.

3. Denning, D.E. and F.B. Schneider. "Master Keys for Group Sharing." *Information Processing Letters 12*, 1 (January 1981), 23-25.

4. Denning D.E., H. Meijer and F.B. Schneider. "More on Master Keys for Group Sharing." *Information Processing Letters 13*, 3 (December 1981), 125-126.

5. Diffie W., and M.E. Hellman. "New Directions in Cryptography." *IEEE Transactions of Information Theory IT-22*, 7 (November 1976), 644-654.

6. Konheim A.G.. *Cryptography A Primer.* John Wiley & sons, New York, 1981.

7. Needham R.M. and M.D. Schroeder. "Using Encryption for Authentication in Large Networks of Computers." *Communications of the ACM 21*, 12 (December 1978), 993-999.

8. Rivest R. , Shamir A. and Adleman, L. "A Method for Obtaining Digital Signatures and Public Key Cryptosystems." *Communications of the ACM 21*, 2 (February 1978), 120-126.

9. Yung M. M. Cryptoprotocols: Subscription to a Public Key, the Secret Blocking and the Multi Player Mental Poker Game. In *Proceedings of Crypto84 (to appear)*, Springer-Verlag, 1984.