# A Practical Protocol for Large Group Oriented Networks

Yair Frankel
Electrical Engineering and Computer Science Department
University of Wisconsin-Milwaukee
Milwaukee, WI 53201

## 1 Introduction

It is infeasible in large networks for every individual to have his own key. In a group oriented society, public keys are needed for communications between one organization and another. The organization might also want the supervisors to read some of the messages received by the employees. In the case of urgent messages, the organization may want any member to be able to read it [3]. This paper proposes a method in which individuals at separate organizations can communicate without the advance coordination of keys between the individuals. Thus, reducing the number of keys needed to communicate between two organizations. Also, the destination company can create its own policy on who can read messages and the type of public, or conventional, key system used within organization, without any involvement with the sending organization. This system will solve one of the many problems presented in [3].

In this system, an individual sends parts of messages to clerks(devices) who proceed to transmit messages to destination organizations using public keys. The clerks need to know key(s) for the source and the destination organization(s), not the individuals within the organizations. These clerks, when not acting in collusion, cannot recover the message but can only determine the destinations.

At the destination organization, clerks do some calculation to the messages using their private keys and send the result to the destination(s). The individuals at the destinations combine the messages received from their clerks to recover the original message. The clerks at the destination must also act in collusion to recover the original message.

Public key cryptosystems [5] can be used in large communication networks. However as the size of the network increases, the quantity of keys increases to the point where key management becomes a major problem. A system that does not

J.J. Quisquater and J. Vandewalle (Eds.): Advances in Cryptology - EUROCRYPT '89, LNCS 434, pp. 56-61, 1990.

require that a message sender know the key(s) of the intended recipients offers a considerable advantage.

Threshold schemes [1, 8] prevent a (small) number of individuals from acting in collusion to view a message. then the threshold. These schemes are used in this paper to provide protection against collusion on the part of clerks and against communication failures.

Tamper proof devices have been used to show relationships between classes of cryptosystems [4] and to simplify implementations of protocols [2]. In the following, tamperproof devices can be used to replace the clerks, therefore further reducing the collusion problem.

# 2 Basic System

The general concept is that the sender breaks a message into separate parts. Half of the message pieces will be encrypted and go to one of the clerks and the key will go to the *other* clerk. The same will be done with the other half of the message. The clerks will not be able to understand the pieces of the message that they receive since the other clerk will have the key needed to decrypt the message that he/she receives. Each clerk will then transmit the message using the destination company's public key. Since keys are only needed between the companies rather than the individuals in the company, the number of keys is drasticly reduced.

Both of the clerks at the destination organization will receive the message and do some calculations on it. For example, they will both receive an encrypted version of one of the keys sent by the source organization's clerk. They, however, will not be able to know the key after they do the calculation. The destination clerks will both give the result of their calculation to the destination. When the individual(s) receive the message from the clerks, he/she will multiply the two results together to get the key. The rest of the message pieces will be received in the same manner, but the destination will use the key that he/she received earlier to decrypt the message.

Four distinct tasks are necessary for the protocol's operation. These tasks are described in greater detail in the following.

## 2.1 Source

The source partitions the message stream into sequence $M_i$ $i = 0, 1, 2 \ldots$ of non-overlapping packets. The source will then generate two keys $K_1$ and $K_2$ in which $K_1$ will be used to encrypt the odd numbered messages and $K_2$ the even numbered messages using a conventional cryptosystem of required strength.

$$C(M_{2i+1}, K_1) = C_{2i+1}$$

and

$$C(M_{2i}, K_2) = C_{2i} \text{ where } i = 0, 1, 2, \ldots.$$

It will become clear that the keys can be selected as often as needed and the source and eventual destination(s) do not need to choose the keys in advance, only the cryptosystem.

Next $K_1$ and all the even number $C_i$'s are signed and then encrypted using the key of one of the clerks, and transmitted to the clerk. Similarly $K_2$ and the odd number $C_i$'s are signed, encrypted using key of *other* clerk, and then transmitted to the *other* clerk.

The cryptosystem protecting the channel between source and clerk is selected at the discretion of the company's security manager. There is no necessity for coordination between companies in the selection of this system.

## 2.2 Clerks at Source Company

When a source clerk receives a message, he/she decrypts it using his/her key and recovers $K_2$ and $C_{2i+1}$'s, or $K_1$ and the $C_{2i}$'s.

At this point the $K_i$'s are exposed. *Note however that each clerk sees the key of the other clerk's C's, not the key necessary to decrypt the cipher text in his possession.* To compromise this system the clerks must act together or both clerks must be bugged.

Then the clerk authenticates the sender and encrypts the result using the public key of the destination company.

$$\hat{C}_{K_1} = (K_1)^\alpha \mathrm{mod} n, \quad \hat{C}_{M_{2i+1}} = (M_{2i+1})^\alpha \mathrm{mod} n$$

and

$$\hat{C}_{K_2} = (K_2)^\alpha \mathrm{mod} n, \quad \hat{C}_{M_{2i}} = (M_{2i})^\alpha \mathrm{mod} n \text{ where } i = 0, 1, 2, \ldots.$$

Each clerk broadcasts his messages to the destination company.

In this system, the transform $\hat{C}$ is the RSA cryptosystem [7] with modulus $n$. The $\alpha$ and $n$ form the public key of the destination organization and are generated by a trusted key distribution center or the destination organization.

## 2.3 Clerks at Destination Company

Each clerk receives and performs a calculation (described below) on their copy of of every packet received by the organization. It does not matter which clerk sent the message from the sending organization.

To do the calculations, each clerk is given a private key which will not decrypt the packet "completely". That is, the clerks will not be able to read the packet since they did not use the key, but they have done part of the calculations needed to decrypt the message without getting any information about the message or

key. The calculation is the same as if one would decrypt a message using RSA with his private keys.

$$\dot{C}_p^t = (\hat{C}_p)^t \mathrm{mod} n, \quad \dot{C}_p^s = (\hat{C}_p)^s \mathrm{mod} n,$$

where $p$ is a packet received by the destination company.

If the private keys of the clerks (at destination) are $s$ and $t$ then the following relation holds:

$$\alpha t + \alpha s \equiv 1 \bmod \phi(n).$$

Barring collusion between the destination clerks, not even $K_1$ or $K_2$ are exposed. Each clerk transmits the packet he did calculations on (i.e. $\dot{C}_p^t$), encrypts, signs and transmit it to the destination. If the organizations policy states that certain individual must read the message also, the clerk will also send message to them. If the clerks want to collude, they must transmit both parts to an unauthorized individual or the other clerk.

The public key system protecting the channel between the clerks and the destination is selected at the discretion of the destination company's security manager. There is no necessity of cooperation between companies in the selection of this system.

## 2.4 Destination

The destination decrypts the incoming packets ($\dot{C}_p^s$, $\dot{C}_p^t$) which are the result of the computation done by destination clerks, and multiplies the packets together to recover the $K_1$, $K_2$ and the $C_i$'s. This is possible since,

$$\begin{aligned} \dot{C}_p^t * \dot{C}_p^s &= (\hat{C}_p)^{\alpha s} * (\hat{C}_p)^{\alpha t} \mathrm{mod} n \\ &= (\hat{C}_p)^{\alpha t + \alpha s} \mathrm{mod} n \\ &= (\hat{C}_p)^1 = M_p \mathrm{or} K_p. \end{aligned}$$

The destination then can easily recover the $M_i$'s using $K_1$ and $K_2$.

# 3 Proofs of Equivalence to Existing Cryptosystems

It is easy to show that this cryptosystem is equivalent to RSA. The approach is similar to the method of Kranakis [6], but is slightly different. However, our system is more secure than [6]. Since if there exists public keys $e_i, e_j$ in the Krankis method such that $\gcd(e_i, e_j) = 1$, it is trivialy breakable.

Since $t + s = \alpha^{-1} \bmod \phi(n)$ and $t$ is chosen randomly, finding $s$ is equivalent to calculating $\alpha^{-1}$.

# 4 Extending the Basic System

The total dependence of the trustworthyness of the source clerks is a weak point of the basic system. To remove this deficiency and to increase the reliability, multiple source clerks can be employed. The cryptosystem $C$ above is expanded to require $M$ keys, and the number of source clerks is increased to $N$. Each clerk has sufficient numbers of keys and the copies of the keys are distributed in such a fashion that any set of $K$ clerks have all $N$ keys.

$$C(M_i, K_j) = C_i \text{ where } i = 0, 1, 2, \ldots \text{ and } j = 0, 1, 2, \ldots, M.$$

Since the above key distribution will operate in the face of $N$ minus $K$ inoperative clerks, the partitioned message stream can distributed to the clerks such that every message block is sent to exactly $N - K + 1$ clerks. If the $M$ keys are randomly generated and are exclusive or'd to form a single key as part of the encipherment $C$ the the knowledge of any $M$ minus 1 keys given an enemy no knowledge of the actual key. This method becomes impractical due to message expansion.

In the scheme presented in section 2, collusion on the part of the two destination clerks, in the basic system results in compromising the message. To prevent this more clerks can be employed but as the number grows the reliability of the system degrades. Each destination clerk $j$ posses $k$ keys such that

$$\alpha \sum_{i=1}^{k} x_{j,i} \equiv 1 \bmod \phi(n) \quad \text{where} j = 1, \ldots, n.$$

No clerk possess more than one $x_{j,i}$ from the same congruence. When a clerk receives a message he generates $k$ messages by raising the message to the power of each key mod $n$. The destination can choose any $k$ messages that resulted from the same message (that the destination clerks *received*) provided that the keys are from the same congruence.

Tamper proof devices can be used to replace the clerks by performing all their duties. These devices must have at their disposal an authenticated routing table and public/private key database.

# 5 Conclusion

This protocol is a method in which one can use RSA in a large network. Since our society is organized into groups, this system is not only practical but represents a robust method since the techniques allow a single sender or receiver to communicate with a company rather than require companies to commincate only.

# 6 Acknowledgements

The author wishes to extend his thanks to Professor Yvo Desmedt for his suggestions and references. The author wishes to thank Brian Matt for his many helpful discussions and suggestions.

# REFERENCES

[1] G. R. Blakley. Safeguarding cryptographic keys. In *Proc. Nat. Computer Conf. AFIPS Conf. Proc.*, pages 313–317, 1979. vol.48.

[2] G.I. Davida and B.J. Matt. Arbitration in tamper proof systems. In C. Pomerance, editor, *Advances in Cryptology, Proc. of Crypto'87 (Lecture Notes in Computer Science 293)*, pages 216–223. Springer–Verlag, 1988. Santa Barbara, California, U.S.A., August 16–20.

[3] Y. Desmedt. Society and group oriented cryptography : a new concept. In C. Pomerance, editor, *Advances in Cryptology, Proc. of Crypto'87 (Lecture Notes in Computer Science 293)*, pages 120–127. Springer–Verlag, 1988. Santa Barbara, California, U.S.A., August 16–20.

[4] Y. Desmedt and J.-J. Quisquater. Public key systems based on the difficulty of tampering (Is there a difference between DES and RSA?). In A. Odlyzko, editor, *Advances in Cryptology, Proc. of Crypto'86 (Lecture Notes in Computer Science 263)*, pages 111–117. Springer–Verlag, 1987. Santa Barbara, California, U.S.A., August 11–15.

[5] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, IT–22(6):644–654, November 1976.

[6] E. Kranakis. A class of cryptosystems equivalent to RSA. *Dept. of Computer Science, Yale Univ. tech report 315*, April 1984.

[7] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. *Commun. ACM*, 21:294 – 299, April 1978.

[8] A. Shamir. How to share a secret. *Commun. ACM*, 22:612 – 613, November 1979.