SOCIETY AND GROUP ORIENTED CRYPTOGRAPHY: A NEW CONCEPT

Yvo Desmedt
Dépt. I.R.O., Université de Montréal
Montréal (Québec), H3C 3J7 Canada

1 Introduction

Messages are frequently addressed to a group of people, *e.g.*, board of directors. Conventional and public key systems (in the sense of Diffie and Hellman [4]) are not adapted when messages are *intended for a group* instead of for an individual. To deeply understand the lack of usefulness of the above cryptosystems in the case that messages are intended for (or are originating from) a group of people, let us now nevertheless attempt to use these systems. When conventional and public key systems are used to protect privacy, the legitimate receiver(s) has (have) to know the secret key to decrypt. This means that, a first solution could be, to send the message to all members of the group, *e.g.*, using their public keys. A second is that the secret key is known to all members and that the message is sent only once. All other solutions using a conventional or public key system, are combinations of the above two solutions. We now explain briefly why these two obvious solutions are not adapted to security needs specific to the protection of information intended for groups.

Both solutions are not adequate when members of the group *have* to share *the same responsibility*. To illustrate this we focus on companies who develop software. It is well known, that sometimes supervisors, with *crucial information* about large software developing projects, leave a company to start their own one. If the group, who is developing the software product, had only one supervisor, then frequently the group has to restart at zero, as a consequence of the lack of the crucial information. To avoid this problem, modern software developing companies use two supervisors, such that projects can continue when one of supervisors leaves [6]. One has however to be certain that both supervisors have the same knowledge about the project. In such case the first solution means sending the messages to both supervisors such that the messages are encrypted with their own different public keys. This solution is unsecure when one of them is able to intercept the message for the other. In other words this solution does not guarantee that both supervisors *have to share* the *same* responsibilities. A possible solution could be to use anti-jamming techniques, however this does not exclude that the other supervisors destroys the messages once they arrived. The second solution does not satisfy because the one who reads the message first, can then destroy it to prevent that the other learns it. So this solution does again not guarantee that responsibilities have to be shared.

The purpose of this paper is to solve the above problem and some other similar problems. Before doing this we will wonder what a group of people is and classify these groups (see Section 2). In the same section we will then wonder what kind of protection needs exist for messages intended for groups, or for the messages originating from groups. Each of the protection needs corresponds

C. Pomerance (Ed.): Advances in Cryptology - CRYPTO '87, LNCS 293, pp. 120-127, 1988.

with a new open problem. In Section 3 we will describe and extend the cryptographic notion of trust. These new notions of trust will be related with the needs for protection of messages intended for groups. In Section 4 we solve some of the new open problems, others remain unsolved. Before attempting to solve these new open problems we restrict ourselves to avoid impractical solutions as much as possible. We however do not pretend at all that the solutions discussed are very practical.

We finally note that the problems which are discussed in this paper, arose by the author from the lecture of Yao [7]. The relation of our paper with his work [7], will be explained in Section 4.

2 Different kind of groups and their cryptographic needs

In this section we discuss the needs. This does not necessarily means that we know how to implement a system which satisfies the discussed needs. So we are discussing the ideal situation.

We mainly discuss (in this section) the needs of groups related to the security of messages intended for the group. In Section 3 we discuss the needs of groups related to messages which originates from the group.

2.1 Groups in our society

Groups play a crucial role in our modern world. Examples of such groups are: the fire department of a city, the board of directors of a company, the Senate and the Congress of the U.S.A. and similar organizations in democratic and non-democratic countries. Each counsel is also a group of people. Other examples can be found in banks having several services, *e.g.*, the service new accounts, the service loans and the service change. In most companies similar structures exist, a well-known example is the financial department, who pays the salaries.

Remark that all the above groups of people have one common aspect: their existence is more known than the name of the people who are members. Remark that the functional aspect of a group is (or must be) independent of its members. Therefor we will call such a group of people *a group with anonymous membership*. The word anonymous here has not to be taken strictly. Indeed mostly it is no secret who is a member of the group, but it is not necessary to know it. Letters addressed to the group mostly start with the well-known expressions: "Dear Sirs" and the fact that the letters are treated is (has to be) independent from the members. To be more general and extend the notion of groups with anonymous membership we remark that all officials, who once have had the same duty in one office, form a group of people. This is even true if the present incumbent is the only one still living. All these people form a group in time and the functional aspects of the official is (for the greater part) independent from who is in charge. For this reason most kings start their official documents with: "*We* king of ...". To make a distinction between a group of people consisting with more than one member at the same moment and the other ones, we will refer to the first as *groups with anonymous members*, and we call the second group *officials*.

Not all groups in our society have anonymous membership. Sometimes a group of people designates a group of well-known members, *e.g.*, The Beatles. We will refer to these as *groups with known members*.

Before that we discuss what the cryptographic needs are of these groups, we wonder how groups access information intended for them.

2.2 Access of information by groups

If an encrypted message is intended for a group of people, then there exist several methods to access this information. It is evident that not all these methods satisfy the needs of security. To analyze these needs we assume that some members of the group are competitors and that they try to access the information inappropriately or try to hide some information from their colleagues.

Influenced by the example of two supervisors (see Section 1), one could argue that the best method of access to information by a member of the group, for who the message is intended, is that access is only possible if all members agree to open it (decipher it) at exactly the same moment. To demonstrate that this is possible, we assume for the moment that the message is put in a safe with a secret combination of 128 bits. Each of the supervisors knows only 64 bits of the 128. To open the safe both have to agree to meet each other at the safe and have then to bring in their part of the combination to open it and to read the message. We remark briefly that banks use frequently this precaution with safes. A non-mechanical solution to this problem will be discussed later (see Section 4).

The above solution is however sometimes unsecure. A trivial but very important counter-example is the urgent message which is sent to the fire department: "Fire at location ...". Here each member of the fire department must be able to read the message, even if the other members of the fire department are out. Another example is when a fire is destroying the computer of the software developing group. In that case the message is so urgent, that it does not matter that the other supervisor, who has just a day off, does not read it! *So groups can have different needs at different moments*!

Another need is a consequence of the fact that not all members have the same position in the group. A group may have one (or two) supervisors (or presidents). Messages can be intended for the whole group, but only after that the supervisor has read the message. It is normal that the president of the board of directors is informed first that his company has just received an important contract. It would be strange that all other members know it before the president.

To implement the last need some additional memory is necessary, to remember that the president has read the message.

It is evident that after having discussed the three above examples (message is readable if *all* members agree to decipher at the same moment; message is readable by *each* member separately; *if* the president has read the message, *then* it is readable by all members of the group), many other possibilities exist for access by the members of the group. This is trivial to understand if one takes into consideration that a group of people may correspond with a hierarchical society having one president and two vice-presidents. The best method for access could be that messages are first read by the president, then by the vice-presidents and then by the members.

It is clear from the example of the fire in the software developing company, that methods of access can vary from moment to moment. To study this, we now wonder who decides how the information may be accessed.

2.3 Several methods of access: who decides?

Before discussing who decides how the group will access information, intended for them, let us briefly discuss the aspect of keys which will be used when an outsider sends an encrypted message to the group.

In the case that the group has anonymous members, it is evident that it is impossible to use the public keys of all members. So the group will have a common public key, which we call the *group public key* (remark that we do not yet discuss the deciphering).

Several methods can be used to decide how and in which order that the information can be accessed. The first idea is that the group decides it and publishes a group public key corresponding with their needs. The second solution is that a group publishes different group public keys, *e.g.*, one for normal use and one for emergency. In the case of the supervisors, this means that both have to access at the same moment the messages which are sent using the group public key intended for normal use. When the group public key intended for emergency is used, then both can access the information immediately. Let us now discuss the last solution.

When several different access methods have to be used, it would be impractical to publish as many group public keys as there are *possible* access methods. Indeed there are exponentially many methods of access when the group is large. So if the group decides that access methods will differ frequently, then the ideal situation would be that the sender of the information can add a few bits to the message such that these few bits enable the group to access the information *using the method he has decided and no other one*! This means that the group has only to publish one public key and that it can be used in cases that all members have to access the information simultaneously as well as in case of emergency and all other possible cases (depending of what the sender decides). *We remember that we are speaking in this section about needs, what does not necessarily indicate that we know how to build such a system.*

One need related to group encryption needs still to be discussed.

2.4 Who knows about the decision

We have just discussed who decides what the protection need for the messages is (this means how and in which order that they will be accessed). It is clear that those who have decided, know about their decision. But do others know it too? To illustrate the problem let us fix an example. Suppose that the group decided that the messages has first to be accessed by the supervisor and then afterwards by all other members at the same moment. The group publishes the corresponding public key. The question now is: *does the publication of this key and of the encryption method, reveal what the decision of the group is*? If that would be the case, then everybody knows which hierarchy the group has, or more generally knows which kind of society that corresponds with the group.

The last problem is clearly strongly related to the implementation. Does the implementation reveal how the group will access the information, or does there exist a solution which keeps this information secret?

The author is not able to answer this last question for the moment.

2.5 Members leaving

A major problem which was not yet discussed is the problem of members who are leaving the group. Indeed it would be completely impractical that the group public key has to be modified each time that a member leaves the group, or when a new member joins the group! This is certainly true if the group is large.

So an important need is that the group public key has to be modified only if (*e.g.*) the majority of the initial group members is modified.

3 What is trust

All actual cryptosystems rely at the last end on trust. Indeed if you use a key to send information to somebody (using a conventional or public key system) you have to trust the authenticity of the key you use. Even if you use a secret key which you have agreed with your correspondent when you last met him, you rely on trust. Indeed who guarantees you that the person with who you have spoken is not an impersonator!

Problems as the authenticity of the file of the public keys are frequently solved in cryptography by using a so called *trusted party*, *e.g.*, a notary public. However in our society there exist other forms of trust, *e.g.*, two witnesses. Frequently it is better to use two trusted parties instead of one. But what is the optimal form of trust? Again this depend from application to application. In a bank the clerk can handle small transactions on his own. But in most banks the clerk needs a approval from his supervisor for large transactions. The supervisor in a small branch of the bank needs also an approval for wholesale transactions. This means that for banks the ideal situation would be that clerks can use the cryptosystem of the bank on their own for signing (or confirming) small transaction. But that the same cryptosystem refuses to accept a signature for a large transaction, except when countersigned by the supervisor. This means that for outsiders the bank would have only *one* public key, and everybody can verify a signature made by the bank. However making such a signature can depend from the application.

It is clear that there are as many needs as there are forms of society (hierarchical, partially hierarchical, and so on). So the same remarks can be made as above. For example is it necessary that a customer of the bank knows at what level the difference is between small and medium transaction? Or can cryptosystems be made which keep this secret.

4 Solutions

4.1 Introduction

Proposing a cryptosystem which satisfy all these requirements would probably be a revolution in the cryptographic society. Indeed many applications would exist for cryptosystems which satisfy all the above needs. One of the examples we have already discussed was the signature by bank clerks.

But even if such a cryptosystem would be found, this does not guarantee that it is practical. So the author restricted himself to systems which forbid that a ping-pong protocol is used between

sender and receiver. Indeed such systems are in many applications impractical due to the delay of the communication. Indeed nevertheless the fact that electronic mail, used between the U.S.A. and Europe, is mostly faster than normal mail, it is impossible to apply ping-pong protocols. Therefor we have restricted the use of such ping-pong protocol. So we *exclude* the use of a ping-pong protocol *between sender and receivers*, but allow that the receivers can use it among themselves. This last concession does mostly not affect practical implementations if the members of the group are on some Local Area Network.

4.2 Solution based on the difficulty of tampering

It is not difficult to make cryptosystems which satisfy *most* of the above needs if one uses systems based on the difficulty of tampering [3]. The reader can easily figures this out by reading [3]. Indeed these systems simplify many implementations (see [2]).

If one has more faith in cryptosystems which security is mainly only based on mathematics, then the discussed approach has to be rejected.

4.3 Solution based on complexity theory

In this case it is no longer true that the author knows solutions for most needs.

Before we discus some solutions, let us first discuss the context in which the ideas have originated. Yao [7] has recently claimed he has a method such that two people (Alice and Bob) can make a number n which is the product of exactly two primes and such that neither Alice nor Bob know what these primes are, except when they both want to recover at the same moment what these primes are. The author has found the following similar problem. The above n would correspond with the public key of the two people and it would be impossible for both to read, without the collaboration of the other, messages encrypted with that key. In order to do this both would have to collaborate such that after the interaction they both know the message, but not the secret key. This last condition is a consequence of the exclusion of a ping-pong protocol between sender and receivers. Before tackling the problem, the author worked on generalizing the needs of encryption of messages intended for (or originating from) groups. Meanwhile the author has realized that one of the problems can easily be solved by using the well-known Blum protocol [1] and another problem by using [5]. We now briefly discuss these two problems and their solutions.

4.3.1 In the case of known members

Suppose that the members of the group are known, *e.g.*, Alice and Bob, and that an outsider *e.g.*, Brigitte, wants to send a message such that Alice and Bob are unable to read it except when they collaborate and such that *no ping-pong* is allowed between the sender and the receivers.

It is easy to solve the above problem by using [1]. We assume that the discrete log problem and that the factoring problem are difficult and that the Blum protocol is secure. In our solution Brigitte encrypts M such that the ciphertext $C \equiv M^x \pmod{p}$, where p is a standard prime. x corresponds with x_1 concatenated with x_2, where x_1 and x_2 are (save) primes. Brigitte enciphers x_1 and x_2 by using respectively the public key of Alice and Bob. So she obtains $E_A(x_1)$ and $E_B(x_2)$, where the index A and B correspond with the public key of Alice and Bob. Then Brigitte

signs the complete message consisting in $(M^x, E_A(x_1), E_B(x_2), x_1 * a_1, x_2 * a_2)$, where a_1 and a_2 are random (safe) primes chosen by Brigitte. This signature operation is extremely important to avoid that Bob modifies the $E_A(x_1)$ (and similar related to Alice and $E_B(x_2)$). Brigitte sends the above signed message to Alice and Bob. Neither Alice nor Bob can decipher the ciphertext, but can recover respectively x_1 and x_2. Alice and Bob then use the Blum protocol (or an improved version) to discover the other x_i. Then both Alice and Bob can calculate $x^{-1} \pmod{p-1}$.

It isn't difficult to make variants of the solution. Such variants can have theoretical advantages, certainly that one could argue that the above x has a special form and could help the cryptanalyst. To avoid this x could be divided into its bits. Each of these bits could correspond with one bit of a different prime. These primes are then multiplied with a_i.

4.3.2 Anonymous members

Let us describe the specifications of the necessary cryptosystem. An outsider is sending a message to the group, using the group public key. The group decided that the only possible access to the message is that *if all members of the group are honest*, then access is only possible if all members are willing to read simultaneously. The solution has to take into consideration: that members are leaving and new ones are coming, that if one member dies (or leaves), then the group must be able to continue with the successor (who was unknown before). To realize this last condition the solution may allow that if more than 50% of the members (or ex-members) of the group are becoming dishonest, then this subgroup can do a coup to take over the power of the other members.

The solution is nothing else than an application of [5]. Assume that the group contains always m members (not necessarily the same). First of all the members assume that only u members will become dishonest (where $u/m < 1/2$), while they are a member of the group. All members agree on the public key (or better probabilistic public key) system that will be used. Each member of the group (*e.g.*, member i) chooses a random secret value x_i and large enough. The members agree how they will make the group public key starting from the x_i. In the case of the RSA the public key corresponds with the so called e and n. So for the last case they agree on some Boolean functions, which they will use to go from the x_i to the public key. In order to do so and keeping the x_i secret, they use [5]. The members of the group also discussed how to use these x_i to decipher each message. This means, in the case of the RSA, that they have described how the x_i will be used to perform the operation $M^d \pmod{n}$, where d and n are in fact functions of the x_i. *It is extremely important to remark here that the secret key itself will never be calculated by the group.* So, this means that each time that an outsider has sent a message, the group has to use their secret x_i to finally find the message, certainly not the secret key! So each time that a message arrives, the group applies [5]. So remark that the protocol of [5] is used $k+1$ times, where k is the number of messages received and where 1 corresponds with the use of the protocol for the calculation of the group public key.

When a member leaves the group, one asks him to reveal his secret to his successor. The successor tests then his x_i. This test can be set-up by encrypting a random message and by testing that the group can decrypt it with the value know to the successor, or a zero-knowledge test can be done. In fact in their idea [5] each x_i was encrypted, so that a correct x_i can be

checked. If the leaving member refuses to give his old x_i then the other members of the group can use a protocol of [5] to recover the x_i and give it to the new member (they could also do this in a way that nobody else of the members will know what the x_i is). Suppose that there now exist y (ex-members), such that $(y+u)/m$ approaches $1/2$, then it is time for the group to make a new public key and announce it.

The same ideas can be used when the group is willing to make signatures, which have to be signed by at least 50% of the members to be valid. (To be correct the word members has to be replaced by: members or ex-members. If we assume again that only u members will become dishonnest a similar approach as just explained is possible).

5 Conclusion

The needs of encryption systems intended for groups has been analyzed. Solutions were adapted from the literature to solve some of the needs. Practical concerns as limitation of ping-pong were taken into consideration. Nevertheless, these solutions are still theoretical, but show that solutions exist for some of the above needs. However it would probably be worse if a ping-pong protocol would have been used between sender and receiver.

The needs of groups are certainly worth to be further analyzed and cryptosystems worked out, which satisfies these needs.

Acknowledgement

The author wants to thank Claude Crepeau (MIT) for having informed the author of the existence of [5] and for the fruitful discussions related to Section 4.3.2 of this paper and [5].

References

[1] M. Blum. How to exhange (secret) keys. *ACM Trans. on Computer Systems*, 1(2):175–193, May 1983.

[2] G. Davida and B. Matt. Arbitration in tamper proof systems. Presented at the same conference (Crypto'87).

[3] Y. Desmedt and J.-J. Quisquater. Public key systems based on the difficulty of tampering (Is there a difference between DES and RSA?). Presented at CRYPTO'86, Santa Barbara, California, U. S. A., August 11–15, 1986, extended abstract will appear in Advances in Cryptology, Proc. of Crypto'86, Lecture Notes in Computer Science, Springer–Verlag, 1987.

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

[5] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proceedings of the Nineteenth ACM Symp. Theory of Computing, STOC*, pages 218 – 229, May 25–27, 1987.

[6] G. M. Schneider and S. C. Bruell. *Advanced programming and problem solving with Pascal.* Wiley, N.Y., second edition, 1987.

[7] A. C. Yao. How to generate and exchange secrets. In *The Computer Society of IEEE, 27th Annual Symp. on Foundations of Computer Science (FOCS)*, pages 162–167, IEEE Computer Society Press, 1986. Toronto, Ontario, Canada, October 27–29, 1986.