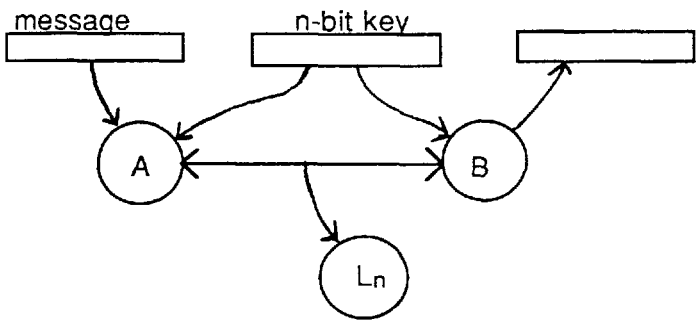
A Basic Theory of Public and Private Cryptosystems

by
Charles Rackoff
Dept. of Computer Science
University of Toronto

Not since the early work of [DH], [RSA], and [GM] has there been a great deal of work on the basic definition of "normal" cryptography, and on what it means for a cryptosystem to be secure. By normal cryptogaphy, I mean not protocols to accomplish sophisticated goals, but merely the situation where party *A* wishes to send a message to party *B* over a line which is being tapped. Existing definitions of such a system, when they aren't too vague, are overly restrictive; existing definitions of security of such systems, when given rigorously, are usually overly liberal. In this paper I'll present what seem to me to be the proper definitions, give statements of the basic theorems I know about these definitions, and raise some very fundamental open questions. Most of the definitions and results appeared in [R].

A cryptosystem looks like the following picture.



A and *B* are probabilistic interacting Turing Machines. Normally, if there is a private key, one only allows *A* to talk to *B*, but I allow *A* and *B* to talk back and forth. *n* is called the *security parameter*. The intuition behind the (plain-text) message is that it should consist of *all* bits to be sent by the cryptosystem until the universe dies; usually this is taken to be some fixed polynomial in *n*, but I find it more pleasing to let it be infinite. *B* must output its guess at the i^{th} bit of the message in time

S. Goldwasser (Ed.): Advances in Cryptology - CRYPTO '88, LNCS 403, pp. 249-255, 1990.

polynomial in n and i, including the computing time of A. We also need some kind of "on-line" condition; a natural (although not completely necessary) one is that A doesn't read bit i+1 of the message until B has output its guess at bit i. (For convenience, assume that the time at which A reads bit i of the message just depends on i and n.) A and B also read an n -bit key. The two sensitive issues are correctness and security. My definition of correctness allows a small probability of error for B. Although most definitions don't allow for this possibility, many suggested cryptosystems have in fact had this feature (since "prime" numbers that were used might not really be prime). My definition of security is intended to encompass all reasonable attacks by an eavesdropper, including chosen plain text attack. Most definitions given in the past (including those discussed in [MRS]), have the property that a secure system can be modified to be secure according to the other definition, but trivially breakable using a chosen plain text attack. The reader will note that neither the definition of correctness nor of security assumes any distribution on the message space. We will always assume (unless stated otherwise) that a cryptosystem is correct.

Correctness: For every $c,d \in \mathbf{N}$, for every sufficiently large $n \in \mathbf{N}$, for every $\alpha \in \{0,1\}^*$ of length n^c, if A tries to send a message beginning with α and the key is randomly chosen, (and the random bits of A and B are randomly chosen,) then the probability that B outputs α is $>1-(1/n^d)$.

Security: Let $L=\{L_1,L_2,...\}$ be a family of polynomial size "circuits". Actually, what L_n can do is as follows: it sees the communication between A and B up until the time A reads message bit 0; L_n then fixes message bit 0; L_n then sees the communication between A and B up until the time A reads message bit 1; L_n then fixes message bit 1; this continues up until some message bit i (i determined by L_n), is chosen randomly from {0,1} (but not seen by L_n); L_n then sees the communication up until the time A reads message bit i+1; L_n then fixes message bit i+1; this continues until L_n chooses to output its guess at message bit i. Let p_n be the probability that L_n is successful at guessing this bit. Then for every d and sufficiently large n, $p_n<(1/2)+(1/n^d)$.

Given a definition of security, it is easy to prove many of the facts normally assumed in the folklore. Theorems 1 and 2 below are examples.

Another example is that a secure (in the sense of [GGM]) pseudo-random number generator implies the existence of a secure cryptosystem. Such a theorem, however, is only as meaningful as the definition of security is good.

In the definition of security, the listener is modeled as nonuniform "circuits" rather than as probabilistic algorithms. This isn't very important, but it makes theorems easier to prove and various things become cleaner. For example, with nonuniform circuits, it is not necessary to add probabilism to L since this would not affect the power of the listener. Certain other aspects of the definitions are there for cleanness and convenience. For example, if we had a system which was secure but only 3/4 correct (instead of $1-(1/n^d)$), the "majority" trick could be used to make it correct with probability exponentially close to 1, and still secure. If we had a system which was correct, but only secure if we replaced $(1/2)+(1/n^d)$ by 3/4, then the "exclusive-or" trick could be used to convert it to one which is secure (although not, as far as we know, "exponentially close to 1/2" secure), and still correct.

If the key, instead of being chosen randomly, is chosen to be 0^n, then I call the system a *public* cryptosystem. Presumably, when people talk about a secure "key exchange protocol", what they mean is a public cryptosystem which is secure for sending (say) n message bits; these bits can be sent as a single block (rather than one bit at a time), possibly speeding things up by a factor of n, but the question of the existence of such a protocol appears to be equivalent to the question of the existence of a secure public cryptosystem. Theorem 1 shows that the open question about the existence of secure public cryptography can be formulated as a question of sending only 1 message bit securely. If A only sends to B, the system is called "1-pass"; if B sends to A and then A sends to B, the system is called "2-pass"; "j-passes" is defined in the obvious way. A 2-pass public system is what is often called a "public key cryptosystem", where the string sent by B is called the "public key". Theorem 2 is part of the basis of "public key cryptography".

Theorem 1: Let (A,B) be a public cryptosystem in which the first message bit is sent securely. Then a secure public cryptosystem (for all the bits) can be obtained by *independently* running (A,B) on each of the message bits (that is, each time, A and B start over and choose new random bits).

Theorem 2: Let (A,B) be a 2-pass public cryptosystem in which the first message bit is sent securely. Then a secure (for all the bits) 2-pass public cryptosystem can be obtained by running B once to see the string ß that B would send; then run A independently on each of the message bits, where each time A uses new random bits, but the same string ß fom B.

Theorem 3 is the analogue of Theorem 1 for (private) cryptosystems.

Theorem 3: Let (A,B) be a (private) cryptosystem in which the first $n+1$ message bits are sent securely, where n is the security parameter. Then a secure (for all the bits) cryptosystem can be obtained as follows: say that α_0 is the key and that the message is $b_0b_1b_2\ldots$; A generates random n -bit strings $\alpha_1,\alpha_2,\ldots$; (A,B) is run with key α_0 on the $n+1$-bit message $\alpha_1 b_0$, then (A,B) is run with key α_1 on the $n+1$ bit message $\alpha_2 b_1$, etc. Note that if (A,B) is 1-pass, then so is the new cryptosystem. However, the new cryptosystem will be probabilistic, even if A and B are deterministic.

Open Questions: Can it help to have more than 1 pass in a (private) cryptosystem? Can it help to have more than 2 passes in a public cryptosystem? Can it help to have more than 3 passes in a public cryptosystem? For each question, either prove a negative answer, or give a convincing example where the extra passes appear to help.

It is interesting to note that there are settings, other than those discussed here, where one can either prove or give good evidence that extra interaction helps. An interesting example, of relevance to cryptography, appears in [BBR].

Theorem 5 below shows that at the moment, our ability to prove security of cryptosystems is severely limited. It is known that with a one-time pad, one can sent n message bits securely with an n-bit random key. If P=NP, which we are unable to disprove, then this is essentially the best we can do. Theorem 5 can be proven by observing that in the proof of Theorem 4, all the listener had to be able to do was "approximate counting"; this task is in the polynomial time hierarchy by a result of [St] and [Si]. Theorem 5, at least in the case $g(n)=0$, has also been observed by other people. A version of theorem 4 was first prove by Shannon [Sh].

Theorem 4: If we remove the polynomial time restriction on the listener, then there is no secure cryptosystem. In fact, a stronger result can be proved. Let g be a function such that $g(n)$ is computable in time polynomial in n, and $0 \leq g(n) \leq n$. Then there is no cryptosystem which sends $g(n)+1$ bits securely (against an unrestricted time listener) if only the first $g(n)$ bits of the key are chosen randomly (and the rest are fixed, say, to be 0).

Theorem 5: If P=NP, then for g as in Theorem 4, even if the listener is restricted to polynomial time, there is no cryptosystem which sends $g(n)+1$ securely if only the first $g(n)$ bits of the key are chosen randomly.

Probably the most important open issue in *all* of cryptography concerns the conjectures can be used to prove the existence of secure cryptosystems. The asumption P≠NP is certainly necessary, but probably not sufficient. The only natural assumptions currently in use relate to the difficulty of integer factorization or discrete log. One possible thing to search for is a "complete" cryptosystem $\mathfrak{R}$: one whose insecurity would imply the insecurity of every other system $\mathfrak{R}'$. Using ideas of Levin, such a system can be constructed by a kind of diagonalization. Such a "complete" system can also be constructed for the class of 1-pass cryptosystems, for the class of public cryptosystems, and for the class of 2-pass public cryptosystems. I will not define this notion of "complete" precisely, since in any case, it has the following problem: the time to break $\mathfrak{R}'$, given an oracle for breaking $\mathfrak{R}$, requires time only polynomial in n, but *exponential* in the size of the description of $\mathfrak{R}'$.

Open Question: Is there a cryptosystem whose security problem is "complete" in an appropriate sense?

Lastly, I'd like to point out what I have *not* talked about here. I haven't discussed the scenerio where there is a group of mutually distrusting people, each pair of which wishes to communicate in the presence of a listener. Although many soulutions to this (and more complicated) problems have been proposed, I have seen no rigorous definition of this scenerio, let alone any definition of what security would mean in such a setting. Of course, there are related subproblems which have been rigorously studied: examples are signature schemes ([GMR]) and the problems studied in this paper. But it appears to be very difficult to

talk about the more complicated situation, and it can be very dangerous to think that security can necessarily be understood in terms of security in simpler situations. For example [GMT] point out that if a secure 2-pass public cryptosystem is used in the obvious way to create a "public key network of users", the result might wind up being insecure.

The difficulties involved in understanding the relatively simple situation discussed in this paper imply that one must approach the more complicated (and realistic) situations very slowly and with a great deal of care.

References

[BBR] C. H. Bennett, G. Brassard, J. Robert, "Privacy amplification by public discussion", *SIAM J. on Comput.,* 17, 1988, 210-229.

[DH] W. Diffie, M. E. Hellman, "New directions in cryptography", *IEEE Trans. Informat. Theory,* IT-22, 1976, 644-654.

[GGM] O. Goldreich, S. Goldwasser, S. Micali, "How to construct random functions", *JACM*, 33, 1986, 792-807.

[GM] S. Goldwasser, S. Micali, "Probabilistic encryption", *J. Comput. System Sci.*, 28, 1984, 270-299.

[GMR] S. Goldwasser, S. Micali, R. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks", *SIAM J. on Comput.,* 17, 1988, 281-308.

[GMT] S. Goldwasser, S. Micali, P. Tong, "Why and how to establish a private code on a public network", *Proc. 23 IEEE Symp. on Foundations of Computer Science*, 1982, 134-144.

[MRS] S. Micali, C. Rackoff, B Sloan, "The notion of security for probabilistic cryptosystems", *SIAM J. on Comput.*, 17, 1988, 412-426.

[R] C. Rackoff, Class notes on Cryptography, 1985.

[RSA] R. Rivest, A. Shamir, L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", *Comm. ACM*, 21, 1978, 120-126.

[Sh] C. E. Shannon, "Communication theory of secrecy systems", *Bell Syst. Tech. J.*, vol.28, 1949, 656-715.

[Si] M. Sipser, "A complexity theoretic approach to randomness", *Proc. 15 ACM Symp. on Theory of Computing*, 1983, 330-335.

[St] L. Stockmeyer, "On approximation algorithms for #P", *SIAM J. on Comput.*, 14, 1985.