Towards Practical Public Key Systems Secure Against Chosen Ciphertext attacks

Ivan Damgård
University of Aarhus
Matematisk Institut, Ny Munkegade
DK 8000 Århus C, Denmark
ivan@daimi.aau.dk

Abstract

We present two efficient constructions aimed at making public key systems secure against chosen ciphertext attacks. The first one applies to any deterministic public key system and modifies it into a system that is provably as hard to break under a passive attack as the original one, but has the potential of making a chosen ciphertext attack useless to an enemy. The second construction applies to the El Gamal/Diffie-Hellman public key system. Again, the modified system is provably as hard to break under a passive attack as the original one, and under an additional cryptographic assumption, a chosen ciphertext attack is provably useless to an enemy. We also point out a connection between such public-key systems and efficient identification schemes.

1 Introduction

The question of whether public key encryption schemes can be secure against chosen ciphertext attacks has received at lot of attention in the last 12 years. The problem first came up when Rabin presented his variant of RSA in 1978 based on modular squaring [Ra]. He proved that decrypting a random ciphertext in this system is reducable to factoring. The good news here is that consequently Rabins system has maximal security against passive attacks: the problem of decryption can never be harder than that of finding the private key from the public one. The bad news is that this also implies that the system breaks down completely under a chosen ciphertext attack.This fact mislead many researchers into thinking that no public key system could be secure against a chosen ciphertext attack if the problem of decrypting was reducable to the problem of finding the private key from the public one. A similar "paradox" for public key signature schemes was also discussed in the folklore.

The folkore "proofs" of these "theorems", however, implicitely relied on the assumption that only one trapdoor is used in the system. Goldwasser, Micali and Rivest were the first to observe that the problem could be solved if two independently chosen trapdoors were used. This lead to construction of the first signature scheme secure against an

J. Feigenbaum (Ed.): Advances in Cryptology - CRYPTO '91, LNCS 576, pp. 445-456, 1992.
© Springer-Verlag Berlin Heidelberg 1992

adaptive chosen message attack [GMR]. Later, Naor and Yung [NY] combined the use of two trapdoors with non-interactive zero-knowledge proof systems to build the first public key encryption scheme provably as hard to break under a chosen ciphertext attack as under a passive attack, where the enemy simply observes ciphertexts and tries to decrypt them.

Thus, as far as theoretical results are concerned, the matter is closed. For *practical* schemes, however, we are still very far from a satisfactory solution. The scheme of [NY] relies heavily on non-interactive zero-knowledge, which is a very nice theoretical tool, but in general leads to schemes that no one would try to implement because of the enormous expansion that takes place when going from plaintext to ciphertext.

This paper makes a first step in the direction of finding truely practical public key encryption schemes with optimal security. This will be done by showing how to make modifications of any deterministic public key system (which includes RSA and Rabin), and of the El Gamal/Diffie-Hellman public key system. All the modifications preserve the security of the original system under a passive attack, while extra assumptions are needed to ensure that a chosen ciphertext attack will not help an enemy. The modifications typically require 1 extra encryption/decryption operation of the system in question, and communication of 1 extra multiprecision number.

The model of a chosen ciphertext attack that we will consider here follows that of [NY]: the enemy may specify any (polynomial) number of ciphertexts and receives the corresponding plaintexts. Then he gets a ciphertext as input and must try to decrypt it by himself. Other researchers have suggested different models [RS], where the enemy knows in advance the ciphertext he must attack, but where his choice of ciphertexts to ask for decryptions of is limited in various ways. This model requires that also senders of messages possess secret/public key pairs.

2 Deterministic Public Key Systems

A public key encryption scheme is said to be deterministic, when the ciphertext is uniquely determined from the cleartext and the public encryption key. Thus the basic form of the RSA and Rabin systems are deterministic.

For simplicity, we think of deterministic public key systems as trapdoor one-way permutations of bit strings of a given length, although systems like RSA actually are injections into such a set (using injections would complicate the notation, but not change any of the results).

Saying something meaningful about the security of a deterministic system requires that one specifies a distribution of the cleartexts. Throughout this paper, we will assume that the encryption operates on k-bit blocks, and that the cleartexts are uniformly distributed k-bit strings. This is a natural assumption if no particular application is considered, and is the only reasonable in several cases, e.g. when the system is being used for exchange of keys to a conventional cryptosystem.

Definition 1

A *family of one-way trapdoor permutations* is a countably infinite family of finite sets $F = \{I_k\}_{k=0}^{\infty}$. An element of I_k is a permutation on the set of k-bit strings and is called an instance of size k. F must satisfy the following:

1. There exists a probablistic poynomial time algorithm Gen_F called a *generator for* F, which on input k outputs a pair (f, f^{-1}). f is a polynomial time algorithm for computing a permutation randomly chosen from I_k, while f^{-1} is a polynomial time algorithm for computing the inverse of that permutation.
2. For any probabilistic polynomial time algorithm A, let $p_A(k)$ be the probability that A on input $f \in I_k$ chosen randomly by Gen_F and a uniformly chosen k-bit x manages to compute $f^{-1}(x)$. Then $p_A(k)$ is superpolynomially small as a function of k.

It should be clear that this is just a formalization of the properties that we hope for example RSA has: the permutations would be exponentiations modulo k-bit RSA-moduli, and the generator would output randomly chosen k-bit moduli, together with the public, resp. secret exponent.

In a chosen ciphertext attack against a system like this, an enemy effectively gets an oracle for f^{-1}, for example he may have been able to get temporary access to the deciphering equipment, without being able to get to the secret key itself. By choosing cleverly the ciphertexts to decrypt, he may be able to figure out the entire description of f^{-1}. That is precisely what happens with the Rabin system.

A standard practitioner's way of protecting against this would be to require that all cleartexts satisfy a certain redundancy rule, and program the deciphering algorithm such that it refuses to answer, if the cleartext produced does not satisfy this rule. The idea would be that an enemy now cannot produce ciphertexts that he can get decrypted unless

he starts by choosing the cleartext, in which case the attack becomes useless: the enemy already knows the cleartexts he gets.

However, such a redundancy rule will be of no use, unless it ensures that a uniformly chosen bit string satisfies the rule with only negligible probability. The problem now is that even if our permutation family satisfies the above Definition 1, the permutations might still become easy to invert, if we restrict the input to the negligibly small subset of messages satisfying the redundancy rule! For example, using this idea on the Rabin system would mean that we would loose the proof of the equivalence of decryption to factoring. Hence we would like to have a solution that makes a chosen ciphertext attack difficult to exploit, *without* having to change the cleartext distribution.

Consider therefore the following public key encryption scheme, which is constructed from two (not necessarily distinct) families F and G as above:

To generate keys, run the generators Gen_F and Gen_G on input k to get outputs $(f, f^{-1}), (g, g^{-1})$, respectively. Let h be an arbitrary but fixed easy-to-invert permutation on k-bit strings (we assume for simplicity that h is a generic description of a permutation algorithm that works for any k). Let f, g, h be the public key and store f^{-1} as the secret key (g^{-1} may be discarded). The enciphering function E operates on k-bit strings and is defined by:

$$E(m) = (f(r), g(h(r)) \oplus m),$$

where r is a uniformly chosen k-bit string. To deciper, we have the function D, defined by:

$$D(c, d) = g(h^{-1}(f^{-1}(c))) \oplus d$$

It is clear that $D(E(m)) = m$. We then have the following definition and result:

Definition 2

An algorithm A is said to break (F, G, h) under a passive attack, if A finds m with probability more than a polynomial fraction on input a description of E and $E(m)$. The probability is taken over a random choice of E as above, and a uniform choice of k-bit string m.

Theorem 1

If the family F satisfies Definition 1, then no probabilistic polynomial time algorithm breaks (F, G, h) under a passive attack.

Proof

Suppose A breaks (F, G, h), and let f and x be chosen as in condition 2 of Definition 1. Run Gen_G to get an algorithm for a permutation g and its inverse. Choose a uniform k-bit string s, and give to A the description of E constructed from f and g and the ciphertext (x, s). Let m be the output of A. Then as our guess at $f^{-1}(x)$, we output $h^{-1}(g^{-1}(m \oplus s))$. It is clear that the distribution of the public key f, g, h and the ciphertext (x, s) is precisely the one A expects to see. Therefore we get a correct answer m with nonnegligible probability. Finally, it is trivial that if m is correct, then our answer is also correct□

This theorem says that modifying the public key system defined by F, by using G and h as above, does not hurt the security against passive attacks. However, we may hope that the security of (F, G, h) is even better than that of F against chosen ciphertext attacks: suppose that there exists an algorithm that will find the secret key f^{-1} if it is given the public f and a black box that evaluates f^{-1}, i.e. the F-system has maximal security under a passive attack, but is breakable under a chosen ciphertext attack.

To see how the (F, G, h) system behaves in this respect, observe that what the enemy gets in a chosen ciphertext attack against (F, G, h) is the ability to specify any x to a black box and get back $g(h(f^{-1}(x)))$. Since g is one-way, it is not at all clear that the enemy can find $f^{-1}(x)$ from this information, and therefore (F, G, h) may be secure against a chosen ciphertext attack, even though it provably has maximal security against a passive attack.

It is tempting to conjecture that as long as f and g are independently chosen, seeing $g(h(f^{-1}(x)))$ does not give the enemy any useful extra knowledge, and might in fact as well be a random value, as far as a polynomial time enemy is concerned. This is too optimistic however: suppose we wanted to improve the Rabin system in a simplistic way by letting both F and G be modular squaring and h be the identity. Then an enemy could from a chosen ciphertext attack obtain residues of the same value r^2 modulo two different moduli, where r is the square root he is looking for. But from this, the integer r^2 (and hence r) can easily be found by the Chinese Remainder theorem, and we are no better off than for the original Rabin system!

Thus the functions must be chosen with more care: for example we can define h to be some efficient easy-to-invert bit-scrambling function. One nice way to construct such a function would use a pseudo-random bit generator ϕ taking bit strings of length $l < k$

as seed. Then we can define

$$h(r) = r_{k-l} \oplus \phi(r_k)||r_k,$$

where r_{k-l}, resp. r_k are the most significant $k-l$, resp. the least significant k bits of r, and $||$ denotes concatenation. For concreteness, one can think of $l = 56$ and ϕ = DES in output feedback mode, but many ideas are possible here. Another idea is to simply encrypt r under a random, but fixed key using your favorite conventional cipher.

With such a construction, it seems quite resonable to conjecture that r and $h(r)$ will appear to be unrelated as far as modular arithmetic is concerned, and that therefore the above problem will go away. Alternatively, one could let $h = id$, but define G to be RSA with random (large) public exponent.

This leads to the following concrete suggestion for an encryption function E:

$$E(m) = (r^2 \ mod \ n_1, \ m \oplus (h(r)^2 \ mod \ n_2)),$$

where h is constructed as above, n_1, n_2 are Blum-integers and r is a random square modulo n_1 (this ensures that the receiver can reconstruct r from $r^2 \ mod \ n_1$). Decryption is left to the reader. For this construction, the extra security potential comes at a price of very little extra computation - evaluation of h and a modular squaring for both sender and receiver. The bandwidth required is twice that of ordinary Rabin encryption. For applications such as exchange of conventional keys this will often be perfectly acceptable.

Of course, proving chosen ciphertext security for such concrete constructions is probably difficult. Even defining precisely what properties we should demand for F and G is non-trivial. We suggest the construction of F, G and h such that security against chosen ciphertext attacks of (F, G, h) can be proved as an interesting open problem.

3 The El Gamal/Diffie-Hellman System

This public key encryption scheme was suggested by El Gamal [ElGa] as a variant of the Diffie-Hellman key exchange. The system requires an infinite family of cyclic groups $\{G_k\}$ such that discrete logarithms are hard to compute in G_k. One can use the multiplicative groups modulo large primes here, but also other groups might be used, e.g. the groups on some elliptic curves, or the multiplicative groups of extension fields.

To run the system, we are given a generator g of G_k of order d, where d is in $O(2^k)$, such that group elements may be represented as k-bit strings. The secret key in the

system is a number x, chosen uniformly in $[0..d-1]$. The public key is $y = g^x$. Let E denote the enciphering operation, D the deciphering. Then

$$E(m) = (g^r, y^r \oplus m),$$

where r is uniform in $[0..d-1]$.

$$D(c_1, c_2) = c_2 \oplus c_1^x.$$

In this description, the $\oplus$-operation may be replaced by any easy to invert group operation on k-bit strings. It is clear that this system is not deterministic, but probabilistic: the encryption involves a random choice. For such systems, one may sometimes be able to prove that the system is secure against passive attacks, *independently* of the plaintext distribution. See for example the system of Blum and Goldwasser [BlGo]. However, such a result is not known for the El Gamal system. Moreover, although its security against chosen ciphertext attacks is unknown in general, Bert den Boer [Bo] has shown that for some primes, the Diffie-Hellman problem modulo these primes is equivalent to discrete log, and hence that in these cases the system is breakable under a chosen ciphertext attack.

Hence also for this system, it is of interest to improve its security against chosen ciphertext attacks, without having to change the plaintext distribution. To this end, we propose the following modified version of El Gamal:

The private key consists of x and z, chosen uniformly in $[0..d-1]$. The public key is $y = g^x$ and $w = g^z$. Let E' and D' denote the encryption and decryption operations, resp. Then

$$E(m) = (w^r, g^r, m \oplus y^r),$$

where r is uniform in $[0..d-1]$.

$$D(c_1, c_2, c_3) = c_3 \oplus c_2^x, \ if \ \ c_1 = c_2^z, \ NULL \ otherwise.$$

Here, $NULL$ is a special symbol which can be distinguished from ordinary plaintext, and can be thought of as meaning "no answer".

Security against passive attacks for El Gamal and Modified El Gamal is defined in the same way as in Definition 2, except that the probabilities are also taken over the random coins used in the enciphering. We first have the following easy lemma:

Lemma 1

Modified El Gamal is as hard to break as El Gamal under a passive attack.

Proof

Suppose algorithm A breaks Modified El Gamal. Then given an El Gamal public key y and a ciphertext (c_1, c_2), choose z at random in $[0..d-1]$, and give y, g^z as public key to A and (c_1^z, c_1, c_2) as ciphertext. It is trivial to see that the input we generate for A will have the distribution it expects, and that if A is successful, it decrypts the original El Gamal ciphertext.

Intuitively, the reason why this variant may be more secure against a chosen ciphertext attack, is that given only w, g, it seems hard to generate a pair of the form w^r, g^r, unless one starts by simply choosing r. Hence it will be hard for an enemy to come up with a ciphertext that will produce a non-null output from D, unless he already knows the plaintext. Formalizing this, we suggest the following assumption:

Assumption 1

Let A be a probabilistic polynomial (in k) time algorithm which receives as input $w, g \in G_k$ and outputs a pair of group elements a, b. Then there exists another probabilistic polynomial time algorithm A', which uses the same input and the same random coins as A. Except with superpolynomially small probability taken over the random coins, A' will output a, b, r, whenever A on the same input produces $(a, b) = (w^r, g^r)$.

The function that maps r to (w^r, g^r) is an example of what one could call *one-way functions with sparse image*: only a very small fraction of the pairs of numbers are in the image, and an element in the image cannot be found in any other way than by computing the function on some input value. Such functions would be extremely useful in other contexts too, e.g. identification protocols, and it is an interesting open problem to find out whether they can be proved to exist.

Definition 3

A *chosen ciphertext enemy* is a probabilistic polynomial time algorithm that repeatedly gets to choose a ciphertext and receive the output from the decryption algorithm on this ciphertext. It finally takes a random ciphertext as input and tries to decrypt it. The cryptosystem is said to be *secure against chosen ciphertext attacks*, if any chosen ciphertext enemy succeeds with only superpolynomially small (in k) probability.

Assumption 1 is enough to prove this type of security for Modified El Gamal:

Theorem 2

Assume Assumption 1 and that El Gamal is secure under a passive attack. Then Modified El Gamal is secure under a chosen ciphertext attack.

Proof

By contradiction, let A be a successful chosen ciphertext enemy. Let $C_1, C_2, \ldots$ be the sequence of ciphertexts of which A requests the decryption. Let A_i be the algorithm that simulates A until the output of C_i and then stops. We can now show by induction that for all i, A_i can be simulated *without* access to a decryption oracle: A_1 is clear. To do A_{i+1}, observe that by induction, A_i can be simulated without the oracle. Therefore Assumption 1 guarantees us the existence of an algorithm A_i' that outputs a quadrouple (c_1, c_2, c_3, r), such that $C_i = (c_1, c_2, c_3)$, and that with large probability, $(c_1, c_2) = (w^r, g^r)$ whenever C_i produces a non-null output from the decryption. Knowledge of r suffices to decrypt C_i, and therefore we can simulate also the last steps of A_{i+1}. From A we can therefore build an algorithm that breaks the system under a passive attack, and we are done by Lemma 1.

Note that, contrary to Theorem 1, this result works for any plaintext distribution. Note also that we are talking about the ability of an enemy to decrypt entire messages, not whether he can get partial information about the cleartext. However, if we had a result about security of single bits in El Gamal encryption similar to the one for RSA and Rabin bits, it would be easy to reformulate and prove Theorem 2 in terms of probabilistic public key systems, following [NY].

A final remark concerns the model for chosen ciphertext attacks: at first sight, it may seem like a strange condition that the enemy does not get to see the ciphertext he is to decrypt until after his usage of the decryption oracle is over. However, if he knows initally that he eventually wants to decrypt ciphertext C, then he might as well give C to the oracle in stead of trying to figure it out himself!

In [RS], it is argued that one could solve this problem in a natural way by allowing the enemy to know C initially, but introduce a restriction on his choice of ciphertexts for decryption in the first phase. One concrete possibility is to demand that he asks the oracle for anything but C. It is not clear, however, that this makes the model more natural or realistic: the enemy may well be the only player who knows which ciphertext he is attacking, and in this case, how could the rest of the system possibly impose on him the restriction required by the model?

It is quite possible that we have not yet found the best model to describe this type of problem, and that a final solution would have to include some conditions on the timeliness of messages.

4 Connection to Identification Protocols

In this section we point out an interesting duality between the conditions of Definition 3 and the properties of a secure identification system as defined Feige, Fiat and Shamir in [FFS].

Concretely, from any public key encryption scheme that satisfies Definition 3, it is easy to build an efficient identification system: each individual knows a secret key for which the corresponding public key is known by everybody. A prover can identify himself by demonstrating his ability to decrypt messages that were encrypted under the corresponding public key, i.e. the verifier encrypts a random message m, sends the encryption to the prover, who decrypts and returns m.

By Definition 3, even a cheating verifier who interacts with the honest prover P a polynomial number of times will not afterwards be able to impersonate P with non-negligible probability of success.

Such systems are very efficient in terms of the number of rounds used: only two messages have to be sent. This adds to the interest of solving the open problems listed below: their solution would also lead to construction of very efficient identification schemes.

Note that if Assumption 1 holds, then the protocol in which a verifier receives a public key for the Modified El Gamal system, chooses a ciphertext and receives the decryption, would in fact be zero-knowledge. Despite the small number of messages sent, this does not contradict the result of Goldreich and Krawzyk [GK] because the simulation we get from Assumption 1 is not black-box: it depends on the verifier that participates.

5 Conclusion and Open Problems

We have seen that truely practical public key systems can be constructed, for which chosen ciphertext attacks seem totally useless for an enemy, and for which security against passive attacks is provably equivalent to that of well known systems like RSA, Rabin and El Gamal/Diffie-Hellman.

Open problems: prove or disprove Assumption 1. Construct other functions that satisfy an assumption similar to Assumption 1. Find permutation families F and G for which the (F, G, h) system of Section 2 is provably secure against chosen ciphertext attacks.

References

[BlGo] M.Blum and S.Goldwasser: *An Efficient Probabilistic Public-Key Encryption Scheme which Hides all Partial Information*, Proc. of Crypto 84, Springer Verlag.

[Bo] B. den Boer: *Diffie-Hellman is as Strong as Discrete Log for Certain Primes*, Proc. of Crypto 84, Springer Verlag.

[ElGa] T. El Gamal: *A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*, IEEE Trans. on Inf. Theory, vol.IT-31, 1985.

[FFS] U.Feige, A.Fiat and A.Shamir: *Zero-Knowledge Proofs of Identity*, J.Crypt, vol. 1, 1988, Springer Verlag.

[GK] O.Goldreich and H. Krawczyk: *On the Composition of Zero-Knowledge Proof Systems*, Proc. of ICALP 90.

[GMR] S.Goldwasser, S.Micali and R.Rivest: *A "Paradoxical" Solution to the Signature Problem*, proc. of FOCS 84.

[NY] M. Naor and M. Yung: *Public-Key Cryptosystems Provably Secure against Chosen Ciphertext Attacks*, Proc. of FOCS 90.

[Ra] M.O. Rabin: *Digital Signatures and Public Key Encryption as Intractable as Factorization*, Tech. report, MIT/LCS/TR-212. M.I.T., 1978.

[RS] C. Rackoff and D. Simon: *Non-Interactive Zero-Knowledge Proofs of Knowledge and Chosen Ciphertext Attacks*, these proceedings.