On the Security of ElGamal Based Encryption

Yiannis Tsiounis[1] and Moti Yung[2]

[1] GTE Laboratories Inc., Waltham MA. e-mail: ytsiounis@gte.com
[2] CertCo, NY, NY. e-mail: moti@certco.com

Abstract. The ElGamal encryption scheme has been proposed several years ago and is one of the few probabilistic encryption schemes. However, its security has never been concretely proven based on clearly understood and accepted primitives. Here we show directly that the decision Diffie-Hellman assumption implies the security of the original ElGamal encryption scheme (with messages from a subgroup) without modification. In addition, we show that the opposite direction holds, i.e., the semantic security of the ElGamal encryption is actually *equivalent* to the decision Diffie-Hellman problem. We also present an exact analysis of the efficiency of the reduction.

Next we present additions on ElGamal encryption which result in non-malleability under adaptive chosen plaintext attacks. Non-malleability is equivalent to the decision Diffie-Hellman assumption, the existence of a random oracle (in practice a secure hash function) or a trusted beacon (as needed for the Fiat-Shamir argument), and one assumption about the unforgeability of Schnorr signatures. Our proof employs the tool of message awareness.

1 Introduction

Discrete-log based building blocks are heavily used in protocols. For example in ElGamal encryption and signatures, Schnorr signatures or discrete-log based bit commitments. However, most of these sub-protocols have not been shown equivalent to some natural discrete-log or Diffie-Hellman variant, thus several practical systems are forced to rely on a multitude of arbitrary security assumptions. In this work we prove security which is a step towards reducing some of these sub-protocols to more natural and widely used assumptions by showing that the semantic security of the ElGamal encryption [ElG85] is equivalent to the decision Diffie-Hellman assumption. Note that obtaining the full plaintext of an ElGamal encrypted ciphertext is equivalent to the Diffie-Hellman assumption [ElG85,SS98]; thus the fact that the *semantic security* [GM84] of ElGamal encryption is equivalent to the decision Diffie-Hellman problem provides a natural analogy. In fact, ElGamal has stated this result as a conjecture in his earlier work [ElG98].

We also discuss efficiency (security degradation) of our reduction. As a result of the proof, one can deduce the availability (under a proper assumption) of having a very efficient semantically secure scheme which is ElGamal-based

(unlike the typically more theoretical constructions). The scheme can be very efficient since encryption can be done with preprocessing of the exponentiation operations, and decryption involves one exponentiation which can be accelerated using preprocessing (e.g., as in [BGM93]).

This result can also be seen as a "shortcut" to the provability of the ElGamal encryption, bypassing the need to first prove security under the random oracle model [BR94,BR97] and then construct "practical random oracles" under, e.g., variants of the decision Diffie-Hellman assumption as was advocated and done in [Can97].

Recently, some results have utilized the decision Diffie-Hellman assumption or some of its variants (see [FTY96,Bea97,Bea96,NR97,Can97,CS98]). The last three works also claim the correctness of the first result we report here, but as a side issue (since they mainly deal with other interesting problems) and without a proof. We believe that the exact proof of security together with related results presented here is a central issue which deserves publicity (indeed, we first reported our work in [TY97]).

We then proceed in the second part of the paper to show extensions of the ElGamal encryption; in particular, an instatiation of a non-malleable [DDN91] under adaptive chosen plaintext attacks [RS92] variant of ElGamal which is provably secure under the random oracle model, the decision Diffie-Hellman assumption, and one assumption about the security of Schnorr signatures. Other instatiations (for semantic security under chosen ciphertext attacks or message awareness) have been proposed earlier [ZS93,BR94,BR97] (our constructions have some similarity to those in [ZS93]) but their security proofs rely on random-oracle like hash functions which hide all partial information and (for [ZS93]) some other assumptions. Our results use the random oracle in a different way than [ZS93,BR94,BR97]; the only use of the oracle is for the Fiat-Shamir argument, i.e., as an unpredictable beacon. Thus, the common practice of substituting random oracles with collision-resistant hash functions is more suited to our assumptions. Recently, [CS98] presented another variation of ElGamal encryption which remarkably is semantically secure under adaptive chosen plaintext attacks (thus also non-malleable), based on the decision Diffie-Hellman assumption and Collision Intractable Hash Functions, i.e., in the standard model (without the use of random oracles).

Organization: We begin with the basic definitions in section 2, and proceed with the proofs in both directions in sections 3 and 4. Then we discuss the efficiency of our reductions in section 5, and we show security extensions, in particular a provably non-malleable and message aware scheme under the random oracle model, in section 6.

2 Preliminaries

In this section we provide a consistent background for the proofs in the sequel.

对于安全参数n，对于特定常量δ选取素数P和Q满足δ+n = P-1,对于特定的y，P = yQ+1，一个独特的乘法循环群ZP的子群GQ，order为Q，

2.1 The Diffie-Hellman problem and ElGamal encryption

First we formally define the decision Diffie-Hellman problem and the ElGamal encryption scheme.

The following setup is common for all definitions.

Setup: For security parameter n, primes P and Q are chosen such that $|P-1| = \delta + n$ for a specified constant δ, and $P = \gamma Q + 1$, for a specified integer γ. Then a unique subgroup G_Q of prime order Q of the multiplicative group Z_P^* and generator g of G_Q are defined.

> **(Decision Diffie-Hellman problem)** For security parameter n, P a prime with $|P-1| = \delta + n$ for a specified constant δ, for $g \in Z_P^*$ a generator of prime order $Q = (P-1)/\gamma$ for a specified integer γ and for $a, b \in_R Z_Q$ random, given $[g^a, g^b, y]$ output 0 if $y \equiv g^{ab} \pmod P$ and 1 otherwise, with probability better than $1/2 + 1/n^c$ for any constant c for large enough n.

For example if P is a strong prime $P = 2Q + 1$, then g can be a generator which generates all the quadratic residues.

The *decision Diffie-Hellman assumption* (decision D-H) states that it is infeasible for a p.p.t. adversary to solve the decision Diffie-Hellman problem.

Next we define the ElGamal public-key encryption scheme (modified to have messages from a subgroup). The ElGamal encryption scheme [ElG85] is based on the Diffie-Hellman assumption and it is a probabilistic encryption scheme, i.e., a specific message has many—exponential in the security parameter—possible encryptions. Formally,

Definition 1. (ElGamal public-key encryption scheme) *The* ElGamal public key encryption *scheme is defined by a triplet (G, E, D) of probabilistic polynomial time algorithms, with the following properties:*

- *The* system setup algorithm, S, *on input 1^n, where n is the security parameter, outputs the system parameters (P, Q, g), where (P, Q, g) is an instance of the DLP collection, i.e., P is a uniformly chosen prime of length $|P| = n + \delta$ for a specified constant δ, and g is a uniformly chosen generator of the subgroup G_Q of prime order Q of Z_P^*, where $Q = (P-1)/\gamma$ is prime and γ is a specified small integer.*
- *The* key generating algorithm, G, *on input (P, Q, g), outputs a public key, $e = (P, Q, g, y)$, and a private key, $d = (P, Q, g, x)$, where*
  - *x is a uniformly chosen element of Z_Q, and*
  - *$y \equiv g^x \bmod P$.*

  (Note: In the proofs below we abuse the notation and assume that the input on G is simply 1^n.)
- *The* encryption algorithm, E, *on input (P, Q, g, y) and a message $m \in G_Q$, uniformly selects an element k in Z_Q and outputs*

$$E((P, Q, g, y), m) = (g^k \pmod P, my^k \pmod P) .$$

- *The* decryption algorithm, D, *on input* (P, Q, g, x) *and a ciphertext* (y_1, y_2), *outputs*

$$D((P, g, x), (y_1, y_2)) = y_2 (y_1{}^x)^{-1} \pmod{P} .$$

For example for P a strong prime $P = 2Q + 1$ and where g is a generator which generates all the quadratic residues, the above system may be useful to send messages which are quadratic residues mod P; one can have a message with added random field which is chosen so as to make the entire sent message a residue. (Other modifications are easily obtainable to use the variant of ElGamal over a subgroup).

2.2 Semantic security 语义安全

Here we reiterate the definition of semantic security.

Semantic security for an encryption scheme [GM84,Gol93] is defined as follows:

一个加密方案是语义安全的：对于属于多项式随机变量的每个集体X，对于每个多项式函数h，对于每个函数f，对于每个概率多项式算法，存在一个概率多项式算法对于每个常量c>0，有一下结论

Definition 2. (semantically secure encryption) *An encryption scheme* (G, E, D) *is said to be* semantically secure *if, for every ensemble* $X = \{X_n\}_{n \in N}$ *of polynomial random variables, for every polynomial function* h, *for every function* f, *and for every probabilistic polynomial time algorithm* A, *there exists a probabilistic polynomial time algorithm* A' *such that for every constant* $c > 0$ *and for every sufficiently large* n,

$$Pr\left[A(E_{G(1^n)}(X_n), h(X_n), 1^n) = f(X_n)\right] \leq Pr[A'(h(X_n), 1^n) = f(X_n)] + \frac{1}{n^c} ,$$

where the probability is taken over the coin tosses of A *(resp.* A'*),* E *and* G, *and the distribution of* X.

Intuitively, given any a-priori information, $h(X_n)$, no algorithm A can obtain some information, $f(X_n)$, from the ciphertext that could not have been efficiently computed by A' without the ciphertext.

There is an alternative way to define secure encryption, which sometimes proves more useful in practice. The definition is based on indistinguishability; intuitively, if it is infeasible for an adversarial algorithm to distinguish between the encryptions of *any* two messages, even if these messages are given, then the encryption should not reveal any information about the messages (in the uniform case it suffices that two such messages cannot be efficiently found). Here we include, for completeness, the definition of security in the sense of indistinguishability (from [Gol89]).

Definition 3. (encryption secure in the sense of indistinguishability) *An encryption scheme* (G, E, D) *is said to be* secure in the sense of indistinguishability *if, for every probabilistic polynomial time algorithm* F *(for "Find"), for every probabilistic polynomial time algorithm* A, *for every constant* $c > 0$ *and for every sufficiently large* n,

$$Pr\left[F(1^n) = (\alpha, \beta, \gamma) \ s.t. \ \Omega(\alpha, \beta, \gamma) > \frac{1}{n^c}\right] < \frac{1}{n^c} ,$$

with

$$\Omega(\alpha,\beta,\gamma) = \left|Pr\left\{A((\gamma), E_{G(1^n)}(\alpha)) = 1\right\} - Pr\left\{A(\gamma, E_{G(1^n)}(\beta)) = 1\right\}\right| ,$$

where the probability is taken over the coin tosses of F, A, E and G.

It has been proven that *an encryption scheme secure in the sense of indistinguishability is semantically secure* [GM84]. The opposite direction was shown in [MRS88].[1] In the sequel we show the equivalence of the decision Diffie-Hellman with the security in the sense of indistinguishability of ElGamal encryption. In the non-uniform model this is equivalent to semantic security.

3 ElGamal is at least as hard as the decision D-H

如果Elgamal加密方案在不可区分性上是不安全的，那么现有的 a ppt tm就可以以极大的概率解决DDH问题

Theorem 1. *If the ElGamal encryption scheme is not secure in the sense of indistinguishability, then there exists a p.p.t. TM that solves the decision Diffie-Hellman problem with overwhelming probability.*

Proof. We show the uniform case. If the ElGamal encryption is not secure in the sense of indistinguishability then there exists a p.p.t. adversarial algorithm A (which can be seen as the "oracle" that "breaks" the ElGamal scheme), a (polynomial) random variable Z_n and two independent (polynomial) random variables (X_n, Y_n) that have the same distribution, such that:

存在c>0,N

$\exists\, c > 0, \exists\, N$, s.t. for infinitely many $n > N$, $\Pr[(X_n Y_n Z_n) \in B_n^c] > \frac{1}{n^c}$, where

攻击者可以有效区分两个密文的差异的概率

$$B_n^c = \left\{(\alpha,\beta,\gamma) : \left|\Pr\left[A(\gamma, E_{G(1^n)}(\alpha)) = 1\right] - \Pr\left[A(\gamma, E_{G(1^n)}(\beta)) = 1\right]\right| > \frac{1}{n^c}\right\} ,$$

where the probabilities are taken over the coin tosses of the key generating algorithm G, the encryption algorithm $E_{G(1^n)}$, the adversarial algorithm A, and the selection of (α,β,γ).

We now show how this adversarial algorithm can be used by a translator T to solve an instance of the decision D-H problem.

Translation:

The translator is given a triplet Translator的目的是确定这是不是一个DH的三元组

$$[g^a \pmod P, g^b \pmod P, y] ,$$

for g a generator of G_Q, $y \stackrel{\text{def}}{\equiv} g^x \pmod P$, and $a, b \not\equiv 0 \pmod Q$. Its purpose is to decide (non-negligibly better than random guessing) whether it is a correct Diffie-Hellman triplet (i.e., whether $x \equiv ab \pmod Q$) or not.

The translator then performs the following steps, for the above P, Q, g.

[1] These results were generalized for the uniform case in [Gol89,Gol93], but the proof of the reverse direction (i.e., semantic security implying distinguishable encryptions) requires the additional assumption of *decomposability* (given $E(\alpha)$ one should be able to find $E(\beta)$ for any suffix β of α with $|\beta| = 1/3|\alpha|$).

T试图找到一对消息，使得敌手算法可以辨别这对消息。
T选择一个随机的消息对，然后评估用预言机辨别他们的效率
基于不可区分性的矛盾，这一步保证在多项式步骤内成功

1. **Preparation stage:** The translator first tries to find a pair of messages (m_1, m_2) that can be distinguished by the adversarial algorithm ("oracle"). Intuitively, the translator chooses random message pairs and evaluates the oracle's effectiveness on distinguishing them. It is shown that this step is guaranteed to succeed in polynomial steps, based on the contradiction of indistinguishability of encryptions.
   Specifically, the translator chooses a triplet (m_1, m_2, γ) from the distribution (X_n, Y_n, Z_n) and tries to estimate the difference
   特别的，T从分布（）中选择一个三元组（），并试图来评估他们的区别
   $$\Delta(m_1, m_2, \gamma) = \left|\Pr[A(\gamma, E_{G(1^n)}(m_1)) = 1] - \Pr[A(\gamma, E_{G(1^n)}(m_2)) = 1]\right| ,$$
   with accuracy better than $\frac{1}{4n^c}$. This accuracy can be achieved with overwhelming probability $(1 - 2^{-n})$ with $s_1 = 64 \ln 2(n+1)n^{2c}$ experiments (using the Hoefding inequality, with each experiment allowing new coin tosses for A, E and G). (We refer to section 5 for more details). If the difference is greater than $\frac{3}{4n^c}$ then the pair is accepted. In this case the translator also records the calculated probability for $[A(\gamma, E_{G(1^n)}(m_1)) = 1]$, as this is to be used in the next phase. It is then guaranteed (with overwhelming probability) that the actual difference $\Delta(m_1, m_2, \gamma)$ for this pair is at least $\frac{1}{2n^c}$. In other words, for this particular pair (m_1, m_2) the oracle finds a difference whose expected (mean) value is at least $\frac{1}{2n^c}$.
   If the estimate for the difference is smaller or equal than $\frac{3}{4n^c}$ then the pair is rejected and a new one is tried. From the properties of the oracle (A) it is guaranteed that the probability of finding such a pair with the required difference is at least $\frac{1}{n^c}$, thus an average of at most $\frac{n^c}{2}$ experiments will be performed.
2. **Testing phase:** In this stage the translator tries to see if the oracle is successful in distinguishing between m_1 and uniformly chosen messages.
   Specifically, the translator uniformly chooses messages $m \in G_Q$ and estimates the value
   $$\Pr[A(\gamma, E_{G(1^n)}(m)) = 1] ,$$
   with accuracy better than $\frac{1}{32n^c}$. For an error that is negligible in n (i.e., probability of success $(1 - 2^{-n})$) we need $s_2 = 512 \ln 2(n+1)n^{2c}$ experiments, where a different m is used in each experiment (and of course $E_{G(1^n)}(m)$ is created using new coin tosses for E). Then it calculates the difference
   $$\Delta(m_1, m, \gamma) = \left|\Pr[A(\gamma, E_{G(1^n)}(m_1)) = 1] - \Pr[A(\gamma, E_{G(1^n)}(m)) = 1]\right| ,$$
   again with accuracy $\frac{1}{32n^c}$; some of the calculations of the first phase can be reused here (see section 5). If the difference is greater than $\frac{3}{16n^c}$ then the actual difference is at least $\frac{1}{8n^c}$, and the oracle can distinguish between m_1 and random messages (i.e., in comparing m_1 with random messages the oracle finds a difference whose expected (mean) value is at least $\frac{1}{8n^c}$). Otherwise the actual difference is less than $\frac{1}{4n^c}$.
   Now we show that if the oracle does not distinguish between m_1 and random messages it must be the case that it can distinguish between m_2 and random

messages. To see this, consider that for any message m we have

$$\begin{aligned}
\frac{1}{2n^c} &< \Delta(m_1, m_2, \gamma) \\
&= \left|\Pr[A(\gamma, E_{G(1^n)}(m_1)) = 1] - \Pr[A(\gamma, E_{G(1^n)}(m_2)) = 1]\right| \\
&= \left|\Pr[A(\gamma, E_{G(1^n)}(m_1)) = 1] - \Pr[A(\gamma, E_{G(1^n)}(m)) = 1] + \right. \\
&\quad \left. \Pr[A(\gamma, E_{G(1^n)}(m)) = 1] - \Pr[A(\gamma, E_{G(1^n)}(m_2)) = 1]\right| \\
&\leq \left|\Pr[A(\gamma, E_{G(1^n)}(m_1)) = 1] - \Pr[A(\gamma, E_{G(1^n)}(m)) = 1]\right| + \\
&\quad \left|\Pr[A(\gamma, E_{G(1^n)}(m)) = 1] - \Pr[A(\gamma, E_{G(1^n)}(m_2)) = 1]\right| \\
&= \Delta(m_1, m, \gamma) - \Delta(m, m_2, \gamma) \ .
\end{aligned}$$

Thus, for m_i uniformly chosen (i.e., $\Pr[m_i] = \frac{1}{|G_Q|}$), we have

$$\begin{aligned}
\Sigma_i \left[\Delta(m_1, m_i, \gamma) - \Delta(m_i, m_2, \gamma)\right] > \Sigma_i \frac{1}{2n^c} &\iff \\
\Sigma_i \Delta(m_1, m_i, \gamma) - \Sigma_i \Delta(m_i, m_2, \gamma) > |G_Q| \frac{1}{2n^c} &\iff \\
\Sigma_i \frac{\Pr[m_i]}{\Pr[m_i]} \Delta(m_1, m_i, \gamma) - \Sigma_i \frac{\Pr[m_i]}{\Pr[m_i]} \Delta(m_i, m_2, \gamma) > |G_Q| \frac{1}{2n^c} &\iff \\
|G_Q| \Sigma_i \Pr[m_i] \Delta(m_1, m_i, \gamma) - |G_Q| \Sigma_i \Pr[m_i] \Delta(m_i, m_2, \gamma) > |G_Q| \frac{1}{2n^c} &\iff \\
\mathrm{Exp}[\Delta(m_1, m_i, \gamma)] - \mathrm{Exp}[\Delta(m_i, m_2, \gamma)] > \frac{1}{2n^c} \ , &
\end{aligned}$$

where the expected value is taken over the choice of messages m_i (and the last step holds based on the uniform choice of m_i's).
Therefore if $\mathrm{Exp}\Delta(m_1, m, \gamma) < \frac{1}{4n^c}$ then it must be that $\mathrm{Exp}\Delta(m, m_2, \gamma) > \frac{1}{4n^c}$.
This step requires $P_2 = 992 \ln 2 (n+1) n^{2c}$ executions of the oracle (see section 5 for details).

3. **Decision phase:** Here the translator proceeds according to the result of the testing phase.
   - If the oracle can distinguish between m_1 and a random message then the translator "randomizes" m_1 to m_1' and runs the oracle on (m_1, m_1'). This randomization is based on the given triplet, such that $m_1' = m_1$ if the triplet is a correct D-H triplet, or m_1' is a uniformly chosen message otherwise. The randomization also has to guarantee that $E(m_1)$ is independent of $E(m_1')$ (i.e., the coin tosses of E are not affected by the selection of m_1').
   - If the oracle cannot distinguish between m_1 and random messages then (as we saw in the testing phase) it can distinguish between m_2 and random messages. Thus the translator randomizes m_2 and runs the oracle on (m_2', m_2).

   If the oracle manages to distinguish between the values then the D-H triplet is incorrect (i.e., $x \not\equiv ab \pmod{Q}$); and it is a correct triplet otherwise.

For this step we first show how the randomization is performed and then how the translator tries to distinguish between m and m' (where m is either m_1 or m_2, based on the result of the testing phase).

(a) **Randomization:** Given a message $m \in G_Q$ (and the candidate triplet $[g^a, g^b, y]$), the translator uniformly selects exponents $u, v, t \in_R Z_Q^*$, and outputs the ElGamal ciphertexts

$$E(m) = [m(g^b)^{wu}, g^u], \quad E(m') = [my^{wt}(g^b)^{wv}, (g^a)^t g^v = g^{(at+v)}] ,$$

based on public key g^{bw} and generator g, for $w \in_R Z_Q^*$. This transformation results in a random and independently selected (from m) message m' when the D-H triplet is incorrect, while it produces the same message $m' = m$ when the given triplet is a correct D-H triplet. To see this, observe the following:

- The random coin tosses of the key generating algorithm G are simulated by the selection of w: g^{bw} is now a uniformly chosen public key, since $g^b \not\equiv 1 \pmod P$ is a generator of G_Q.
- The plaintext m' is equal to m when $y \equiv g^{ab} \pmod P$, but if $x \not\equiv ab \pmod Q$ the message is $m' \equiv mg^{(x-ab)tw} \pmod P$ because the oracle sees the ciphertext as

$$c \stackrel{\text{def}}{=} [(g^{bw})^{(at+v)} m', g^{(at+v)}] .$$

  It is also easy to verify that $m \equiv m' \pmod P \iff g^{tw(x-ab)} \equiv 1 \pmod P \iff tw(x-ab) \equiv 0 \pmod Q$, that is $x \equiv ab \pmod Q$.
- If $x - ab \not\equiv 0 \pmod Q$, i.e., $m \not\equiv m' \pmod P$, we have that $g^{w(x-ab)} \not\equiv 1 \pmod P$ is a generator of G_Q; thus by changing t the "message" $m' \stackrel{\text{def}}{=} mg^{tw(x-ab)}$ can get any value in G_Q, i.e., $m' \in_R G_Q$, and furthermore it is independent of m (due to the random choice of t).
- Finally, $E(m')$ is independent of $E(m)$, due to the additional choice of v.

(b) **Distinguishing:** Here the difference between m and m' is estimated

$$\Delta(m, m', \gamma) = \left|\Pr[A(\gamma, E_{G(1^n)}(m)) = 1] - \Pr[A(\gamma, E_{G(1^n)}(m')) = 1]\right| ,$$

with accuracy better than $\frac{1}{16n^c}$ if $m = m_2$ or $\frac{1}{32n^c}$ if $m = m_1$. The number of experiments required for obtaining such accuracy with error probability less than 2^{-n} is $s_3 = 992 \ln 2(n+1)n^{2c}$ or $s_3 = 3584 \ln 2(n+1)n^{2c}$. (Each experiment requires a different randomized m', so that the approximation of $\mathrm{Exp}[\Delta(m, m', \gamma)]$ is found).

Now since the real difference is either at least $\frac{1}{4n^c}$, if $m \not\equiv m' \pmod P$ (resp. $\frac{1}{8n^c}$ for $m = m_1$), or 0 if $m = m'$, the estimate can either be greater than $\frac{3}{16n^c}$ (resp. $\frac{3}{32n^c}$) or lower than $\frac{1}{16n^c}$ (resp. $\frac{1}{32n^c}$). In the first case the triplet given is an incorrect D-H triplet and in the second it is a correct triplet.

Finally, the expected number of oracle calls for this step is on the average $P_3 = 2288 \ln 2(n+1)n^2 c$. □

4 Decision D-H is at least as hard as ElGamal

For this proof we show that if there exists an oracle solving the decision Diffie-Hellman problem then the ElGamal encryption is not secure in the sense of indistinguishability, and therefore it is not semantically secure. This part completes section 3, to show that the semantic security of the ElGamal encryption and the decision Diffie-Hellman assumption are equivalent.

Note that this direction is much easier and intuitive than the previous one. Also notice that the decision D-H oracle allows us to build a very strong ElGamal oracle that distinguishes between any two messages; that is, there are no restrictions in terms of the probability distribution of the messages to be distinguished (i.e., the messages *need not* be constructed in any particular way).

Theorem 2. *If there exists an oracle $\mathcal{O}$ which solves the decision Diffie-Hellman problem with probability non-negligibly better than random guessing then the ElGamal encryption scheme is not secure in the sense of indistinguishability.*

Proof. In order to show that an encryption algorithm is not secure in the sense of indistinguishability it suffices to show that we can find, with non-negligible probability, a pair of plaintext messages such that their encryptions can be distinguished with non-negligible probability of success.

Let $y \equiv g^x \pmod{P}$ be the public key of a party in an ElGamal encryption scheme. Our adversarial algorithm selects random $m_0, m_1 \in_R G_Q$.

Then given the ElGamal encryptions of these messages, i.e.,

$$((P,Q,g,y),[y^{r_0}m_i, g^{r_0}]) \ , i \in_R \{0,1\} \text{ and}$$

$$((P,Q,g,y_1),[y^{r_1}m_{1-i}, g^{r_1}]) \ ,$$

where $r_0, r_1 \in_R Z_Q$, we only need to show that given the decision D-H oracle we can distinguish non-negligibly better than random guessing which ciphertext encrypts which message, i.e., find i.

To this effect we employ a translator which constructs an instance of the decision D-H problem in such a way that solving this instance allows us to distinguish the ciphertext for messages m_0 and m_1.

Translation:

Given the above ciphertexts and the messages m_0, m_1 the translator selects random $v \in_R Z_Q^*$ and outputs:

$$g^{r_0} \pmod{P} \ , \ yg^v \equiv g^{x+v} \pmod{P} \text{ and}$$

$$g^{r_0 v} y^{r_0} m_i / m_0 \pmod{P} \ .$$

It is now easy to see that if $m_i \equiv m_0 \pmod{P}$ we have $i = 0$ except with negligible probability (namely, i can be 1 when $m_1 = m_0$), and the first ciphertext encrypts the first message; then the decision D-H oracle would output 0 (i.e., "correct triple") with probability non-negligibly better than random guessing, since the input would be a (uniformly distributed, since r_0 and v are randomly

chosen) correct D-H triplet. Otherwise, if $m_i \not\equiv m_0 \pmod{P}$, we have that $i = 1$, and the input to the oracle would still be valid and uniformly distributed, but the output would be 1, again with probability non-negligibly better than random guessing. Therefore the oracle can be used to determine i with probability non-negligibly better than random guessing. □

5 Efficiency of reductions

For an exact treatment of our reductions we analyze here the amount of computation that the translator has to perform in order to solve the decision Diffie-Hellman problem given an ElGamal oracle.[2]

We concentrate on the number of calls to the ElGamal oracle, as well as the number of exponentiations that need to be performed. For concreteness we assume that we have an oracle for which $\Pr[(X_n Y_n Z_n) \in B_n^c] > \epsilon$. The analysis proceeds with the steps of the reduction.

1. **Preparation stage.** The difference $\Delta(m_1, m_2, \gamma)$ is estimated using the Hoefding inequality. For completeness we repeat the definition of the latter:

   Hoefding inequality: Let $X_1, X_2, \ldots, X_n$ be n independent random variables with identical probability distribution, each ranging over the (real) interval $[a, b]$, and let μ denote the expected value of each of these variables. Then,

   $$\Pr\left(\left|\frac{\Sigma_{i=1}^n X_i}{n} - \mu\right| > \delta\right) < 2 \cdot e^{-\frac{2\delta^2}{b-a} \cdot n} .$$

   The estimation proceeds as follows. First we define

   $$W_i = \Pr[A^i(\gamma, E^i_{G^i(1^n)}(m_1)) = 1] ,$$

   for $i = 1 \ldots k$ where A^i, E^i, G^i denote the i-th call of algorithms A, E, G, such that in each call the algorithms are allowed a new, independent set of coin tosses.
   Then, the above $W_1, W_2, \ldots, W_k$ are independent random variables with identical probability distribution, each ranging over the interval $[0, 1]$. If $\mathrm{Exp}(W_i)$ is the expected value of each of those variables then, from the Hoefding inequality, substituting for $\delta = \frac{1}{d \cdot n^c}$, we have

   $$\Pr\left(\left|\frac{\Sigma_{i=1}^k W_i}{k} - \mathrm{Exp}(W_i)\right| > \frac{1}{d \cdot n^c}\right) < 2 \cdot e^{-\frac{2}{d^2 \cdot n^{2c}} \cdot k} = P_k .$$

   Thus, with probability $1 - P_k$, the average value of k experiments is an estimate of the expected value of each W_i with accuracy $\frac{1}{dn^c}$. Since we want

[2] The other direction is easier and the efficiency of the reduction apparent.

$1-P_k$ to be at least $1-2^{-n}$ we can find the number of required experiments by solving the following inequality:

$$P_k = 2 \cdot e^{-\frac{2}{d^2 \cdot n^{2c}} \cdot k} \leq 2^{-n} ,$$

which results in $k \geq \frac{\ln 2 \cdot (n+1) \cdot d^2 \cdot n^{2c}}{2}$.
For the estimation we compute the estimate of $\mathrm{Exp}(W_i)$ and $\mathrm{Exp}(V_i)$ with accuracy $\frac{1}{8n^c}$, where

$$V_i = \Pr[A^i(\gamma, E^i_{G^i(1^n)}(m_2)) = 1] ,$$

and we subtract the two estimates to find $\mathrm{Exp}\Delta(m_1, m_2, \gamma)$ with accuracy $\frac{1}{4n^c}$.
For each estimate we need $\frac{\ln 2 \cdot (n+1) \cdot 8^2 \cdot n^{2c}}{2} = 32 \cdot \ln 2 \cdot (n+1) \cdot n^{2c}$ experiments and each experiment requires two modular exponentiations (i.e., computing one ElGamal encryption).
Since we need $\frac{1}{\epsilon}$ experiments on the average to find two messages that can be distinguished, we have that in average this step requires $P_1 = 64 \cdot \ln 2 \cdot (n+1) \cdot n^{2c} \cdot \epsilon^{-1}$ oracle calls, and $D_1 = 2P_1 = 128 \cdot \ln 2 \cdot (n+1) \cdot n^{2c} \cdot \epsilon^{-1}$ modular exponentiations.

2. **Testing phase.** Here the estimates for $\Pr[A(\gamma, E_{G(1^n)}(m)) = 1]$ and $\Pr[A(\gamma, E_{G(1^n)}(m_1)) = 1]$ are computed, with accuracy $\frac{1}{32 \cdot n^c}$. The estimate for m_1 is stored for the next phase. Some number of experiments for m_1 have already been conducted in the preparation phase, so this step requires $P_2 = (2 \cdot 512 - 32) \cdot \ln 2 \cdot (n+1) \cdot n^{2c} = 992 \cdot \ln 2 \cdot (n+1) \cdot n^{2c}$ oracle calls and $D_2 = 1984 \ln 2 (n+1) n^{2c}$ modular exponentiations.
3. **Decision phase.** (*Distinguishing* step.) Here the oracle calls for estimating $\Pr[A(\gamma, E_{G(1^n)}(m')) = 1]$ with accuracy $\frac{1}{32 \cdot n^c}$ or $\frac{1}{64 \cdot n^c}$ are either $512 \cdot \ln 2 \cdot (n+1) \cdot n^{2c}$ or $2048 \cdot \ln 2 \cdot (n+1) \cdot n^{2c}$. In the first case some experiments for m_2 can be reused from the preparation phase, so $(512-32) \cdot \ln 2 \cdot (n+1) \cdot n^{2c} = 480 \cdot \ln 2 \cdot (n+1) \cdot n^{2c}$ calls are needed; similarly, if $m = m_1$, $(2048-512) = 1536 \cdot \ln 2 \cdot (n+1) \cdot n^{2c}$ calls are needed. Each new experiment requires 2 modular exponentiations for each of m, m'. In total, and considering that the oracle can distinguish with the same probability between m_1 and random m' or m_2 and random m', we have that the (average) total number of oracle calls is $P_3 = \frac{1}{2}(512+2048+480+1536) \cdot \ln 2 \cdot (n+1) \cdot n^{2c} = 2288 \cdot \ln 2 \cdot (n+1) \cdot n^{2c}$ while the (average) total number of exponentiations is $D_3 = 4576 \cdot \ln 2 \cdot (n+1) \cdot n^{2c}$.

Thus the reduction requires, on the average, a total of $P_0 = P_1 + P_2 + P_3 = (5568 + \epsilon^{-1}) \cdot \ln 2 \cdot (n+1) \cdot n^{2c} \approx (3859 + \epsilon^{-1}) \cdot (n+1) \cdot n^{2c}$ oracle calls and $D_0 = 2 \cdot P_0$ exponentiations for solving the decision Diffie-Hellman problem, given an ElGamal oracle. We note that the reductions can be made more efficient (e.g., by requiring that the error of an estimate is $\frac{1}{8 \cdot n^c} - \frac{1}{128 \cdot n^c}$ instead of $\frac{1}{16 n^c}$), but the above numbers are meant to simplify calculations. In general we would have $P_0 = (C + \epsilon^{-1}) \cdot (n+1) \cdot n^{2c}$ oracle calls, for $C \leq 3859$, and $D_0 = 2 \cdot P_0$ modular exponentiations.

6 Security extensions

We now extend the basic scheme to provide enhanced security. Our goal is non-malleability under chosen ciphertext attacks; this is achieved using non-malleable non-interactive zero knowledge proofs of knowledge of the plaintext under the random oracle model (namely, message-awareness is employed). The notion of *chosen ciphertext security* was first defined and implemented for public keys in [NY90]; a more generalized (adaptive) attack was formalized in [RS92]. Informally, it states that an active adversary does not obtain any advantage in breaking the system by asking for decryptions of arbitrarily chosen ciphertexts.

Non-malleability, first defined in [DDN91], is a security notion stronger than semantic security. Informally, it requires that it is infeasible, given a ciphertext, to create a *different* ciphertext such that their plaintexts are related. The difference may be simply the claim that the ciphertext came from party B instead of party A (and indeed self-protecting of a party by the "use of its unique name" is the basic motivation for non-malleability). Thus, for non-malleability to hold, the least requirement is that of unique names. In practice, each party should be allowed to choose a unique (although not necessarily certified) name.

Non-malleability is an extension of semantic security in that it considers security and self-protection of senders in the context of a network of users, and not simply between one sender and one receiver. For example, consider a chosen-ciphertext secure scheme for which it has been proven that the party which constructed the encryption is "aware" of what he is encrypting ("message awareness"). But this does not imply that a *third* party is also aware of the plaintext. Thus, in a network setting, it may be the case that a man-in-the-middle (i.e., an adversary other than the original sender) is not aware of the plaintext. For a concrete example consider the scheme of [Dam91], where the encryption of m is $E(m) = g^u, y^u \cdot m, Y^u$, where Y is a public value. Under some assumptions this scheme is (semantically) secure against ("lunch-time," [NY90]) chosen ciphertext attacks [Dam91], but it is easy to see that a man-in-the-middle can, given $E(m) = [A, B, C]$, produce $E(m') = [A \cdot g^v, B \cdot y^v/t, C \cdot Y^v]$, a randomized encryption of a related message $m' = m/t$. Thus the scheme is not non-malleable; furthermore, if the man-in-the-middle is not the party that constructed the original encryption $E(m)$ then s/he does not know the plaintext of $E(m')$ and therefore the scheme is not message-aware. Again, the reason is that in a network setting it is not only important to show a proof of knowledge, but a proof of knowledge with respect to some identity (i.e., a *non-malleable* proof of knowledge [DDN91]).

We now proceed to show the non-malleable scheme. The tool we use is message-awareness with respect to the sender's identity (i.e., the party which included the identity is also aware of the plaintext).

Finally, as we also note in the next section, proof of origin of messages is typically given by a digital signature (i.e., by a step additional to non-malleable encryption). Our scheme can also easily integrate a signature scheme together with non-malleable encryption, to provide a proof of message origin (i.e., "non-malleable signcryption").

7 Non-malleable encryption

Setup:
As discussed above (and in more detail in [DDN91]) for non-malleability it is necessary for each party to have a unique name (or some unique information that can be traced back to that party). We denote the name of party S (the sender of the encryption) with ID_S.

In what follows we show how to achieve non-malleability concisely, by resorting to proofs of knowledge of discrete logarithms. For concreteness we demonstrate the scheme using Schnorr proofs of knowledge [Sch91] but other protocols may be used instead; for example Fiat-Shamir proofs [FS87]. We present Schnorr proofs here simply for their efficiency advantages. For security we require an assumption about the unforgeability of Schnorr signatures which will be formalized in the body of the proof.

We will get the following:

Theorem 3. *Based on the decision Diffie-Hellman and assumption 1, the scheme presented below is non-malleable, in the random oracle model.*

Encryption:
The idea here is that the sender sends a zero-knowledge (ZK) proof of knowledge of the randomness used, but the ZK proof is non-malleable, i.e., it includes her/his chosen name. Using random oracles this can be done concisely by including the name in the input of the random oracle:

$$\begin{aligned} A = g^u\ , \quad B = y^u \cdot m\ , \quad F = g^{u'}\ , \\ ID_S = \text{ Name, other information }\ , \\ C = u \cdot H(g, A, B, F, ID_S) + u'\ , \\ E^n(m) = [A, B, F, C, ID_S]\ , \end{aligned}$$

where $u' \in_R Z_Q$ is randomly chosen and H is a random oracle.
Note: It is important to note that the oracle above is not used to hide information (a property investigated in [Can97] and utilized in [BR94,BR97]) but rather only as an unpredictable chellenge generator (the Fiat-Shamir construction which is used for the proofs in [PS96]). Thus the properties required of the oracle are unpredictability rather than secrecy; which means that also a "trusted beacon" can be employed.
Decryption:
The receiver obtains the ciphertext $[A, B, F, C, ID_S]$ and decrypts as in the original ElGamal scheme:

$$m = B/A^x\ ,$$

(we remind that $y = g^x$ is the receiver's public key). The receiver only accepts this encryption if the following equation is satisfied:

$$g^C = A^{H(g,A,B,F,ID_S)} \cdot F\ ,$$

otherwise it rejects and outputs *reject.*

7.1 Proof of non-malleability

Here we sketch the proof of non-malleability under adaptive chosen ciphertext attacks. The proof proceeds in two steps: (1) first we show that the semantic security is equivalent to the decision D-H assumption, i.e., the addition of the proof of knowledge does not affect semantic security; (2) then we assume that the scheme is not non-malleable and, using an assumption about Schnorr signatures, proceed to get a contradiction on its (proven) semantic security.

The semantic security of E^n is equivalent to the decision D-H :

We do not yet consider chosen ciphertext attacks; thus we can refer directly to the proofs of section 3. In the first direction (E^n is as hard as the decision D-H) the proof follows the same steps as the proof of section 3; we omit repetition for conciseness. It is straightforward to verify that the proof carries over in all parts, with the only exception being the randomization part of the decision phase (step 3(a)). Here we have to show that this step can be repeated and still results in a randomly generated encryption, while the message m' is randomly chosen if $y \not\equiv g^{ab}$ and equal to the original message m otherwise.

To this effect, the translator computes, for each ciphertext to be generated, the identity of the sender ID_S. Now we can see how the translator can generate the appropriate encryptions:

$$E^n(m) = [A = g^u,\ B = m \cdot (g^b)^{w \cdot u},\ F = g^{u'},\\ C = u \cdot H(g, A, B, F, ID_S) + u',\ ID_S\ ]\ ,$$

and

$$E^n(m') = [A' = (g^a)^t g^v = g^{(a \cdot t + v)},\ B' = m \cdot y^{w \cdot t} (g^b)^{w \cdot v},\ F' = g^{u''},\\ C' = (a \cdot t + u) \cdot H(g, A', B', F', ID_S) + u'',\ ID_S\ ]\ ,$$

where w, u, v, t, u' are random numbers, and the choice of F', u'' is discussed below.

The main issue we have to guarantee here is that the translator can actually produce $C' = (at + u) \cdot H(g, A', B', F', ID_S) + u''$, since it does not know a (meanwhile it is easy to verify that the rest of the values can be produced and are of the required form). To this effect, the translator computes $F' = (g^a)^s \cdot g^{s'}$, effectively setting $u'' = as + s'$, where s, s' are chosen at random. If we substitute this value in C' we have

$$C' = a \cdot t \cdot H(g, A', B', F', ID_S) + a \cdot s + s' = a \cdot [t \cdot H(g, A', B', F', ID_S) + s] + s'\ .$$

Now we force the output of the function H above such that the part that is multiplied by a becomes zero, and thus the translator can simply output s' (which it knows); i.e.,

$$t \cdot H(g, A', B', F', ID_S) + s \equiv 0 \pmod{Q} \iff$$

$$H(g, A', B', F', ID_S) \equiv \frac{-s}{t} \pmod{Q}\ .$$

This manipulation of H is possible due to the properties of random oracles. Specifically, when the translator calls the ElGamal oracle on the above entries it also supplies the oracle with a random oracle that has the desired output; this "tweaking" of the oracle cannot be detected since (1) the output supplied is random ($-s/t$ where both s,t are random numbers), so it still resembles the output of a random oracle, while (2) any random oracle is as good as any other random oracle, i.e., the ElGamal oracle cannot detect the difference and "change" its response. Notice that the main trick here is that the translator gets to "pick" its own oracle, since it is only performing a simulation, i.e., it does not need to "share" this oracle with another party in advance, but can generate the oracle outputs "on-the-fly" as needed (much in the way a "non-random oracle" ZK simulation proceeds for the Fiat-Shamir argument).

Now it is easy to verify that, as required, the randomization properties of both m' and $E^n(m')$ with respect to $m, E^n(m)$ are satisfied.

On the second direction the translator can simply ignore the values F, C, ID_S.

Non-malleability :

Now assume that the scheme is not non-malleable. That is, there exists an adversary A which (1) firsts adaptively queries the deciphering algorithm on ciphertexts of her choice; (2) then selects a distribution $\mathcal{M}$ of messages and is given a challenge ciphertext $c = E^n(m)$ for a message $m \in_R \mathcal{M}$; (3) and finally adaptively queries the deciphering algorithm on ciphertexts of her choice (other than c) and tries to produce a ciphertext $c' = E^n(m')$ such that a polynomial-time computable relation $R(m, m')$ holds. We will show that, under assumption 1, this contradicts the semantic security of $E^n(m)$ which was shown above.

First, observe that the triplet (which is a subset of the encryption $E^n(m)$)

$$[A = g^u, F = g^{u'}, C = u \cdot H(g, A, B, F, ID_S) + u'] ,$$

in combination with the verification of the receiver (deciphering oracle)

$$g^C = A^{H(g,A,B,F,ID_S)} \cdot F ,$$

forms a Schnorr signature on the message (g, A, B, F, ID_S), with public key $A = g^u$. This signature is existentially unforgeable against adaptive chosen plaintext attacks [PS96] under the discrete logarithm assumption *(DLA)*, which is of course a weaker assumption than the decision D-H assumption. However, the proof of this unforgeability depends on the external queries of the adversarial algorithm (in our case, queries to the decryption oracle), which have to be answered by a simulator that does not possess the decryption (private) key. We capture this difference in the following assumption.

Assumption 1. Let $\mathcal{A}$ be a p.p.t. adversary that succeeds with non-negligible probability in an existential forgery of Schnorr signatures under a public key P of its choice, when it is given some adaptively obtained information $\mathcal{I}$. Then there exists a p.p.t. adversary $\mathcal{A}'$ having access to the same information $\mathcal{I}$ that succeeds with non-negligible probability in extracting the private key corresponding to public key P.

In fact this assumption is stronger from what our proof requires, but it is phrased more generally to cover all applications of Schnorr signatures. The intuition is that if the adversary can forge a signature, then there is a modified adversarial algorithm which: (1) constructs a random oracle H and runs the adversary until she produces a forged signature (A, F, C) on "message" $M = (g, A, B, F, ID_S)$; (2) fabricates a second random oracle H' which is identical to H except for its output on M (i.e., $H(M) \neq H'(M)$) and re-runs the adversary on the same inputs; (3) outputs the private key u corresponding to the Schnorr signature, and from this computes the plaintext $m = B/y^u$. In other words, if the adversary can produce a signature, then it is within her computational power (via the modification above) to compute the private key corresponding to this signature. However, this is not a complete argument as the assumption must be proven depending on the "adaptively obtained information $\mathcal{I}$". In particular if $\mathcal{I}$ is obtained from adaptive plaintext attacks against a signing oracle then the assumption holds [PS96]. For our case we need assumption 1 to hold when $\mathcal{I}$ is the information returned from the decryption oracle.

Now, under assumption 1 the encryption is *message (plaintext) aware* with respect to the name ID_S, since the party which included that name in the encryption (i.e., the party who produced the Schnorr signature) can compute the "private key" u corresponding to the signature, and from this compute the plaintext $m = B/y^u$. Therefore the adaptive chosen ciphertext attack in step (1) above ("lunch-time attack" [NY90]) provides no information to the adversary, if she has produced the ciphertexts by herself. If she has not produced the ciphertexts herself but has instead asked for decryptions of previously seen/captured ciphertexts, then this is equivalent as having some a-priori information; this is handled by the semantic security proof (see definitions of semantic security and indistinguishability of encryptions in section 2.2).

Now also note that if the adversary changes any part of the ciphertext $c = E^n(m)$ then she needs to obtain a signature on the "message" $(g, A', B', F', ID_{S'}) \neq (g, A, B, F, ID_S)$ which she has not seen before; therefore, again from assumption 1, she is required to know (or be able to efficiently compute) v (where $A' = g^v$). Thus for any *modified* ciphertexts submitted by the adversary to the deciphering oracle the adversary already knows v and therefore the plaintext; thus the adaptive ciphertext attack in step (3) above provides no additional information to the adversary, since she is not allowed to submit the same ciphertext that she has been challenged with in step (2). Therefore we can relax the attack model to a no-message attack, under which (as proven above) the scheme is semantically secure.

To complete the proof observe that if the adversary manages to create a ciphertext $c' = E^n(m') = [A', B', F', C', ID_{S'}] \neq [A, B, F, C, ID_S] = E^n(m) = c$ such that a poly-time computable relation $R(m, m')$ holds, then the adversary has effectively produced a Schnorr signature on the message $(g, A', B', F', ID_{S'}) \neq (g, A, B, F, ID_S)$ and, again from assumption 1 (and since she has not seen this signature before, as is required in step (3) of the attack model above), she must know the discrete logarithm of A' base g (i.e., the v for which $A' = g^v$),

and therefore she must be able to obtain m'. But this means that the adversary knows some information about m, since she knows m' and the polynomial time computable relation $R(m, m')$; this contradicts the semantic security of the scheme. QED

Note: In practical encryption applications we would like a transmitted message to be both authenticated and secret. In such a setting non-malleability is not by itself sufficient, since it does not incorporate a signature of the sender: in effect the sender only states a name and binds the encryption to that name, but any other party could bind an encryption to the same name (i.e., impersonate the sender); therefore the transmission is not authenticated.[3] A digital signature is thus still required for authentication of a sent message for strong origin authentication; alternatively a combination of encryption and signature can be used, to create a "signcryption" scheme [Zhe97] in which the encryption part is non-malleable (in our scheme a Schnorr signature can be added smoothly).

Acknowledgements

We would like to thank Victor Shoup for pointing out the need for assumption 1; Berry Schoenmakers for pointing out an inconsistency in an earlier version; and Yair Frankel for helpful discussions.

References

[Bea96] D. Beaver. Plausible deniability. In *Advances in Cryptology — PraguoCrypt '96 Proceedings*, Prague, Czech Republic, 1996.

[Bea97] D. Beaver. Plug and play cryptography. In *Advances in Cryptology — CRYPTO '97 Proceedings, LLNCS 1294*, Santa Barbara, CA, August 17–21 1997. Springer-Verlag.

[BGM93] E. F. Brickell, D. Gordon, and K. S. McCurley. Fast exponentiation with precomputation. In *Advances in Cryptology — Eurocrypt '92, Proceedings (Lecture Notes in Computer Science 658)*. Springer-Verlag, 1993.

[BR94] M. Bellare and P. Rogaway. Optimal assymetric encryption— how to encrypt with RSA. In A. De Santis, editor, *Advances in Cryptology, Proc. of Eurocrypt '94, (Lecture notes in Computer Science Volume 950)*, Perugia, Italy, May 9–12 1994. Springer-Verlag.

[BR97] M. Bellare and P. Rogaway. Minimizing the use of random oracles in authenticated encryption schemes. In *ISICS '97*, 1997.

[Can97] R. Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In B. Kaliski, editor, *Advances in Cryptology — CRYPTO '97 Proceedings, LLNCS 1294*, pages 455–469, Santa Barbara, CA, August 17–21 1997. Springer-Verlag.

[CS98] R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack, 1998. Preprint. Available at http://www.cs.wisc.edu/ shoup/papers/.

[3] We divert from [BR97] who define "authenticated encryption" as "plaintext awareness + semantic security," or intuitively knowledge of the plaintext; here we consider authentication of both the sender and the message, as required in a network setting.

[Dam91] I. B. Damgård. Towards practical public key systems against chosen ciphertext attacks. In J. Feigenbaum, editor, *Advances in Cryptology, Proc. of Crypto '91 (Lecture Notes in Computer Science 576)*, pages 445–456. Springer-Verlag, 1991.

[DDN91] O. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. In *Proceedings of the 23rd Symposium on Theory of Computing, ACM STOC*, 1991.

[ElG85] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inform. Theory*, 31:469–472, 1985.

[ElG98] T. ElGamal, January 1998. Personal communication.

[FS87] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. Odlyzko, editor, *Advances in Cryptology, Proc. of Crypto '86 (Lecture Notes in Computer Science 263)*, pages 186–194. Springer-Verlag, 1987. Santa Barbara, CA, August 11–15.

[FTY96] Y. Frankel, Y. Tsiounis, and M. Yung. Indirect discourse proofs: achieving fair off-line e-cash. In *Advances in Cryptology, Proc. of Asiacrypt '96 (Lecture Notes in Computer Science 1163)*, pages 286–300, Kyongju, South Korea, November 3–7 1996. Springer-Verlag. http://yiannis.home.ml.org/pubs.html

[GM84] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, April 1984.

[Gol89] O. Goldreich. Foundations of cryptography, 1989. Class notes. Available at http://www.wisdom.weizmann.ac.il/people/homepages/oded/ln89.html.

[Gol93] O. Goldreich. A uniform-complexity treatment of encryption and zero-knowledge. *Journal of Cryptology*, 6(1):21–53, 1993.

[MRS88] S. Micali, C. Rackoff, and B. Sloan. The notion of security for probabilistic cryptosystems. *SIAM Journal of Computing*, 17:412–426, 1988.

[NR97] M. Naor and O. Reingold. On the construction of pseudo-random permutations: Luby-Rackoff revisited. In *38th Annual Symp. on Foundations of Computer Science (FOCS)*, 1997.

[NY90] M. Naor and M. Yung. Public-key cryptosytems provably secure against chosen ciphertext attack. In *Proceedings of the twenty second annual ACM Symp. Theory of Computing, STOC*, pages 427–437, May 14–16, 1990.

[PS96] D. Pointcheval and J. Stern. Security proofs for signature schemes. In U. Maurer, editor, *Advances in Cryptology– Eurocrypt '96*, pages 387–398, Zaragoza, Spain, May 11–16, 1996. Springer-Verlag.

[RS92] C. Rackoff and D. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Advances in Cryptology–Crypto '91 (LLNCS 576)*, pages 433–444, Santa Barbara, CA, 1992. Springer-Verlag.

[Sch91] C. P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.

[SS98] K. Sakurai and H. Shizuya. Relationships among the computational powers of breaking discrete log cryptosystems. *Journal of Cryptology*, 1998. To appear.

[TY97] Y. Tsiounis and M. Yung. The semantic security of El Gamal encryption is equivalent to the decision Diffie-Hellman. Technical Report, GTE Laboratories Inc., May 1997.

[Zhe97] Y. Zheng. Digital signcryption or how to achieve cost(signature & encryption) << cost(signature) + cost(encryption). In B. Kaliski, editor, *Advances in Cryptology–Crypto '97 (Lecture Notes in Computer Science 1294)*, pages 165–179, Santa Barbara, CA, August 17–21 1997. Springer-Verlag.

[ZS93] Y. Zheng and J. Seberry. Immunizing public key cryptosystems against chosen ciphertext attacks. *IEEE Journal on Selected Areas in Communications*, 11(5):715–724, June 1993.