A new security proof for Damgård's ElGamal

Kristian Gjøsteen

December 20, 2004

Abstract

We provide a new security proof for a variant of ElGamal proposed by Damgård, showing that it is secure against non-adaptive chosen ciphertext. Unlike previous security proofs for this cryptosystem, which rely on somewhat problematic assumptions, our computational problem is similar to accepted problems such the Gap and Decision Diffie-Hellman problems.

1 Introduction

Damgård [7] defined a variant of the ElGamal public key cryptosystem, and proposed a proof of security against non-adaptive chosen ciphertext attacks based on an assumption now commonly known as the knowledge-of-exponent assumption [10, 2, 3]. Unfortunately, the knowledge-of-exponent assumption is a somewhat strange and impractical assumption, and it would be better if we could do without it.

In [3], one proof is given for Damgård's cryptosystem and one for the so-called CS-lite scheme [6]. Since the security of CS-lite clearly follows from the security of Damgård's scheme, the universal-2 hash proof system in CS-lite indicates exactly what is needed to prove Damgård's scheme secure without the knowledge-of-exponent assumption.

We therefore propose a new problem similar to conventional problems such as the Gap Diffie-Hellman problem. If the new problem is hard, we are able to show that Damgård's cryptosystem is semantically secure against a non-adaptive chosen ciphertext attack.

Our new assumption is defined in Section 2 and the new security proof is in Section 3. But first we need to recall the definition of a smooth hash proof systems in Section 1.1.

We actually define a family of cryptosystems. Asymptotically, the fastest instantiation is probably Damgård's cryptosystem instantiated with elliptic curves. For concrete security parameters, the scheme based on the problem described in Section 2.3 may be faster.

While all of the instantiations described are homomorphic, the scheme based on the Decision Composite Residuosity problem is additively homomorphic (multiplying ciphertexts corresponds to adding messages), which is a very useful property.

1.1 Smooth projective hash proof systems

We refer the reader to [5] for more information about projective hash families and hash proof systems.

Let G be a set and let H be a subset of G. We say that a set W is a *witness set* for H if there is an easily computable bijection $\rho : W \to H$. This bijection allows one to prove that an element $x \in G$ really is in H by presenting an element $w \in W$ such that $\rho(w) = x$. This obviously assumes that it is easy to recognise elements of W.

For two sets S, S', denote by $\mathrm{Map}(S, S')$ the set of maps from S to S'. Let L be a finite, abelian group. We are interested in looking at maps from G to L. There is a natural map $\mathrm{Map}(G, L) \to \mathrm{Map}(H, L)$ given by restriction. From ρ we get a bijection $\rho^* : \mathrm{Map}(H, L) \to \mathrm{Map}(W, L)$. We also denote the natural map $\mathrm{Map}(G, L) \to \mathrm{Map}(W, L)$ by ρ^*.

A *projective hash family* is a tuple $(G, H, L, L', W, \rho, M)$, where G is a set, H is a subset of G, L is a group, L' is a subgroup of L, W is a witness set for H with isomorphism ρ, M is a subset of $\mathrm{Map}(G, L)$.

Let x be sampled randomly from $G \setminus H$, let f be sampled uniformly at random from M, and let y be sampled uniformly at random from L'. We say that the projective hash family is (t, ϵ)-*smooth* if for any algorithm A with run-time less than t that accepts as input triples from $(G \setminus H) \times \mathrm{Map}(W, L) \times L$ and outputs a bit b, we have that

$$\mathrm{Adv}_A^{SHF(G,H,L,L',W,\rho,M)} = |\Pr[A(x, \rho^*(f), f(x)) = 1] - \Pr[A(x, \rho^*(f), f(x)y)]| < \epsilon.$$

The algorithm A is given access to an oracle that evaluates f at any point in H.

We note that the definitions given in [5] are statistical, but our definition is computational. For statistical results, we omit the time bound t.

A *hash proof system* P for (G, H) is a projective hash family $(G, H, L, L', W, \rho, M)$ along with efficient algorithms for sampling M and W, and evaluating the functions $f \in M$ and $\rho^*(f) \in \mathrm{Map}(W, L)$ (given descriptions of the functions).

The sampling algorithms sample from some distribution that is δ-close to the uniform distribution. We say that δ is the hash proof system's *approximation error*.

We shall provide constructions for smooth hash proof systems in the next section.

2 Gap subgroup membership problems

Let G be a finite, abelian group. Let H be a non-trivial, proper subgroup of G, and suppose that there is a subgroup J of G such that $H \cap J$ is trivial and $HJ = G$. Then $H \times J \simeq G$.

There is an isomorphism $H \times J \to G$ given by $(x, y) \mapsto xy$. Let $\sigma : G \to H \times J$ be the inverse of this isomorphism. The *splitting* problem for (G, H, J) is to compute this map for a given random $z \in G$.

The *subgroup membership* problem for (G, H, J) is to decide if $(x, y) \in G \times G$ is the splitting of $z \in G$, given that z is sampled uniformly from G and (x, y) is either equal to $\sigma(z)$ or sampled uniformly from $(G \setminus H) \times J$, subject to $xy = z$. Obviously, we can

ignore z and y, and the problem is to decide if a *challenge element* $x \in G$ is sampled uniformly from H or from $G \setminus H$. (A stronger assumption [9] samples (x, y) uniformly either from $H \times J$ or from $(G \times G) \setminus (H \times J)$.)

The *gap splitting* problem for (G, H, J) is the same as the splitting problem for, but any solver is given access to an oracle that for any element $x \in G$ returns 1 if $x \in H$, and 0 otherwise. This oracle is called the *gap* oracle.

We propose a new problem, *gap subgroup membership* problem. First, the adversary receives (G, H, J), and he has free access to a gap oracle for (G, H, J). After some computation, he will request the challenge element $x \in G$. He must then decide if $x \in H$ or if $x \in G \setminus H$. But after he receives the challenge, he no longer has access to gap oracle. (If he did, the problem would be trivial.)

We let $\mathrm{Adv}_A^{GSM(G,H,J)}$ denote the advantage of the adversary A against the gap subgroup membership problem.

Note that the gap splitting problem and the gap subgroup membership problem are interactive problems. Given the splitting and subgroup membership problems, the gap problems follow immediately by adding the appropriate oracles.

2.1 Diffie-Hellman

Let L be a cyclic group of order p generated by g. We write the group multiplicatively.

The *discrete logarithm to base* g is the group isomorphism $\log_g : L \to \mathbb{Z}_p$ given by $\log_g g = 1$.

The *Computational Diffie-Hellman (CDH)* problem [8] in L is, for random $x, y \in L$, find z such that $\log_g z = \log_g x \log_g y$.

The *Decision Diffie-Hellman (DDH)* problem [4] in L is, for random $x, y \in L$ and z, decide if z is random or if $\log_g z = \log_g x \log_g y$.

The *Strong Diffie-Hellman (SDH)* problem [1] for G and x is the same as the GDH problem, except that the problem solver is given access to a DDH oracle (on input of $(y', z') \in L \times L$, it decides if $\log_g z' = \log_g x \log_g y'$).

Let $G = L \times L$, let H be the subgroup generated by (g, x) (where x is a random, non-trivial element), and let J be the subgroup generated by $(1, g)$. Then CDH is the splitting problem (G, H, J), DDH is the subgroup membership problem (G, H, J), and SDH is the gap splitting problem (G, H, J).

Remark. To get a proper equivalence of problems, we need to consider the group H as sampled from some distribution.

Remark. There is a stronger version of SDH known as the *Gap Diffie-Hellman* problem [11], where the gap oracle answers queries for arbitrary x.

Hash proof system We shall describe a projective hash family $(G, H, L, L', W, \rho, M)$.

Let $L' = L$. Let (g_1, g_2) be a generator for H. Let $W = \mathbb{Z}_p$, and $\rho(w) = (g_1, g_2)^w$. Finally, let M be the set homomorphisms $G \to L$ of the form $(x, y) \mapsto x^{k_1} y^{k_2}$, where $(k_1, k_2) \in \mathbb{Z}_p \times \mathbb{Z}_p$.

We note that for any $f(x,y) = x^{k_1}y^{k_2}$ in M, the element $f(g_1, g_2) \in L$ is sufficient to allow efficient computation of $\rho^*(f)$, since $\rho^*(f)(w) = (g_1^w)^{k_1}(g_2^w)^{k_2} = (g_1^{k_1}g_2^{k_2})^w = f(g_1, g_2)^w$.

This family is statistically 0-smooth [5]. Any sampling can be done uniformly at random by sampling uniformly from $\{0, \ldots, p-1\}$, so the approximation error is 0.

2.2 Composite Residuosity problem

Let $n = pq$ be an RSA modulus such that p and q are safe primes and $\gcd(n, \phi(n)) = 1$. Let G be the subgroup of quadratic residues in $\mathbb{Z}_{n^2}^*$, let H be the subgroup isomorphic to the subgroup of quadratic residues in $\mathbb{Z}_n^*$, and let J be the subgroup generated by residue class containing $1 + n$.

We get a splitting problem (G, H, J) called the *Computational Composite Residuosity (CCR)* problem, a subgroup membership problem (G, H, J) called the *Decision Composite Residuosity (DCR)* problem. These were first proposed by Paillier [12].

Hash proof system We shall describe a projective hash family $(G, H, L, L', W, \rho, M)$.

Let g be a generator for H. Let $L = G$, $L' = J$, $M = \mathrm{Hom}(G, G)$, and $W = \mathbb{Z}_{\phi(n)}$. Let $[w] \in \mathbb{Z}_n$ be the residue class represented by $w \in \mathbb{Z}$. Then $\rho([w]) = g^w$.

Any $f \in \mathrm{Hom}(G, G)$ is of the form $x \mapsto x^k$. Therefore, the elements in the set $\{0, \ldots, n\phi(n)/4 - 1\}$ are useful descriptions of the homomorphisms. Again, $f(g)$ is a useful description of $\rho * (f)$, since $\rho * (f)([w]) = f(g^w) = (g^k)^w = f(g)^w$.

Assume $p < q$. As was shown in [5], this hash-family is $1/p$-smooth.

Unless we know p and q, we cannot sample elements uniformly from $\mathrm{Hom}(G, G) \simeq \mathbb{Z}_{\phi(n)}$, but we can sample $4/(p-1)$-close to uniformly by sampling uniformly from the set $\{0, \ldots, \lfloor n/4 \rfloor - 1\}$.

Given only n, we cannot sample uniformly from W, but by sampling uniformly from $\{0, \ldots, \lfloor n/4 \rfloor - 1\}$, we get a sample distribution on W that is $4/p$-close to uniform. Therefore, the approximation error is $4/p$.

2.3 Symmetric subgroup membership problems

Let $n = pq$ be an RSA modulus such that $p' = 2n + 1$ is a prime. Let G be the subgroup of quadratic residues in $\mathbb{F}_{p'}^*$, let H be the subgroup of order p and J be the subgroup of order q.

Alternatively, let a, b, c, d be primes such that $p = 2ab + 1$ and $q = 2cd + 1$ are prime. Set $n = pq$. Let G be the subgroup of $\mathbb{Z}_n^*$ with Jacobi symbol 1, let H be the subgroup of order $2ac$ and J be the subgroup of order bd.

These symmetric subgroup membership problems are further discussed in [9].

Projective hash family We shall describe a projective hash family $(G, H, L, L', W, \rho, M)$.

Let g be a generator for H. Let $L = G = L'$, $W = \mathbb{Z}_{|H|}$, $\rho([w]) = g^w$, where $[w]$ is the residue class represented by $w \in \mathbb{Z}$, and $M = \mathrm{Hom}(G, G)$.

Any $f \in \mathrm{Hom}(G,G)$ is of the form $x \mapsto x^k$. Therefore, the elements in the set $\{0,\ldots,|G|-1\}$ are useful descriptions of the homomorphisms. Again, $f(g)$ is a useful description of $\rho^*(f)$, since $\rho^*(f)(w) = f(g^w) = (g^k)^w = f(g)^w$.

Let ℓ be the smallest prime dividing $|J|$. As was shown in [9], this hash-family is $(t, 1/\ell + \epsilon')$-smooth, where ϵ' is the advantage of any algorithm with run-time at most t against the gap subgroup membership problem (G, J, H).

Unless we know $|G|$, we cannot sample elements uniformly from $\mathrm{Hom}(G,G) \simeq \mathbb{Z}_{|G|}$. In the finite field case we know $|G|$, and in ring case $n/4$ is a good approximation to $|G|$.

Given only n, we cannot sample uniformly from W. But if we know that $|H| < 2^t$, we can sample 2^{-t_0}-close to uniform by sampling from $\{0,\ldots,2^{t+t_0}-1\}$. This trick also works for $\mathrm{Hom}(G,G)$, so we can take the approximation error to be 2^{-t_0} for any reasonable $t_0 > 0$.

3 The cryptosystem

First we describe the cryptosystem based on a subgroup membership problem (G,H,J) and a hash proof system P with projective hash family (G,H,L,L',W,ρ,M). The key generation algorithm samples a function f from M using the sampling algorithms of P. The public key is $pk = (G,L,L',W,\rho,\rho^*(f))$, the private key is $sk = (G,H,L,L',f)$. The description of H given to the key generation algorithm and stored in the private key should allow the decryption algorithm to decide if an element is in H or not.

To encrypt $m \in L'$, w is sampled uniformly at random from W. The ciphertext is then $(\rho(w), \rho^*(f)(w)m)$.

To decrypt a ciphertext $(x,y) \in G \times L$, the algorithm first verifies that $x \in H$. If $x \notin H$, then failure is reported and the ciphertext is discarded. Otherwise, the message $yf(x)^{-1} \in L'$ is returned.

Theorem 1. *Let (G,H,J) be a gap subgroup membership problem and let P be a hash proof system with projective hash family (G,H,L,L',W,ρ,M) and approximation error δ. Let A be a non-adaptive chosen ciphertext adversary against the semantic security of the cryptosystem based on P. Then*

$$\mathrm{Adv}_A \leq \mathrm{Adv}_{A'}^{GSM(G,H,J)} + \mathrm{Adv}_{A''}^{SHF(G,H,L,L',W,\rho,M)} + 2\delta,$$

where the algorithms A' and A'' have essentially the same run-time as A.

Proof. We use the standard techniques of game-hopping.

Game 0 The initial game is the usual non-adaptive chosen ciphertext attack against semantic security.

Game 1 The first modification we make is to sample the element of W used for creating the challenge ciphertext from the uniform distribution, not via P's sampling algorithm. The difference in game behaviour is bounded by the approximation error δ.

Game 2 Next, instead of using $\rho^*(f)$ and w in the encryption, we sample x from the uniform distribution on H and apply f. This is a purely conceptual change, and the game behaviour does not change.

Game 3 Now we sample not from H, but from $G \setminus H$ when creating the challenge ciphertext. We claim that there is a adversary A' against the gap subgroup membership problem (G, H, J) whose advantage is equal to the change in behaviour.

The algorithm A' takes (G, H, J) as input. To simulate the key generation, the hash proof system's sampling algorithms are used to construct the public and private keys. Obviously, the description of H given to the algorithm does not contain enough information about H to decide membership. Therefore, the private key will be deficient and the decryption algorithm must be changed.

When the adversary requests decryptions, the gap oracle is used to check that the group element really is in H. If it is, f is used to decrypt the message.

When the adversary submits its messages, the algorithm requests its challenge element x and computes the challenge ciphertext is $(x, f(x)m)$. If the adversary guesses correctly, we output 1, otherwise 0.

If $x \in H$, everything proceeds as in Game 2. If $x \in G \setminus H$, everything proceeds as in Game 3.

Game 4 To prepare for the adversary against the hash proof system, we sample the function f from the uniform distribution on M. The difference in behaviour is bounded by the approximation error δ.

Game 5 In this game we compute the ciphertext as $f(x)ym$, where y is sampled uniformly from L'. We claim that there is an adversary A'' against the hash proof system whose advantage is equal to the change in behaviour.

The algorithm takes $(G, H, L, L', W, \rho, M)$, $x \in G \setminus H$, $\rho^*(f)$ and $z \in G$ as input. It constructs the public key from its input. To answer decryption queries (x', y'), it passes x' onto its evaluation oracle. If the oracle refuses to answer, then $x' \notin H$ and the ciphertext does not decrypt. If the oracle replies with z, the decryption $y'z^{-1}$ is returned.

When the adversary submits its messages, the challenge ciphertext is (x, zm). If the adversary guesses correctly, we output 1, otherwise 0.

If $z = f(x)$, then everything proceeds as in Game 4. If $z = f(x)y$, where y is a random element of L', then everything proceeds as in Game 5.

To conclude the proof, we simply note that in Game 5, the distribution of the ciphertext is independent of the message, the adversary gets no information about the message, and therefore he has no advantage. □

It is worthwhile to note that all of the instantiations of this scheme are homomorphic, so they are not secure against adaptive chosen ciphertext attacks. When instantiated

with the Decision Composite Residuosity problem, the scheme is actually additively homomorphic, a property that is very useful.

Damgård's scheme Finally, we show that Damgård's cryptosystem is really the same as our cryptosystem when it is instantiated with the Diffie-Hellman group structure.

In this case, the key generation algorithm does as follows. It selects a group L of order p with a generator g_1, samples a uniformly from $\{0, \ldots, p-1\}$ and sets $g_2 = g_1^a$. The subgroup H is generated by (g_1, g_2). The number a now contains enough information to decide subgroup membership. Then the key generation algorithm samples k_1 and k_2 uniformly from $\{0, \ldots, p-1\}$ and computes $s = g_1^{k_1} g_2^{k_2}$. The public key is then (L, g_1, g_2, s), the private key is (L, a, k_1, k_2).

The encryption algorithm samples w uniformly from $\{0, \ldots, p-1\}$, computes the ciphertext as $(x_1, x_2, y) = (g_1^w, g_2^w, s^w m)$.

The decryption algorithm checks that $(x_1, x_2) \in H$ by checking that $x_1^a = x_2$. If it is, it returns the message $y x_1^{-k_1} x_2^{-k_2}$.

In Damgård's original scheme, the public key also consists of (L, g_1, g_2, s), but s is computed as g_1^b for a random b. It is quite clear that our key generation algorithm yields the exact same key distribution as Damgård's scheme. Indeed, if $b \equiv k_1 + a k_2 \pmod{p}$ the public keys would be equal.

References

[1] M. Abdalla, M. Bellare, and P. Rogaway. DHIES: An encryption scheme based on the Diffie-Hellman problem, 2001. `http://www.cs.ucsd.edu/users/mihir/papers/dhies.html`.

[2] Mihir Bellare and Adriana Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. Cryptology ePrint Archive, Report 2004/008, 2004. `http://eprint.iacr.org/`.

[3] Mihir Bellare and Adriana Palacio. Towards plaintext-aware public-key encryption without random oracles. Cryptology ePrint Archive, Report 2004/221, 2004. `http://eprint.iacr.org/`.

[4] D. Boneh. The Decision Diffie-Hellman problem. In *Proceedings of the Third Algorithmic Number Theory Symposium*, volume 1423 of *LNCS*, pages 48–63. Springer-Verlag, 1998.

[5] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *Proceedings of EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64. Springer-Verlag, 2002.

[6] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.

[7] Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In Joan Feigenbaum, editor, *Proceedings of CRYPTO '91*, volume 576 of *LNCS*, pages 445–456. Springer-Verlag, 1992.

[8] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22:644–654, 1976.

[9] Kristian Gjøsteen. Symmetric subgroup membership problems. In *Proceedings of Public Key Cryptography 2005*, LNCS. Springer-Verlag. To appear.

[10] Satoshi Hada and Toshiaki Tanaka. On the existence of 3-Round zero-knowledge protocols. In Hugo Krawczyk, editor, *Proceedings of CRYPTO '98*, volume 1462 of *LNCS*, pages 408–423. Springer-Verlag, 1998.

[11] Tatsuaki Okamoto and David Pointcheval. The gap-problems: A new class of problems for the security of cryptographic schemes. In Kwangjo Kim, editor, *Proceedings of Public Key Cryptography 2001*, volume 1992 of *LNCS*, pages 104–118. Springer-Verlag, 2001.

[12] P. Paillier. Public-key cryptosystems based on composite degree residue classes. In Jacques Stern, editor, *Proceedings of EUROCRYPT '99*, volume 1592 of *LNCS*, pages 223–238. Springer-Verlag, 1999.