Security of Signed ElGamal Encryption

Claus Peter Schnorr[1] and Markus Jakobsson[2]

[1] Fachbereich Mathematik/Informatik, Universität Frankfurt, PSF 111932, D-60054 Frankfurt am Main, Germany. schnorr@cs.uni-frankfurt.de
[2] Information Sciences Laboratory, Bell Laboratories Murray Hill, New Jersey 07974. markusj@research.bell-labs.com

Abstract. Assuming a cryptographically strong cyclic group G of prime order q and a random hash function H, we show that ElGamal encryption with an added Schnorr signature is secure against the *adaptive chosen ciphertext attack*, in which an attacker can freely use a decryption oracle except for the target ciphertext. We also prove security against the novel *one-more-decyption attack*. Our security proofs are in a new model, corresponding to a combination of two previously introduced models, the Random Oracle model and the Generic model. The security extends to the distributed threshold version of the scheme. Moreover, we propose a very practical scheme for private information retrieval that is based on blind decryption of ElGamal ciphertexts.

1 Introduction and Summary

We analyse a very practical public key cryptosystem in terms of its security against the strong *adaptive chosen ciphertext attack* (CCA) of [RS92], in which an attacker can access a decryption oracle on arbitrary ciphertexts (except for the target ciphertext.) Let a *signed ElGamal encryption* of a message be an ElGamal ciphertext together with a Schnorr signature of that ciphertext — the public signature key is given by the ElGamal ciphertext. We prove that this scheme is secure against generic attacks where both the group G and the random hash function H are black boxes.

The traditional versus the new security model. Assuming a strong cyclic group G and a random hash function H we prove tight bounds on the success probability of a generic attacker performing some t generic steps. Our approach has practical consequences. It yields very practical cryptographic schemes that are provably secure in a reasonable, new security model, the *random oracle and generic model* (ROM+GM). The ROM goes back to Fiat and Shamir [FS86] and has been further enhanced by Bellare and Rogaway [BR93], while the generic model (GM) goes back to Nechaev [Ne94] and Shoup [Sh97]. We introduce the combination of these two models, the result of which seems to cover all practical attacks at hand. Namely, security in ROM+GM allows a separation of potential weaknesses of the group G, the hash function H and the cryptographic protocols using G and H. It allows a modular replacement of weak hash functions or groups without forcing changes to the cryptographic protocols. Whereas

the security guarantees of most efficient groups and hash functions are merely heuristics based on the absence of known attacks, we obtain tight bounds on the success of arbitrary generic attacks. While we do not have to rely on any unproven assumption, it is the case that our security guarantees hinge on the existence of strong hash functions H and groups G for which the combination (G, H) has no weaknesses. On the other hand, we do *not* assume that the discrete logarithm (DL) problem or to the Diffie-Hellman problem is hard — our security proof contains a hardness proof of the DL-problem in the generic model.

The new ROM+GM is a powerful tool for proving security against interactive attacks. In this paper we merely consider encryption. For security in ROM+GM of Schnorr signatures — in particular security of blind signatures against the one-more signature forgery — see [SJ99]. Recently, it has been shown [Sc00] that the generation of secret DL-keys from short random seeds through a strong hash function is secure in GM.

Notions of security. Let G be a cyclic group of prime order q with generator g, and let $\mathbf{Z}_q$ be the field of integers modulo q. A Diffie-Hellman key pair consists of a random secret key $x \in \mathbf{Z}_q$ and the corresponding public key $h = g^x \in G$. Diffie-Hellman keys give rise to many cryptographic schemes, for example ElGamal *encryption* [E85]. An ElGamal ciphertext of message $m \in G$ is a pair $(g^r, mh^r) \in G^2$ for random $r \in \mathbf{Z}_q$. ElGamal encryption is *indistinguishable* [GM84] — it is secure against a passive, merely eavesdroping adversary. Formally, an attacker, given distinct messages m_0, m_1 and a corresponding target ciphertext cip_b for random $b \in \{0, 1\}$, cannot guess b better than with probability $\frac{1}{2}$. However, ElGamal encryption is completely insecure against various active attacks, where a decryption oracle can be used under appropriate conditions.

A powerful active attack is the CCA-attack of Rackoff and Simon [RS92]. CCA-security means indistinguishability against an adversary that can freely use a decryption oracle except for the target ciphertext. Dolev, Dwork and Naor [DDN91] propose another notion of security against active attacks, called *non-malleability*. Here the adversary — which is given a decryption oracle — tries to create another ciphertext that is related in an interesting way to the target ciphertext. Non-malleability and CCA-security have been shown to be equivalent [DDN98].

Previous work. The public key encryption schemes of Shoup, Gennaro [SG98], Cramer, Shoup [CS98], Abdalla, Bellare, Rogaway [ABR98], Fujisaki, Okamoto [FO99], Shoup [Sh00] and Zheng, Seberry [ZS92] all extend variants of ElGamal encryption by an added signature or tag. This idea first appears in [ZS92] without a security proof. CCA-security has been proved in [SG98, CS98, ABR98, FO99, Sh00]. The schemes in [SG98, CS98, ABR98, Sh00] either use an involved tag construction or key generation to simplify the reduction to the discrete log or to the Diffie-Hellman problem, the tag in [ABR98] uses symmetric encryption. We consider the very practical, signed extension of ElGamal encryption, which was independently proposed by Tsiounis and Yung [TY98] and Jakobsson [J98]. Herein, an ElGamal ciphertext (g^r, mh^r) is completed by a Schnorr *signature* [Sc91] providing a proof of knowledge of

the plaintext m and of the secret r — the public signature key g^r is given by the ciphertext. CCA-security of this *signed ElGamal encryption* has been shown in [TY98] under the assumption that the signer really "knowns" the secret signature key r. That assumption holds in the ROM if there is only a logarithmic number of interactions with the decryption oracle.[1]

Our results. We "validate" the [J98,TY98]-assumption that the signer really "knows" the secret key r in the ROM+GM. We give a plaintext extractor, and we prove security against a generic CCA-attacker performing some number $t = o(\sqrt{q})$ of interactions and generic group steps. A CCA-attacker can freely use a decryption oracle except for the target ciphertext. We show that a generic CCA-attacker using t generic steps, and given distinct messages m_0, m_1, a target ciphertext cip_b for random $b \in_R \{0,1\}$, cannot predict b with probability better than $\frac{1}{2} + t^2/q$. This probability is over the random hash function H, the random public encryption key h, the coin tosses of the encipherer, and the random bit b. This bound is almost tight, as a generic attacker, given the public key h, can compute the secret decryption key with probability $\binom{t}{2}/q$ in t generic steps. This result improves the known security guarantees for signed ElGamal encryption. Moreover, our security proofs extend to a straightforward distributed threshold version of signed ElGamal encryption, see [SG98] for the threshold setting.

Furthermore, we introduce the *one-more decryption attack* and we show that signed ElGamal encryption is secure against this attack. In the one-more decryption attack the adversary attempts to partially decrypt $\ell + 1$ ciphertexts by asking a decryption oracle some ℓ times. The new attack is not covered by the adaptive chosen ciphertext attack, as the latter relates to a single target ciphertext. Interestingly, security against the one-more attack follows from plaintext awareness (PA) as defined in [BR94]. Proving PA is the core of the proof of Theorem 1 and 2. [2] For motivation of the one-more decryption attack, we propose a practical scheme for private information retrieval. It is based on blind decryption and security against the *random one-more attack* — which is a weak version of the one-more decryption attack.

Generalized (signed) ElGamal encryption. Finally, we propose a more general variant of (signed) ElGamal encryption with two major advantages. *Firstly*, for long messages our generalized encryption is very fast and its data expansion rate approaches 1. *Secondly*, the generalized encryption does not require messages to be encoded into the group generated by the public key h.[3] Let the message space

[1] The FFS-extractor of Feige-Fiat-Shamir, in the oracle replay mode of Pointcheval and Stern [PS96], extracts the secret signature key from signed ElGamal encryptions. The FFS-extractor has a constant delay factor, and thus can in polynomial time at most be iterated a logarithmical number of times.

[2] It seems that PA is the most important security notion for encryption. E.g., [BDPR98] show that PA and IND-CPA imply CCA-security while the converse does not hold. PA requires the ROM, security proofs without assuming the ROM do not prove PA.

[3] Encoding of arbitrary bit sequences into sequences of group elements is easy for particular groups such as $\mathbf{Z}_q^*$ that correspond to an interval of integers. For general groups, even for subgroups of $\mathbf{Z}_N^*$ or subgroups of elliptic curves, an encoding into

be an arbitrary additive group M, e.g., $M = \mathbf{Z}_q^n$ for some n. Let a generalized ElGamal ciphertext be a pair $(g^r, m + H_M(h^r))$ for random $r \in \mathbf{Z}_q$, where $H_M : G \to M$ is a random hash function. We then add a Schnorr signature (using the public signature key g^r) to the ciphertext $(g^r, m + H_M(h^r)) \in G \times M$. This signed generalized ElGamal encryption has provably the same security as signed ElGamal encryption, without any further assumptions.

The structure of the paper. In Section 2, we introduce the generic model for interactive algorithms that use a hash oracle and an oracle for decryption. We propose a setup for the GM that slightly differs from the [Sh97] proposal in that we do not assume a random binary encoding of group elements. We examplify the difference of the two setups for the baby-step-giant-step DL-algorithm. While our generic algorithms do not allow for efficient sorting of group elements this does not affect the number of generic steps as equality tests of group elements are free of charge.

In Section 3, we review signed ElGamal encryption, which is based on the original ElGamal encryption. Moreover, we generalize the original and the signed ElGamal encryption. Then we introduce the main tools for proving security in the GM. We show in Lemma 1 and 2 that a collision-free, non-interactive generic attacker $\mathcal{A}$ gets no information on the secret random data — the secret key, the random number r , etc. — except that $\mathcal{A}$ observes the absence of collisions. Lemma 1 bounds the probability for non-trivial collisions. This bound also covers the leakage of secret information through the absence of collisions.

Section 4 presents the proof of CCA-security of signed ElGamal encryption in the ROM+GM. It gives a generic extractor that extracts the signature key $\bar{r} = \log_g \bar{h}$ from a signed ElGamal ciphertext $(\bar{h}, \bar{f}, c, z)$, produced by the attacker. We also prove security against the one-more decryption attack. We motivate this novel attack by interesting services for trading encrypted data.

2 The Random Oracle and the Generic Model

The Random Oracle Model (ROM). Let G be a group of prime order q with generator g, a range M of messages, and let $\mathbf{Z}_q$ denote the field of integers modulo q. Let H be an *ideal* hash function with range $\mathbf{Z}_q$, modelled as an oracle that given an input (query) in $G \times M$, outputs a random number in $\mathbf{Z}_q$. Formally, H is a random function $H : G \times M \to \mathbf{Z}_q$ chosen at random over all functions of that type with uniform probability distribution. There is an ongoing debate on whether the assumption of a random hash function is realistic or too generous. The problem is that random functions can in principle not be implemented by public algorithms. Canetti, Goldreich, Halevi [CGH98] present an artificial "counter-example" that is provably secure in the ROM but which cannot be implemented in a secure way replacing the random oracle by a computable

group elements is impractical. Known extensions of ElGamal encryption — see e.g., [MOV] section 8.26 — do not solve this encoding problem.

function family.[4] Nevertheless, the security achievable in the ROM seems to in practice eliminate all attacks at hand.

The Generic Model (GM). Generic algorithms for G do not use the binary encodings of the group elements, as they access group elements only for group operations and equality tests. NECHAEV [Ne94] proves that the discrete logarithm problem is hard in such a model. The generic model of algorithms was further elaborated on by SHOUP [Sh97]. We present the Shoup model in a slightly different setup[5] and we extend it to algorithms that interact with a decryption oracle. Encryptions are for the private/public key pair (x, h), where x is random in $\mathbf{Z}_q$ and $h = g^x$. We describe the extended generic model in detail, first focusing on non-interactive algorithms and thereafter on algorithms interacting with oracles for hashing and decryption.

The *data of a generic algorithm* is partitioned into group elements in G and non-group data. The *generic steps* for group elements are multivariate exponentiations:

- mex: $\mathbf{Z}_q^d \times G^d \to G,\ (a_1, ..., a_d, g_1, ..., g_d) \mapsto \prod_i g_i^{a_i}$ with $d \geq 0$.

The cases $d = 2, a_1 = 1, a_2 = \pm 1$ present multiplication/division. The case $d = 0$ presents *inputs* in G — e.g., g, h are inputs for the DL-computation.

Def. A (non-interactive) *generic algorithm* is a sequence of t generic steps[6]

- $f_1, \ldots, f_{t'} \in G$ (inputs) $1 \leq t' < t$,
- $f_i = \prod_{j=1}^{i-1} f_j^{a_j}$ for $i = t' + 1, \ldots, t$, where $(a_1, \ldots, a_{i-1}) \in \mathbf{Z}_q^{i-1}$ depends arbitrarily on i, the non-group input and the set $\mathcal{CO}_{i-1} := \{(j, k) \mid f_j = f_k,\ 1 \leq j < k \leq i-1\}$ of previous *collisions* of group elements.

Typical non-group inputs are represented by elements in $\mathbf{Z}_q$ — which we assume to be given — various integers in $\mathbf{Z}_q$ contained in given ciphertexts or signatures. $\mathcal{CO}_t$ is the set of all collisions of the algorithm.

Some group inputs f_i depend on random coin flips, e.g., the random public key $h = g^x$ depends on the random secret key $x \in_R \mathbf{Z}_q$. The *probability space* consists of the random group elements of the input. The logarithms $\log_g f_i$ of the

[4] In [CGH98] a mechanism for the implementation of random hash functions has been added to the ROM. The artificial "counter-example" is defined relative to that mechanism using the function ensemble that implements the random oracle.

[5] We count the same generic steps as in [Sh97]; however, we allow arbitrary multivariate exponentiations while Shoup merely uses multiplication and division. The technical setup in [Sh97] looks different as groups G are *additive* and associated with a random injective encoding $\sigma : G \to S$ of the group G into a set S of bit strings — the generic algorithm performs arbitrary computations on these bit strings. Addition/subtraction is done by an oracle that computes $\sigma(f_i \pm f_j)$ when given $\sigma(f_i), \sigma(f_j)$ and the specified sign bit. As the encoding σ is random it contains only the information about which group elements coincide — this is what we call the set of *collisions*.

[6] We can allow a generic algorithm to perform a number t of generic steps, where t varies with the input. We can let the algorithm decide after each step whether to terminate depending arbitrarily on the given non-group data. Then the number t of generic steps depends on the computed non-group data.

random inputs f_i play the role of *secret parameters*. Information about the secret parameters can only be revealed by collisions. E.g., $g^a = f_i^b$ implies $\log_g f_i = a/b$. We let the non-group input and the generator g not depend on random bits.

The *output* of a generic algorithm consists of

- non-group data that depend arbitrarily on the non-group input and on the set $\mathcal{CO}_t$ of all collisions,
- group elements $f_{\sigma_1}, \ldots, f_{\sigma_d}$ where the integers $\sigma_1, \ldots, \sigma_d \in \{1, \ldots, t\}$ depend arbitrarily on the non-group input and on $\mathcal{CO}_t$.

For the sake of clairifying the GM, we give an example of a generic algorithm:

The baby-step-giant-step DL-algorithm. This algorithm is given q and $g, h \in G$ and computes $\log_g h \in \mathbf{Z}_q$ in $2\sqrt{q}$ generic steps.

1. Compute $k := \lceil \sqrt{q} \rceil$, $l := \lceil q/k \rceil$ so that $lk - k < q \leq lk$. The computation of the non-group data k, l is for free.

2. Form the lists $L_1 := \{g^i \mid 0 \leq i < k\}$ in $k - 1$ multiplications and $L_2 := \{hg^{jk} \mid 0 \leq j < l\}$ in l multiplications. Clearly, $L_1 \cap L_2 \neq \emptyset$.

3. Find a collision by testing all equalities $g^i = hg^{jk}$. Note that the detection of the collision is for free. An equality implies $\log_g h = i - jk \bmod q$.

While this algorithm performs $\#L_1 \times \#L_2$ "free" equality tests, the corresponding Turing machine — in the [Sh97]-setup — constructs a collision differently, using only $O(\sqrt{q} \log_2 q)$ equality tests. It sorts the binary encodings of the g^i and inserts the encodings of hg^{jk} into the sorted list.

Going back to the description of the model we work in, we now elaborate on **interactive, generic algorithms**. We count the following generic steps :

- group operations, $\quad \mathrm{mex}\colon \mathbf{Z}_q^d \times G^d \to G, \quad (a_1, ..., a_d, g_1, ..., g_d) \mapsto \prod_i g_i^{a_i}$,
- queries to the hash oracle H,
- interactions with a decryption oracle (*decryptor* for short) — see 3.1 [7].

A *generic adversary* $\mathcal{A}$ — attacking an encryption scheme — is an interactive algorithm that interacts with a decryptor. It performs some t generic steps resulting in $t' \leq t$ group elements $f_1, ..., f_{t'}$. $\mathcal{A}$ iteratively selects the next generic step — a group operation, a query to H, an interaction with the decryptor — depending arbitrarily on the non-group input and on previous collisions of group elements.

The *input* consists of the generator g, the public key $h \in G$, the group order q, a collection of messages and ciphertexts and so on, all of which can be broken down into group elements and non-group data.

The computed *group elements* $f_1, ..., f_{t'} \in G$ are the group elements contained in the input, such as g, h. When counting the number of group operations, we count each input as one operation. As a decryptor interaction is counted as a generic step the number t' of group elements is bounded by the number t of generic steps, $t' \leq t$. We have $t = t'$ for a non-interactive $\mathcal{A}$.

[7] Other types of interactions are possible for other signature/encryption schemes, other cryptographic protocols using groups of non-prime order, groups of unknown order or using several distinct groups.

The given *non-group data* consists of the non-group data contained in the input, the previous hash replies $H(Q)$ of queries Q, and the set of previous collisions of group elements.

A *decryptor interaction* (defined in subsection 3.1) is a two round deterministic protocol. A claimed ciphertext is send to the decryptor, which performs a generic group operation using the secret decryption key x, verifies the Schnorr signature using the public key g^r contained in the ciphertext, and — in case that this signature is correct — outputs the decrypted message. If the signature is invalid the decryptor outputs a random element of G. $\mathcal{A}$'s interactions with the decryptor are sequential as the interleaving of these two-round interactions is necessarily trivial.

$\mathcal{A}$'s *output* and *transmission* to the decryptor consists of non-group data NG and previously computed group elements f_σ, where NG and σ, $1 \leq \sigma \leq t'$, depend arbitrarily on given non-group data.

$\mathcal{A}$'s *transmission* to the hash oracle H depends arbitrarily on given group elements and given non-group data. The *probability space* consists of the random H and the random input group elements.

The *restriction of the generic model* is that $\mathcal{A}$ can use group elements only for generic group operations, equality tests and for queries to the hash oracle, whereas non-group data can be arbitrarily used without charge. The computed group elements $f_1, ..., f_{t'}$ are given as explicit multiplicative combinations of group elements in the input and from decryptor interactions. Let the group elements in the input and from decryptor interactions be $g_1, ..., g_\ell$. By induction on j, a computed $f_j \in G$ is of the form $f_j = g_1^{a_{j,1}} ... g_\ell^{a_{j,\ell}}$, where the exponents $a_{j,1}, ..., a_{j,\ell} \in \mathbf{Z}_q$ depend arbitrarily on given non-group data. $\mathcal{A}$ can arbitrarily use the coefficients $a_{j,1}, ..., a_{j,\ell}$ from this explicit representation of f_j. A generic adversary is deterministic, which is not a restriction as its coin flips would be useless.[8]

Trivial collisions. We call a collision $(i, j) \in \mathcal{CO}_t$ *trivial* if $f_i = f_j$ holds with probability 1, i.e., if it holds for all choices of the secret data such as the secret key x and the random bits r of the encipherer. We write $f_i \equiv f_j$ for a trivial collision.[9] Trivial collisions do not release any information about the secret data while non-trivial collisions can completely release some secret data. Trivial collisions can be ignored, and so, we can exclude them from $\mathcal{CO}_t$ so that $\mathcal{CO}_t$ consists only of non-trivial collisions.

[8] $\mathcal{A}$ could select interior coin flips that maximize the probability of success — there is always a choice for the internal coin flips that does not decrease $\mathcal{A}$'s probability of success. It is useless for $\mathcal{A}$ to generate random group elements — in particular ones with unknown DL. Using one generic step, $\mathcal{A}$ could replace random elements in G by some deterministic g^a where $a \in \mathbf{Z}_q$ is chosen as to maximize the probability of success.

[9] Trivial collisions occur in testing correctness of an ElGamal ciphertext (g^r, mh^r) and its message m. In case of a correct message-ciphertext pair the test results in a trivial collision. Also, identical repetitions of a group operation yield a trivial collision.

3 Signed ElGamal Encryption, Non-interactive Attacks.

We define Schnorr signatures, based on an ideal hash function $H : G \times M \to \mathbf{Z}_q$, where M is the set of messages. Hereafter we define signed ElGamal encryption as well as the generalized concepts of the original and of signed ElGamal encryption.

Lemma 1 and 2 are our main tools for proving security in GM. These show — for a collision-free attacker — that the secret data x, r, etc. are stat. indep. of all non-group data. There is, however, a minor leakage of secret information as the secret data are not perfectly random in the absence of collisions. We show in Prop. 2 that ElGamal encryption is indistinguishable (or semantically secure) against generic non-interactive attacks. Prop. 2 is part of the CCA-security proof of Theorem 1.

Private/public key for signatures. The *private key* x is random in $\mathbf{Z}_q$. The corresponding *public key* $h = g^x \in G$ is random in G, $x = \log_g h$.

A SCHNORR *signature* on a message m is a triple $(m, c, z) \in M \times \mathbf{Z}_q^2$ such that $H(g^z h^{-c}, m) = c$. In order to *sign* a message $m \in M$, pick a random $r \in_R \mathbf{Z}_q$, compute g^r, $c := H(g^r, m)$ and $z := r + cx$. Output the *signature* (m, c, z).

In order to *verify* a signature (m, c, z) check that $H(g^z h^{-c}, m) = c$. The signing protocol produces a correct signature since $g^z h^{-c} = g^{r+cx} h^{-c} = g^r$.

3.1 Definition of Signed ElGamal Encryption.

The private/public key pair for encryption is x, $h = g^x$ where x is random in $\mathbf{Z}_q$. The basic encryption scheme is for messages in $M = G$, ElGamal ciphertexts are in $G \times M$, the added Schnorr signature signs pairs in $G \times M$ and uses a random hash function $H : G^2 \times M \to \mathbf{Z}_q$. We also propose a generalized scheme, where the message space M is an arbitrary additive group.

In order to *encipher* a message $m \in G$, we pick random $r, s \in_R \mathbf{Z}_q$, compute g^r, $m\,h^r$, $c := H(g^s, g^r, mh^r)$ and $z := s + cr$ and output the *ciphertext* $(g^r, mh^r, c, z) \in G^2 \times \mathbf{Z}_q^2$.

A decryption oracle (decryptor) is a function that decrypts valid ciphertexts: The user sends a claimed ciphertext $(\bar{h}, \bar{f}, c, z)$ to the decryptor. The decryptor checks that $H(g^z \bar{h}^{-c}, \bar{h}, \bar{f}) = c$ and sends, if that test succeeds, $m := \bar{f}/\bar{h}^x$ to the user. If the test fails the decryptor sends a random message in G. For simplicity, we disregard the impact of that random message to the probability.

The decryption is correct as $\bar{h} = g^r$, $\bar{f} = m\,h^r$ yields $\bar{f}/\bar{h}^x = m\,g^{rx} g^{-rx} = m$.

Remarks 1. A signed ciphertext (g^r, mh^r, c, z) consists of an ElGamal ciphertext (g^r, mh^r) and a Schnorr signature (c, z) of the "message" (g^r, mh^r) for the public signature key g^r. The signature (c, z) does not contain any information about m as (c, z) depends on m exclusively via some hash value that is statistically independent of m.

2. Threshold Distributed Version. The validity of the ciphertext $(\bar{h}, \bar{f}, c, z)$ is tested prior to and separate from decryption. Hence, the security properties of the scheme are preserved in the more general setting of threshold cryptography, see [SG98]. It is possible for a distributed entity to perform the decryption in a

controlled manner after each server first having verified that indeed the decryption is allowed i.e., that the signature in the ciphertext is valid. If this were not locally verifiable, it would make a threshold decryption severely more complex.
3. Comparison with other secure DL-cryptosystems. We count the number of exponentiations per encryption/decryption and the number of on-line exp. per enc. (exponentiations not depending on the message). [10]

	exp./enc.	on-line/enc.	exp./dec.
Signed ElGamal enc.	3	0	2
[FO99] El Gamal	2	2	2
[ABR 98]	2	0	1
[CS98], [Sh00]	4	1	2
[SG98], TDH1, TDH2	5	2	5

The relative efficiency of [FO99], [ABR98] is due to the usage of further cryptographic primitives. [FO99] uses private encryption, [ABR98] uses private encryption and message authentication code. Signed ElGamal encryption and TDH1, TDH2 of [SG98] are amenable to a secure distributed threshold decryption. Signed EG-encryption and the [FO99] EG-scheme are plaintext aware. Signed ElGamal encryption virtually combines all the good properties.

Generalized (signed) ElGamal encryption. Let the message space M be an arbitrary additive group, e.g., $M = \mathbf{Z}_q^n$. Let $H : G^2 \times M \to \mathbf{Z}_q$ be a random hash function and let $H_M : G \to M$ be a second random hash function that is statistically independent of H. Then replace in the basic encryption scheme $mh^r \in G$ by $m + H_M(h^r) \in M$.

The generalized ElGamal ciphertext is $(g^r, \bar{f})$, where $\bar{f} = m + H_M(h^r)$, the generalized signed ElGamal ciphertext is $(g^r, \bar{f}, c, z)$, and $c = H(g^s, g^r, \bar{f})$, $z = s + cr$. Decrypt a signed ciphertext $(\bar{h}, \bar{f}, c, z)$ into $\bar{f} - H_M(\bar{h}^x)$ provided that the signature (c, z) of $(\bar{h}, \bar{f})$ is correct, i.e., $H(g^z \bar{h}^{-c}, \bar{h}, \bar{f}) = c$.

For $M = \mathbf{Z}_q^n$ the bit length of the ciphertext is $\log_2 \|G\| + (n+2)\log_2 q$, the message is $n \log_2 q$ bits long and $\|G\|$ is the bit length of the group elements. The data expansion rate is $1 + \frac{2}{n} + \frac{\log_2 \|G\|}{n \log_2 q}$ which is near to 1 for large n.

The short generalized ciphertexts are as secure as the original ones. Encryption requires only a long[11] and a short hash as well as a long and a short addition. The three exponentiations g^r, h^r, g^s can be done beforehand.

3.2 Basic Tools for Proving Security in GM

This subsection studies to a generic, non-interactive adversary $\mathcal{A}$ that performs some t generic steps in attacking the indistinguishability of ElGamal encryption.

[10] We count an expression $g^z \bar{h}^{-c}$ as 1 exponentiation even though it is slightly more expensive than a full exponentiation.

[11] Long hash values in $\mathbf{Z}_q^n$ can be generated using a random hash function $H_M : G \to \mathbf{Z}_q^n$ according to the following, or some related, approach: $(H_M(f,1), \ldots, H_M(f,n))$.

Given q, the public key $h = g^x$, two messages $m_0, m_1 \in G$ and an ElGamal ciphertext $cip_b = (g^r, m_b h^r)$ for random $r, x \in_R \mathbf{Z}_q$ and $b \in_R \{0,1\}$, $\mathcal{A}$ guesses b. We show that $\mathcal{A}$ does not succeed better than with probability $\frac{1}{2} + 2\binom{t}{2}/q$.

The probability space consists of the random group elements $g^r, g^x, m_b g^{rx}$, or equivalently of the random $r, x \in_R \mathbf{Z}_q$ and $b \in_R \{0,1\}$. Let $\mathcal{A}$ compute the group elements $f_1, ..., f_t$. We let the *Main Case* be the part of the probability space where there are no non-trivial collisions among $f_1, ..., f_t$, i.e., $\mathcal{CO}_t = \emptyset$.

Lemma 1. *Non-trivial collisions among $f_1, ..., f_t$ occur at most with probability $2\binom{t}{2}/q$. The probability refers to the random b, r, x.*

Proof. In order to prove the claim we show for $i < j$, $f_i \not\equiv f_j$, for constant b and random $r, x \in \mathbf{Z}_q$ that $\Pr_{r,x}[f_i = f_j] \leq \frac{2}{q}$. This implies

$$\Pr_{r,x}[\mathcal{CO}_t \neq \emptyset] \leq \sum_{1 \leq i < j \leq t} \Pr_{r,x}[f_i = f_j] \leq 2\binom{t}{2}/q.$$

The input group elements are $g, g^r, h, m_b h^r, m_0, m_1$. Let $\log_g m_0, \log_g m_1$ be given, then all computed group elements are explicit combinations of $(g_1, g_2, g_3, g_4) = (g, g^r, h, m_b h^r)$, thus $f_j = \prod_{\nu=1}^{4} g_\nu^{a_{j,\nu}}$ where the exponents $a_{j,1}, ..., a_{j,4} \in \mathbf{Z}_q$ depend arbitrarily on given non-group data, but not on b, r, x. Consider r, x as formal variables over $\mathbf{Z}_q$. Then $\log_g f_j$ is a polynomial in $\mathbf{Z}_q[r, x]$ of the form $a_{j,1} + a_{j,2} r + a_{j,3} x + a_{j,4}(\log_g m_b + rx)$. The *difference polynomial* $\log_g f_i - \log_g f_j \in \mathbf{Z}_q[r, x]$ has total degree $d \geq 1$ as we assume that trivial collisions have been eliminated. Importantly, trivial collisions do not depend on b. [12] As $1 \leq d \leq 2$, the probability that $f_i(r, x) = f_j(r, x)$ for random r, x is[13] at most $\frac{2}{q}$, thus proving the claim. Here we use a Lemma attributed to SCHWARTZ [Sch80] [14] □

The leakage of secret information through the absence of collisions. Here we pay attention to the fact that b, r, x are not perfectly random if $\mathcal{CO}_t = \emptyset$. By Lemma 1 a $2\binom{t}{2}/q$-fraction of the probability space is excluded in the Main Case. The SHANNON entropy of the secret parameters b, r, x decreases accordingly. We can neglect this minor leakage of secret information through the absence of collisions. Thus, for a "collision-free" attacker the secret data are statistically indepependent of the computed non-group data:

Lemma 2. *In the Main Case the random b, r, x are stat. indep. of the computed non-group data except that the b, r, x leading to collisions are excluded.*

Proof. The random b, r, x, enter into the generic computation only via the group elements $g^r, g^x, m_b g^{rx}$. Therefore, b, r, x enter into non-group data only via non-trivial collisions of group elements. □

[12] The formal polynomial $\log_g f_i - \log_g f_j \in \mathbf{Z}_q[r, x]$ is of the form $c_1 + c_2 r + c_3 x + c_4(\log_g m_b + rx)$. The coefficients $c_1, ..., c_4 \in \mathbf{Z}_q$ only depend on q and previous non-trivial collisions. If $f_i \equiv f_j$ holds for some $b \in \{0,1\}$ then $c_4 = 0$ and $f_i \equiv f_j$ holds for all $b \in \{0,1\}$. Hence the identity $f_i \equiv f_j$ does not depend on b.

[13] The factor 2 disappears if $m_b h^r$ is removed from the input — then the difference polynomial has total degree at most 1.

[14] Lemma [Sch80] A multivariate polynomial $F \in \mathbf{Z}_q[X_1, ..., X_k]$ of total degree d satisfies for random $x_1, ..., x_k \in \mathbf{Z}_q$ that $\Pr_{x_1,...,x_k}[F(x_1, ..., x_k) = 0] \leq d/q$.

Proposition 1. Generic DL-Complexity Lower Bound [Ne94,Sh97]. *Let $\mathcal{A}$, upon input g and $h = g^x \in_R G$, output $y \in \mathbf{Z}_q$. Then* $\Pr_x[y = \log_g h] \leq \binom{t}{2}/q + \frac{1}{q}$.

Proof. We use Lemma 1 and 2 for a generic $\mathcal{A}$ with input g, h — without inputs $g^r, m_b h^r$. The factor 2 in Lemma 1 disappears as the polynomials $\log_g f_j$ have total degree ≤ 1. For a collision-free $\mathcal{A}$, x is statistically independent of the non-group output y, and thus $\Pr_h[y = \log_g h] = \frac{1}{q}$. By Lemma 1, non-trivial collisions occur at most with probability $\binom{t}{2}/q$. □

Proposition 2. Indistinguishability. *Let a generic, non-interactive $\mathcal{A}$ be given g, h, two messages $m_0, m_1 \in G$ and a ciphertext $(g^r, m_b h^r)$ for random $r \in_R \mathbf{Z}_q$ and $b \in_R \{0, 1\}$. Let $\mathcal{A}$ output a guess b' for b. Then* $\Pr_{b,x,r}[b' = b] \leq \frac{1}{2} + 2\binom{t}{2}/q$.

Proof. In the Main Case b, r, x are stat. indep. of the non-group output b', thus $\Pr_{b,x,r}[b' = b] = \frac{1}{2}$. The Main Case occurs except with probability $2\binom{t}{2}/q$. □

Extension 1. Obviously Prop.2 extends to generalized ElGamal ciphertexts $(g^r, m + H_M(h^r))$ for a random function $H_M : G \to M$. Whereas $\mathcal{A}$ can arbitrarily use the hash values $H_M(f_1), ..., H_M(f_t)$ of the computed group elements these hash values are statistically independent random numbers except for collisions $f_i = f_j$.

Extension 2. Prop. 2 extends to signed ElGamal encrytion and to generalized signed ElGamal encrytion. This is because the added Schnorr signature does not contain any information about the plaintext.

4 Security Against Interactive Attacks

We study the security of signed ElGamal encryption in ROM+GM. Signed ElGamal encryption was independently proposed by Tsiounis and Yung [TY98] and Jakobsson [J98]. We show in Theorem 1 that this scheme is indistinguishable against the adaptive chosen ciphertext attack (CCA). This is equivalent to non-malleability against CCA [DDN98]. We refer to non-malleability as defined in [DDN98] and to the strong chosen ciphertext attack proposed by Rackoff and Simon [RS92]. The adversary has access to a decryption oracle which can be used arbitrarily except for the target ciphertext.

Moreover, we introduce the one-more-decryption attack and we show in Theorem 2 that signed ElGamal encryption is secure against this attack. An adversary can — after some ℓ interactions with the decryption oracle — not decrypt more than ℓ ciphertexts. More precisely, he gets non-negligible information about at most ℓ encrypted plaintexts. The core of the proof of Theorems 1 and 2 shows that signed ElGamal encryption is plaintext aware. Therefore, the attackers decryption requests for self-constructed ciphertexts can be eliminated.

Theorem 1 proves indistinguishability against a CCA-adversary $\mathcal{A}$. The adversary is given a target ciphertext cip_b and a decryption oracle for the decryption of arbitrary ciphertexts except for cip_b. The attack is called adaptive because the queries to the decryption oracle may depend on the challenges and their corresponding answers. We let the generic adversary $\mathcal{A}$ perform some t generic

steps: group operations, inputs in G, queries to the oracle H, and queries to the decryption oracle not including the target ciphertext.

Theorem 1. *Let the attacker $\mathcal{A}$ be given g, h, distinct messages m_0, m_1, a target ciphertext cip_b corresponding to m_b for a random bit $b \in_R \{0,1\}$, and oracles for H and for decryption. Then a generic $\mathcal{A}$ using t generic steps cannot predict b with a better probability than $\frac{1}{2} + t^2/q$. The probability space consists of the random x, H, b and the coin tosses r of the encipherer.*

Proof. We present a generic extractor $\mathcal{E}$ that extracts the secret key $\bar{r} = \log_g \bar{h}$ from a signed ciphertext $(\bar{h}, \bar{f}, c, z)$ produced by $\mathcal{A}$. Given $\bar{r}, \bar{h}, \bar{f} = \bar{m}\bar{h}^{\bar{r}}$ the plaintext $\bar{m}$ can be extracted in one generic step. Thus, signed ElGamal encryption is — in a generic way — plaintext aware as defined in [BR94].

Let $(\bar{h}, \bar{f}, c, z)$ be the first claimed ciphertext that $\mathcal{A}$ transmits to the decryptor. $\mathcal{A}$ has produced it without interacting with the decryptor. Let this non-interactive, generic computation compute group elements $f_1, ..., f_{t'}$, $t' \leq t$. By Lemma 1 non-trivial collisions among $f_1, ..., f_{t'}$ occur with probability no more than $2\binom{t}{2}/q$. By Lemma 2 the secret b, r, x are statistically independent of the non-group data of a collision-free computation of $(\bar{h}, \bar{f}, c, z)$.

In the ROM the equation $c = H(g^z \bar{h}^{-c}, \bar{h}, \bar{f})$, required for a valid signature, necessitates that $\mathcal{A}$ selects c from given hash values $H(f_\sigma, f_j, \bar{f})$ for given group elements f_σ, $f_j = \bar{h}$, $\bar{f}$. Otherwise, the equation $c = H(g^z \bar{h}^{-c}, \bar{h}, \bar{f})$ holds with probability $\frac{1}{q}$ as H is random. $\mathcal{A}$ gets $c = H(f_\sigma, f_j, \bar{f})$ from the hash oracle[15] and must compute z so that $g^z \bar{h}^{-c} = f_\sigma$, i.e., $\mathcal{A}$ must compute $z = \log_g(f_\sigma f_j^c)$.

The computed z does not depend on x, r whereas $\log_g(f_\sigma f_j^c)$ may depend. We distinguish between the two values as follows: We let $z' := \log_g(f_\sigma f_j^c)$ denote the value required for a signature, whereas the computed z is from $\mathcal{A}$'s transmission $(f_j, \bar{f}, c, z)$.

Let the target ciphertext be $cip_b = (g^r, m_b h^r, c_b, z_b)$, where the random $r, x \in_R \mathbf{Z}_q$, $b \in_R \{0,1\}$ are secret and $h = g^x$. Let $\log_g m_0, \log_g m_1$ be given, then $\mathcal{A}$'s group steps refer to the given group elements $(g_1, g_2, g_3, g_4) := (g, g^r, h, m_b h^r)$. $\mathcal{A}$ computes $f_i := \prod_{\nu=1}^{4} g_\nu^{a_{i,\nu}}$ for $i = 1, ..., t'$ using exponents $a_{i,1}, ..., a_{i,4} \in \mathbf{Z}_q$ that arbitrarily depend on given non-group data, but not on b, r, x. Hence z' is of the form

$$\begin{aligned} z' &= \log_g(f_\sigma f_j^c) \\ &= a_{\sigma,1} + c\, a_{j,1} + (a_{\sigma,2} + c\, a_{j,2})\, r + (a_{\sigma,3} + c\, a_{j,3})\, x + (a_{\sigma,4} + c\, a_{j,4})(\log_g m_b + rx). \end{aligned}$$

Considering r, x as formal variables over $\mathbf{Z}_q$, z' is a polynomial in $\mathbf{Z}_q[r, x]$. The random b, c, r, x are statistically independent of $a_{\sigma,1}, ..., a_{\sigma,4}, a_{j,1}, ..., a_{j,4}$.

Obviously z' has total degree $d = 0$ if and only if $a_{\sigma,k} + c\, a_{j,k} = 0$ for $k = 2, 3, 4$. If the total degree d is non-zero then $1 \leq d \leq 2$, and thus $z' = z(c)$ holds with probability at most $\frac{2}{q}$ for random r, x and arbitrary functions $z(c)$. There are two subcases of the case $d = 0$: *either* $f_j = g^{a_{j,1}}$, $f_\sigma = g^{a_{\sigma,1}}$ *or* $a_{\sigma,k} = -c a_{j,k}$ for $k = 2, 3, 4$. The second case occurs with probabilty $\leq \frac{1}{q}$ as the hash value c is statistically indepependent of $a_{\sigma,k}, a_{j,k}$.

[15] $\mathcal{A}$'s choice of c, σ is determined by the claimed ciphertext $(\bar{h}, \bar{f}, c, z)$ via $f_\sigma = g^z \bar{h}^{-c}$.

Thus, a collision-free $\mathcal{A}$ succeeds not better than with probability $\frac{3}{q}$ in generating a correct signature (c, z) except that $\mathcal{A}$ sets $f_j = g^{a_{j,1}}$, $f_\sigma = g^{a_{\sigma,1}}$. So, let the extractor $\mathcal{E}$ compute $\bar{r} := a_{j,1}$ by mimicking $\mathcal{A}$'s computation of $\bar{h} = f_j$.

Eliminating all interactions with the decryptor. The plaintext corresponding to $(\bar{h}, \bar{f}, c, z)$ is $\bar{f}/h^{\log_g \bar{h}} = \bar{f}/h^{a_{j,1}}$ except for a probability $\frac{3}{q}$. This eliminates the first interaction with the decryptor and the call for $H(f_\sigma, f_j, \bar{f})$[16] using one generic step for computing $\bar{f}/h^{a_{j,1}}$. This decreases the number of generic steps and reduces $\mathcal{A}$'s probability of success by at most $\frac{3}{q}$. Let there be ℓ interactions with the decryptor. We iteratively eliminate them by the above method.[17] This transforms $\mathcal{A}$ into a non-interactive generic $\mathcal{A}'$ that performs $t - \ell$ generic steps. Proposition 2 applies to the non-interactive $\mathcal{A}'$, because the Schnorr signature in cip_b is useless for decryption.[18] Also, the oracle H is useless without a decryptor. Thus, the non-interactive $\mathcal{A}'$ predicts b with a probability not exceeding $(t - \ell)^2/q + \frac{1}{2}$. This proves Theorem 1 as $(t-\ell)^2 + 3\ell \leq (t-\ell+\ell)^2 = t^2$ for $t - \ell \geq 3$. Note that $t - \ell \geq 4$ due to the input group elements $g, g^r, h, m_b h^r$. □

Theorem 1 can easily be extended to the one-more decryption attack.

Theorem 2. *Let the attacker $\mathcal{A}$ be given g, h, ciphertexts $cip_1, ..., cip_d$, the corresponding messages $m_1, ..., m_d$ in random order and oracles for H and for decryption. Let the generic $\mathcal{A}$ perform t generic steps including some $\ell < d$ arbitrary queries to the decryption oracle. Then $\mathcal{A}$ cannot produce $\ell + 1$ message-ciphertext pairs with a probability better than $\frac{1}{d-\ell} + t^2/q$. The probability space consists of the random x, H, the coin tosses of the encipherer and the random ordering of the messages.*

Proof. We have shown that signed ElGamal encryption is plaintext aware, and the attacker can only construct ciphertexts corresponding to known plaintexts. In particular, the adversary $\mathcal{A}$ can be transformed into a generic adversary $\mathcal{A}'$ that does not query the decryptor about any self-constructed ciphertext, performs t generic steps and succeeds essentially with the same probability as $\mathcal{A}$. $\mathcal{A}'$ can only query the decryption oracle about ℓ of the input ciphertexts. These ℓ decryptions give no information about the $d - \ell$ remaining input ciphertexts. This is because the random bits of the ciphertexts are stat. indep. We can therefore eliminate the ℓ decryptions and the resulting ℓ message-ciphertext pairs. This transforms $\mathcal{A}'$ into a non-interactive adversary where the argument of Lemma 1 applies.

[16] The transformed $\mathcal{A}$ gets the plaintext $\bar{f}/h^{a_{j,1}}$ of the first decryptor interaction without using the signature and its hash value required for the decryption request. If $\mathcal{A}$ does not get c from the oracle H we remove the call for decrypting $(\bar{h}, \bar{f}, c, z)$ decreasing the number of generic steps and decreasing $\mathcal{A}$'s probability of success by at most $\frac{1}{q}$.

[17] This iterative elimination is impossible in the ROM without assuming the GM, see footnote 1.

[18] The signature contained in a ciphertext does not reveal any information about the message m. The signature depends on m exclusively via the hash value c that is statistically independent of m.

Consider the impact of a random permutation of the remaining $d - \ell$ messages for a collision-free attacker. By Lemma 2 the random permutation is statistically independent of $\mathcal{A}'$'s guess of a correct message-ciphertext pair. Therefore, $\mathcal{A}'$ cannot guess a correct pair with a probability better than $\frac{1}{d-\ell}$. By Lemma 1 non-trivial collisions occur with probability at most $2\binom{t}{2}/q$, hence the claim. □

Trading encrypted information. Suppose a user wants to buy sensitive digital information, e.g., digital music, videos, pictures, stock market analysis, etc. Let the digital information be freely accessible in encrypted form in a public data bank. For simplicity, let each encrypted package cost \$1. Let the users have access to a public decryption oracle that charges \$1 per decryption. For the security of such trade of encrypted information the encryption scheme must be secure against the one-more decryption attack.

This type of service does not require CCA-security. However, it would be nice to have an encryption that allows for blind decryption so that no information is revealed in a decryptor interaction. Blind decryption guarantees anonymity of the buyer of digital information. It is well known that the original ElGamal ciphertexts allow for blind decryption.[19] Even though, ElGamal encryption is insecure against the one-more decryption attack we show below that it is secure against the weaker *random one-more attack*, where the enciphered plaintexts are statistically independent messages — e.g. secret keys that are unknown to the attacker.

Efficient scheme for private information retrieval (PIR). Let the information packages m_i of the public data bank be each encrypted under a private key k_i of a secure symmetric encryption scheme. Let m_i contain a content description $descr_i$ of m_i and a signed ElGamal ciphertext $cip(k_i) = (g^r, k_i h^r, c, z)$ of the key $k_i \in G$. Let (c, z) be a signature of $(g^r, k_i h^r, descr_i)$ with public key g^r. Suppose a user wants to anonymously buy ℓ packages m_i of his choice. He checks the Schnorr signature (c, z) of $cip(k_i) = (g^r, k_i h^r, c, z)$ in package m_i and stops if the signature is invalid. Otherwise, he blinds the ElGamal ciphertext $(g^r, k_i h^r)$ into $(g^{r+s}, u k_i h^{r+s})$ for random $s \in \mathbf{Z}_q$, $u \in G$, and asks the decryption oracle to decrypt $(g^{r+s}, u k_i h^{r+s})$. As the blinded ciphertext is statistically independent of $(g^r, k_i h^r)$ no information is revealed about which k_i he gets. As the user pays for ℓ decryptions it is important that he cannot get $\ell + 1$ keys k_i.

Security against the random one-more attack. Consider the above PIR for random $k_i \in G$. Clearly, $\ell+1$ keys $k_i \in_R G$ have SHANNON *entropy* $(\ell+1)\log_2 q$. But each decryption reveals no more than $\log_2 q$ bits of a plaintext in G, $|G| = q$. Thus, ℓ decryptions cannot reveal $\ell + 1$ statistically independent keys k_i.

Another application would be an electronic service for delivering sensitive, possibly unpleasant messages like court orders, summons, admonitions and so on. Such messages can be sent in encrypted form, given access to a decryption oracle that combines the decryption with an acknowledgement of the receipt

[19] Blind decryption of the ElGamal ciphertext (g^r, mh^r): The user picks random $u \in G$ and $s \in \mathbf{Z}_q$ and asks for decryption of (g^{r+s}, umh^{r+s}). He gets m from the plaintext um transmitted by the decryptor by multiplication with u^{-1}.

of the decrypted message. This makes sure that a recipient can only read the message by acknowledging receipt. For such a service it would be important that the encrytion is CCA-secure, so that the receipt correctly specifies the revealed message. However, we also need security against the one-more decryption attack as users may want to decrypt several ciphertexts. Signed ElGamal encryption can be used for such a service.

Security of Schnorr signatures with short hash values. Let the hash values of H be random in an interval $[0, 2^k[\subset [0, q[\cong \mathbf{Z}_q$. The size of that interval enters into the proof of Theorem 1 merely at the point, where we argue that the case $a_{\sigma,k} = -ca_{j,k}$ for $k = 2, 3, 4$ has probability $\leq \frac{1}{q}$. For random hash values $c \in_R [0, 2^k[$ that case has probability $\leq 2^{-k}$.

Consequently, in the case of Theorem 1 a CCA-attacker does not succeed better than with probability $\frac{1}{2} + t^2/q + \ell(2^{-k} - \frac{1}{q})$, where ℓ is the number of decryptor interactions. This shows that random hash values can securely range over a set of $\sqrt{q}$ values.

Security of Schnorr Signatures in the ROM+GM. The proof of Theorem 1 contains a security proof for Schnorr signatures in the ROM+GM:

Corollary 1. *Let $\mathcal{A}$ be a generic algorithm that is given g, the public signature key $h \in_R G$ and a random hash oracle. Using t generic steps — group operations and hash queries — $\mathcal{A}$ cannot produce a Schnorr signature with a probability better than $\frac{3}{q} + \binom{t}{2}/q$. The probability space consists of the random h, H.*

Security against the chosen message attack. Corollary 1 extends to the case that the adversary $\mathcal{A}$ has a signature oracle and can ask the oracle for signatures on messages of its choice. An interaction with the signature oracle is counted as generic step. The goal of the attack is to generate a new signature which is not produced by the signature oracle. The proof of the extension is straightforward.

Unlike the case of Theorems 1 and 2, Corollary 1 and its extension have a counterpart in the ROM without assuming the GM, see POINTCHEVAL AND STERN [PS96]. Howewer, the security theorems and their proofs in the ROM use completely different arguments — the probability bounds are less tight.

References

[ABR98] *M. Abdalla, M. Bellare and P. Rogaway*: DHES: An Encryption Scheme Based on the Diffie-Hellman Problem. Contributions to P1363, ftp://stdgbbs.ieee.org/pub/p1363/contributions/aes-uhf.ps

[BDPR98] *M. Bellare, A. Desai, D. Pointcheval and P. Rogaway*: Plaintext Awareness, Non-Malleability, and Chosen Ciphertext Security: Implications and Separations. Crypto'98, LNCS 1462, pp. 26–45, 1998.

[BL96] *D. Boneh and R.J. Lipton*: Algorithms for black-box fields and their application in cryptography. Crypto'96, LNCS 1109, pp. 283–297, 1996.

[BR93] *M. Bellare and P. Rogaway*: Random Oracles are Practical: a Paradigms for Designing Efficient Protocols. 1st ACM Conference on Computer Communication Security, pp. 62–73, 1993.

[BR94] *M. Bellare and P. Rogaway* : Optimal Asymmetric Encryption. Eurocrypt'94, LNCS 950, pp. 92–111, 1995.

[CGH98] *R. Canetti, O. Goldreich and S. Halevi* : The Random Oracle Methodology, Revisited. STOC'98, ACM Press, pp. 209–218, 1998.

[CS98] *R. Cramer and V. Shoup* : A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. Crypto'98, LNCS 1462, pp. 13–25, 1998.

[DDN91] *D. Dolev, C. Dwork and M. Naor* : Non-Malleable Cryptography. STOC'91, ACM Press pp. 542–552, 1991.

[DDN98] *D. Dolev, C. Dwork and M. Naor* : Non-Malleable Cryptography. Manuscript (updated, full length version of STOC paper), 1998.

[E85] *T. ElGamal* : A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. IEEE Trans. Inform. Theory, 31, pp. 469–472, 1985.

[FO99] *E. Fujisaki and T. Okamoto* : Secure Integration of Asymmetric and Symmetric Encryption Schemes. Crypto'99, LNCS 1666, pp. 537–554, 1999.

[FFS88] *U. Feige, A. Fiat and A. Shamir* : Zero-knowledge proofs of identity. J. Cryptology, 1 , pp. 77–94, 1988.

[FS87] *A. Fiat and A. Shamir* : How to Prove Yourself: Practical Solutions of Identification and Signature Problems. Proc. Crypto'86, LNCS 263, pp. 186–194, 1987.

[GM84] *S. Goldwasser and S. Micali* : Probabilistic Encryption. J. Computer and System Sciences, 28, pp. 270–299,1984.

[J98] *M. Jakobsson* : A Practical Mix. Eurocrypt'98, LNCS 1403, pp. 448–461, 1998.

[MOV96] *A. Menezes, P. van Oorschot and S. Vanstone* : Handbook of Applied Cryptography. CRC Press, Inc., 1996.

[Ne94] *V.I. Nechaev*: Complexity of a Determinate Algorithm for the Discrete Logarithm. Mathematical Notes 55, pp. 165-172, 1994.

[RS92] *C. Rackoff and D.R. Simon* : Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. Crypto'91, LNCS 576, pp. 433–444, 1992.

[Sch80] *J. Schwartz* : Fast probabilistic algorithms for verification of polynomial identities. J. ACM, 27(4), pp. 701–717, 1980.

[Sc91] *C.P. Schnorr* : Efficient Signature Generation for Smart Cards. Journal of Cryptology 4 (1991), pp. 161–174.

[SJ99] *C.P. Schnorr and M. Jakobsson* : Security of Discrete Log Cryptosystems in the Random Oracle and Generic Model. TR report University Frankfurt and Bell Laboratories 1999.

[Sc00] *C.P. Schnorr* : Small Generic Hardcore Subsets for the Discrete Logarithm: Short Secret DL-Keys. Presented at rump session of Eurocrypt'2000.

[Sh97] *V. Shoup* : Lower Bounds for Discrete Logarithms and Related Problems. Eurocrypt'97, LNCS 1233, pp. 256-266, 1997.

[Sh00] *V. Shoup* : Using Hash Functions as a Hedge against Chosen Ciphertext Attack. Eurocrypt'2000, LNCS 1807, pp. 275–288, 2000.

[SG98] *V. Shoup and R. Gennaro* : Securing Threshold Cryptosystems against Chosen Ciphertext Attacks. Eurocrypt'98, LNCS 1404, pp. 1–16, 1998.

[TY98] *Y. Tsiounis and M. Yung*, On the Security of ElGamal Based Encryption. PKS'98, LNCS 1431, pp. 117-134, 1998.

[ZS92] *Y. Zheng and J. Seberry*, Practical Approaches to Attaining Security against Adaptively Chosen Ciphertext Attacks. Crypto'92, LNCS 740, pp. 292-304, 1992.