Security of Discrete Log Cryptosystems in the Random Oracle and the Generic Model

Claus Peter Schnorr
Fachbereich Mathematik/Informatik
Universität Frankfurt, Germany
and Bell Laboratories
schnorr@cs.uni-frankfurt.de

Markus Jakobsson
Information Sciences Laboratory
Bell Laboratories
Murray Hill, New Jersey 07974
markusj@research.bell-labs.com

April 28, 2000

Abstract

We introduce novel security proofs that use combinatorial counting arguments rather than reductions to the discrete logarithm or to the Diffie-Hellman problem. Our security results are sharp and clean with no polynomial reduction times involved. We consider a combination of the *random oracle model* and the *generic model*. This corresponds to assuming an *ideal hash function* H given by an oracle and an *ideal group* of prime order q, where the binary encoding of the group elements is useless for cryptographic attacks

In this model, we first show that Schnorr signatures are secure against the *one-more signature forgery*: A generic adversary performing t generic steps including ℓ sequential interactions with the signer cannot produce $\ell+1$ signatures with a better probability than $\binom{t}{2}/q$. We also characterize the different power of sequential and of parallel attacks.

Secondly, we prove signed ElGamal encryption is secure against the *adaptive chosen ciphertext attack*, in which an attacker can arbitrarily use a decryption oracle except for the challenge ciphertext. Moreover, signed ElGamal encryption is secure against the one-more decryption attack: A generic adversary performing t generic steps including ℓ interactions with the decryption oracle cannot distinguish the plaintexts of $\ell + 1$ ciphertexts from random strings with a probability exceeding $\binom{t}{2}/q$.

1 Introduction and Summary

Proving security for cryptographic primitives like signatures and encryption is a challenging problem in particular for an interactive setting, where an active adversary interferes in the interaction. We introduce novel security proofs for discrete log cryptosystems that use combinatorial counting arguments rather than reductions to the discrete logarithm or to the Diffie-Hellman problem. Our security results are sharp and clean with no polynomial reduction times involved. Our approach separates in a better way cryptographic weaknesses of the hash function, the group and the cryptographic protocols. This separation is crucial. If an attack is possible for a specific hash function or group we need a stronger hash function or group while keeping the cryptographic protocols. As NIST has proposed strong hash functions and strong groups it makes sense to analyze cryptographic protocols assuming that the hash function and the group have no cryptographic weaknessses. So we merely consider attacks that work for all hash functions and for all groups. If an attack occurs that works

for a specific hash function or group in use, then the latter must be replaced. Formally, we assume the random oracle model (ROM) for the hash function and the generic model for the group. This is a combination of two already accepted models. We do NOT assume that the discrete log problem is hard, our security proofs contain a hardness proof for the discrete log problem in the generic model. Our approach has practical consequences.

Previous security proofs for discrete log cryptosystems do not prove security for the most simple, unbroken discrete log schemes. Additional control mechanisms have been introduced into yet unbroken cryptographic protocols in order to simplify the reductions to the discrete log or to the Diffie-Hellman problem. We prove that the most simple, unbroken discrete log schemes are secure in a reasonable security model.

Traditional security proofs for discrete log signatures in the random oracle model [PS96a,PS96b,P98] polynomially transform successful attacks into discrete log computations. In [PS96a] it is shown that various DL-signatures, in particular Schnorr signatures, are secure against the adaptive chosen message attack. The security proofs in [PS96b, P98] apply to blind signatures in the interactive setting. The security proof in [PS96b] requires that the number of interactions of the parallel attacker with the signer are poly-logarithmically bounded. In [P98] a third party — the *checker* — has been introduceded, and it was shown that the resulting three-party signature protocol is secure for a polynomial number of *synchronized* signer interactions, where the synchronization forces the completion of each step for all the different protocol invocations before the next step of any other invocation is started. Moreover, these schemes use more complicated signatures — Okamoto-Schnorr signatures with multiple key components. Our novel security proofs are for the most simple discrete log signatures covering general parallel attacks, we exemplify them for Schnorr signatures.

For public key encryption schemes we refer to the schemes of Shoup, Gennaro [SG98], Cramer, Shoup [CS98], Abdalla, Bellare, Rogaway [ABR98] and Zheng, Seberry [ZS92]. All these schemes extend ELGamal encryption by a signature or tag. This idea first appears in [ZS92] without a security proof. Security against strong chosen ciphertext attacks has been proved in [SG98, CS98, ABR98]. The schemes in [SG98, CS98, ABR98] use an involved tag construction and key generation to simplify the reduction to the discrete log and to the Diffie-Hellmamn problem, the tag in [ABR98] uses symmetric encryption. We consider the most simple extension of ElGamal encryption that was independently proposed by Tsiounis and Yung [TY98] and Jakobsson [J98]. Herein, a Schnorr signature providing a proof of knowledge of the plaintext and of the secret encryption parameter r is added to the ElGamal ciphertext. We call this encryption the *signed* ElGamal encryption.

Our results on signatures. The most powerful attack is the *one-more signature forgery* introduced in [PS96b, P98], where security means that an attacker cannot obtain $\ell + 1$ valid signatures from ℓ interactions with the signer. The most general case are parallel attacks, where the parallel interactions are non-synchronous and arbitrarily interleaved. Security against this attack is important in e-commerce, where it translates into that an adversary cannot "create additional money". We prove security of plain Schnorr signatures against the one more signature forgery. We present a sharp security bound for sequential attacks, where a generic adversary using t generic steps cannot succeed better than with probability $\binom{t}{2}/q$. Parallel attacks surpassing the power of sequential attacks must solve an established hard problem: solving — from a system of t distinct linear equations in ℓ variables, where the inhomogenities are random integers modulo a prime q, and $\ell < t < q$ — more than ℓ linear equations modulo q. This holds for ℓ arbitrary interactions with the signer with arbitrary interleaving. The fastest known algorithm that solves more than ℓ equations does not succeed

better than with probability $\binom{T}{2}/q$ using T arithmetic steps modulo q. By the famous work of HASTAD [H97], the corresponding problem with constant inhomogenities is NP-hard in worst case.[1] We note that the problem with random inhomogenities seems to be hard for *all* instances. This makes it a natural and attractive hard NP-problem.

Our results on encryption. We prove that signed ElGamal encryption is secure against the strong adaptive chosen ciphertext attack, where an attacker can use a decryption oracle — except for the challenge ciphertext. We show that a generic attacker using t generic steps cannot decrypt the challenge ciphertext better than with probability $\binom{t}{2}/q$. Moreover, signed ElGamal encryption is secure against the *one-more decryption attack*.[2] If a generic adversary performs t generic steps, is given statistically independent ciphertexts and access to a decryption oracle some ℓ times, he cannot decrypt $\ell + 1$ ciphertexts better than with probability $\binom{t}{2}/q$. Our security bounds are sharp, the same optimal upper bound $\binom{t}{2}/q$ holds for the success probability all attacks. The probability is for a random private key x, a random hash function H, and the coin flips of the encipherer.

Finally, and of possible independent interest, is a scheme for fast encryption of long messages that we propose. The scheme is based on El-Gamal encryption, and can be proved to have the same security as our basic encryption scheme, without any further assumptions.

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 $\mathbf{Z}_q$. Informally, the hash function H is modelled as an oracle that given an input (query) of the appropriate form outputs a random number in $\mathbf{Z}_q$. We will use different input formats onwards, inputs in $G \times M$ for plain signatures and inputs in G^n for encryption. Formally, H is a random function either of type $H : G \times M \to \mathbf{Z}_q$, or of type $H : G^n \to \mathbf{Z}_q$ for some n — chosen at random over all functions of that type with uniform probability distribution. The ROM goes back to FIAT AND SHAMIR [FS86] and has been further developed by BELLARE AND ROGAWAY [BR93].

The Generic Model (GM). Let the group G be *ideal* in that the binary encoding of the group elements is useless for cryptographic attacks.[3] By the ideal group assumption, the adversary is not given the binary encoding of group elements, but can access group elements only for group operations and equality tests. The generic model of algorithms goes back to NECHAEV [Ne94] who proves in this model that the discrete logarithm problem is hard. Generic algorithms have been further elaborated by SHOUP [Sh97].[4] We slightly modify the

[1] **Theorem 2.4** [H97]. *For any $\varepsilon > 0$ and prime q, it is NP-hard to approximate the maximal number of satisfiable equations modulo q within a factor $q - \varepsilon$.*

[2] The one-more decryption attack is not covered by the adaptive chosen ciphertext attack. An adversary can use the challenge ciphertexts in an arbitrary way for queries to the decryption oracle. This is excluded in the adaptive ciphertext attack.

[3] The ideal group assumption is believed to hold for random elliptic curves and for subgroups $G \subset \mathbf{Z}_p^*$ of the multiplicative group $\mathbf{Z}_p^*$ of integers modulo a prime p, provided that G is of prime order q, q and $(p-1)/q$ are random and p/q is so large that sieving methods for discrete log computations are inefficient.

[4] Our generic model is close to that of Shoup [Sh97] but differs as follows. Shoup counts equality tests and group operations (performed by oracles), whereas we only count group operations allowing a more general class of group operations. The [Sh97] algorithms use internal coin flips and a random encoding of group elements, we eliminate these random sources as they are useless. Our generic complexity lower bounds are a bit stronger than those for the Shoup model.

Shoup model and we extend it to algorithms that interact with a signature/decryption oracle. Signatures and encryption are for the private/public key pair is (x, h), where x is random in $\mathbf{Z}_q$ and $h = g^x$. We describe the extended generic model in detail. (Other extensions are possible using different types of interactions, e.g. for different signature/encryption schemes.)

Generic steps. In the ROM + GM we count for *generic steps*:

- group operations, i.e. multivariate exponentiations
  $\text{mex}_{\mathbf{a}} : G^d \to G, \quad (g_1, ..., g_d) \mapsto \prod_i g_i^{a_i}$ with $\mathbf{a} = (a_1, ..., a_d) \in \mathbf{Z}_q^d$,
- queries to the hash oracle H,
- interactions with a signature/decryption oracle (*signer/decryptor* for short).

A *generic adversary* $\mathcal{A}$ — attacking a signature/encryption scheme — is an interactive algorithm that interacts with a signer/decryptor. It performs a straight-line program consisting of some t generic steps resulting in $t' \leq t$ group elements $f_1, ..., f_{t'}$ and non-group data (containing no group elements). $\mathcal{A}$ iteratively selects the next generic step — a group operation, a query to H, an interaction with the signer/decryptor — arbitrarily depending on the previously computed non-group data. The hash queries in $G \times M$ respectively in G^n may arbitrarily depend on previously computed group elements *and* non-group-data.

The *input* consists of the generator g, the public key $h \in G$, and e.g. a collection of messages, signatures and ciphertexts consisting of group elements and non-group data.

The *group elements* $f_1, ..., f_{t'} \in G$ are: the group elements contained in the input, the results of the group operations $\text{mex}_{\mathbf{a}}$, the group elements that an interactive $\mathcal{A}$ receives from the signer/decryptor (the signer/decryptor replies with at most one group element per interaction and thus $t' \leq t$). The sequence $f_1, ..., f_{t'}$ starts with the group elements contained in the input $f_1 = g, f_2 = h$ etc., the input group elements are counted as generic steps.

The *non-group data* are: the non-group data contained in the input, the exponent vectors $\mathbf{a}$ of the generic group operations $\text{mex}_{\mathbf{a}}$, the hash replies $H(Q)$ of queries Q, the equalities $f_i = f_j$ (called *collisions*) and inequalities $f_i \neq f_j$ of all previously computed f_i, f_j, and the non-group data that $\mathcal{A}$ receives in signer/decryptor interactions.

In a *signer/decryptor interaction*, described in subsequent sections, the signer/decryptor performs a generic group operation depending on the secret signature/decryption key x and, in case of the signer, on a random number $r \in_R \mathbf{Z}_q$ selected by the signer.

$\mathcal{A}$'s *output* and his *transmission* in interactions with the signer/decryptor consists of previously computed group elements and non-group data. Interactions are *sequential* if there is at most one interaction at a time, in that case the attack is called sequential. Non-sequential attacks are called *parallel*. We consider the most general form of a parallel attack, where $\mathcal{A}$ can interleave parallel interactions in an arbitrary way, $\mathcal{A}$ can start the second rounds of the interactions in an order that is independent of the order of the first rounds.

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. On the other hand generic algorithms can arbitrarily transform non-group data without charge.

The *probability space* consists of the random group elements in the input as the public key $h \in_R G$ etc., the random H and the coin flips of the signer (the decryptor is deterministic). A generic adversary is deterministic. This is not a restriction as its coin flips would be useless. $\mathcal{A}$ can select interior coin flips that maximize the probability of success.[5]

[5]There always exists a choice for the internal coin flips that does not decrease $\mathcal{A}$'s probability of success.

A *standard form of generic group steps.* In a generic group operation $\mathcal{A}$ computes some $f_j \in G$ of the form $f_j = \prod_{i<j} f_i^{a_i}$, where $f_1, ..., f_{j-1}$ are the previously computed group elements including the f_i received from the signer/decryptor. The exponents $a_1, ..., a_{j-1} \in \mathbf{Z}_q$ depend arbitrarily on previously computed non-group data. Let the group elements in the input and from signer/decryptor interactions be $g, h, g_1, ...g_\ell$. Then by induction, the f_j are of the *standard form* $f_j = g^{\alpha_{j,0}} h^{\alpha_{j,1}} g_1^{\alpha_{j,2}} ... g_\ell^{\alpha_{j,\ell+1}}$, where the exponent vector $\boldsymbol{\alpha_j} = (\alpha_{j,0}, ..., \alpha_{j,\ell+1}) \in \mathbf{Z}_q^{\ell+2}$ depends arbitrarily on the previously computed non-group data. We require that $\mathcal{A}$ does not identically repeat a generic group operation.[6]

3 Schnorr Signatures, ElGamal Encryption.

We study signer interactions, an interactive protocol that enables a user to generate Schnorr signatures of messages of its choice. We first describe the setting and the structure of the signatures, after which we review the protocol for generation of signatures. We later show how this can be used to generate blind signatures of the same type.

Signatures will be based on an ideal hash function $H : G \times M \to \mathbf{Z}_q$, where M is the set of messages.

Private/public key pairs. The *private key* x of the signer is random in $\mathbf{Z}_q$. The corresponding *public key* is $h = g^x \in G$, a random group element. We have $x = \log_g h$.

Signatures. A Schnorr signature on a message m is a triple $(m, c, z) \in M \times \mathbf{Z}_q^2$ such that $H(g^z h^{-c}, m) = c$. For this paper, we let signatures (m, c, z) comprise the message.

Signing 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)

The result is a valid signature since we have $g^z h^{-c} = g^{r+cx} h^{-c} = g^r$, and thus $H(g^z h^{-c}, m) = c$. We call a signature (m, c, z) constructed by this protocol a *standard signature.*

A signer interaction is an interactive protocol between the signer and a user consisting of three rounds:

signer		*user*
pick $r \in_R \mathbf{Z}_q$		
compute g^r	$\xrightarrow{g^r}$	
	$\xleftarrow{c}$	select $c \in \mathbf{Z}_q$
$z := r + cx$	$\xrightarrow{z}$	

The user can generate the standard signature (m, c, z) by selecting $c := H(g^r, m)$, but he has more options than that. We will study all possibilities to produce signatures by a sequence of arbitrary interactions with the signer. We let $(r, c, z) \in \mathbf{Z}_q^3$ denote the signer interaction consisting of the signer's random choice r, the user's *challenge* c and the signer's *response* z.

Non-interactive proof of knowledge. Schnorr signatures provide a non-interactive proof of knowledge of the secret key $x = \log_g h$. Pointcheval and Stern [PS96] show in the ROM how to transform in polynomial time signature forgeries into a discrete log computation.

[6] The exponent vectors $\boldsymbol{\alpha_i} \in \mathbf{Z}_q^{\ell+2}$ must be pairwise distinct for $i = 1, ..., t'$ and for all instances of the previously computed non-group data. If $\boldsymbol{\alpha_i} = \boldsymbol{\alpha_j}$, $i < j$ we can remove f_j referring to f_i instead.

Blind Signature Protocol.

signer		*user*
picks $r \in_R \mathbf{Z}_q$		picks $\alpha, \beta \in_R \mathbf{Z}_q$
computes g^r	$\xrightarrow{g^r}$	
	$\xleftarrow{c}$	$c := H(g^{r+\alpha}h^{-\beta}, m) - \beta$
$z := r + cx$	$\xrightarrow{z}$	$z' := z + \alpha,\ c' := c + \beta$

A signer interaction (r, c, z) can be used to generate the *standard signature* (m, c, z) or a transformation (m, c', z') of this signature. We call the signature protocol *blind* if it generates a signature (m, c', z') that is statistically independent of the interaction corresponding to the triple (r, c, z). The user can generate such an independent signature (m, c', z') from random numbers $\alpha, \beta \in_R \mathbf{Z}_q$.

Signature Validity. For the output of the interaction $(m, c', z') = (m, c+\beta, z+\alpha)$ we have $g^{z'}h^{-c'} = g^{r+cx+\alpha}h^{-c-\beta} = g^{r+\alpha}h^{-\beta}$. Hence $H(g^{z'}h^{-c'}, m) = c + \beta = c'$, and thus (m, c', z') is a valid signature.

Blindness Property. The generated signature $(m, c+\beta, z+\alpha)$ is — for a constant interaction (r, c, z) — uniformly distributed over all signatures on message m due to the random $\alpha, \beta \in_R \mathbf{Z}_q$. Each signature (m, c', z') is produced for a unique pair $\alpha,\ \beta$, namely $\alpha = z' - z,\ \beta = c' - c$.

3.1 The Power of the Security Model, an Illustration.

This subsection refers to a generic adversary $\mathcal{A}$ that interacts with a signer and computes t' group elements $f_1, ..., f_{t'}$. We let the *Main Case* be the part of the probability space, where there are no collisions among $f_1, ..., f_{t'}$.[7]

Lemma 1. *Collisions among $f_1, ..., f_{t'}$ occur at most with probability $\binom{t'}{2}/q$. The probability refers to the random h, H and the coin flips of the signer.*

Proof. We show for $i < j$ that $\Pr_{x,\mathbf{r},H}[f_i = f_j] \leq \frac{1}{q}$ under the condition that there is no prior collision of group elements. So let us assume that there is no such prior collision. The main point is to show that f_i, f_j either are *s.i.* or f_i/f_j is constant with $f_i \neq f_j$. Let f_j be in standard form $f_j = g^{\alpha_{j,0}}h^{\alpha_{j,1}}g^{r_1\alpha_{j,2}}...\,g^{r_\ell\,\alpha_{j,\ell+1}}$, where the exponent vector $\boldsymbol{\alpha_j} = (\alpha_{j,0}, ..., \alpha_{j,\ell+1}) \in \mathbf{Z}_q^{\ell+2}$ depends arbitrarily on the previously computed non-group data. The signer transmits g^{r_k} in round k, $\mathbf{r} = (r_1, ..., r_\ell)$ consists of the signers coin tosses. Considering x and $r_1, ..., r_\ell$ as indeterminates over $\mathbf{Z}_q$, $\log_g f_j = \langle \boldsymbol{\alpha_j}, (1, x, \mathbf{r}) \rangle$ [8] is a polynomial in $\mathbf{Z}_q[x, r_1, ..., r_\ell]$ of maximal degree 1.

For a *non-interactive* $\mathcal{A}$, where $\ell = 0$ and $\mathbf{r}$ is empty we have $f_i = f_j$ iff $\langle \boldsymbol{\alpha_i} - \boldsymbol{\alpha_j}, (1, x) \rangle = \mathbf{0}$, i.e., iff $\alpha_{i,0} - \alpha_{j,0} + x(\alpha_{i,1} - \alpha_{j,1}) = 0$. Here, $\boldsymbol{\alpha_i}$, $\boldsymbol{\alpha_j}$ depend on x only via random hash values. Therefore, x is *s.i.* of $\boldsymbol{\alpha_i}$, $\boldsymbol{\alpha_j}$, and thus $\Pr_{x,H}[f_i = f_j] \leq \frac{1}{q}$.[9]

[7] A $\binom{t'}{2}/q$-fraction of the probability space, the instances that lead to collisions, is excluded in the Main Case. In the following, we neglect this restriction of the probability space. The impact of this restriction on $\mathcal{A}$'s probability of success is already covered by the probability $\binom{t'}{2}/q$ for collisions.

[8] Let $\langle\ ,\ \rangle$ denote the standard inner product in $\mathbf{Z}_q^{\ell+2}$, $\langle \boldsymbol{\alpha_j}, (1, x, \mathbf{r}) \rangle = \alpha_{j,0} + x\alpha_{j,1} + \sum_{i=2}^{\ell+1} r_i \alpha_{j,i+1}$.

[9] The equality $f_i = f_j$ holds with zero probability if $\alpha_{i,0} \neq \alpha_{j,0}$ and $\alpha_{i,1} = \alpha_{j,1}$. The case that $\boldsymbol{\alpha_i} = \boldsymbol{\alpha_j}$ has been excluded, in this case f_j identically recomputes f_i.

Next consider an *interactive* $\mathcal{A}$. We call r_k *prior* to f_j if $\boldsymbol{\alpha_j}$ depends on the signer response $z_k = r_k + c_k x$, otherwise r_k is *subsequent* to f_j, i.e., $\boldsymbol{\alpha_j}$ does not depend on z_k. The probability space of $f_j = g^{\langle \boldsymbol{\alpha_j}, (1,x,\mathbf{r})\rangle}$ consist of x, H and the r_k subsequent to f_j — the $r_k = z_k - c_k x$ prior to f_j are linear in x, with constants z_k, c_k. Consider $\langle \boldsymbol{\alpha_j}, (1, x, \mathbf{r})\rangle$ as a function in x and the r_k subsequent to f_j. The vector $\boldsymbol{\alpha_j} \in \mathbf{Z}_q^{\ell+2}$ depends on $x, H, \mathbf{r}$ only via prior r_k and via random hash values, and thus x and the subsequent r_k are *s.i.* of $\boldsymbol{\alpha_j}$. Therefore, $\langle \boldsymbol{\alpha_i} - \boldsymbol{\alpha_j}, (1, x, \mathbf{r})\rangle$ is either constant or uniformly distributed over $\mathbf{Z}_q$. The case that $\langle \boldsymbol{\alpha_i} - \boldsymbol{\alpha_j}, (1, x, \mathbf{r})\rangle = 0$ holds for all x and all r_k subsequent to f_j has been excluded.[10] This shows that $\Pr_{x,\mathbf{r},H}[f_i = f_j] \leq \frac{1}{q}$, which implies the claim of Lemma 2 as there are $\binom{t'}{2}$ pairs $i < j$. □

Lemma 2. *Let $\mathcal{A}$ be given the input g and $h = g^x \in_R G$. In the Main Case we have that*

- *the* p.d. (*probability distribution*) *of the non-group data is constant as h varies.*
- *the random $h \in_R G$ is* s.i. (*statistically independent*) *of the computed non-group data.*

Proof. Collisions of group elements $f_i = f_j,\ i < j$ are non-group data that may depend on $h = g^x$, in particular if f_i, f_j contain powers of h. Now suppose that there is no such collision. Then the public key $h = g^x$, respectively $x = \log_g h$, enters into $\mathcal{A}$'s non-group data only by queries to the hash oracle and by interactions with the signer.

In the ROM hash values $H(f_i, m)$ are random, their distribution does not change with the query (f_i, m). In a signer interaction $\mathcal{A}$ gets the pair (g^{r_k}, z_k), where $r_k \in \mathbf{Z}_k$ is random and $z_k = r_k + c_k x$. Due to the random r_k the distribution of z_k is constant as $h = g^x$ varies. Therefore, the *p.d.* of the non-group data generated from hash values and signer responses is constant as h varies. In particular, the public key $h = g^x$ and thus x is *s.i.* of all non-group data ($h = g^x$ is NOT *s.i.* of (g^{r_k}, z_k), however g^{r_k} enters into the computation of non-group data only by collisions of group elements and via random hash values). □

Proposition 3. Complexity Lower Bound for Discrete logarithm [Ne94,Sh97].
Let $\mathcal{A}$, upon input g and $h \in_R G$, output $y \in \mathbf{Z}_q$. Then $\Pr_h[y = \log_g h] \leq \binom{t'}{2}/q + \frac{1}{q}$.

Proof. In the Main Case h is by Lemma 2 *s.i.* of the non-group output y, and thus $\Pr_h[y = \log_g h] = \frac{1}{q}$. By Lemma 1, collisions occur at most with probability $\binom{t'}{2}/q$. □

An *ElGamal ciphertext* of a message $m \in G$ with the public key $h \in G$ is a pair $(g^r, mh^r) \in G^2$, where $r \in_R \mathbf{Z}_q$ is random.

Proposition 4. Semantic Security of ElGamal Encryption. *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* $\Pr_{b,h,r}[b' = b] \leq \binom{t'}{2}/q + \frac{1}{2}$.

Proof. Consider the impact of permuting m_0, m_1 in $\mathcal{A}$'s input without changing g, h, b, m_b. We apply the argument of Lemma 2 to the transformed input. In the Main Case that transform does not affect the *p.d.* of the non-group output b'. In the Main Case b is *s.i.* of b' and thus $\Pr_{b,h,r}[b' = b] = \frac{1}{2}$. The Main Case occurs except with probability $\binom{t'}{2}/q$. □

[10] Then f_j identically repeats — under the condition $z_k = r_k + c_k x$ — the computation of f_i, so we can remove f_j from the computation referring to f_i instead.

4 Security of Signed ElGamal Encryption

We study the security of signed ElGamal encryption in the ROM + GM. Signed ElGamal encryption was independently proposed by Tsiounis and Yung [TY98] and Jakobsson [J98]. This scheme is semantically secure against the adaptive chosen ciphertext attack (Theorem 5). This is equivalent to non-malleability against chosen ciphertext attacks [DDN91]. We refer to non-malleability as defined in [DDN91] 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 challenge ciphertext. We let the adversary interact with a decryption oracle without that any further party is involved. Therefore, the adversary can only use ciphertexts that are either given for input or are self-produced.

We show in Theorem 6 that signed ElGamal encryption is secure against the one-more decryption attack: An adversary can — after arbitrary ℓ interactions with the decryption oracle — not decrypt more than ℓ ciphertexts, more precisely he gets non-zero information on at most ℓ of the corresponding plaintexts. The adversary can arbitrarily use the challenge ciphertexts for queries to the decryptor which is excluded in adaptive chosen ciphertext attacks.

The private/public key pair x, $h = g^x$ of the decryptor is as for the signer: x is random in $\mathbf{Z}_q$. We let messages be contained in G or more generally in G^n for an arbitrary natural number n. We use an ideal hash function $H : G^{n+2} \to \mathbf{Z}_q$. The domain G^{n+2} of H is different from the hash function for signing.

Enciphering a message $m \in G$ by signed ElGamal encryption. Pick random $r, s \in_R \mathbf{Z}_q$, compute $\bar{h} := g^r$, $\bar{f} := m\,h^r$, $c := H(g^s, g^r, mh^r)$ and $z := s + cr$.
Output the ciphertext $(g^r, mh^r, c, z) = (\bar{h}, \bar{f}, c, z)$.

A decryption oracle is a function that decrypts valid ciphertexts :

user		*decryptor*
$(\bar{h}, \bar{f}, c, z)$	$\xrightarrow{\bar{h}, \bar{f}, c, z}$	$H(g^z \bar{h}^{-c}, \bar{h}, \bar{f}) \stackrel{?}{=} c$
		if "yes" $F := \bar{h}^x$
$m := \bar{f}/F$	$\xleftarrow{F}$	if "no" $F := ?$

The decryption is correct as we have for $\bar{h} = g^r$, $\bar{f} = m\,h^r$ that $\bar{f}/F = m\,g^{rx}g^{-rx} = m$.

Remarks **1.** The ciphertext $(\bar{h}, \bar{f}, c, z)$ consists of an ElGamal ciphertext $(\bar{h}, \bar{f})$ and a Schnorr signature (c, z) on the "message" $(\bar{h}, \bar{f})$. The signature is for the private/public key pair $(r, \bar{h})$.

2. *Fast encryption with a data expansion rate near* 1. We let the messages have a different format. Let the message be a sequence $\mathbf{m} = (m_1, ..., m_n) \in \mathbf{Z}_q^n$ and let $H_n : G \to \mathbf{Z}_q^n$ be a hash function that provides "long" hash values[11] in $\mathbf{Z}_q^n$. Encipher a message $\mathbf{m}$ into $(\bar{h}, \bar{f}, c, z)$ where $\bar{h} = g^r$, $\bar{f} = \mathbf{m} \oplus_q H_n(h^r)$, $c = H_n(g^s, \bar{h}, \bar{f})$, $z = s + cr$ for $r, s \in_R \mathbf{Z}_q$. Here $\oplus_q$ is the component-wise addition in $\mathbf{Z}_q^n$. Decrypt the ciphertext $(\bar{h}, \bar{f}, c, z)$ into $\bar{f} \oplus_q H_n(\bar{h}^x)$ provided that (c, z) is a signature of the message $(\bar{h}, \bar{f})$ with public key $\bar{h}$, i.e. $c = H_n(g^z \bar{h}^{-c}, \bar{h}, \bar{f})$.

[11] Long hash values can be generated using a standard hash function H according to the following, or some related, approach: $H_n(m) = (H(m, 1), \ldots, H(m, n))$, where n is the constant length (in terms of short hash function outputs) of the desired hash function.

The bit length of the ciphertext is $\log_2 \|G\|+(n+2)\log_2 q$, where 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 ciphertexts are as secure as the original ones. Encryption requires only a long 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. *Without encoding of messages.* Signed ElGamal encryption requires that messages m are encoded into the group G so that we can form mh^r. Not all groups allow a natural algorithm that encodes bit strings into group elements. Remark 2. shows, how to replace that encoding by a hash function $H_n : G \to \mathbf{Z_q}$.

4.1 Security against Interactive Attacks

Adaptive chosen ciphertext attacks. The next theorem proves semantic security against an adversary $\mathcal{A}$ mounting an adaptive chosen ciphertext attack. In this attack the adversary is given a challenge ciphertext and a decryption oracle for the decryption of arbitrary ciphertexts except for the challenge. 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 t generic steps: group operations, inputs in G, queries to the oracle H, queries to the decryption oracle not including the challenge ciphertext.

Theorem 5. *Let $\mathcal{A}$ be given g, h, distinct messages m_0, m_1, a challenge 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 $\frac{1}{2}+\binom{t}{2}/q$. The probability space consists of the random $h = g^x, H, b$ and the coin tosses of the encipherer.*

The next theorem shows security against the one-more decryption attack. This is interesting in a setting where users are charged for interactions with the decryption oracle.

Theorem 6. *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 $\ell < d$ arbitrary queries to the decryption oracle. Then $\mathcal{A}$ cannot produce $\ell + 1$ message-ciphertext pairs with a better probability than $\frac{1}{d-\ell} + \binom{t}{2}/q$. The probability space consists of the random h, H, the coin tosses of the encipherer and the random ordering of the messages.*

Proof of Theorem 5. For a non-interactive adversary $\mathcal{A}$ the claim follows from Prop. 4 because the Schnorr signature in cip_b does not help $\mathcal{A}$ to decrypt.

We transform $\mathcal{A}$ into a non-interactive adversary $\mathcal{A}'$ in that we successively eliminate the first interaction with the decryptor. Let the first query to the decryptor be about the ciphertext $(\bar{h}, \bar{f}, c, z)$. Let this be a valid ciphertext — $\mathcal{A}$ can check that validity and does not get any information for invalid ciphertexts. Then $\mathcal{A}$ has produced that ciphertext without interacting with the decryptor. Recall that $((\bar{h}, \bar{f}), c, z)$ is a Schnorr signature of the "message" $(\bar{h}, \bar{f})$ under the private/public key pair $(\log_g \bar{h}, \bar{h})$. The equation $c = H(g^z \bar{h}^{-c}, \bar{h}, \bar{f})$, required for a valid signature, necessitates that the hash value $H(g^z \bar{h}^{-c}, \bar{h}, \bar{f})$ is determined prior to c, z — otherwise the equation holds with probability $\frac{1}{q}$ as the hash value is random. We see that the hash value is of the form $H(f, \bar{h}, \bar{f})$, where f is among the computed group elements

$f_1, ..., f_{t'}$. $\mathcal{A}$ gets $c = H(f, \bar{h}, \bar{f})$ from the hash oracle and must compute z so that $g^z \bar{h}^{-c} = f$, i.e., $z = \log_g f + c \log_g \bar{h}$.

Now, consider the *Main Case* in the computation of $(\bar{h}, \bar{f}, c, z)$, where there is no collision among the computed group elements $f_1, ..., f_{t'}$. In the Main Case the *p.d.* of z does not depend on $h, \bar{h}$ whereas $\log_g f + c \log_g \bar{h}$ may depend on h and $\bar{h}$. We distinguish the two values. We let $z' = \log_g f + c \log_g \bar{h}$ denote the value required for a signature, whereas z denotes the computed value in $(\bar{h}, \bar{f}, c, z)$. Let the challenge ciphertext be $cip_b = (g^r, m_b h^r, c_b, z_b)$, where $r \in_R \mathbf{Z}_q$ is secret, and let $\mathcal{A}$ be given $\log_g(m_0), \log_g(m_1)$ (this can only facilitate $\mathcal{A}$'s task). Then $\mathcal{A}$'s group steps refer exclusively to the group elements $(f_1, f_2, f_3, f_4) = (g, h, g^r, m_b h^r)$, all given group elements are known powers of $f_1, ..., f_4$. $\mathcal{A}$ computes group elements of the form $f_i := \prod_{\nu=1}^{4} f_\nu^{\alpha_{i,\nu}}$ for $i = 1, ..., t'$, where the exponent vector $\boldsymbol{\alpha_i} = (\alpha_{i,1}, ..., \alpha_{i,4}) \in \mathbf{Z}_q^4$ depends arbitrarily on previous non-group data. Let $f = f_i$, $\bar{h} = f_j$, then we have

$$z' = \log_g f_i + c \log_g f_j = \langle \boldsymbol{\alpha_i} + c\boldsymbol{\alpha_j}, (1, x, r, \log_g(m_b) + xr) \rangle.$$

Considering x, r as formal variables, z' is a polynomial in $\mathbf{Z}_q[x, r]$ that is linear in r, x. The required value z' evaluates that polynomial at instances x, r that are *s.i.* of the polynomial coefficients. By the argument of Lemma 2, the *p.d.* of z does not depend on $\bar{h} = g^r, h = g^x$ and thus does not depend on r, x. We conclude that r, x must cancel out in the formal polynomial z' or else we have $\Pr[z = z'] \leq \frac{1}{q}$. More in details we show the

Fact. If $z = z'$ then we have $\tilde{\boldsymbol{\alpha}} =_{\text{def}} (\alpha_{i,2}, \alpha_{i,3}, \alpha_{i,4}, \alpha_{j,2}, \alpha_{j,3}, \alpha_{j,4}) = \mathbf{0}$ and $\log_g \bar{h} = \alpha_{j,0}$ except for an event of probability $\frac{1}{q}$.

Proof. By the argument of Lemma 2, the random g^r, g^x, r, x are *s.i.* of the non-group data $z, \boldsymbol{\alpha_i}, \boldsymbol{\alpha_j}$ — the *p.d.* of z does not depend on $\bar{h} = g^r, h = g^x$ and thus does not depend on r, x. If $\tilde{\boldsymbol{\alpha}} \neq 0$ then $z' = \langle \boldsymbol{\alpha_i} + c\boldsymbol{\alpha_j}, (1, x, r, \log_g(m_b) + xr) \rangle$ is — due to the random r, x — uniformly distributed over $\mathbf{Z}_q$. By the argument of Lemma 2, g^x, g^r, x, r are *s.i.* of the non-group data z, and thus z' is *s.i.* of z or else r, x must cancel out in z'. As z' contains the linear terms $\alpha_{i,2}x + c\alpha_{j,2}x + \alpha_{i,3}r + c\alpha_{j,3}r$ and c is random, if x, r cancel out, we have $\alpha_{i,2} = \alpha_{i,3} = \alpha_{j,2} = \alpha_{j,3} = 0$. Similarly, $\alpha_{i,4} = \alpha_{j,4} = 0$ is necessary to cancel out the monomial xr in z'. We see that $\Pr_{r,h,H}[z = z', \tilde{\boldsymbol{\alpha}} \neq 0] \leq \frac{1}{q}$. This proves the claim as $\tilde{\boldsymbol{\alpha}} = \mathbf{0}$ implies $\log_g \bar{h} = \log_g f_j = \alpha_{j,0}$.

Eliminating the interactions with the decryptor. The plaintext corresponding to $(\bar{h}, \bar{f}, c, z)$ is $\bar{f}/h^{\log_g \bar{h}} = \bar{f}/h^{\alpha_{j,0}}$ except for an event of probability $\frac{1}{q}$. This eliminates the first interaction with the decryptor without increasing $\mathcal{A}$'s number of generic steps, and reduces $\mathcal{A}$'s probability of success by $\frac{1}{q}$ in worst case. (This method is impossible in the Turing machine model.[12]) Let there be ℓ interactions with the decryptor. We iteratively eliminate them by the above method. This transforms $\mathcal{A}$ into a non-interactive generic $\mathcal{A}'$ that performs t generic steps. By Prop. 4 the non-interactive $\mathcal{A}'$ predicts b not better than with probability $\binom{t'}{2}/q + \frac{1}{2}$. We have seen that the probabilities of success of $\mathcal{A}$ and $\mathcal{A}'$ differ by ℓ/q in worst case. Therefore, $\mathcal{A}$ predicts b not better than with probability $\ell/q + \binom{t'}{2}/q + \frac{1}{2} \leq \binom{t}{2}/q + \frac{1}{2}$ as $t' + \ell \leq t$.

Plaintext awareness. The above method — of extracting from $\mathcal{A}$ the signature key $\log_g \bar{h}$ and the plaintext corresponding to the constructed ciphertext — shows that signed ElGamal

[12] In the Turing machine model, eliminating an interaction with the decryptor increases the number of Turing machine steps by a constant factor δ^{-1}, where δ is $\mathcal{A}$'s probability of success in producing a valid ciphertext $(\bar{h}, \bar{f}, c, z)$. Here the extractor has to try δ^{-1} statistically independent hash functions in place of H.

encryption is plaintext aware in the sense defined in [BR94]. □

Proof sketch of Theorem 6. As signed ElGamal encryption is plaintext aware, 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 statistically independent. 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 Prop. 4 applies. Consider in the Main Case, where there is no collision of group elements, the impact of a random permutation of the indices i of the remaining $d-\ell$ messages m_i. By the argument of Lemma 2, that random permutation of the indices is *s.i.* of $\mathcal{A}'$'s guess of a correct message-ciphertext pair (m_i, cip_j). Therefore, $\mathcal{A}'$ cannot guess a correct pair (m_i, cip_j) better than with probability $\frac{1}{d-\ell}$. The Main Case occurs except with probability $\binom{t}{2}/q$, hence the claim. □

5 Security of Signatures against Interactive Attacks

Theorems 7 and 8 show that Schnorr signatures are secure against the one-more signature forgery in the ROM + GM. These theorems cover blind signatures as required for anonymous electronic cash. This is the first sharp security result for simple discrete log signatures in the interactive setting. We characterize the different power of sequential and of parallel attacks. Parallel attacks that beat the success rate $\binom{t}{2}/q$ of sequential attacks must solve an established hard problem [H97]: they must find an intersection point of $\ell+1$ hyperplanes in $\mathbf{Z}_q^\ell$ out of a collection of randomized hyperplanes.

Theorem 7. *Let a generic adversary $\mathcal{A}$ interact with the signer and be given the generator g, the public key h and an oracle for H. Let $\mathcal{A}$ perform t generic steps including ℓ* sequential *signer interactions. Then $\mathcal{A}$ cannot produce $\ell+1$ signatures with a better probability than $\binom{t}{2}/q$. The probability space consists of h, H and the coin flips of the signer.*

Theorem 8. *If a generic adversary $\mathcal{A}$ surpassess in a* parallel *attack the $\binom{t}{2}/q$-bound of* Theorem 7*, he solves the following problem. Given an oracle for a random function F, $\mathcal{A}$ finds distinct vectors $\boldsymbol{\alpha_i} \in \mathbf{Z}_q^\ell$ for $i = 1, ..., t$ and an intersection point of $\ell+1$ of the t hyperplanes $H_i = \{\mathbf{x} \in \mathbf{Z}_q^\ell | \langle \boldsymbol{\alpha_i}, \mathbf{x}\rangle = F(\boldsymbol{\alpha_i})\}$. The fastest known non-generic method to find such an intersection point does not succeed better than with probability $\binom{T}{2}/q$ when using T arithmetic steps over $\mathbf{Z}_q$.*

Proof of Theorems 7 and 8. *Notation.* We let (r_k, c_k, z_k) for $k = 1, ..., \ell$ denote the interactions of $\mathcal{A}$ with the signer. The signer correctly transmits g^{r_k} and responds to $\mathcal{A}$'s challenge c_k by $z_k = r_k + c_k x$. We abbreviate $\mathbf{r} := (r_1, ..., r_\ell)$ and $g^{\mathbf{r}} := \{g^{r_1}, ..., g^{r_\ell}\}$. We let $f_1, ..., f_{t'}$ denote the group elements of $\mathcal{A}$'s computation. In the generic model $\mathcal{A}$ computes f_k of the form $f_k = g^{\alpha_{k,0}} h^{\alpha_{k,1}} g^{r_1\alpha_{k,2}+...+r_\ell\alpha_{k,\ell+1}} = g^{\langle \boldsymbol{\alpha_k}, (1,x,\mathbf{r})\rangle}$, where the *exponent vector* $\boldsymbol{\alpha_k} = (\alpha_{k,0}, ..., \alpha_{k,\ell+1}) \in \mathbf{Z}_q^{\ell+2}$ merely depends on previous collisions $f_i = f_j$, $i < j < k$, on previous signer responses and hash values. We prove in Lemma 9 that the group element $g^{z'_i} h^{-c'_i}$ corresponding to a signature (m'_i, c'_i, z'_i) must be among $f_1, ..., f_{t'}$ — we let i' denote

its index, $g^{z'_i}h^{-c'_i} = f_{i'} = g^{\langle \boldsymbol{\alpha}_{\boldsymbol{i'}},(1,x,\mathbf{r})\rangle}$. Let there be t'' (distinct) queries to the hash oracle resulting in t'' independent hash values in $\mathbf{Z}_q$. By assumption we have $t' + t'' = t$.

Informal survey. It is assumed that $\mathcal{A}$ outputs distinct triples $(m'_i, c'_i, z'_i) \in M \times \mathbf{Z}_q^2$ for $i = 1, ..., \ell + 1$. The event that these are $\ell + 1$ signatures is claimed to have no better probability then $\binom{t}{2}/q$ for random $h, H, \mathbf{r}$. By Lemma 9 we have $g^{z'_i}h^{-c'_i} = f_{i'}$ for some $i' \leq t'$ and $\mathcal{A}$ receives c'_i as the hash value $H(f_{i'}, m'_i)$. $\mathcal{A}$ must find the corresponding z'_i which by $g^{z'_i}h^{-c'_i} = g^{\langle \boldsymbol{\alpha}_{\boldsymbol{i'}},(1,x,\mathbf{r})\rangle}$ satisfies $z_i = \langle \boldsymbol{\alpha}_{\boldsymbol{i'}}, (1, x, \mathbf{r})\rangle + c'_i x$. Hence, $\mathcal{A}$ must evaluate the linear polynomial $\langle \boldsymbol{\alpha}_{\boldsymbol{i'}}, (1, x, \mathbf{r})\rangle + c'_i x$ at $(x, \mathbf{r})$, where by Lemma 2 x is *s.i.* of $(\boldsymbol{\alpha}_{\boldsymbol{i'}}, c'_i)$, except that there is a prior collision $f_j = f_k$. By Lemma 1, collisions $f_j = f_k$ occur at most with probability $\binom{t'}{2}/q$. Consider the *Main Case*, where there are no collisions. In the Main Case, $\mathcal{A}$ can only succeed with (m'_i, c'_i, z'_i) if x cancels out in the polynomial $\langle \boldsymbol{\alpha}_{\boldsymbol{i'}}, (1, x, \mathbf{r})\rangle + c'_i x$. In order to cancel x out the challenges $c_1, ..., c_\ell$ must by Lemma 9 satisfy the equations $c'_i = -\alpha_{i',1} + \sum_{k=1}^{\ell} \alpha_{i',k+1} c_k$. This yields for each signature (m'_i, c'_i, z'_i) a linear equation for $c_1, ..., c_\ell$.

We separately study the complexity of sequential and of parallel attacks. In a *sequential* attack the system of $\ell + 1$ equations for $c_1, ..., c_\ell$ is solvable with probability at most $\binom{t}{2}/q$. The fastest known *parallel*, non-generic algorithm for solving the $\ell + 1$ equations for $c_1, ..., c_\ell$ succeeds with probability $\binom{T}{2}/q$ using T arithmetic steps .

Lemma 9. *Let the output (m'_i, c'_i, z'_i) be a signature with a better probability than $\frac{1}{q}$. Then c'_i coincides with the value $H(f, m)$ corresponding to some hash query $(f, m) = (g^{z'_i}h^{-c'_i}, m'_i)$, where $f = f_{i'}$ for some $1 \leq i' \leq t'$. Moreover, c'_i, z'_i, i' satisfy the equations $z'_i = \alpha_{i',0} + \sum_{k=1}^{\ell} \alpha_{i',k+1} z_k$ and $c'_i = -\alpha_{i',1} + \sum_{k=1}^{\ell} \alpha_{i',k+1} c_k$.*

Proof. The first claim follows from the equation $c'_i = H(g^{z'_i}h^{-c'_i}, m'_i)$ required for signatures (m'_i, c'_i, z'_i). In the ROM this equation necessitates that $\mathcal{A}$ has queried that hash value — otherwise the equality only holds with probability $\frac{1}{q}$ as the hash value is random. Thus the hash value $c'_i = H(f, m)$ results from a query $(f, m) \in G \times M$, where $(f, m) = (g^{z'_i}h^{-c'_i}, m'_i)$ holds for the output (m'_i, c'_i, z'_i). We let $1 \leq i' \leq t'$ denote the index of f among the computed group elements $f_1, ..., f_{t'}$, and thus $f_{i'} = g^{z'_i}h^{-c'_i} = g^{\langle \boldsymbol{\alpha}_{\boldsymbol{i'}},(1,x,\mathbf{r})\rangle}$.[13] The latter equations and $r_k = z_k - c_k x$ imply

$$\begin{aligned} z'_i &= \log_g g^{z'_i}h^{-c'_i} + c'_i x = \langle \boldsymbol{\alpha}_{\boldsymbol{i'}}, (1, x, \mathbf{r})\rangle + c'_i x \\ z'_i &= \alpha_{i',0} + \textstyle\sum_{k=1}^{\ell} \alpha_{i',k+1} z_k + (\, \alpha_{i',1} - \sum_{k=1}^{\ell} \alpha_{i',k+1} c_k \; + c'_i\,) x, \end{aligned} \tag{1}$$

$\mathcal{A}$ can easily compute the correct z'_i in the particular case that $c'_i = -\alpha_{i',1} + \sum_{k=1}^{\ell} \alpha_{i',k+1} c_k$. The secret key x cancels out in this case and we have $z'_i = \alpha_{i',0} + \sum_{k=1}^{\ell} \alpha_{i',k+1} z_k$, where the signer responses $z_1, ..., z_\ell$ and the coefficients $\alpha_{i',0}, \ldots, \alpha_{i',\ell+1}$ are known to $\mathcal{A}$.

Conversely, $\mathcal{A}$ must select $c_1, ..., c_\ell$ so that the secret key x cancels out in (1). Otherwise, the equation (1) holds merely with probability $\frac{1}{q}$ as x is by Lemma 2 *s.i.* of the non-group data $z'_i, \alpha_{i',0}, ..., \alpha_{i',\ell+1}, c_1, ..., c_\ell$. If x does not cancel out, $\mathcal{A}$'s probability of success is not

[13] The index $1 \leq i' \leq t'$ is determined by the output (m'_i, c'_i, z'_i) via the equation $f_{i'} = g^{z'_i}h^{-c'_i}$. We use that collisions of hash values are quite unlikely. The event that there is either a collision of group elements or a collision of hash values occurs with probability at most $\binom{t}{2}/q$. For simplicity we abbreviate $f_{i'} = g^{z'_i}h^{-c'_i}$ even though that equation only holds a posteriori, i' and $\boldsymbol{\alpha}_{\boldsymbol{i'}}$ depend on $h, H, \mathbf{r}$.

better than $\frac{1}{q}$. This proves the Lemma as the claimed equations hold if x cancels out. $\square$

Now suppose that $\mathcal{A}$ outputs $\ell+1$ signatures (m'_i, c'_i, z'_i) for $i = 1, ..., \ell+1$. W.l.o.g. let $\mathcal{A}$ query the hash oracle about the t'' values $H(f_i, m_i)$ for $i = 1, ..., t''$. $\mathcal{A}$'s selection of $\ell+1$ hash values $H(f_{i'}, m'_i)$ out of these t'' values is defined by the output signatures (m'_i, c'_i, z'_i) via $c'_i = H(f_{i'}, m'_i)$, where $f_{i'} = g^{\langle \boldsymbol{\alpha_{i'}}, (1,x,\mathbf{r})\rangle}$. $\mathcal{A}$'s challenges $c_1, ..., c_\ell$ must solve the equations of Lemma 9:

$$H(f_{i'}, m'_i) = -\alpha_{i',1} + \textstyle\sum_{k=1}^{\ell} \alpha_{i',k+1} c_k \quad \text{for } i = 1, ..., \ell+1 \tag{2}$$

Each satisfied equation yields a signature. Each of the t'' hash queries defines by the corresponding equation (2) a hyperplane of solutions $(c_1, ..., c_\ell)$ in $\mathbf{Z}_q^\ell$. $\mathcal{A}$'s challenges $(c_1, ..., c_\ell)$ form an intersection point of $\ell+1$ of these hyperplanes.

Complexity of parallel attacks. Consider a particular selection of $\ell+1$ hash values $H(f_{i'}, m'_i)$ out of the t'' given hash values $H(f_i, m_i)$. The corresponding $\ell+1$ equations (2) in the unknowns $c_1, ..., c_\ell$ are solvable with probability at most $\frac{1}{q}$. The probability refers to the random inhomogenities $H(f_{i'}, m'_i) + \alpha_{i',1}$ for $i = 1, ..., \ell+1$ and does not depend on $\mathcal{A}$'s choice of $f_{i'}, m'_i, \boldsymbol{\alpha_i}$. As there are $\binom{t''}{\ell+1}$ possible choices of the $\ell+1$ out of t'' hash values, the equations (2) are solvable with no better probability than $\binom{t''}{\ell+1}/q$. Finding an intersection point of $\ell+1$ hyperplanes is free of costs in the generic model. But now consider the non-generic costs in terms of arithmetic steps:

A simple method to find a common intersection point of $\ell+1$ hyperplanes tries all $\binom{t''}{\ell}$ choices of ℓ out of the t'' hyperplanes — in general ℓ of the hyperplanes have a common intersection point — sorts these $\binom{t''}{\ell}$ intersection points and checks for a collision. This requires at least $\binom{t''}{\ell}$ arithmetic steps over $\mathbf{Z}_q$. That method is *optimal* for $\ell = 1$.[14]

In a variant of the above method one forms for a subfamily of some T — from the $\binom{t''}{\ell}$ choices of ℓ out of t'' hyperplanes — the T intersection points of the corresponding ℓ-tuples of hyperplanes, and one searches for a collision of the intersection points. Two intersection points coincide with probability at most $\frac{1}{q}$. Hence, the restricted search succeeds not better than with probability $\binom{T}{2}/q$ using T arithmetic steps over $\mathbf{Z}_q$ — no matter, how we select the family of T ℓ-tuples of hyperplanes. The success probability $\binom{T}{2}/q$ is analogous to the success probability $\binom{t}{2}/q$ in Theorems 7 and 8, however T counts arithmetic steps while t counts generic steps. We pose it as an open problem to beat that $\binom{T}{2}/q$ bound.

From the work of Hastad [H97] we know that beating the naive method — in maximizing a subset of hyperplanes with a common intersection point — is NP-hard. The approximation problem considered in [H97] is NP-hard in worst case — our problem seems to be hard for almost all instances. This is because the t'' inhomogenities $H(f_i, m_i) + \alpha_{i,1}$ of the hyperplanes are independent random numbers due the random oracle H. It seems to be irrelevant that the adversary can choose himself the vectors $\boldsymbol{\alpha_i}$.

Complexity of sequential attacks. A sequential attack iterates a sequence of parallel attacks with $\ell = 1$. In the Main Case sequential attacks do not succeed better than with probability $\binom{t''}{2}/q$. As the Main Case occurs except with probability $\binom{t'}{2}/2$, and $\binom{t''}{2} + \binom{t'}{2} \le \binom{t}{2}$, sequential attacks do not succeed better than with probability $\binom{t}{2}/q$. Sequential attacks can

[14] For $\ell = 1$ there is an intersection point of 2 of the t'' hyperplanes iff there is a collision among the $\mathbf{Z}_q$-numbers $(H(f_i, m_i) - \alpha_{i,2})/\alpha_{i,1}$ for $i = 1, ..., t''$. These $\mathbf{Z}_q$-numbers are pairwise *s.i.* due to the random H. A collision occurs with probability $\binom{t''}{2}/q$, it can be found in $O(t'' \log t'')$ arithmetic steps by sorting.

be performed efficiently on the non-group data.[15] □

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

[BM89] *M. Bellare and S. Micali*: How to Sign Given Any Trapdoor Function. Proc. Crypto'88, LNCS 403, Springer-Verlag, pp. 200–215, 1989.

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

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

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

[CGH98] *R. Canetti, O. Goldreich and S. Halevi*: The Random Oracle Methodology, Revisited. Proc. 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. Proc. Crypto'98, LNCS 1462, Springer-Verlag, pp. 13–25, 1998.

[DH76] *W. Diffie and M.E. Hellman*: New Directions in Cryptography. In IEEE Transactions on Information Theory, vol IT-22, no. 6, pp. 644–654, November 1976.

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

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

[G85] *ElGamal*: A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. IEEE Trans. Inform. Theory, 31, pp. 469–472, 1985.

[GHY89] *Z. Galil, S. Haber, and M. Yung*, A Secure Public-Key Authentication Scheme. Proc. Eurocrypt'89, LNCS 434, Springer-Verlag, pp. 3-15, 1990.

[H97] *J. Hastad*: Some Optimal Approximability Reseults. Proc. ACM Symposium on Theory of Computing 1997, ACM Press, pp. 1–10, 1997.

[15] One equation (2) can be satisfied per choice of c_i: set $\alpha_{i',i} = 1$, $\alpha_{i',i+1} = ...\alpha_{i',\ell} = 0$ and determines c_i as to satisfy the i-th equation (1). With probability ${}^{t''}_{2}/q$ there will be a collision satisfying an additional equation (2). There are $O(t'' \log t'')$ arithmetic steps on non-group data.

[J98] *M. Jakobsson*: A Practical Mix. Proc. Eurocrypt'95, LNCS 1403, Springer-Verlag, pp. 448–461, 1998.

[JLO97] *A. Juels, M. Luby, and R. Ostrovsky*: Security of Blind Digital Signatures. Proc. Crypto'97, LNCS 1294, Springer-Verlag, pp. 150–164, 1997.

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

[N94] *NIST*: "Digital Signature Standard (DSS), Federal Information Processing Standard" PuB 186, 1994 May 19.

[O92] *T. Okamoto*: Provably Secure Identification Schemes and Corresponding Signature Schemes. Proc. Crypto'92, LNCS 740, Springer-Verlag, pp. 31–53, 1992.

[PS96a] *D. Pointcheval and J. Stern*: Security Proofs for Signature Schemes. Proc. Eurocrypt'96, LNCS 1070, Springer-Verlag, pp. 387–398, 1996.

[PS96b] *D. Pointcheval and J. Stern*: Provably Secure Blind Signature Schemes. Proc. Asiacrypt'96, LNCS 1163, Springer Verlag, pp. 387–393, 1996.

[P98] *D. Pointcheval*: Strengthened Security for Blind Signatures. Proc. Eurocrypt'98 LNCS 1403, Springer Verlag, pp. 391–405, 1998.

[RS92] *C. Rackoff and D. Simon*: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. Proc. Crypto'92, LNCS 576, Springer-Verlag, pp. 433–444.

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

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

[Sh98] *V. Shoup,* personal communication.

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

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

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