Security of DL-encryption and signatures against generic attacks—a survey

Claus Peter Schnorr

Abstract. We survey recent results on the security of DL-cryptosystems and DL-signatures against generic attacks [Sh97, SJ99, SJ00] assuming the random oracle model (ROM) and the generic group model (GM). We comment on the relevance of these results towards applications.

Key words and phrases: Generic algorithm, generic group model, random oracle model, signed ElGamal encryption, generalized signed ElGamal encryption, one-more decryption attack, chosen ciphertext attack, digital signature, DL-signature, one-more signature forgery, blind Schnorr signature, interleaved, parallel attack.

1. Introduction and summary

While there are no security proofs without unproven assumptions the most important practical attacks against DL-signatures and DL-cryptosystems are generic in that either the hash function or the cyclic group or both are modelled by black-box oracles. The random oracle model (ROM) assumes an ideal random hash function while the generic group model (GM) assumes an ideal group G of prime order q with group operations done by a group oracle. The ROM is a stronger assumption than assuming the hash function to be collision resistant—collisions of random functions have negligibly small probability. The GM is a stronger assumption than assuming that the discrete logarithm is intractable—it has been shown that the discrete logarithm has exponential complexity in the GM [Ne94, Sh97]. The GM is appropriate for elliptic and hyperelliptic curves as the known attacks are all generic. The GM is less appropriate for groups of units as it does not cover attacks using sieving or index calculus. The ROM goes back to Fiat and Shamir [FS86] and has been further enhanced by Bellare and Rogaway [BR93], while the generic group model goes back to Nechaev [Ne94] and Shoup [Sh97].

Assuming a combination of the ROM and the GM we study security against attacks that do not exploit particular weaknesses of the hash function H or of the group G or of a combination of G and H. While we do not have to rely on any unproven assumption, it is the case that our security guarantee hinges on the existence of strong hash functions H and groups G for which the combination

(G, H) has no weaknesses. The strong assumption of the ROM+GM makes sense in cases where traditional security proofs fail. Prominent examples are the security of Schnorr identification against active adversaries shown in [Sh97], security of signed ElGamal encryption [SJ00] and of blind Schnorr signatures [SJ99]. A security proof in the ROM+GM encourages further analysis under relaxed assumptions. For instance it is reasonable to ask whether the random function H can be replaced by Canetti's "oracle hashing"—replacing $H(z)$ by $(h(z,r),r)$ for a one-way h and a random one-time key r. It would be nice to show security under this replacement.

Security of signed ElGamal encryption. It has long been believed that a suitably tagged ElGamal ciphertext is secure against the strong adaptive chosen ciphertext attack (CCA) of Rackoff and Simon [RS92]. CCA-security means indistinguishability of the plaintext m against any other message when given a ciphertext of m—the target ciphertext—and free access to a decryption oracle except for the target ciphertext. The idea of tagging an ElGamal ciphertext appears first in [ZS92], and one of the proposed schemes can be proven to be CCA-secure. CCA-secure schemes together with a security proof have been proposed in [SG98, CS98, ABR98, FO99, Sh00]. The schemes in [ZS92, 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 a private key encryption scheme.

Signed ElGamal encryption is a very practical encryption scheme where an ElGamal encryption [E85] is tagged by a Schnorr signature [Sc91]—the public signature key is part of the ElGamal ciphertext. Signed ElGamal encryption was independently proposed by Tsiounis and Yung [TY98] and Jakobsson [J98]. Herein, an ElGamal ciphertext (g^r, mh^r) is authenticated 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 part of the ciphertext. CCA-security of *signed ElGamal encryption* has been shown in [TY98] under the assumption that the signer really "knows" 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)].

While CCA-security of signed ElGamal encryption is still open under traditional assumptions Schnorr and Jakobsson [SJ00] prove CCA-security of signed ElGamal encryption in the ROM+GM. Moreover, it is shown in [SJ00] that signed ElGamal encryption is secure against the novel *one more decryption attack.* That proof is based on a novel and strong concept of generic plaintext awareness that is inspired by the notion of plaintext awareness set forth in [BR94]. It is possible to extract from a generic algorithm that generates a valid signed ElGamal ciphertext the corresponding plaintext m along with the one-time keys r, s used for encryption and for the tagged signature. There is a generic extractor that controls the

1) The FFS-extractor of Feige-Fiat-Shamir, in the oracle replay mode of Pointcheval and Stern [PS96a], 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.

generic steps of the generic attacker. The explicit description of the attacker is given to the extractor not just the black-box of the attacker.

The CCA-security of signed ElGamal encryption in the ROM+GM favorably contrasts with the partially unsettled situation of OAEP proposed in [BR94]. Recently, it was shown in [FOPS00] that the RSA-OAEP is CCA-secure in the ROM under the RSA-assumption filling a proof gap pointed out in [Sh00b]. But the exact CCA-security of RSA-OAEP remains open—the reduction to the RSA-assumption in [FOPS00] is not tight—also CCA-security for general one-way trapdoors is still unsettled. The CCA-security of signed ElGamal encryption in the ROM+GM indicates that signed ElGamal encryption is CCA-secure for all strong groups G and strong H.

The one-more decryption attack. In the ROM+GM it is possible to prove security against the one-more decryption attack in which an attacker interacts some l times with a decryption oracle and gets partial information about $l+1$ plaintexts corresponding to given ciphertexts. Various applications of e-commerce are possible provided that a decryption oracle is accessible that decrypts arbitrary ciphertexts against a corresponding fee [SJ00]. For such an application it is important that an attacker cannot get even partial information about more than l plaintexts by asking the decryption oracle some l times.

Security of blind Schnorr signatures. We study the security of blind Schnorr signatures against the one-more signature forgery in which an attacker interacts some l times with the legitimate signer and produces from these l interactions $l+1$ signatures. We first present a novel, parallel attack that succeeds in a one-more signature forgery against blind Schnorr and blind Okamoto-Schnorr signatures. The attack merely requires a solution of the ROS-problem, a possibly intractable problem: find an overdetermined, solvable system of linear equations modulo q with random inhomogeneities (right sides). Specifically, given a system of $t \gg l$ linear equations modulo q in l unknowns with random right sides, find a solvable subsystem of $l+1$ equations—a solvable subsystem corresponds to an $(l+1)\times(l+1)$-submatrix of rank l.

The generic parallel attack has the interesting feature not to depend on the public key. Traditional security proofs do not seem to work in the presence of such an attack. Usually, traditional security proofs use the attacker to solve a DL-problem or a decisional Diffie-Hellman-problem associated with the public key. As the generic parallel attack uses a solution of the ROS-problem that is not related to the public key, the attack cannot help to solve a DL- or a DDH-problem. How then could [PS00, P00, PS96b] prove security? The known security results only cover cases where solutions of the ROS-problem exist with negligible probability. The generic parallel attack points to an inherent weakness of the security result of Pointcheval and Stern.

The main result Theorem 26 of [PS00] shows[2)] that an attacker mounting a one-more signature forgery with a probability of success $\varepsilon > 4Q^{l+1}/q$ can be used

2) The security proofs of Pointcheval, Stern show that blind Okamoto-Schnorr signatures are secure against parallel interactive attacks [PS96b, PS00] provided that

to compute a discrete logarithm. Here Q is the number of hash queries, l is the number of interactions with the signer and q is the prime order of G. For an elliptic curve G of order $\leq 2^{200}$ and $Q = 2^{50}$ we must have $l \leq 3$ since $\varepsilon \leq 1$. For a subgroup G of units of order $\leq 2^{1000}$ we must have $l \leq 20$.

The generic parallel attack shows that any improved security guarantee, covering larger values of l, requires that the ROS-problem is intractable. The security of blind DL-signatures against the one-more signature forgery hinges on the intractability of the ROS-problem. The ROS-problem deserves further study because restricting the signer to merely a few parallel interactions is impractical for applications in e-commerce. It is important that the server issuing anonymous digital cash can operate in an unrestricted parallel way.

Practically meaningful security guarantees against the one-more signature forgery require in addition to the ROM and the intractability of the DL-problem also the intractability of the ROS-problem. Conversely, assuming the intractability of the ROS-problem a practical security guarantee for blind Schnorr signatures holds in the ROM+GM, see Theorem 5. A generic attacker performing t generic steps, including some l interactions with the signer, cannot produce $l+1$ signatures with a better probability than $\binom{t}{2}/q$. For elliptic curves G of order $q \approx 2^{200}$ this guarantee covers up to $t = 2^{100}$ generic steps including up to 2^{100} parallel signer interactions that can be interleaved in an arbitrary way. Moreover, this shows that blind Schnorr signatures have the same security level in the ROM+GM as the double-keyed blind Okamoto-Schnorr signatures, thus blind Schnorr signatures can save a considerable overhead. In conclusion our result suggests to use blind Schnorr signatures in connection with strong elliptic curves rather than double-keyed blind Okamoto-Schnorr signatures using subgroups of units.

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/signatures. In Section 3 we survey and comment on the security of signed ElGamal encryption presented in [SJ00]. In Section 4 we consider the security of blind Schnorr signatures against interactive attacks based on [SJ99].

2. The random oracle and the generic model

The random oracle model (ROM). Let H be an *ideal* hash function, modelled as an oracle, that given an input (query) outputs a random number in the range of H. Hash functions for signatures are of the type $H : G \times M \to \mathbf{Z}_q$, where G

the number of interactions with the signer is polylogarithmic—polylog($|q|$) for the binary length $|q|$ of q. The polylog bound on the number of signer interactions has not been explicitly mentioned in [P00] but it is required as the proof is based on the results of [PS00]. In [P98] a third party—the *checker*—has been introduced, and it is 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.

is a group of prime order q, M is a range of messages and $\mathbf{Z}_q$ denotes the field of integers modulo 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 function family. 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. Nevertheless, the security achievable in the ROM seems to in practice eliminate all attacks at hand.

The Generic Model (GM). Let G be an *ideal* group of prime order q with generator g. 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[3)] and we extend it to algorithms that interact with a decryption oracle. We do not assume a random binary encoding of group elements. We exemplify 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.

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:

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

3) 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*.

Definition. A (non-interactive) *generic algorithm* is a sequence of t generic steps[4)]

- $f_1, \ldots, f_{t'} \in G$ (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 given elements 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 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 clarifying 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, $\mathrm{mex} : \mathbf{Z}_q^d \times G^d \to G$, $(a_1, \ldots, a_d, g_1, \ldots, g_d) \mapsto \prod_i g_i^{a_i}$,
- queries to the hash oracle H,

4) 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.

- interactions with a decryption/signature oracle—see 3.2, 4.2 [5].

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 $f_1, \ldots, f_{t'}$. $\mathcal{A}$ 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* $f_1, \ldots, f_{t'} \in G$ 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 $\mathcal{A}$.

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.

An interaction with the *decryptor/signer* (defined in subsection 3.2/4.2) is a two round deterministic protocol. A claimed ciphertext is sent 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 $f_1, \ldots, f_{t'}$ 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 $g_1, \ldots, g_l$. By induction on j, a computed $f_j \in G$ is of the form $f_j = g_1^{a_{j,1}} \cdots g_l^{a_{j,l}}$, where the exponents $a_{j,1}, \ldots, a_{j,l} \in \mathbf{Z}_q$ depend arbitrarily on given non-group data. $\mathcal{A}$ can arbitrarily use the coefficients

5) 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.

$a_{j,1}, \ldots, a_{j,l}$ from this explicit representation of f_j. A generic adversary is deterministic, which is not a restriction as its coin flips would be useless[6].

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[7]. 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.

2.1. Results for the GM

This subsection refers to a generic, non-interactive adversary $\mathcal{A}$ that performs some t generic steps in attacking the indistinguishability of ElGamal encryption. Given q, the public key $h = g^x$, two messages $m_0, m_1 \in G$ and an ElGamal ciphertext $\text{cip}_b = (g^r, m_b h^r)$ for random $r, x \in_R \mathbf{Z}_q$ and $b \in_R \{0,1\}$, $\mathcal{A}$ guesses b. There is no hash function involved, we assume the GM but not the ROM. 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 $f_1, \ldots, f_t$. We let the *Main Case* be the part of the probability space where there are no non-trivial collisions among $f_1, \ldots, f_t$, i.e., $\mathcal{CO}_t = \emptyset$.

Lemma 1 ([SJ00]). *Non-trivial collisions among $f_1, \ldots, f_t$ occur with no better probability than $2\binom{t}{2}/q$. The probability refers to the random b, r, x.*

The factor 2 is the degree of the polynomial $\log_g m_b h^r = rx \log_g m_b \in \mathbf{Z}_q[r,x]$. If $m_b h^r$ is removed from the input then all polynomials $\log_g f_j \in \mathbf{Z}_q[r,x]$ are linear, and the factor 2 disappears. The degree of the polynomials $\log_g f_j \in \mathbf{Z}_q[r,x]$ enters into Lemma 1 via Lemma 2 that is attributed to Schwartz [Sch80].

Lemma 2. *A multivariate polynomial $F \in \mathbf{Z}_q[X_1, \ldots, X_k]$ of total degree d satisfies for random $x_1, \ldots, x_k \in \mathbf{Z}_q$ that $\Pr_{x_1,\ldots,x_k}[F(x_1, \ldots, x_k) = 0] \leq d/q$.*

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

6) $\mathcal{A}$ 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.

7) 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.

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 independent of the computed non-group data:

Lemma 3. *If there are no non-trivial collisions of group elements then the random b, r, x are statistically independent 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. □

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 3 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 non-interactive generic algorithm $\mathcal{A}$ 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. Suppose that there is no non-trivial collision of group elements in $\mathcal{A}$'s computation. Then, b, r, x are, by Lemma 2, statistically independent of the non-group output b', thus $\Pr_{b,x,r}[b' = b] = \frac{1}{2}$. Non-trivial collisions occur with probability $\leq 2\binom{t}{2}/q$. □

The decisional Diffie-Hellman problem (DDH-problem). This problem is to distinguish g^{xy} and g^z when given g, g^x, g^y for random $x, y, z \in_R \mathbf{Z}_q$. Proposition 2 implies the hardness of the DDH-problem in GM which was first shown by Shoup [Sh97].

Proposition 3 ([Sh97]). *Let a generic algorithm $\mathcal{A}$ be given g, g^x, g^y and $\{g^{xy}, g^z\}$ in random order for random $x, y, z \in_R \mathbf{Z}_q$ and output a guess for the order of g^{xy}, g^z. Then $\mathcal{A}$ succeeds with no better probability than $\frac{1}{2} + 2\binom{t}{2}/q$.*

2.2. Security of Schnorr identification, following [Sh97]

An identification scheme is an interactive protocol that allows one party $\mathcal{P}$ to prove its identity to another party $\mathcal{V}$. To do this $\mathcal{P}$ has a private key and a corresponding public key. Such a scheme is considered secure if an attacker cannot feasibly con-

vince $\mathcal{V}$ it is conducting the protocol with $\mathcal{P}$. One can allow the adversary to first interact with $\mathcal{P}$, pretending to be $\mathcal{V}$ (but not necessarily following $\mathcal{V}$'s protocol), in order to gain some useful information about $\mathcal{P}$'s secret key. Such an attack is called "active". An attack where no prior interaction with $\mathcal{P}$ is allowed is called "passive".

The Schnorr identification scheme [Sc91] runs as follows. Let G be a group of prime order q, with a publicly known generator g. $\mathcal{P}$'s private key is a random element $x \in_R \mathbf{Z}_q$, the corresponding public key is $h = g^x$. In the first step of the protocol $\mathcal{P}$ picks a random $r \in_R \mathbf{Z}_q$, computes $h' = g^r \in G$ and sends h' to $\mathcal{V}$. Upon receiving h', $\mathcal{V}$ picks a random challenge $c \in_R \mathbf{Z}_q$ and sends c to $\mathcal{P}$. Upon receiving c, $\mathcal{P}$ computes $y = r + xc \in \mathbf{Z}_q$ and sends y to $\mathcal{V}$. $\mathcal{V}$ accepts if $g^y = h'h^c$ and otherwise rejects.

It has been shown in [Sc91] that this identification scheme is secure against passive attacks provided that the DL-problem is hard. Shoup shows security against active attacks in the GM. An interaction with $\mathcal{P}$ is counted as generic step.

Proposition 4 ([Sh97]). *Let a generic adversary $\mathcal{A}$ be given $g, h = g^x$ and q and performing some t generic steps, including interactions with $\mathcal{P}$. Then $\mathcal{A}$ cannot make $\mathcal{V}$ accept with a better probability than t^2/q.*

2.3. Zeroknowledge versus generic verifiers

Let $\mathcal{L}$ be a language in the class NP of non-deterministic polynomial time languages. An interactive proof of membership for $\mathcal{L}$ is an interactive protocol between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$. For a common input y, $\mathcal{P}$ wants to convince $\mathcal{V}$ of a proof of membership for "$y \in \mathcal{L}$" known to $\mathcal{P}$. The zeroknowledge property of the interactive protocol $(\mathcal{P}, \mathcal{V})$ shows that $\mathcal{V}$ cannot learn anything about $\mathcal{P}$'s proof of membership by executing the protocol with $\mathcal{P}$. The protocol $(\mathcal{P}, \mathcal{V})$ is called *zeroknowledge* if there is a simulator $\mathcal{S}$ which given y generates the same distribution of data as do $\mathcal{P}$ and $\mathcal{V}$, without using $\mathcal{P}$'s proof of membership.

A zeroknowledge simulator $\mathcal{S}$ for an interactive proof of membership needs some control over the verifier $\mathcal{V}$. For many protocols it is sufficient that $\mathcal{S}$ uses the verifier as a black-box—we call $\mathcal{S}$ a *black-box simulator*. A simulator $\mathcal{S}$ that is given the Turing machine of $\mathcal{V}$ is more powerful as $\mathcal{S}$ completely controls all steps of $\mathcal{V}$. For generic verifiers in the generic group model it makes sense to let the simulator control the generic group steps of $\mathcal{V}$—we call $\mathcal{S}$ a *generic verifier simulator*.

M. Fischlin [F00] demonstrates the power of generic verifier simulators. He presents a three round negligible-error proof of membership for arbitrary NP-languages together with a polynomial time zeroknowledge simulator $\mathcal{S}$. Fischlin's generic verifier simulator $\mathcal{S}$ achieves something that is impossible for black-box simulators and arbitrary Turing machine verifiers. Goldreich and Krawczyk [GK96] show that polynomial time black-box simulators for three round negligible-error proofs of membership merely exist for languages in BPP—which is conjectured to be a proper subclass of NP.

Fischlin's example shows that generic verifier zeroknowledge does not imply black-box TM verifier zeroknowledge. This is no surprise as there are two reasons: generic verifiers are in their generic steps more restricted than TM-verifiers, and black-box simulators are less powerful than generic verifier simulators that control the generic group steps of the verifier.

3. Signed ElGamal encryption

3.1. Introduction

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.)

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 $r \in \mathbf{Z}_q$. Assuming the intractability of the DDH-problem, ElGamal encryption is *indistinguishable*, and equivalently *semantically secure* [GM84]—it is secure against a passive, merely eavesdropping 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 [DDN91,00] 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 [DDN00].

Security in the ROM+GM. The [TY98]-assumption that the signer of a Schnorr signature really "knows" the secret key can be proved in the ROM+GM. The core of the security proof in [SJ00] is a generic extractor that extracts from a generic CCA-attacker the plaintext and the secret one-time keys of a ciphertext constructed by the attacker, see Theorem 1. Theorem 2 shows security against a generic CCA-attacker performing some $t = o(\sqrt{q})$ 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 $l+1$ ciphertexts by asking a decryption oracle some l times. For motivation of the one-more decryption attack, we propose a practical scheme for buying digital information anonymously. It is based on blind decryption and security against the *random one-more attack*, a weak version of the one-more decryption attack, but does not require CCA-security.

3.2. Definition of signed ElGamal encryption

We first 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.

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

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

In order to *verify* a signature (m, c, z) check that $H(g^z \bar{h}^{-c}, m) = c$. The signature protocol produces a correct signature since $g^z \bar{h}^{-c} = g^{s+cr} \bar{h}^{-c} = g^s$ [8].

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 tagged Schnorr signature signs pairs in $G \times M$ and uses a random hash function $H : G^2 \times M \to \mathbf{Z}_q$.

In order to *encipher* a message $m \in G$, 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

8) Signatures (c, z) with $c = 0$ have the undesirable property that they hold for all public keys h. It makes sense to exclude such signatures even though they appear only with negligible probability.

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 $\bar{f}/\bar{h}^x = m\,g^{rx}g^{-rx} = m$.

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 secret signature key r is the one-time key for encryption. In the ROM 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.

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.

Comparison with other secure DL-cryptosystems. We count the number of exponentiations[9] per encryption/decryption and the number of on-line exponentiations per encryption (exponentiations not depending on the message).

	exp./enc.	on-line/enc.	exp./dec.
Signed ElGamal enc.	3	0	2
[FO99]	2	2	2
[ABR98]	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 is plaintext aware in a strong sense. Signed ElGamal encryption virtually combines all the good properties.

Generalized (signed) ElGamal encryption. Our security results extend to the variant of ElGamal encryption proposed by Naor, Reingold [NR97]. This variant has 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

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

the public key h [10]. The generalized encryption solves the security flaw in textbook ElGamal encryption pointed out in [BJN00].

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.

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

Second extension of Proposition 2. Assuming the ROM, Proposition 2 extends to signed ElGamal encryption and to generalized signed ElGamal encryption. This is because the added Schnorr signature does not contain any information about the plaintext.

3.3. Security against interactive attacks

Consider the security of signed ElGamal encryption in ROM+GM. Theorem 1 shows that signed ElGamal encryption is plaintext aware in a strong sense. The proof of Theorem 1 is implicit in [SJ00], it is the core for proving Theorems 2 and 3.

Theorem 1 (Plaintext awareness [SJ00]). *Let a generic adversary $\mathcal{A}$ be given for input g, h and signed ElGamal ciphertexts* $\mathrm{cip}_1, \ldots, \mathrm{cip}_d$. *Let $\mathcal{A}$ produce within t*

10) 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 group elements is impractical. Known extensions of ElGamal encryption—see e.g., [MOV97] section 8.26—do not solve this encoding problem.

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: $(H_M(f,1), \ldots, H_M(f,n))$.

generic steps a signed ElGamal ciphertext cip_{d+1}. *Then there is a generic algorithm* $\mathcal{A}'$ *which given the same input produces within* t *generic steps the plaintext and the one-time keys corresponding to* cip_{d+1}. $\mathcal{A}'$ *succeeds except for an event of probability* $3/q$. *The probability refers to the random* h, H *and the random one-time keys of the encipherer.*

$\mathcal{A}'$ succeeds in extracting the plaintext corresponding to cip_{d+1} whenever $\mathcal{A}$ has produced a correct ciphertext cip_{d+1}—except for an event of probability $\frac{3}{q}$. Moreover, $\mathcal{A}'$ extract the one-time keys $r, s \in \mathbf{Z}_q$ from the construction of $\text{cip}_{d+1} = (g^r, mh^r, c, z), g^s = g^z h^{-c}$.

The generic extractor is more efficient than known black-box extractors for DL-signatures and DL-identification. The black-box extractor of [FFS88] has a constant delay factor compared to the attacker. The generic extractor does not induce any delay, the number of generic steps does not increase compared to the attacker. Only the attacker's probability of success decreases by at most $\frac{3}{q}$. The power of the generic extractor relies on the fact that it can be iterated some T number of times decreasing the attacker's probability of success by at most $3T/q$. Theorem 1 can be extended from given ciphertexts $\text{cip}_1, \ldots, \text{cip}_d$ to ciphertexts that are produced in an adaptive way by querying an enciphering oracle about ciphertexts of chosen messages.

Theorem 2 shows indistinguishability against a CCA-adversary $\mathcal{A}$. This is equivalent to non-malleability against CCA [DDN00]. We refer to non-malleability as defined in [DDN00] 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. 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 2 ([SJ00]). *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 one-time keys of the encipherer.*

Theorem 3 shows that signed ElGamal encryption is secure against the one more decryption 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.

Theorem 3 ([SJ00]). *Let the attacker* $\mathcal{A}$ *be given* g, h, *ciphertexts* $\text{cip}_1, \ldots, \text{cip}_d$, *the corresponding messages* $m_1, \ldots, m_d$ *in random order and oracles for* H *and for decryption. Let the generic* $\mathcal{A}$ *perform* t *generic steps including some* $l < d$ *arbitrary queries to the decryption oracle. Then* $\mathcal{A}$ *cannot produce* $l+1$ *message-ciphertext pairs with a probability better than* $\frac{1}{d-l} + t^2/q$. *The probability space*

consists of the random x, H, the one-time keys of the encipherer and the random ordering of the messages.

The notions of CCA-security and of security against the one-more decryption attack are of independent interest. Consider the case of blind decryption where the decryption oracle is queried about a blinded ciphertext that is statistically independent of the ciphertext that is actually decrypted. In this case—which is important for acquiring digital information anonymously—CCA-security is meaningless as there is no well defined target ciphertext.

CCA-security implies a weak form of security against the one-more decryption attack. A one-more decryption attack can be transformed into a CCA-attack by guessing the ciphertext for which the attacker forms a correct message-ciphertext pair without querying the decryption oracle. However, this transform induces a big delay factor as large as the number of given ciphertexts in the data bank.

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 via 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[12]. 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 anonymously buying digital information (ABDI). 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 $\text{cip}(k_i) = (g^r, k_i h^r, c, z)$ of the key $k_i \in G$. Let (c, z) be a signature of $(g^r, k_i h^r, \text{descr}_i)$ 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 $\text{cip}(k_i) = (g^r, k_i h^r, c, z)$ 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 $l+1$ keys k_i.

12) 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}.

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

Of course, the private key encryption scheme must be secure. Specifically, ciphertexts are supposed not to reveal any information about the secrete encryption keys. The security argument does not require CCA-security of the public-key encryption scheme, but merely semantical security. Therefore, ElGamal encryption can be replaced by other self-reducible encryption schemes [AFK89]. Our proposal can be considered as an implementation of a $\binom{T}{l}$-oblivious transfer, where T is the number of encrypted packages in the data bank and l is the number of decryption requests.

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 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 encryption 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 with short hash values. Let the hash values of H be random in an interval $[0, 2^k[\subset [0, q[\cong \mathbf{Z}_q$. In the situation of Theorem 2 a CCA-attacker does not succeed better than with probability $\frac{1}{2}+t^2/q+l(2^{-k}-\frac{1}{q})$, where l is the number of decryptor interactions. This shows that random hash values can securely range over a set of $\sqrt{q}$ values.

4. Security of blind Schnorr signatures

4.1. Introduction and survey

We are interested in blind signatures as required for anonymous digital cash. Blind signatures are generated by an interaction with the signer who controls the private signature key. We study security against the *one-more signature forgery* introduced in [PS96b], where security means that an attacker cannot obtain $l+1$ 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 first present the generic parallel attack which escaped in all previous works.

The generic parallel attack uses a solution of the ROS-problem: find a random, overdetermined, solvable system of $l+1$ linear equations in l unknowns modulo q

with random inhomogeneities (right sides). This may well be an intractible problem. Given a solution of the ROS-problem, the attacker easily succeeds in a parallel one-more signature forgery. The generic, parallel attack works in the same way for Schnorr and for Okamoto-Schnorr signatures, it does not even use the public key. Therefore, the attacker cannot help to solve a DL-, DH- or a DDH-problem. Theorem 4 evaluates the average number of solutions of the ROS-problem. A solution exists and the parallel attack is possible with 4 signer interactions and 2^{50} hash queries for an elliptic curve of order $q \approx 2^{200}$.

Assuming the intractability of the ROS-problem, Theorem 5 gives a practical security guarantee against the one-more signature forgery in the ROM+GM. A generic adversary using t generic steps cannot succeed better than with probability $\binom{t}{2}/q$.

4.2. Interactive generation of blind Schnorr signatures

We describe 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 $x = \log_g h$.

Recall the definition of Schnorr signatures from Subsection 3.2. A triple (m, c, z) is a Schnorr signature for the public key $h = g^x$ if $H(g^z h^{-c}, m) = c$.

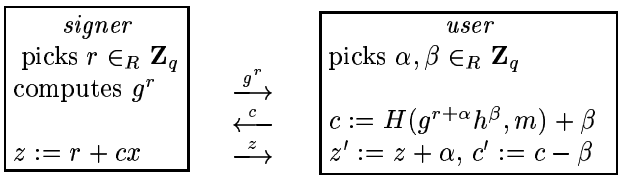
A signer interaction is an interactive protocol between the signer and a user consisting of three rounds. The signer picks a random $r \in_R \mathbf{Z}_q$ computes g^r and sends the commitment g^r. The user selects $c \in \mathbf{Z}_q$ and sends the challenge c. The signer responses by sending $z := r + cx$.

The user can generate from this protocol 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 coin r, the user's *challenge* c and the signer's *response* z.

Non-interactive proof of knowledge. Schnorr signatures provide in the ROM a non-interactive proof of knowledge of the secret key $x = \log_g h$ [PS96a].

Blind signature protocol. 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) pro-

viding the view of the signer. The user can generate such an independent signature (m, c', z') from random numbers $\alpha, \beta \in_R \mathbf{Z}_q$.



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 α, β, namely $\alpha = z' - z$, $\beta = c - c'$.

4.3. The generic parallel attack

We present a variant of the attack that does not use the generator g and the public key h. We first present the attack for Schnorr signatures. Thereafter, we extend it to Okamoto-Schnorr signatures. This shows that Okamoto-Schnorr signatures do not protect better against the generic parallel than plain Schnorr signatures.

The signer sends commitments $g_1 = g^{r_1}, \ldots, g_l = g^{r_l}$. The attacker $\mathcal{A}$ computes $f_k = g_1^{a_{k,1}} \cdot \ldots \cdot g_l^{a_{k,l}}$ and $H(f_k, m_k)$ for $k = 1, \ldots, t''$. Then $\mathcal{A}$ solves $l + 1$ of the t'' equations (1) in the unknowns $c_1, \ldots, c_l$ over $\mathbf{Z}_q$.

$$H(f_k, m_k) = \sum_{\ell=1}^{l} a_{k,\ell}\, c_\ell \text{ for } k = 1, \ldots, t''. \tag{1}$$

$\mathcal{A}$ sends the solutions $c_1, \ldots, c_l$ as challenges to the signer. The signer sends back $z_\ell := r_\ell + c_\ell x \in \mathbf{Z}_q$ for $\ell = 1, \ldots, l$. For each solved equation (1), the attacker gets a valid signature (m_k, c'_k, z'_k) by setting

$$c'_k := \sum_{\ell=1}^{l} a_{k,\ell}\, c_\ell = H(f_k, m_k) \text{ and } z'_k := \sum_{\ell=1}^{l} a_{k,\ell} z_\ell.$$

In the ROM the values $H(f_k, m_k)$ are random. The coefficients $a_{k,\ell}$ selected by the attacker are arbitrary values. The solution $(c_1, \ldots, c_l)$ of $l + 1$ of the t equations (1) does not depend on g, h. As $\mathcal{A}$ does not use g, h it is impossible that $\mathcal{A}$ helps in black-box mode to solve a DL-, DH- or DDH-problem for G.

The generic parallel attack is intrinsic parallel. By Theorem 4 the number l of parallel interactions with the signer must be at least logarithmic in q. Otherwise, the probability $\binom{t''}{l+1}/q$ for the existence of an ROS-solution is negligible.

The generic parallel attack against Okamoto-Schnorr signatures. We follow the notation of [PS00]. There are two public keys h and $y = g^{-r}h^{-s}$ for random secret keys $r, s \in_R \mathbf{Z}_q$ while $\log_g h$ is unknown. A signature of message m is a tuple $(m, \varepsilon, \sigma, \rho) \in M \times \mathbf{Z}_q^3$ satisfying $H(g^\rho h^\sigma y^\varepsilon, m) = \varepsilon$.

The signer picks random $t_\ell, u_\ell \in_R \mathbf{Z}_q$ and sends commitments $g_\ell = g^{t_\ell} h^{u_\ell}$ for $\ell = 1, \ldots, l$ to the attacker $\mathcal{A}$. $\mathcal{A}$ selects coefficients $a_{k,\ell} \in \mathbf{Z}_q$, computes $f_k = g_1^{a_{k,1}} \cdots g_l^{a_{k,l}}$ and $H(f_k, m_k)$ for $k = 1, \ldots, t''$. $\mathcal{A}$ solves $l+1$ of the t'' linear equations (1) modulo q in the unknowns $c_1, \ldots, c_l$. $\mathcal{A}$ sends the solutions $c_1, \ldots, c_l$ as challenges to the signer. The signer sends back $R_\ell := t_\ell + c_\ell r$, $S_\ell := u_\ell + c_\ell s \in \mathbf{Z}_q$ for $\ell = 1, \ldots, l$. For each solved equation (1) $\mathcal{A}$ gets a valid signature $(m_k, \varepsilon, \rho_k, \sigma_k)$ by setting

$$\varepsilon_k = H(f_k, m_k) = \sum_{\ell=1}^{l} a_{k,\ell}\, c_\ell,\ \ \rho_k = \sum_{\ell=1}^{l} a_{k,\ell}\, R_\ell,\ \ \sigma_k = \sum_{\ell=1}^{l} a_{k,\ell}\, S_\ell.$$

Correctness. From the equations (1) we get that

$$g^{\rho_k} h^{\sigma_k} y^{\varepsilon_k} = \prod_{\ell=1}^{l} g_\ell^{a_{k,\ell}} = f_k \text{ and } H(g^{\rho_k} h^{\sigma_k} y^{\varepsilon_k}, m_k) = \varepsilon_k.$$

Conclusion. The attacker $\mathcal{A}$ does not use the public g, h, y. Thus, it is impossible to use a successful $\mathcal{A}$ to solve a DL- DH- or DDH-problem. The generic parallel attack has been excluded in Theorem 26 [PS00] by assuming that the attacker has a probability of success $4t''^{(l+1)}/q$ which is greater than the probability $\binom{t''}{l+1}/q$ for the existence of a solution of the ROS-problem. By Theorem 4 a solution of a subsystem of $l+1$ equations (1) exists and the generic parallel attack is possible with $l = 4$ parallel interactions and 2^{50} hash queries for a 200 bit order q of an elliptic curve. Thus it is impossible to get a practical security guarantee for elliptic curves of order $\leq 2^{200}$ by the proof methods of [PS00].

The average number of solutions for the ROS-problem. Consider distinct equations

$$a_{k,1}\, c_1 + \ldots + a_{k,l}\, c_l = F(a_{k,1}, \ldots, a_{k,l}) \text{ for } k = 1, \ldots, t'' \tag{2}$$

in the unknowns $c_1, \ldots, c_l$ over $\mathbf{Z}_q$ for a random function $F : \mathbf{Z}_q^l \mapsto \mathbf{Z}_q$. We evaluate the expected number of solvable subsystems consisting of $l+1$ equations.

Theorem 4 ([SJ99]). *For arbitrary coefficients $a_{k,\ell}$ the average number of solvable subsystems of $l+1$ out of the t'' equations (2) is at most $\binom{t''}{l+1}/q$. For statistically independent coefficients $a_{k,\ell} \in_R \mathbf{Z}_q$ the average number of solvable subsystems with $l+1$ equations (2) is $\binom{t''}{l+1} q^{-1}(1 - q^{-1} + O(q^{-2}))$.*

How to solve the ROS*-problem in the Turing machine model.* A simple method to find all solvable subsystems of $l+1$ out of t'' equations (2) tries all $\binom{t''}{l}$ choices for $1 \leq \sigma_1 < \ldots < \sigma_l < t''$—with overwhelming probability each subsystem has at most one solution—sorts these solutions and checks for a collision. A collision

yields a solution of $l+1$ equations. Conversely, a solution of $l+1$ equations coincides with a collision. This method requires at least $\binom{t''}{l}$ arithmetic steps over $\mathbf{Z}_q$ and succeeds with probability $\leq \binom{t''}{l+1}/q$. That method is *optimal* for $l = 1$ [13].

In a variant of the above method one forms for a subfamily of some T—from the $\binom{t''}{l}$ choices of l out of t'' equations—sorts the solutions of the corresponding l-tuples of equations, and searches for two colliding solutions. Two solutions collide with probability at most $\frac{1}{q}$. The restricted search does not succeed with a better probability than $\binom{T}{2}/q$ using T arithmetic steps over $\mathbf{Z}_q$—for an arbitrary selection of the coefficients $a_{k,\ell}$ and the T subsystems. We pose it as an open problem to beat that $\binom{T}{2}/q$ bound.

From the work of Håstad [H97] we know that beating the naive method—in maximizing a subset of solvable linear equations of an overdetermined system—is NP-hard. This associated NP-hard problem seems to indicate that the ROS-problem is not trivial. It seems to be irrelevant that the adversary can choose himself the coefficients $a_{k,\ell}$ of the equations (2) as the inhomogeneities $F(a_{k,0}, \ldots, a_{k,\ell})$ are independent random numbers. Moreover, the complexity of the ROS-problem hardly depends on the selected coefficients $a_{k,\ell}$. The ROS-problem for particular coefficients seems to be as hard as in worst case.

4.4. Security of blind signatures against interactive attacks

This subsection refers to a generic adversary $\mathcal{A}$ performing some t generic steps—including some l interactions $(r_1, c_1, z_1), \ldots, (r_l, c_l, z_l)$ with the signer—some t' group elements and some t'' queries to the hash oracle. We let $\mathbf{r} = (r_1, \ldots, r_l)$ denote the signers random coins. Let $f_1 = g$, $f_2 = h = g^x$, $f_3, \ldots f_{t'} \in G$ denote the group elements of $\mathcal{A}$'s computation. The generic $\mathcal{A}$ computes $f_j = g^{a_{j,-1}} h^{a_{j,0}} g_1^{a_{j,1}} \cdots g_l^{a_{j,l}}$, where $g_1 = g^{r_1}, \ldots, g_l = g^{r_l}$ are the signer's commitments and the exponents $a_{j,l} \in \mathbf{Z}_q$ depend arbitrarily on the previously computed non-group data. As each signer interaction yields one group element g^{r_ℓ} we have that $t'' = t - t' \geq 0$ is the number of interactions with the hash oracle. We first present the basic Lemma 1′ and 2′ that extend Lemma 1 and 2 from a non-interactive attacker to an adversary using an hash oracle and a signature oracle.

Interactions with the signature oracle, (signer *for short*). The generic algorithm sends to the signer a message $m \in M$ and a public key $h \in G$, and receives from the signer a random signature $(m, e, y) \in M \times \mathbf{Z}_q^2$ with public key h. The signer picks a random $r \in_R \mathbf{Z}_q$ forms $e := H(g^r, m)$ by asking the hash oracle, and sets $y := r + ex$, where $x = \log_g h$ is the secret key.

13) For $l = 1$ there is a solution of two of the t'' equations (2) if and only if there is a collision among the $\mathbf{Z}_q$-numbers $F(a_{k,1}, \ldots, a_{k,l})/a_{k,1}$ for $k = 1, \ldots, t''$. These $\mathbf{Z}_q$-numbers are pairwise statistically independent due to the random F. A collision occurs with probability $\binom{t''}{2}/q$, it can be found in $O(t'' \log t'')$ arithmetic steps by sorting.

Lemma 1' ([SJ99]). Collisions among $f_1, \ldots, f_{t'}$ occur at most with probability $\binom{t'}{2}/q$. The probability refers to the random h, H and the random coins $\mathbf{r}$ of the signer.

Lemma 2' ([SJ99]). If there are no collisions among $f_1, \ldots, f_{t'}$ the random x is statistically independent of the computed non-group data except that the random coins $(\mathbf{r}, x)$ leading to collisions are excluded.

Theorems 4 and 5 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 DL-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 the ROS-problem: find an overdetermined solvable system of linear equations modulo q with random inhomogeneities.

Theorem 5 ([SJ99]). *Let a generic adversary $\mathcal{A}$ be given the generator g, the public key h, an oracle for H and let $\mathcal{A}$ interact with the signer some l times. Let $\mathcal{A}$ perform t generic steps including l parallel signer interactions. If a generic adversary $\mathcal{A}$ succeeds in a one-more signature forgery with a better probability of success than $\binom{t}{2}/q$ then $\mathcal{A}$ must solve the* ROS*-problem: given an oracle for a random function F find $l+1$ distinct vectors $(a_{k,1}, \ldots, a_{k,l}) \in \mathbf{Z}_q^l$ and a solution $c_1, \ldots, c_l \in \mathbf{Z}_q$ of the equations $\sum_{l=1}^{l} a_{k,l}\, c_l = F(a_{k,1}, \ldots, a_{k,l})$ for $k = 1, \ldots, l+1$.*

Security of Schnorr signatures against the chosen message attack (CMA). Theorem 5 implies for $l = 0$ that Schnorr signatures are secure in ROM+GM against existential forgery, where an attacker is given the public signature key and tries to produce some correct signature (m, c, z). This security result extends to the case that the adversary $\mathcal{A}$ has a signature oracle (signer) and can ask the oracle for signatures of messages of its choice. An interaction with the signer is counted as generic step. The goal of the attack is to generate a new signature which is not produced by the signer.

Theorem 6 (Security against CMA). *Let $\mathcal{A}$ be a generic algorithm that is given g, the public signature key $h \in_R G$ and oracles for H and signatures. Using t generic steps—group operations and hash queries and queries to the signer—$\mathcal{A}$ cannot produce a new Schnorr signature with a probability better than $\frac{3}{q} + \binom{t}{2}/q$. The probability space consists of the random h, H.*

Theorem 6 is closely related to the security in GM of Schnorr identification against active attacks proved by Shoup [Sh97]. From that proof the CMA-security of Schnorr signatures follows by a very close argument replacing the random queries of the verifier in the Schnorr identification by random hash values and using the oracle replay mode proposed in [PS96a].

However, the security of Schnorr identification in GM does not bear any security against the one-more signature forgery. The latter security means that an active

attacker against the Schnorr identification scheme cannot transform l parallel interactions with an identification oracle into $l+1$ identifications. This would be a form of the man-in-the-middle-attack which has been explicitly excluded in [Sh97]. Moreover, such a security result requires the intractability of the ROS-problem.

Acknowledgement. I thank Marc Fischlin for useful comments while reading the manuscript.

References

[AFK89] Abadi, M., Feigenbaum, J., Kilian, J., On hiding information from an oracle. J. Computer System Sci. 39 (1989), 21–50.

[ABR98] Abdalla, M., Bellare, M., Rogaway, P., DHES: An encryption scheme based on the Diffie-Hellman problem. Contributions to P1363, ftp: `//stdgbbs.ieee.org/pub/p1363/contributions/aes-uhf.ps`

[BDPR98] Bellare, M., Desai, A., Pointcheval, D., Rogaway, P., Relations among notions of security for public-key encryption schemes. In: Advances in Cryptology – CRYPTO '98 (ed. by H. Krawczyk; Lecture Notes in Comput. Sci. 1462), 26–45. Springer, Berlin 1998.

[BR93] Bellare, M., Rogaway, P., Random oracles are practical: a paradigm for designing efficient protocols. In: Proceedings of the 1st ACM Conference on Computer and Communication Security (CCS '93), 62–73. ACM Press, New York 1993.

[BR94] —, —, Optimal asymmetric encryption – how to encrypt with RSA. In: Advances in Cryptology – EUROCRYPT '94 (ed. by A. De Santis; Lecture Notes in Comput. Sci. 950), 92–111. Springer, Berlin 1995.

[BJN00] Boneh, D., Joux, A., Nguyen, P.Q., Why textbook ElGamal and RSA encryption are insecure. In: Advances in Cryptology – ASIACRYPT 2000 (ed. by T. Okamoto; Lecture Notes in Comput. Sci. 1976), 30–43. Springer, Berlin 2000.

[BL96] Boneh, D., Lipton, R.J., Algorithms for black-box fields and their application in cryptography. In: Advances in Cryptology – CRYPTO '96 (ed. by N. Koblitz; Lecture Notes in Comput. Sci. 1109), 283–297. Springer, Berlin 1996.

[C97] Canetti, R., Towards realizing random oracles: hash functions that hide all partial information. In: Advances in Cryptology – CRYPTO '97 (ed. by B.S. Kaliski; Lecture Notes in Comput. Sci. 1294), 455–469. Springer, Berlin 1997.

[CGH98] Canetti, R., Goldreich, O., Halevi, S., The random oracle methodology, revisited. In: STOC '98, 209–218. ACM Press, New York 1999. `http://www.acm.org/pubs/contents/proceedings/stoc/276698/`

[CS98] Cramer, R., Shoup, V., A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Advances in Cryptology – CRYPTO '98 (ed. by H. Krawczyk; Lecture Notes in Comput. Sci. 1462), 13–25. Springer, Berlin 1998.

[DH76] Diffie, W., Hellman, M.E., New directions in cryptography. IEEE Trans. Inform. Theory 22 (1976), 644–654.

[DDN91] Dolev, D., Dwork, C., Naor, M., Non-malleable cryptography. In: STOC '91, 542–552. ACM Press, New York 1991.

[DDN00] —, —, —, Non-malleable cryptography. SIAM J. Comput. 30 (2000), 391–437.

[E85] ElGamal, T., A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inform. Theory 31 (1985), 469–472.

[FFS88] Feige, U., Fiat, A., Shamir, A., Zero-knowledge proofs of identity. J. Cryptology 1 (1988), 77–94.

[FS87] Fiat A., Shamir, A., How to prove yourself: practical solutions to identification and signature problems. In: Advances in Cryptology – CRYPTO '86 (ed. by A.M. Odlyzko; Lecture Notes in Comput. Sci. 263), 186–194. Springer, Berlin 1987.

[F00] Fischlin, M., A note on security proofs in the generic model. In: Advances in Cryptology – ASIACRYPT 2000 (ed. by T. Okamoto; Lecture Notes in Comput. Sci. 1976), 458–469. Springer, Berlin 2000.

[FO99] Fujisaki, E., Okamoto, T., Secure integration of asymmetric and symmetric encryption schemes. In: Advances in Cryptology – CRYPTO '99 (ed. by M. Wiener; Lecture Notes in Comput. Sci. 1666), 537–554. Springer, Berlin 1999.

[FOPS00] Fujisaki, E., Okamoto, T., Pointcheval, D., Stern, J., RSA-OAEP is still alive!. Cryptology ePrint Archive 2000/061.

[GK96] Goldreich, O., Krawczyk, H., On the composition of zero-knowledge proof systems. SIAM J. Comput. 25 (1996), 169–192.

[GM84] Goldwasser, S., Micali, S., Probabilistic encryption. J. Comput. System Sci. 28 (1984), 270–299.

[H97] Håstad, J., Some optimal inapproximability results. STOC '97, 1–10. ACM Press, New York 1999.
`http://www.acm.org/pubs/contents/proceedings/stoc/258533/`

[J98] Jakobsson, M., A practical mix. In: Advances in Cryptology – EUROCRYPT '98 (ed. by K. Nyberg; Lecture Notes in Comput. Sci. 1403), 448–461. Springer, Berlin 1998.

[MOV97] Menezes, A., van Oorschot, P., Vanstone, S., Handbook of Applied Cryptography. CRC Press, Boca Raton 1997.

[NR97] Naor, M., Reingold, M., Number-theoretic constructions of pseudo-random functions. In: 38th Symposium on Foundations of Computer Science, 458–467. IEEE, Washington 1997.

[Ne94] Nechaev, V.I., On the complexity of a deterministic algorithm for a discrete logarithm (Russian). Mat. Zametki 55.2 (1994), 91–101. English transl.: Math. Notes 55 (1994), 165–172.

[O92] Okamoto, T., Provably secure identification schemes and corresponding signature schemes. In: Advances in Cryptology – CRYPTO '92 (ed. by E.F. Brickell; Lecture Notes in Comput. Sci. 740), 31–53. Springer, Berlin 1993.

[P98] Pointcheval, D., Strengthened security for blind signatures. In: Advances in Cryptology – EUROCRYPT '98 (ed. by K. Nyberg; Lecture Notes in Comput. Sci. 1403), 391–405. Springer, Berlin 1998.

[P00] —, The composite discrete logarithm and secure authentication. In: Public Key Cryptography (PKC 2000, Melbourne) (ed. by H. Imai, Y. Zhang; Lecture Notes in Comput. Sci. 1751), 113–128. Springer, Berlin 2000.

[PS96a] Pointcheval, D., Stern, J., Security proofs for signature schemes. In: Advances in Cryptology – EUROCRYPT '96 (ed. by U. Maurer; Lecture Notes in Comput. Sci. 1070), 387–398. Springer, Berlin 1996.

[PS96b] —, —, Provably secure blind signature schemes. In: Advances in Cryptology – ASIACRYPT '96 (ed. by K. Kim, T. Matsumoto; Lecture Notes in Comput. Sci. 1163), 387–393. Springer, Berlin 1996.

[PS00] —, —, Security arguments for digital signatures and blind signatures. J. Cryptology 13 (2000), 361–396.

[RS92] Rackoff, C., Simon, D.R., Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Advances in Cryptology – CRYPTO '91 (ed. by J. Feigenbaum; Lecture Notes in Comput. Sci. 576), 433–444. Springer, Berlin 1992.

[Sc91] Schnorr, C.P., Efficient signature generation for smart cards. J. Cryptology 4 (1991), 161–174.

[Sc01] —, Small generic hardcore subsets for the discrete logarithm: short secret DL-keys. Inform. Process. Lett. 79 (2001), 93–98.

[SJ99] Schnorr, C.P., Jakobsson, M., Security of discrete log cryptosystems in the random oracle + generic model. TR report University Frankfurt and Bell Laboratories 1999. `http:www.mi.informatik.uni-frankfurt.de`.

[SJ00] —, —, Security of signed ElGamal encryption. In: Advances in Cryptology – ASIACRYPT 2000 (ed. by T. Okamoto; Lecture Notes in Comput. Sci. 1976), 73–89. Springer, Berlin 2000.

[Sch80] Schwartz, J., Fast probabilistic algorithms for verification of polynomial identities. J. Assoc. Comput. Mach. 27 (1980), 701–717.

[Sh97] Shoup, V., Lower bounds for discrete logarithms and related problems. In: Advances in Cryptology – EUROCRYPT '97 (ed. by W. Fumy; Lecture Notes in Comput. Sci. 1233), 256–266. Springer, Berlin 1997.

[Sh00] —, Using hash functions as a hedge against chosen ciphertext attack. In: Advances in Cryptology – EUROCRYPT 2000 (ed. by B. Preneel; Lecture Notes in Comput. Sci. 1807), 275–288. Springer, Berlin 2000.

[Sh00b] —, OAEP reconsidered. Cryptology ePrint Archive 2000/060.

[SG98] Shoup, V., Gennaro, R., Securing threshold cryptosystems against chosen ciphertext attacks. In: Advances in Cryptology – EUROCRYPT '98 (ed. by K. Nyberg; Lecture Notes in Comput. Sci. 1403), 1–16. Springer, Berlin 1998.

[TY98] Tsiounis, Y., Yung, M., On the security of ElGamal based encryption. In: Public Key Cryptography (PKC '98) (ed. by H. Imai, Y. Zhang; Lecture Notes in Comput. Sci. 1431), 117–134. Springer, Berlin 1998.

[ZS92] Zheng, Y., Seberry, J., Practical approaches to attaining security against adaptively chosen ciphertext attacks. In: Advances in Cryptology – CRYPTO '92 (ed. by E.F. Brickell; Lecture Notes in Comput. Sci. 740), 292–304. Springer, Berlin 1993.

Fachbereiche Mathematik/Informatik, Universität Frankfurt
PSF 111932
D-60054 Frankfurt am Main, Germany
schnorr@cs.uni-frankfurt.de

Received: January 31, 2001.