Short Signatures Without Random Oracles

Dan Boneh[1]*, and Xavier Boyen[2]

[1] Computer Science Department, Stanford University, Stanford CA 94305-9045
dabo@cs.stanford.edu
[2] Voltage Security, Palo Alto, California
xb@boyen.org

Abstract. We describe a short signature scheme which is existentially unforgeable under a chosen message attack without using random oracles. The security of our scheme depends on a new complexity assumption we call the *Strong Diffie-Hellman* assumption. This assumption has similar properties to the Strong RSA assumption, hence the name. Strong RSA was previously used to construct signature schemes without random oracles. However, signatures generated by our scheme are much shorter and simpler than signatures from schemes based on Strong RSA. Furthermore, our scheme provides a limited form of message recovery.

1 Introduction

Boneh, Lynn, and Shacham (BLS) [BLS01] recently proposed a short digital signature scheme where signatures are about half the size of DSA signatures with the same level of security. Security is based on the Computational Diffie-Hellman (CDH) assumption on certain elliptic curves. The scheme is shown to be existentially unforgeable under a chosen message attack in the random oracle model.

In this paper we describe a signature scheme where signatures are almost as short as BLS signatures, but whose security does not require random oracles. We prove security of our scheme using a complexity assumption we call the Strong Diffie-Hellman assumption, or SDH for short. Roughly speaking, the q-SDH assumption in a group $\mathbb{G}$ of prime order p states that the following problem is intractable: given $g, g^x, g^{(x^2)}, \ldots, g^{(x^q)} \in \mathbb{G}$ as input, output a pair $(c,\, g^{1/(x+c)})$ where $c \in \mathbb{Z}_p^*$. Precise definitions are given in Section 2.3. Using this assumption we construct a signature scheme that is existentially unforgeable under a chosen message attack *without using random oracles*.

Currently, the most practical signature schemes secure without random oracles [GHR99,CS00] are based on the Strong RSA assumption (given an RSA modulus N and $s \in \mathbb{Z}_N^*$ it is difficult to construct a non-trivial pair $(c, s^{1/c})$ where $c \in \mathbb{Z}$). Roughly speaking, what makes Strong RSA so useful for constructing secure signature schemes is the following property: given a Strong RSA problem instance (N, s) it is possible to construct a new instance (N, s') with q known

* Supported by NSF and the Packard Foundation.

solutions $(c_i, (s')^{1/c_i})$, where the construction of any other solution $(c, (s')^{1/c})$ makes it possible to solve the original problem instance. This property provides a way to prove security against a chosen message attack. In Section 3.1 we show that the q-SDH problem has a similar property. Hence, q-SDH may be viewed as a discrete logarithm analogue of the Strong RSA assumption. We believe that the properties of q-SDH make it a useful tool for constructing cryptographic systems and we expect to see many other systems based on it.

To gain some confidence in the q-SDH assumption we provide in Section 5 a lower bound on the computational complexity of solving the q-SDH problem in a generic group model. This shows that no generic attack on q-SDH is possible. Mitsunari, Sakai, and Kasahara [MSK02] previously used a weaker variant of the q-SDH assumption to construct a traitor tracing scheme. The ideas in their paper are nice, and we use some of them here. Unfortunately, their application to tracing traitors is insecure [TSNZ03].

We present our secure signature scheme in Section 3 and prove its security against existential forgery under chosen message attack. The resulting signatures are as short as DSA signatures, but are provably secure in the absence of random oracles. Our signatures also support limited message recovery, which makes it possible to further reduce the total length of a message/signature pair. In Section 4 we show that with random oracles the q-SDH assumption gives even shorter signatures. A related system using random oracles was recently described by Zhang et al. [ZSNS04].

We refer to [BLS01] for applications of short signatures. We only mention that short digital signatures are needed in environments with stringent bandwidth constraints, such as bar-coded digital signatures on postage stamps [NS00,PV00]. We also note that Patarin et al. [PCG01,CDF03] construct short signatures whose security depends on the Hidden Field Equation (HFE) problem.

2 Preliminaries

Before presenting our results we briefly review two notions of security for signature schemes, review the definition for groups equipped with a bilinear map, and precisely state the q-SDH assumption.

2.1 Secure Signature Schemes

A signature scheme is made up of three algorithms, *KeyGen*, *Sign*, and *Verify*, for generating keys, signing, and verifying signatures, respectively.

Strong Existential Unforgeability

The standard notion of security for a signature scheme is called existential unforgeability under a chosen message attack [GMR88]. We consider a slightly stronger notion of security, called strong existential unforgeability [ADR02], which is defined using the following game between a challenger and an adversary $\mathcal{A}$:

Setup: The challenger runs algorithm *KeyGen* to obtain a public key *PK* and a private key *SK*. The adversary $\mathcal{A}$ is given *PK*.

Queries: Proceeding adaptively, $\mathcal{A}$ requests signatures on at most q_S messages of his choice $M_1, \ldots, M_{q_s} \in \{0,1\}^*$, under *PK*. The challenger responds to each query with a signature $\sigma_i = \mathit{Sign}(SK, M_i)$.

Output: Eventually, $\mathcal{A}$ outputs a pair (M, σ) and wins the game if
(1) (M, σ) is not any of $(M_1, \sigma_1), \ldots, (M_{q_s}, \sigma_{q_s})$, and
(2) $\mathit{Verify}(PK, M, \sigma) = \texttt{valid}$.

We define $\mathsf{Adv\,Sig}_{\mathcal{A}}$ to be the probability that $\mathcal{A}$ wins in the above game, taken over the coin tosses made by $\mathcal{A}$ and the challenger.

Definition 1. *A forger $\mathcal{A}$ (t, q_S, ϵ)-breaks a signature scheme if $\mathcal{A}$ runs in time at most t, $\mathcal{A}$ makes at most q_S signature queries, and $\mathsf{Adv\,Sig}_{\mathcal{A}}$ is at least ϵ. A signature scheme is (t, q_S, ϵ)-existentially unforgeable under an adaptive chosen message attack if no forger (t, q_S, ϵ)-breaks it.*

When proving security in the random oracle model we add a fourth parameter q_H denoting an upper bound on the number of queries that the adversary $\mathcal{A}$ makes to the random oracle.

We note that the definition above captures a stronger version of existential unforgeability than the standard one: we require that the adversary cannot even generate a new signature on a previously signed message. This property is required for some applications [ADR02,Sah99,CHK04]. All our signature schemes satisfy this stronger security notion.

Weak Chosen Message Attacks

We will also use a weaker notion of security which we call existential unforgeability under a weak chosen message attack. Here we require that the adversary submit all signature queries before seeing the public key. This notion is defined using the following game between a challenger and an adversary $\mathcal{A}$:

Query: $\mathcal{A}$ sends the challenger a list of q_S messages $M_1, \ldots, M_{q_s} \in \{0,1\}^*$.

Response: The challenger runs algorithm *KeyGen* to generate a public key *PK* and private key *SK*. Next, the challenger generates signatures $\sigma_i = \mathit{Sign}(SK, M_i)$ for $i = 1, \ldots, q_S$. The challenger then gives $\mathcal{A}$ the public key *PK* and the q_S signatures $\sigma_1, \ldots, \sigma_{q_s}$.

Output: Algorithm $\mathcal{A}$ outputs a pair (M, σ) and wins the game if
(1) M is not any of $M_1, \ldots, M_{q_s}$, and
(2) $\mathit{Verify}(PK, M, \sigma) = \texttt{valid}$.

We define $\mathsf{Adv\,W\text{-}Sig}_{\mathcal{A}}$ to be the probability that $\mathcal{A}$ wins in the above game, taken over the coin tosses of $\mathcal{A}$ and the challenger.

Definition 2. *A forger $\mathcal{A}$ (t, q_S, ϵ)-weakly breaks a signature scheme if $\mathcal{A}$ runs in time at most t, $\mathcal{A}$ makes at most q_S signature queries, and $\mathsf{Adv\,W\text{-}Sig}_{\mathcal{A}}$ is at least ϵ. A signature scheme is (t, q_S, ϵ)-existentially unforgeable under a weak chosen message attack if no forger (t, q_S, ϵ)-weakly breaks it.*

2.2 Bilinear Groups

Signature verification in our scheme requires a bilinear map. We briefly review the necessary facts about bilinear maps and bilinear map groups. We follow the notation in [BLS01]:

1. $\mathbb{G}_1$ and $\mathbb{G}_2$ are two (multiplicative) cyclic groups of prime order p;
2. g_1 is a generator of $\mathbb{G}_1$ and g_2 is a generator of $\mathbb{G}_2$;
3. ψ is an isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$, with $\psi(g_2) = g_1$; and
4. e is a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.

For simplicity one can set $\mathbb{G}_1 = \mathbb{G}_2$. However, as in [BLS01], we allow for the more general case where $\mathbb{G}_1 \neq \mathbb{G}_2$ so that we can take advantage of certain families of elliptic curves to obtain short signatures. Specifically, elements of $\mathbb{G}_1$ have a short representation whereas elements of $\mathbb{G}_2$ may not. The proofs of security require an efficiently computable isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$. When $\mathbb{G}_1 = \mathbb{G}_2$ and $g_1 = g_2$ one could take ψ to be the identity map. On elliptic curves we can use the trace map as ψ.

Let thus $\mathbb{G}_1$ and $\mathbb{G}_2$ be two groups as above, with an additional group $\mathbb{G}_T$ such that $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_T|$. A bilinear map is a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ with the following properties:

1. Bilinear: for all $u \in \mathbb{G}_1, v \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}$, $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degenerate: $e(g_1, g_2) \neq 1$.

We say that $(\mathbb{G}_1, \mathbb{G}_2)$ are bilinear groups if there exists a group $\mathbb{G}_T$, an isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$, and a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ as above, and e, ψ, and the group action in $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ can be computed efficiently.

Joux and Nguyen [JN01] showed that an efficiently computable bilinear map e provides an algorithm for solving the Decision Diffie-Hellman problem (DDH). Our results can be stated using a generic algorithm for DDH. Nevertheless, for the sake of concreteness we instead describe our results by directly referring to the bilinear map.

2.3 The Strong Diffie-Hellman Assumption

Before describing the new signature schemes, we first state precisely the hardness assumption on which they are based. Let $\mathbb{G}_1, \mathbb{G}_2$ be two cyclic groups of prime order p, where possibly $\mathbb{G}_1 = \mathbb{G}_2$. Let g_1 be a generator of $\mathbb{G}_1$ and g_2 a generator of $\mathbb{G}_2$.

q-Strong Diffie-Hellman Problem. The q-SDH problem in $(\mathbb{G}_1, \mathbb{G}_2)$ is defined as follows: given a $(q+2)$-tuple $(g_1, g_2, g_2^x, g_2^{(x^2)}, \ldots, g_2^{(x^q)})$ as input, output a pair $(c,\, g_1^{1/(x+c)})$ where $c \in \mathbb{Z}_p^*$. An algorithm $\mathcal{A}$ has advantage ϵ in solving q-SDH in $(\mathbb{G}_1, \mathbb{G}_2)$ if

$$\Pr\left[\mathcal{A}(g_1, g_2, g_2^x, \ldots, g_2^{(x^q)}) = (c,\, g_1^{\frac{1}{x+c}})\right] \geq \epsilon$$

where the probability is over the random choice of x in $\mathbb{Z}_p^*$ and the random bits consumed by $\mathcal{A}$.

Definition 3. *We say that the (q,t,ϵ)-SDH assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$ if no t-time algorithm has advantage at least ϵ in solving the q-SDH problem in $(\mathbb{G}_1, \mathbb{G}_2)$.*

Occasionally we drop the t and ϵ and refer to the q-SDH assumption rather than the (q,t,ϵ)-SDH assumption. As we will see in the next section the q-SDH assumption has similar properties to the Strong RSA problem and we therefore view q-SDH as a discrete logarithm analogue of the Strong RSA assumption.

To provide some confidence in the q-SDH assumption, we prove in Section 5 a lower bound on the complexity of solving the q-SDH problem in a generic group. Furthermore, we note that the Strong Diffie-Hellman problem has a simple random self-reduction in $(\mathbb{G}_1, \mathbb{G}_2)$.

A weaker version of the q-SDH assumption was previously used by Mitsunari, Sakai, and Kasahara [MSK02] to construct a traitor tracing system (see [TSNZ03] for an analysis). Using our notation, their version of the assumption requires Algorithm $\mathcal{A}$ to output $g_1^{1/(x+c)}$ for a *given input value* c. In the assumption above we allow $\mathcal{A}$ to choose c. When c is pre-specified the q-SDH problem is equivalent to the following problem: given $(g_1, g_2, g_2^x, g_2^{x^2}, \ldots, g_2^{x^q})$ output $g_1^{1/x}$. We note that when $\mathcal{A}$ is allowed to choose c no such equivalence is known. The weaker variant of the assumption was recently used to construct an efficient selective identity secure identity based encryption (IBE) system without random oracles [BB04a].

3 Short Signatures Without Random Oracles

We now construct a fully secure short signature scheme in the standard model using the q-SDH assumption. We consider this to be the main result of the paper.

Let $(\mathbb{G}_1, \mathbb{G}_2)$ be bilinear groups where $|\mathbb{G}_1| = |\mathbb{G}_2| = p$ for some prime p. As usual, g_1 is a generator of $\mathbb{G}_1$ and g_2 a generator of $\mathbb{G}_2$. For the moment we assume that the messages m to be signed are elements in $\mathbb{Z}_p^*$, but as we mention in Section 3.5, the domain can be extended to all of $\{0,1\}^*$ using a collision resistant hash function $H : \{0,1\}^* \to \mathbb{Z}_p^*$.

Key generation: Pick random $x, y \xleftarrow{\text{R}} \mathbb{Z}_p^*$, and compute $u \leftarrow g_2^x \in \mathbb{G}_2$ and $v \leftarrow g_2^y \in \mathbb{G}_2$. The public key is (g_1, g_2, u, v). The secret key is (x, y).

Signing: Given a secret key $x, y \in \mathbb{Z}_p^*$ and a message $m \in \mathbb{Z}_p^*$, pick a random $r \in \mathbb{Z}_p^*$ and compute $\sigma \leftarrow g_1^{1/(x+m+yr)} \in \mathbb{G}_1$. Here $1/(x+m+yr)$ is computed modulo p. In the unlikely event that $x + m + yr = 0$ we try again with a different random r. The signature is (σ, r).

Verification: Given a public key (g_1, g_2, u, v), a message $m \in \mathbb{Z}_p^*$, and a signature (σ, r), verify that

$$e(\sigma,\ u \cdot g_2^m \cdot v^r) = e(g_1, g_2)$$

If the equality holds the result is `valid`; otherwise the result is `invalid`.

Signature length. A signature contains two elements (σ, r), each of length approximately $\log_2(p)$ bits, therefore the total signature length is approximately $2\log_2(p)$. When using the elliptic curves described in [BLS01] we obtain a signature whose length is approximately the same as a DSA signature with the same security, but which is provably existentially unforgeable under a chosen message attack without the random oracle model.

Performance. Key and signature generation times are comparable to BLS signatures. Verification time is faster since verification requires only one pairing and one multi-exponentiation. The value $e(g_1, g_2)$ only needs to be computed at initialization time and cached. In comparison, BLS signature verification requires two pairing computations. Since exponentiation tends to be significantly faster than pairing, signature verification is faster than in the BLS system.

Security. The following theorem shows that the scheme above is existentially unforgeable in the strong sense under chosen message attacks, provided that the q-SDH assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$.

Theorem 1. *Suppose the (q, t', ϵ')-SDH assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$. Then the signature scheme above is (t, q_S, ϵ)-secure against existential forgery under a chosen message attack provided that*

$$t \leq t' - o(t'), \qquad q_S < q \qquad \textit{and} \qquad \epsilon \geq 2(\epsilon' + q_S/p) \approx 2\epsilon'$$

Proof. We prove the theorem using two lemmas. In Lemma 1, we first describe a simplified signature scheme and prove its existential unforgeability against *weak* chosen message attacks under the q-SDH assumption. In Lemma 2, we then show that the security of the weak scheme implies the security of the full scheme. From these results (Lemmas 1 and 2), Theorem 1 follows easily. We present the proof in two steps since the construction used to prove Lemma 1 will be used later on in the paper. □

3.1 A Weakly Secure Short Signature Scheme

We first show how the q-SDH assumption can be used to construct an existentially unforgeable scheme under a *weak* chosen message attack. This construction demonstrates the main properties of the q-SDH assumption. In the next section we show that the security of this weak scheme implies the security of the full scheme above.

The weakly secure short signature scheme is as follows. As before, let $(\mathbb{G}_1, \mathbb{G}_2)$ be bilinear groups where $|\mathbb{G}_1| = |\mathbb{G}_2| = p$ for some prime p. As usual, g_1 is a generator of $\mathbb{G}_1$ and g_2 a generator of $\mathbb{G}_2$. For the moment we assume that the messages m to be signed are elements in $\mathbb{Z}_p^*$.

Key generation: Pick random $x \stackrel{\mathrm{R}}{\leftarrow} \mathbb{Z}_p^*$, and compute $v \leftarrow g_2^x \in \mathbb{G}_2$. The public key is (g_1, g_2, v). The secret key is x.

Signing: Given a secret key $x \in \mathbb{Z}_p^*$ and a message $m \in \mathbb{Z}_p^*$, output the signature $\sigma \leftarrow g_1^{1/(x+m)} \in \mathbb{G}_1$. Here $1/(x+m)$ is computed modulo p. By convention in this context we define $1/0$ to be 0 so that in the unlikely event that $x+m=0$ we have $\sigma \leftarrow 1$.

Verification: Given a public key (g_1, g_2, v), a message $m \in \mathbb{Z}_p^*$, and a signature $\sigma \in \mathbb{G}_1$, verify that

$$e(\sigma, v \cdot g_2^m) = e(g_1, g_2)$$

If equality holds output `valid`. If $\sigma = 1$ and $v \cdot g_2^m = 1$ output `valid`. Otherwise, output `invalid`.

We show that the basic signature scheme above is existentially unforgeable under a *weak* chosen message attack. The proof of the following lemma uses a similar method to the proof of Theorem 3.5 of Mitsunari et al. [MSK02].

Lemma 1. *Suppose the (q, t', ϵ)-SDH assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$. Then the basic signature scheme above is (t, q_S, ϵ)-secure against existential forgery under a weak chosen message attack provided that*

$$t \leq t' - O(q^2) \qquad \textit{and} \qquad q_S < q$$

Proof. Assume $\mathcal{A}$ is a forger that (t, q_S, ϵ)-breaks the signature scheme. We construct an algorithm $\mathcal{B}$ that, by interacting with $\mathcal{A}$, solves the q-SDH problem in time t' with advantage ϵ. Algorithm $\mathcal{B}$ is given an instance $(g_1, g_2, A_1, \ldots, A_q)$ of the q-SDH problem, where $A_i = g_2^{(x^i)} \in \mathbb{G}_2$ for $i = 1, \ldots, q$ and for some unknown $x \in \mathbb{Z}_p^*$. For convenience we set $A_0 = g_2$. Algorithm $\mathcal{B}$'s goal is to produce a pair $(c, g_1^{1/(x+c)})$ for some $c \in \mathbb{Z}_p^*$. Algorithm $\mathcal{B}$ does so by interacting with the forger $\mathcal{A}$ as follows:

Query: Algorithm $\mathcal{A}$ outputs a list of distinct q_S messages $m_1, \ldots, m_{q_s} \in \mathbb{Z}_p^*$, where $q_S < q$. Since $\mathcal{A}$ must reveal its queries up front, we may assume that $\mathcal{A}$ outputs exactly $q-1$ messages to be signed (if the actual number is less, we can always virtually reduce the value of q so that $q = q_S + 1$).

Response: $\mathcal{B}$ must respond with a public key and signatures on the $q-1$ messages from $\mathcal{A}$. Let $f(y)$ be the polynomial $f(y) = \prod_{i=1}^{q-1}(y+m_i)$. Expand $f(y)$ and write $f(y) = \sum_{i=0}^{q-1} \alpha_i y^i$ where $\alpha_0, \ldots, \alpha_{q-1} \in \mathbb{Z}_p$ are the coefficients of the polynomial $f(y)$. Compute:

$$g_2' \leftarrow \prod_{i=0}^{q-1} (A_i)^{\alpha_i} = g_2^{f(x)} \quad \text{and} \quad h \leftarrow \prod_{i=1}^{q} A_i^{\alpha_{i-1}} = g_2^{xf(x)} = (g_2')^x$$

Also, let $g_1' = \psi(g_2')$. The public key given to $\mathcal{A}$ is (g_1', g_2', h). Next, for each $i = 1, \ldots q-1$, Algorithm $\mathcal{B}$ must generate a signature σ_i on m_i. To do so, let $f_i(y)$ be the polynomial $f_i(y) = f(y)/(y+m_i) = \prod_{j=1, j\neq i}^{q-1}(y+m_j)$. As before, we expand f_i and write $f_i(y) = \sum_{j=0}^{q-2} \beta_j y^j$. Compute

$$S_i \leftarrow \prod_{j=0}^{q-2} A_j^{\beta_j} = g_2^{f_i(x)} = (g_2')^{1/(x+m_i)} \in \mathbb{G}_2$$

Observe that $\sigma_i = \psi(S_i) \in \mathbb{G}_1$ is a valid signature on m under the public key (g'_1, g'_2, h). Algorithm $\mathcal{B}$ gives $\mathcal{A}$ the $q-1$ signatures $\sigma_1, \ldots, \sigma_{q-1}$.

Output: Algorithm $\mathcal{A}$ returns a forgery (m_*, σ_*) such that $\sigma_* \in \mathbb{G}_1$ is a valid signature on $m_* \in \mathbb{Z}_p^*$ and $m_* \notin \{m_1 \ldots, m_{q-1}\}$ since there is only one valid signature per message. In other words, $e(\sigma_*, h \cdot (g'_2)^{m_*}) = e(g'_1, g'_2)$. Since $h = (g'_2)^x$ we have that $e(\sigma_*, (g'_2)^{x+m_*}) = e(g'_1, g'_2)$ and therefore

$$\sigma_* = (g'_1)^{1/(x+m_*)} = (g_1)^{f(x)/(x+m_*)} \tag{1}$$

Using long division we write the polynomial f as $f(y) = \gamma(y)(y + m_*) + \gamma_{-1}$ for some polynomial $\gamma(y) = \sum_{i=0}^{q-2} \gamma_i y^i$ and some $\gamma_{-1} \in \mathbb{Z}_p$. Then the rational fraction $f(y)/(y + m_*)$ in the exponent on the right side of Equation (1) can be written as

$$f(y)/(y + m_*) = \frac{\gamma_{-1}}{y + m_*} + \sum_{i=0}^{q-2} \gamma_i y^i$$

Note that $\gamma_{-1} \neq 0$, since $f(y) = \prod_{i=1}^{q-1}(y + m_i)$ and $m_* \notin \{m_1, \ldots, m_{q-1}\}$, as thus $(y + m_*)$ does not divide $f(y)$. Then algorithm $\mathcal{B}$ computes

$$w \leftarrow \left(\sigma_* \cdot \prod_{i=0}^{q-1} \psi(A_i)^{-\gamma_i}\right)^{1/\gamma_{-1}} = g_1^{1/(x+m_*)}$$

and returns (m_*, w) as the solution to the q-SDH instance.

The claimed bounds are obvious by construction of the reduction. □

3.2 From Weak Security To Full Security

We now present a reduction from the security of the basic scheme of Lemma 1 to the security of the full signature scheme described at the onset of Section 3. This will complete the proof of Theorem 1.

Lemma 2. *Suppose that the basic signature scheme of Lemma 1 is (t', q_S, ϵ')-weakly secure. Then the full signature scheme is (t, q_S, ϵ)-secure against existential forgery under a chosen message attack provided that*

$$t \leq t' - O(q_S) \qquad \textit{and} \qquad \epsilon \geq 2(\epsilon' + q_S/p) \approx 2\epsilon'$$

Proof. Assume $\mathcal{A}$ is a forger that (t, q_S, ϵ)-breaks the full signature scheme. We construct an algorithm $\mathcal{B}$ that $(t+O(q_S), q_S, \epsilon/2 - q_S/p)$-weakly breaks the basic signature scheme of Lemma 1.

Before describing Algorithm $\mathcal{B}$ we distinguish between two types of forgers that $\mathcal{A}$ can emulate. Let (h_1, h_2, u, v) be the public key given to forger $\mathcal{A}$ where $u = g_2^x$ and $v = g_2^y$. Suppose $\mathcal{A}$ asks for signatures on messages $m_1, \ldots, m_{q_s} \in \mathbb{Z}_p^*$ and is given signatures (σ_i, r_i) for $i = 1, \ldots, q_S$ on these messages. Let $w_i = m_i + y r_i$ and let (m_*, σ_*, r_*) be the forgery produced by $\mathcal{A}$. We distinguish between two types of forgers:

Type-1 forger: a forger that either (i) makes a signature query for the message $m = -x$, or (ii) outputs a forgery where $m_* + yr_* \notin \{w_1, \ldots, w_{q_s}\}$.

Type-2 forger: a forger that (i) never makes a signature query for the message $m = -x$, and (ii) outputs a forgery where $m_* + yr_* = w_i$ for some $i \in \{1, \ldots, q_S\}$.

We show that either forger can be used to forge signatures for the weak signature scheme of Lemma 1. However, the reduction works differently for each forger type. Therefore, initially $\mathcal{B}$ will choose a random bit $c_{\text{mode}} \in \{1, 2\}$ that indicates its guess for the type of forger that $\mathcal{A}$ will emulate. The simulation proceeds differently for each mode.

We are now ready to describe Algorithm $\mathcal{B}$. It produces a forgery for the signature scheme of Lemma 1 as follows:

Setup: Algorithm $\mathcal{B}$ first picks a random bit $c_{\text{mode}} \in \{1, 2\}$. Next, $\mathcal{B}$ sends to its own challenger a list of q_S random messages $w_1, \ldots, w_{q_s} \in \mathbb{Z}_p^*$ for which it requests a signature. The challenger responds with a public key (g_1, g_2, u) and signatures $\sigma_1, \ldots, \sigma_{q_s} \in \mathbb{G}_1$ on these messages. We know that $e(\sigma_i, g_2^{w_i} u) = e(g_1, g_2)$ for all $i = 1, \ldots, q_S$. Then:

- (If $c_{\text{mode}} = 1$). $\mathcal{B}$ picks a random $y \in \mathbb{Z}_p^*$ and gives $\mathcal{A}$ the public key $PK_1 = (g_1, g_2, u, g_2^y)$.
- (If $c_{\text{mode}} = 2$). $\mathcal{B}$ picks a random $x \in \mathbb{Z}_p^*$ and gives $\mathcal{A}$ the public key $PK_2 = (g_1, g_2, g_2^x, u)$.

In either case, we note that $\mathcal{B}$ provides the adversary $\mathcal{A}$ with a valid public key (g_1, g_2, U, V).

Signature queries: The forger $\mathcal{A}$ can issue up to q_S signature queries in an adaptive fashion. In order to respond, $\mathcal{B}$ maintains a list H-list of tuples (m_i, r_i, W_i) and a query counter ℓ which is initially set to 0. Upon receiving a signature query for m, Algorithm $\mathcal{B}$ increments ℓ by one. Then:

- (If $c_{\text{mode}} = 1$). Check if $g_2^{-m} = u$. If so, then $\mathcal{B}$ just obtained the private key for the public key (g_1, g_2, u) it was given, which allows it to forge the signature on any message of its choice. At this point $\mathcal{B}$ successfully terminates the simulation.
Otherwise, set $r_\ell = (w_\ell - m)/y \in \mathbb{Z}_p^*$. In the very unlikely event that $r_\ell = 0$, Algorithm $\mathcal{B}$ reports failure and aborts. Otherwise, Algorithm $\mathcal{B}$ gives $\mathcal{A}$ the signature (σ_ℓ, r_ℓ). This is a valid signature on m under PK_1 since r_ℓ is uniform in $\mathbb{Z}_p^*$ and

$$e(\sigma_\ell,\ U \cdot g_2^m \cdot V^{r_\ell}) = e(\sigma_\ell,\ u \cdot g_2^m \cdot g_2^{yr_\ell}) = e(\sigma_\ell, u \cdot g_2^{w_\ell}) = e(g_1, g_2)$$

- (If $c_{\text{mode}} = 2$). Set $r_\ell = (x + m)/w_\ell \in \mathbb{Z}_p^*$. If $r_\ell = 0$, Algorithm $\mathcal{B}$ reports failure and aborts. Otherwise, give $\mathcal{A}$ the signature $(\sigma_\ell^{1/r_\ell}, r_\ell)$. This is a valid signature on m under PK_2 since r_ℓ is uniform in $\mathbb{Z}_p^*$ and

$$e(\sigma_\ell^{1/r_\ell},\ U \cdot g_2^m \cdot V^{r_\ell}) = e(\sigma_\ell^{1/r_\ell},\ g_2^x \cdot g_2^m \cdot u^{r_\ell}) = e(\sigma_\ell,\ g_2^{w_\ell} u) = e(g_1, g_2)$$

In either case if $\mathcal{B}$ does not stop it responds with a valid signature on m.
In either case Algorithm $\mathcal{B}$ adds the tuple $(m, r_\ell, g_2^m V^{r_\ell})$ to the H-list.

Output: Eventually, $\mathcal{A}$ returns a forgery (m_*, σ_*, r_*), where (σ_*, r_*) is a valid forgery distinct from any previously given signature on message m_*. Note that by adding dummy queries as necessary, we may assume that $\mathcal{A}$ made exactly q_S signature queries. Let $W_* \leftarrow g_2^{m_*} V^{r_*}$. Algorithm $\mathcal{B}$ searches the H-list for a tuple whose rightmost component is equal to W_*. There are two possibilities:

Type-1 forgery: No tuple of the form $(\cdot, \cdot, W_*)$ appears on the H-list.

Type-2 forgery: The H-list contains at least one tuple (m_j, r_j, W_j) such that $W_j = W_*$.

Let $b_{\text{type}} \leftarrow 1$ if $\mathcal{A}$ produced a type-1 forgery, or $\mathcal{A}$ made a signature query for a message m such that $g_2^{-m} = U$. In all other cases, set $b_{\text{type}} \leftarrow 2$. If $b_{\text{type}} \neq c_{\text{mode}}$ then $\mathcal{B}$ reports failure and aborts. Otherwise, $\mathcal{B}$ outputs an existential forgery on the basic signature scheme as follows:

- (If $c_{\text{mode}} = b_{\text{type}} = 1$). If $\mathcal{A}$ made a signature query for a message m such that $g_2^{-m} = U$ then $\mathcal{B}$ is already done. Therefore, we assume $\mathcal{A}$ produced a type-1 forgery. Since the forgery is valid, we have

$$e(g_1, g_2) = e(\sigma_*,\ U \cdot g_2^{m_*} \cdot V^{r_*}) = e(\sigma_*,\ u \cdot g_2^{m_* + y r_*})$$

Let $w_* = m_* + y r_*$. It follows that (w_*, σ_*) is a valid message/signature pair in the basic signature scheme. Furthermore, it is a valid existential forgery for the basic scheme since in a type-1 forgery Algorithm $\mathcal{B}$ did not request a signature on the message $w_* \in \mathbb{Z}_p^*$. Indeed, $\mathcal{B}$ only requested signatures on messages $w_j = m_j + y r_j$ where $(m_j, r_j, g_2^{w_j})$ is a tuple in the H-list, but $g_2^{w_*}$ is not equal to any $g_2^{w_j}$ on the H-list. Algorithm $\mathcal{B}$ outputs (w_*, σ_*) as the required existential forgery.
- (If $c_{\text{mode}} = b_{\text{type}} = 2$). Let (m_j, r_j, W_j) be a tuple on the H-list where $W_j = W_*$. Since $V = u$ we know that $g_2^{m_j} u^{r_j} = g_2^{m_*} u^{r_*}$. Write $u = g_2^z$ for some $z \in \mathbb{Z}_p^*$ so that $m_j + z r_j = m_* + z r_*$. We know that $(m_j, r_j) \neq (m_*, r_*)$, otherwise the forgery would be identical to a previously given signature on the query message m_j. Since $g_2^{m_j} u^{r_j} = g_2^{m_*} u^{r_*}$ it follows that $m_j \neq m_*$ and $r_j \neq r_*$. Therefore, $z = (m_* - m_j)/(r_j - r_*) \in \mathbb{Z}_p^*$. Hence, $\mathcal{B}$ just recovered the private key for the public key (g_1, g_2, u) it was given. Algorithm $\mathcal{B}$ can now forge a signature on any message of its choice.

This completes the description of Algorithm $\mathcal{B}$.

A standard argument shows that if $\mathcal{B}$ does not abort, then, from the viewpoint of $\mathcal{A}$, the simulation provided by $\mathcal{B}$ is indistinguishable from a real attack scenario. In particular, (i) the view from $\mathcal{A}$ is independent of the value of c_{mode}, (ii) the public keys are uniformly distributed, and (iii) the signatures are correct. Therefore, $\mathcal{A}$ produces a valid forgery in time t with probability at least ϵ.

It remains to bound the probability that $\mathcal{B}$ does not abort. We argue as follows:

– Conditionally on the event $c_{\text{mode}} = b_{\text{type}} = 1$, Algorithm $\mathcal{B}$ aborts if $\mathcal{A}$ issued a signature query $m_\ell = w_\ell$. This happens with probability at most q_S/p.
– Conditionally on the event $c_{\text{mode}} = b_{\text{type}} = 2$, Algorithm $\mathcal{B}$ does not abort.

Since c_{mode} is independent of b_{type} we have that $\Pr[c_{\text{mode}} = b_{\text{type}}] = 1/2$. It now follows that $\mathcal{B}$ produces a valid forgery with probability at least $\epsilon/2 - q_S/p$, as required. □

Since in the full scheme a single message has many valid signatures, it is worth repeating that the full signature scheme is existentially unforgeable in the strong sense: the adversary cannot make any forgery, even on messages which are already signed.

3.3 Relation to Chameleon Hash Signatures

It is instructive to consider the relation between the full signature scheme above and a signature construction based on the Strong RSA assumption due to Gennaro, Halevi, and Rabin (GHR) [GHR99]. GHR signatures are pairs $(r, s^{1/H(m,r)})$ where H is a Chameleon hash [KR00], r is random in some range, and arithmetic is done modulo an RSA modulus N. Looking closely, one can see some parallels between the proof of security in Lemma 2 above and the proof of security in [GHR99]. There are three interesting points to make:

– The $m + yr$ component in our signature scheme provides us with the functionality of a Chameleon hash: given m, we can choose r so that $m + yr$ maps to some predefined value of our choice. This makes it possible to handle the chosen message attack. Embedding the hash $m + yr$ directly in the signature scheme results in a much more efficient construction than using an explicit Chameleon hash (which requires additional exponentiations). This is not known to be possible with Strong RSA signatures.
– One difficulty with GHR signatures is that given a solution $(6, s^{1/6})$ to the Strong RSA problem one can deduce another solution, e.g. $(3, s^{1/3})$. Thus, given a GHR signature on one message it possible to deduce a GHR signature on another message (see [GHR99,CN00] for details). Gennaro et al. solve this problem by ensuring that $H(m, r)$ always maps to a prime; However, that makes it difficult to compute the hash (a different solution is given in [CS00]). This issue does not come up at all in our signature scheme above.
– We obtain short signatures since, unlike Strong RSA, the q-SDH assumption applies to groups with a short representation.

Thus, we see that Strong Diffie-Hellman leads to signatures that are simpler, more efficient, and shorter than their Strong RSA counterparts.

3.4 Limited Message Recovery

We now describe another useful property of the signature schemes whereby the total size of signed messages can be further reduced at the cost of increasing the verification time. The technique applies equally well to the fully secure signature scheme as to the weakly secure one.

A standard technique for shortening the total length of message/signature pairs is to encode a part of the message in the signature [MVV97]. Signatures based on trapdoor permutations support very efficient message recovery.

At the other end of the spectrum, a trivial signature compression mechanism that applies to any signature scheme is as follows: Rather than transmit a message/signature pair (M, σ), the sender transmits $(\hat{M}, \sigma)$ where $\hat{M}$ is the same as M except that the last t bits are truncated. In other words, $\hat{M}$ is t bits shorter than M. To verify $(\hat{M}, \sigma)$ the verifier tries all 2^t possible values for the truncated bits and accepts the signature if one of them verifies. To reconstruct the original signed message M, the verifier appends to $\hat{M}$ the t bits for which the signature verified.

This trivial method shows that the pair (M, σ) can be shortened by t-bits at the cost of increasing verification time by a factor of 2^t. For our signature scheme we obtain a better tradeoff: the pair (M, σ) can be shortened by t bits at the cost of increasing verification time by a factor of $2^{t/2}$ only. We refer to this property as limited message recovery.

Limited Message Recovery. Limited message recovery applies to both the full signature scheme and the weakly secure signature scheme of Lemma 1. For simplicity, we only show how limited message recovery applies to the full signature scheme. Assume messages are k-bit strings represented as integers in $\mathbb{Z}_p^*$. Let (g_1, g_2, u, v) be a public key in the full scheme. Suppose we are given the signed message $(\hat{m}, \sigma, r)$ where $\hat{m}$ is a truncation of the last t bits of $m \in \mathbb{Z}_p^*$. Thus $m = \hat{m} \cdot 2^t + \delta$ for some integer $0 \le \delta < 2^t$. Our goal is to verify the signed message $(\hat{m}, \sigma, r)$ and to reconstruct the missing bits δ in time $2^{t/2}$. To do so, we first rewrite the verification equation $e(\sigma,\, u \cdot v^r \cdot g_2^m) = e(g_1, g_2)$ as

$$e(\sigma, g_2)^m = \frac{e(g_1, g_2)}{e(\sigma, u \cdot v^r)}$$

Substituting $m = \hat{m} \cdot 2^t + \delta$ we obtain

$$e(\sigma, g_2)^\delta = \frac{e(g_1, g_2)}{e(\sigma,\, u \cdot v^r \cdot g_2^{\hat{m}2^t})} \tag{2}$$

Now, we say that $(\hat{m}, \sigma, r)$ is valid if there exists an integer $\delta \in [0, 2^t)$ satisfying equation (2). Finding such a δ takes time approximately $2^{t/2}$ using Pollard's Lambda method [MVV97, p.128] for computing discrete logarithms. Thus, we can verify the signature and recover the t missing message bits in time $2^{t/2}$, as required.

Ultra Short Weakly Secure Signatures. Obvious applications of limited message recovery are situations where bandwidth is extremely limited, such as when the signature is an authenticator that is to be typed-in by a human. The messages in such applications are typically chosen and signed by a central authority, so that adaptive chosen message attacks are typically not a concern. It is safe in those cases to use the weakly secure signature scheme of Lemma 1, and apply limited message recovery to further shrink the already compact signatures it produces. Specifically, using t-bit truncation as above we obtain a total signature overhead of $(160 - t)$ bits for common security parameters, at the cost of requiring $2^{t/2}$ pairing computations for signature verification. We emphasize that the security of this system does not rely on random oracles.

3.5 Arbitrary Message Signing

We can extend our signature schemes to sign arbitrary messages in $\{0,1\}^*$, as opposed to merely messages in $\mathbb{Z}_p^*$, by first hashing the message using a collision-resistant hash function $H : \{0,1\}^* \to \mathbb{Z}_p^*$ prior to both signing and verifying. A standard argument shows that if the scheme above is secure against existential forgery under a chosen message attack (in the strong sense) then so is the scheme with the hash. The result is a signature scheme for arbitrary messages in $\{0,1\}^*$. We note that there is no need for a full domain hash into $\mathbb{Z}_p^*$; a collision resistant hash function $H : \{0,1\}^* \to \{1,\ldots,2^b\}$ for $2^b < p$ is sufficient for the security proof. This transformation applies to both the fully and the weakly secure signature schemes described above.

4 Shorter Signatures With Random Oracles

For completeness we show that the weakly secure signature scheme of Lemma 1 gives rise to very efficient and fully secure short signatures in the random oracle model. To do so, we show a general transformation from any existentially unforgeable signature scheme under a *weak* chosen message attack into an existentially unforgeable signature scheme under a *standard* chosen message attack (in the strong sense), in the random oracle model. This gives a very efficient short signature scheme based on q-SDH in the random oracle model. We analyze our construction using a method of Katz and Wang [KW03] which gives a very tight reduction to the security of the underlying signature. We note that a closely related system with a weaker security analysis was independently discovered by Zhang et al. [ZSNS04].

Let (*KeyGen*, *Sign*, *Verify*) be an existentially unforgeable signature under a *weak* chosen message attack. We assume that the scheme signs messages in some finite set Σ and that the private keys are in some set Π. We need two hash functions $H_1 : \Pi \times \{0,1\}^* \to \{0,1\}$ and $H_2 : \{0,1\} \times \{0,1\}^* \to \Sigma$ that will be viewed as random oracles in the security analysis. The hash-signature scheme is as follows:

Key generation: Same as *KeyGen*. The public key is *PK*; The secret key is $SK \in \Pi$.

Signing: Given a secret key *SK*, and given a message $M \in \{0,1\}^*$, compute $b \leftarrow H_1(SK, M) \in \{0,1\}$ and $m \leftarrow H_2(b, M) \in \Sigma$. Output the signature

$(b, Sign(m))$. Note that signatures are one bit longer than in the underlying signature scheme.

Verification: Given a public key PK, a message $M \in \{0,1\}^*$, and a signature (b, σ), output `valid` if $Verify(PK, H_2(b, M), \sigma) = \texttt{valid}$.

Theorem 2 below proves security of the scheme. Note that the security reduction in Theorem 2 is tight, namely, an attacker on the hash-signature scheme with success probability ϵ is converted to an attacker on the underlying signature with success probability approximately $\epsilon/2$. Proofs of signature schemes in the random oracle model are often far less tight. The proof is given in the full version of the paper [BB04b].

Theorem 2. *Suppose (KeyGen, Sign, Verify) is (t', q'_S, ϵ')-existentially unforgeable under a weak chosen message attack. Then the corresponding hash-signature scheme is (t, q_S, q_H, ϵ)-secure against existential forgery under an adaptive chosen message attack, in the random oracle model, whenever $q_S + q_H < q'_S$, and for all all t and ϵ satisfying*

$$t \leq t' - o(t') \qquad \textit{and} \qquad \epsilon \geq 2\epsilon'/(1 - \frac{q'_S}{|\Sigma|}) \approx 2\epsilon'$$

Applying Theorem 2 to the weakly secure scheme of Lemma 1 gives an efficient short signature existentially unforgeable under a *standard* chosen message attack in the random oracle model assuming $(q_S + q_H + 1)$-SDH. For a public key $(g_1, g_2, v = g_2^x)$ and a hash function $H : \{0,1\}^* \to \mathbb{Z}_p^*$ a signature on a message m is defined as the value $\sigma \leftarrow g_1^{1/(x+H(b,m))} \in \mathbb{G}_1$ concatenated with the bit $b \in \{0,1\}$. To verify the signature, check that $e(\sigma, v \cdot g_2^{H(b,m)}) = e(g_1, g_2)$. We see that signature length is essentially the same as in BLS signatures, but verification time is approximately half that of BLS. During verification, exponentiation is always base g_2 which enables a further speed-up by pre-computing certain powers of g_2.

Full Domain Hash. Another method for converting a signature scheme secure under a weak chosen message attack into a scheme secure under a standard chosen message attack is to simply apply *Sign* and *Verify* to $H(M)$ rather than M. In other words, we hash $M \in \{0,1\}^*$ using a full domain hash H prior to signing and verifying. Security in the random oracle model is shown using a similar argument to Coron's analysis of the Full Domain Hash [Cor00]. However, the resulting reduction is not tight: an attacker on this hash-then-sign signature with success probability ϵ yields an attacker on the underlying signature with success probability approximately ϵ/q_S. We note, however, that these proofs are set in the random oracle model and therefore it is not clear whether the efficiency of the security reduction is relevant to actual security in the real world. Therefore, since this full domain hash signature scheme is slightly simpler that the system in Theorem 2 it might be preferable to use it rather than the system of Theorem 2. When we apply the full domain hash to the weakly secure scheme of Lemma 1, we obtain a secure signature under a standard chosen message attack assuming

$(q_S + q_H + 1)$-SDH. A signature is one element, namely $\sigma \leftarrow g_1^{1/(x+H(m))} \in \mathbb{G}_1$. As before, signature verification is twice as fast as in BLS signatures. As mentioned above, a similar scheme was independently proposed by Zhang et al. [ZSNS04]. We also note that, in the random oracle model, security of this full domain hash scheme can be proven under a slightly weaker complexity assumption than q-SDH, namely that the value c in the q-SDH assumption is pre-specified rather than chosen by the adversary. However, the resulting security reduction is far less efficient.

5 Generic Security of the q-SDH Assumption

To provide more confidence in the q-SDH assumption we prove a lower bound on the computational complexity of the q-SDH problem for generic groups in the sense of Shoup [Sho97].

In the generic group model, elements of $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ appear to be encoded as unique random strings, so that no property other than equality can be directly tested by the adversary. Five oracles are assumed to perform operations between group elements, such as computing the group action in each of the three groups $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$, as well as the isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$, and the bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. The opaque encoding of the elements of $\mathbb{G}_1$ is modeled as an injective function $\xi_1 : \mathbb{Z}_p \to \Xi_1$, where $\Xi_1 \subset \{0,1\}^*$, which maps all $a \in \mathbb{Z}_p$ to the string representation $\xi_1(g^a)$ of $g^a \in \mathbb{G}_1$. We similarly define $\xi_2 : \mathbb{Z}_p \to \Xi_2$ for $\mathbb{G}_2$ and $\xi_T : \mathbb{Z}_p \to \Xi_T$ for $\mathbb{G}_T$. The attacker $\mathcal{A}$ communicates with the oracles using the ξ-representations of the group elements only.

Theorem 3. *Let $\mathcal{A}$ be an algorithm that solves the q-SDH problem in the generic group model, making a total of at most q_G queries to the oracles computing the group action in $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$, the oracle computing the isomorphism ψ, and the oracle computing the bilinear pairing e. If $x \in \mathbb{Z}_p^*$ and ξ_1, ξ_2, ξ_T are chosen at random, then the probability ϵ that $\mathcal{A}(p, \xi_1(1), \xi_2(1), \xi_2(x), \ldots, \xi_2(x^q))$ outputs $(c, \xi_1(\frac{1}{x+c}))$ with $c \in \mathbb{Z}_p^*$, is bounded by*

$$\epsilon \le \frac{(q_G + q + 2)^2 q}{p} = O\left(\frac{(q_G)^2 q + q^3}{p}\right)$$

Proof. Consider an algorithm $\mathcal{B}$ that plays the following game with $\mathcal{A}$.

$\mathcal{B}$ maintains three lists of pairs $L_1 = \{(F_{1,i}, \xi_{1,i}) : i = 0, \ldots, \tau_1 - 1\}$, $L_2 = \{(F_{2,i}, \xi_{2,i}) : i = 0, \ldots, \tau_2 - 1\}$, $L_T = \{(F_{T,i}, \xi_{T,i}) : i = 0, \ldots, \tau_T - 1\}$, such that, at step τ in the game, $\tau_1 + \tau_2 + \tau_T = \tau + q + 2$. The $F_{1,i}$ and $F_{2,i}$ are polynomials of degree $\le q$ in $\mathbb{Z}_p[x]$, and the $F_{T,i}$ are polynomials of degree $\le 2q$ in $\mathbb{Z}_p[x]$. The $\xi_{1,i}$, $\xi_{2,i}$, $\xi_{T,i}$ are strings in $\{0,1\}^*$. The lists are initialized at step $\tau = 0$ by taking $\tau_1 = 1$, $\tau_2 = q + 1$, $\tau_T = 0$, and posing $F_{1,0} = 1$, and $F_{2,i} = x^i$ for $i \in \{0, \ldots, q\}$. The corresponding $\xi_{1,0}$ and $\xi_{2,i}$ are set to arbitrary distinct strings in $\{0,1\}^*$.

We may assume that $\mathcal{A}$ only makes oracle queries on strings previously obtained form $\mathcal{B}$, since $\mathcal{B}$ can make them arbitrarily hard to guess. We note that $\mathcal{B}$

can determine the index i of any given string $\xi_{1,i}$ in L_1 (resp. $\xi_{2,i}$ in L_2, or $\xi_{T,i}$ in L_T), breaking ties between multiple matches arbitrarily.

$\mathcal{B}$ starts the game by providing $\mathcal{A}$ with the $q+2$ strings $\xi_{1,0}, \xi_{2,0}, \ldots, \xi_{2,q}$. Queries go as follows.

Group action: Given a multiply/divide selection bit and two operands $\xi_{1,i}$, $\xi_{1,j}$ with $0 \le i, j < \tau_1$, we compute $F_{1,\tau_1} \leftarrow F_{1,i} \pm F_{1,j} \in \mathbb{Z}_p[x]$ depending on whether a multiplication or a division is requested. If $F_{1,\tau_1} = F_{1,l}$ for some $l < \tau_1$, we set $\xi_{1,\tau_1} \leftarrow \xi_{1,l}$; otherwise, we set ξ_{1,τ_1} to a string in $\{0,1\}^*$ distinct from $\xi_{1,0}, \ldots, \xi_{1,\tau_1-1}$. We add $(F_{1,\tau_1}, \xi_{1,\tau_1})$ to L_1 and give ξ_{1,τ_1} to $\mathcal{A}$, then increment τ_1 by one. Group action queries in $\mathbb{G}_2$ and $\mathbb{G}_T$ are treated similarly.

Isomorphism: Given a string $\xi_{2,i}$ with $0 \le i < \tau_2$, we let $F_{1,\tau_1} \leftarrow F_{2,i} \in \mathbb{Z}_p[x]$. If $F_{1,\tau_1} = F_{1,l}$ for some $l < \tau_1$, we set $\xi_{1,\tau_1} \leftarrow \xi_{1,l}$; otherwise, we set ξ_{1,τ_1} to a string in $\{0,1\}^* \setminus \{\xi_{1,0}, \ldots, \xi_{1,\tau_1-1}\}$. We add $(F_{1,\tau_1}, \xi_{1,\tau_1})$ to L_1, give ξ_{1,τ_1} to $\mathcal{A}$, and increment τ_1 by one.

Pairing: Given two operands $\xi_{1,i}$ and $\xi_{2,j}$ with $0 \le i < \tau_1$ and $0 \le j < \tau_2$, we compute the product $F_{T,\tau_T} \leftarrow F_{1,i} F_{2,j} \in \mathbb{Z}_p[x]$. If $F_{T,\tau_T} = F_{T,l}$ for some $l < \tau_T$, we set $\xi_{T,\tau_T} \leftarrow \xi_{T,l}$; otherwise, we set ξ_{T,τ_T} to a string in $\{0,1\}^* \setminus \{\xi_{T,0}, \ldots, \xi_{T,\tau_T-1}\}$. We add $(F_{T,\tau_T}, \xi_{T,\tau_T})$ to L_T, give ξ_{T,τ_T} to $\mathcal{A}$, and increment τ_T by one.

$\mathcal{A}$ terminates and returns a pair $(c, \xi_{2,\ell})$ where $0 \le \ell < \tau_2$. At this point $\mathcal{B}$ chooses a random $x^* \in \mathbb{Z}_p$. The simulation provided by $\mathcal{B}$ is perfect unless the choice of x creates an equality relation between the simulated group elements that was not revealed to $\mathcal{A}$. Thus, the success probability of $\mathcal{A}$ is bounded by the probability that any of the following holds:

1. $F_{1,i}(x^*) - F_{1,j}(x^*) = 0$ for some i, j such that $F_{1,i} \ne F_{1,j}$,
2. $F_{2,i}(x^*) - F_{2,j}(x^*) = 0$ for some i, j such that $F_{2,i} \ne F_{2,j}$,
3. $F_{T,i}(x^*) - F_{T,j}(x^*) = 0$ for some i, j such that $F_{T,i} \ne F_{T,j}$,
4. $(x^* + c)\, F_{2,\ell}(x^*) = 0$.

Since $F_{1,i} - F_{1,j}$ for fixed i and j is a polynomial of degree at most q, it vanishes at a random $x^* \in \mathbb{Z}_p$ with probability at most q/p. Similarly, for fixed i and j, the second case occurs with probability $\le q/p$, the third with probability $\le 2q/p$ (since $F_{T,i} - F_{T,j}$ has degree at most $2q$), and the fourth with probability $\le (q+1)/p$. By summing over all valid pairs (i,j) in each case, we find that $\mathcal{A}$ wins the game with probability $\epsilon \le \binom{\tau_1}{2}\frac{q}{p} + \binom{\tau_2}{2}\frac{q}{p} + \binom{\tau_T}{2}\frac{2q}{p} + \frac{q+1}{p}$. Since $\tau_1 + \tau_2 + \tau_T \le q_G + q + 2$, the required bound follows: $\epsilon \le (q_G + q + 2)^2 (q/p) = O((q_G)^2 (q/p) + q^3/p)$. $\square$

Corollary 1. *Any adversary that solves the q-SDH problem with constant probability $\epsilon > 0$ in generic groups of order p such that $q < o(\sqrt[3]{p})$ requires $\Omega(\sqrt{\epsilon p/q})$ generic group operations.*

6 Conclusions

We presented a number of short signature schemes based on the q-SDH assumption. Our main result is a short signature which is fully secure without using the random oracle model. The signature is as short as DSA signatures, but is provably secure in the standard model. We also showed that the scheme supports limited message recovery, for even greater compactness.

These constructions are possible thanks to properties of the q-SDH assumption. The assumption can be viewed as a discrete logarithm analogue of the Strong RSA assumption. We believe the q-SDH assumption is a useful tool for constructing cryptographic systems and we expect to see many other schemes based on it. For example, we mention a new group signature scheme of Boneh et al. [BBS04].

References

[ADR02] J.H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In *Proceedings of Eurocrypt 2002*, volume 2332 of *LNCS*. Springer-Verlag, 2002.

[BB04a] Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity based encryption without random oracles. In *Proceedings of Eurocrypt '04*, 2004.

[BB04b] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In *Proceedings of Eurocrypt 2004*, 2004. Full version at: `http://crypto.stanford.edu/~dabo/abstracts/sigssdh.html`.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures using strong Diffie-Hellman. Manuscript, 2004.

[BLS01] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In *Proceedings of Asiacrypt 2001*, volume 2248 of *LNCS*, pages 514–32. Springer-Verlag, 2001.

[CDF03] Nicolas Courtois, Magnus Daum, and Patrick Felke. On the security of HFE, HFEv- and Quartz. In *Proceedings of PKC 2003*, volume 2567 of *LNCS*, pages 337–50. Springer-Verlag, 2003.

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In *Proceeings of Eurocrypt 2004*, LNCS. Springer-Verlag, 2004. `http://eprint.iacr.org/2003/182/`.

[CN00] J. Coron and D. Naccache. Security analysis of the Gennaro-Halevi-Rabin signature scheme. In *Proceedings of Eurocrypt 2000*, pages 91–101, 2000.

[Cor00] Jean-Sébastien Coron. On the exact security of full domain hash. In *Proceedings of Crypto 2000*, volume 1880 of *LNCS*, pages 229–35. Springer-Verlag, 2000.

[CS00] Ronald Cramer and Victor Shoup. Signature schemes based on the strong RSA assumption. *ACM TISSEC*, 3(3):161–185, 2000. Extended abstract in Proc. 6th ACM CCS, 1999.

[GHR99] Rosario Gennaro, Shai Halevi, and Tal Rabin. Secure hash-and-sign signatures without the random oracle. In *Proceedings of Eurocrypt 1999*, LNCS, pages 123–139. Springer-Verlag, 1999.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ron Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Computing*, 17(2):281–308, 1988.

[JN01] Antoine Joux and Kim Nguyen. Separating decision Diffie-Hellman from Diffie-Hellman in cryptographic groups. Cryptology ePrint Archive, Report 2001/003, 2001. `http://eprint.iacr.org/2001/003/`.

[KR00] Hugo Krawczyk and Tal Rabin. Chameleon signatures. In *Proceedings of NDSS 2000*. Internet Society, 2000. `http://eprint.iacr.org/1998/010/`.

[KW03] Jonathan Katz and Nan Wang. Efficiency improvements for signature schemes with tight security reductions. In *Proceedings of ACM CCS*, 2003.

[MSK02] Shigeo Mitsunari, Ryuichi Sakai, and Masao Kasahara. A new traitor tracing. *IEICE Trans. Fundamentals*, E85-A(2):481–484, 2002.

[MVV97] Alfred J. Menezes, Paul C. Van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

[NS00] David Naccache and Jacques Stern. Signing on a postcard. In *Proceedings of Financial Cryptography 2000*, 2000.

[PCG01] Jacques Patarin, Nicolas Courtois, and Louis Goubin. QUARTZ, 128-bit long digital signatures. In *Proceedings of RSA 2001*, volume 2020 of *LNCS*, pages 282–97. Springer-Verlag, 2001.

[PV00] Leon Pintsov and Scott Vanstone. Postal revenue collection in the digital age. In *Proceedings of Financial Cryptography 2000*, 2000.

[Sah99] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *Proceeings 40 IEEE Symp. on Foundations of Computer Science*, 1999.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In *Proceedings of Eurocrypt 1997*. Springer-Verlag, 1997.

[TSNZ03] V. To, R. Safavi-Naini, and F. Zhang. New traitor tracing schemes using bilinear map. In *Proceedings of 2003 DRM Workshop*, 2003.

[ZSNS04] Fangguo Zhang, Reihaneh Safavi-Naini, and Willy Susilo. An efficient signature scheme from bilinear pairings and its applications. In *Proceedings of PKC 2004*, 2004.