Provably Secure Non-Interactive Key Distribution Based on Pairings

Régis Dupont* and Andreas Enge*†

Abstract

We define a security notion for non-interactive key distribution protocols. We identify an apparently hard computational problem related to pairings, the Generalised Bilinear Diffie–Hellman problem (GBDH). After extending the pairing based protocol of Sakai–Ohgishi–Kasahara to a slightly more general setting, we show that breaking the system is polynomially equivalent to solving GBDH in the random oracle model and thus establish a security proof.

Keywords: key distribution, non-interactive, identity based cryptography, pairings, security proof, random oracle model.

1 Introduction

A non-interactive key distribution protocol is a way to create a shared secret between two parties, henceforth called "Alice" and "Bob" as usual to avoid confusion. While interactive protocols like the classical Diffie–Hellman key exchange require some communication between Alice and Bob to establish the common secret, this is not the case for non-interactive systems, hence the name.

Without further communication, the only information Alice and Bob have on each other are their respective identities, so that non-interactive cryptography is necessarily identity based, a concept introduced by Shamir in [21]. In such a system, Alice derives the shared secret from her private key and Bob's identity, which can be seen as his public key, and Bob does likewise. Public keys being fixed by the participants' identities, Alice is clearly unable to determine her private key by herself; otherwise, Bob would be able to deduce Alice's private key as well, since he possesses the very same information on Alice's identity as herself. Thus, the help of a trusted third party is needed, the *Private Key Generator (PKG)*, who possesses additional privileged information in the form of a master-key. The role of the PKG is precisely to derive private keys from public identities using the master-key and to issue these private keys to their legitimate holders. Hence, another way of seeing the information flow in a non-interactive system is that the synchronous communication between Alice and Bob is replaced by asynchronous communication with the PKG.

*Laboratoire d'Informatique (CNRS/FRE 2653), École polytechnique, 91128 Palaiseau Cedex, France, {dupont, enge}@lix.polytechnique.fr

†Action TANC, INRIA Futurs, Domaine de Voluceau-Rocquencourt, B.P. 105, 78153 Le Chesnay Cedex, France

In [21], Shamir proposes only an identity based signature scheme, leaving open among others the problem of key distribution. Maurer and Yacobi in [16] suggest the first non-interactive key distribution scheme, based on discrete logarithms in $(\mathbb{Z}/n\mathbb{Z})^\times$ with composite n. However, some version of the protocol is soon shown to be insecure [15]. Even with the improvements of [17] it can be broken by two colluding participants, who with a high probability can retrieve the PKG's secret information, that is the factorisation of n [14]. In the unbroken version, the modulus m is chosen as the product of two primes p such that the maximal prime factor q of $p-1$ is of medium size. To determine a private key, the PKG computes discrete logarithms modulo the prime factors of the $p-1$, which by Pollard's ρ algorithm can be done with a complexity of $O(\sqrt{q})$. An attacker may also profit from the special structure of the primes and factor n by Pollard's $p-1$-method in time essentially $O(q)$. The relatively small difference between the complexities for creating a key and for breaking the system induces an impractically high computational load on the PKG (cf. [15]).

An alternative protocol, suggested by Hühnlein, Jacobson and Weber in [11], uses non-maximal imaginary quadratic orders. The PKG has to solve discrete logarithm problems in the class group of an imaginary quadratic field and in a finite field, and the fastest algorithm for the class group step known to date has a subexponential complexity with exponent $1/2$. A potential attacker is assumed to have to factor the discriminant, which can also be done in subexponential time with exponent $1/2$ by the elliptic curve method. Hence, this scheme also requires that the PKG disposes of an enormous computing power, and the margin between instances not manageable by the PKG and instances vulnerable by attacks is very small. Furthermore, it is uncertain how well a choice of parameters falling into today's small margin of security will resist the exponential growth of computing power predicted by Moore's law.

In his diploma thesis [13], Kügler develops a key distribution system based on the discrete logarithm problem in $(\mathbb{Z}/n\mathbb{Z})^\times$ for composite n, in which the PKG can compute private keys in polynomial time.

None of the above protocols come with a formal proof of security.

The Weil and Tate pairings on elliptic curves have originally been introduced into cryptology to break certain elliptic curve cryptosystems [18, 7]. Recently, it was shown in [12, 20] that these pairings also present a constructive facet, namely that they can be used for establishing a tripartite Diffie–Hellman or a non-interactive key agreement protocol. Again, the protocols come without a formal security proof. Numerous applications have since then emerged, ranging from identity based encryption [3] over interactive key agreement protocols [22, 1] to short [4] or identity based signatures [5, 9].

In this article, we extend the non-interactive identity based key distribution protocol of [20] to the setting of a very general pairing, whose properties are reviewed in Section 2. The protocol itself is described in Section 3. This generalisation is needed, for instance, to implement the protocol using the Weil pairing of ordinary elliptic curves, and it sheds new light on the precise prerequisites for setting up such a pairing based system. We identify an apparently hard problem, the *Generalised Bilinear Diffie–Hellman Problem (GBDH)*, a natural generalisation of the BDH introduced in the long, on-line version of [3]. We proceed by defining a notion of security in Section 4, and we prove that in the random oracle model, breaking the protocol is polynomially equivalent to solving the GBDH problem, see Section 5. In particular, assuming that the GBDH problem is hard, the protocol is secure. Concrete implementations are obtained, for instance, from the Tate or Weil pairings on algebraic curves. In this setting, the PKG can compute private keys in polynomial time by a scalar multiplication on the curve. The effort for an adversary to solve the GBDH problem, however, even

when using the fastest algorithm known to date, is at least subexponential.

2 Pairings and the GBDH problem

In the remaining sections, we let $(G,+)$, $(\hat{G},+)$ and $(V,\times)$ denote groups of prime order ℓ. The sets of their non-neutral elements are denoted by G^*, $\hat{G}^*$ and V^*, respectively. We suppose that $e : G \times \hat{G} \to V$ is a pairing satisfying the following properties:

- **Bilinearity:** $e(aP, bQ) = e(P,Q)^{ab}$ for all $P \in G$, $Q \in \hat{G}$, a, $b \in \mathbb{Z}$.
- **Non-degeneracy:** there are $P \in G$ and $Q \in \hat{G}$ such that $e(P,Q) \neq 1$. In our setting of prime order groups this is equivalent to $e(P,Q) \neq 1$ for all $P \in G^*$, $Q \in \hat{G}^*$.
- **Computability:** given $P \in G$ and $Q \in \hat{G}$, the value $e(P,Q)$ can be efficiently computed.

For instance, the Tate and Weil pairings on elliptic curves have these properties. If E is an elliptic curve defined over a finite field $\mathbb{F}_q$ and ℓ is some prime number (for efficiency reasons taken to be dividing the cardinality of E), then the Tate and Weil pairings can be defined as pairings from the ℓ-torsion points $E[\ell]$ into some field extension $\mathbb{F}_{q^k}^\times$. For k not too large, they can be computed efficiently. Choosing supersingular curves as originally proposed, one always has $k \leq 6$, but it is unknown whether the use of these curves with their special and very rich algebraic structures might lead to security problems. In [20, 19], it is shown how to obtain ordinary curves with certain small values of k. Recently, constructions for ordinary curves with arbitrary values of k have been given [2, 6].

As $E[\ell]$ is of order ℓ^2, and more precisely of type $\mathbb{Z}/\ell\mathbb{Z} \times \mathbb{Z}/\ell\mathbb{Z}$, one has to choose subgroups of order ℓ for G and $\hat{G}$. With the Tate pairing, it is hereby often possible to take $G = \hat{G}$, while the antisymmetry of the Weil pairing in principle forces G and $\hat{G}$ to be distinct. For supersingular curves, one may sometimes define a modified Weil pairing on a single subgroup of order ℓ, see [3, 4, 9]. For ordinary curves, this is not possible, so that our generalisation to distinct G and $\hat{G}$ becomes necessary.

It turns out that the security of the key exchange protocol to be defined in Section 3 relies on the hardness of the following problem, baptised the **Generalised Bilinear Diffie–Hellman Problem (GBDH)**: given (P, Q, aP, bQ, cP, cQ), compute $e(P,Q)^{abc}$. This is the same problem as the Bilinear Diffie–Hellman Problem introduced in the extended on-line version of [3], except that we allow the groups G and $\hat{G}$ to be different. A probabilistic algorithm $\mathcal{A}$ is said to (t,ε)-solve GBDH in $(G, \hat{G}, V, e)$ if $\mathcal{A}$ runs in time at most t and correctly solves the problem with probability at least ε, that is,

$$\mathrm{Prob}\left(\mathcal{A}\left(P, Q, aP, bQ, cP, cQ\right) = e(P,Q)^{abc}\right) \geq \varepsilon.$$

The probability is taken over the uniformly and independently distributed $P \in G^*$, $Q \in \hat{G}^*$ and a, b, $c \in \mathbb{F}_\ell^\times$ and over the random choices of $\mathcal{A}$.

3 The non-interactive key distribution protocol

Formalising the protocol of [20] and extending it to the framework of general pairings introduced in the previous section, the key sharing protocol can be naturally divided into

four distinct algorithms: **Setup**, **Master-key generation**, **Private key distribution** and **Common secret computation**.

- **Setup:** choose G, $\hat{G}$, V and e as in Section 2, and let $H : \{0,1\}^* \to G^*$ and $\hat{H} : \{0,1\}^* \to \hat{G}^*$ be cryptographic hash functions. All these parameters are publicly known.
- **Master-key generation:** the PKG chooses a random master-key $s \in \{1, \ldots, \ell - 1\}$.
- **Private key distribution:** whenever a user A first wishes to use the system, he contacts the PKG and asks for his private key pair. Using A's identity ID_A, the PKG computes A's private key pair $(S_A, \hat{S}_A) = (sH(\mathrm{ID}_A), s\hat{H}(\mathrm{ID}_A))$ and sends it to A.
- **Common secret computation:** suppose that users A and B wish to create a common secret key. A computes B's public key

  $$(P_B, Q_B) = (H(\mathrm{ID}_B), \hat{H}(\mathrm{ID}_B))$$

  and conversely B computes

  $$(P_A, Q_A) = (H(\mathrm{ID}_A), \hat{H}(\mathrm{ID}_A)).$$

  Then A can compute

  $$(e(S_A, Q_B), e(P_B, \hat{S}_A)),$$

  and B can compute

  $$(e(P_A, \hat{S}_B), e(S_B, Q_A)).$$

  The bilinearity of e makes it easy to see that the computed tuples are in fact equal and thus constitute a secret shared between A and B.

4 Definition of security for non-interactive key distribution

In the non-interactive cryptographic setting of the previous section, the only observable traffic is the distribution of private keys. It is thus natural to consider the protocol secure if the corruption of an arbitrary number of private keys does not reveal the shared secret between two further participants. In particular, a colluding group of participants who reveal their private keys to one another then does not gain any insight into other people's common secrets. Precisely, an adversary $\mathcal{A}$ is said to (t, ε)-break the protocol if it runs in time at most t and has advantage at least ε in the following game.

- **Setup:** the challenger publishes the general system parameters $(G, \hat{G}, V, \ell, e, H, \hat{H})$.
- **Extraction queries:** $\mathcal{A}$ issues a number of extraction queries $\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_n$ to the challenger, who, upon receiving the query ID_i, computes the tuple $(sH(\mathrm{ID}_i), s\hat{H}(\mathrm{ID}_i))$ and sends it back to $\mathcal{A}$.
- **Guess:** Once $\mathcal{A}$ decides that it has collected enough information, it picks two identities ID_A and ID_B, different from all the ID_i, and publishes a quadruple $(\mathrm{ID}_A, \mathrm{ID}_B, \alpha, \beta)$.

The attacker $\mathcal{A}$'s advantage is defined as:

$$\mathrm{Adv}(\mathcal{A}) = p_{\mathcal{A},1} + p_{\mathcal{A},2}$$

with

$$p_{\mathcal{A},1} = \mathrm{Prob}\left(e(H(\mathrm{ID}_A), \hat{H}(\mathrm{ID}_B))^s = \alpha\right)$$

and

$$p_{\mathcal{A},2} = \mathrm{Prob}\left(e(H(\mathrm{ID}_B), \hat{H}(\mathrm{ID}_A))^s = \beta\right).$$

5 Security proof

In this section, we show that the GBDH problem and the security of the non-interactive key distribution protocol of Section 3 are polynomially equivalent.

Proposition 1 *If the GBDH problem in some setting $(G, \hat{G}, V, \ell, e)$ can be (t, ε)-solved, then the key distribution protocol in the setting $(G, \hat{G}, V, \ell, e, H, \hat{H})$ can be $(t+\delta, \varepsilon)$-broken. Hereby, δ is the time needed to carry out one extraction query, to compute two hash values of H and of $\hat{H}$ and to carry out $O(\log \ell)$ group operations in G, $\hat{G}$, V and $\mathbb{F}_\ell^\times$.*

Proof: By issuing one key extraction query on an arbitrary identity and computing the hash values H and $\hat{H}$ of this identity, an attacker on the protocol obtains two pairs (P, sP) and (Q, sQ) with $P \in G^*$ and $Q \in \hat{G}^*$. After multiplying these with two uniformly and independently chosen integers from $\{1, \ldots, \ell - 1\}$, we may hereby assume that P and Q are uniformly and independently distributed over G^* resp. $\hat{G}^*$. Multiplying only the right hand sides with a random element $\gamma \in \{1, \ldots, \ell - 1\}$ replaces them by cP and cQ with $c = \gamma s$ uniformly distributed. The attacker then randomly selects two identities ID_A and ID_B and two elements $\alpha, \beta \in \{1, \ldots, \ell - 1\}$ and computes $R = \alpha H(\mathrm{ID}_A)$ and $S = \beta \hat{H}(\mathrm{ID}_B)$. Hereby, $R = aP$ and $S = bQ$ for some (unknown) a and b.

In the GBDH instance (P, Q, R, S, cP, cQ), the random variables P, Q, a, b and c are now uniformly and independently distributed. Solving the GBDH problem provides the attacker with $v = e(H(\mathrm{ID}_A), \hat{H}(\mathrm{ID}_B))^{\alpha\beta\gamma s}$. He then computes $r = (\alpha\beta\gamma)^{-1}$ in $\mathbb{F}_\ell^\times$ and raises v to the power r, which yields the shared secret between A and B. □

Theorem 2 *Let the hash functions H and $\hat{H}$ be given by random oracles. Suppose that there is some adversary $\mathcal{A}$ who (t, ε)-breaks the protocol with parameters $(G, \hat{G}, V, \ell, e, H, \hat{H})$. Assume furthermore that an upper bound q_E on the number of extraction queries issued by $\mathcal{A}$ is known. Then there is an algorithm $\mathcal{B}$ that $(t', \varepsilon/(2\ \exp(1)^2\ (1 + q_E)^2))$-solves the GBDH problem for $(G, \hat{G}, V, \ell, e)$,*

$$t' = Kt(t_1 + t_2 + \log q_E) + t_3,$$

K is a small constant and

t_1	*is the time needed to carry out a scalar multiplication in G or $\hat{G}$ or an exponentiation in V*
t_2	*is the time needed to generate a random bit*
$\log(q_E)$	*is the time needed to locate an entry in an ordered list with at most q_E entries*
t_3	*is the time required to invert an element of $\mathbb{F}_\ell^\times$.*

Notice that in general, t' will be t times some polynomial in $\log \ell$, and $\log \ell \leq t$ since $\mathcal{A}$'s output is an element of V, so that in fact t' is polynomial in t. The assumption that an upper bound q_E on the number of extraction queries of $\mathcal{A}$ or, *a forteriori*, on its running time $t \geq q_E$ be known by $\mathcal{B}$, certainly shows limitations of the theorem. However, it seems to be commonly adopted in the literature, cf. [3, 4].

Proof: $\mathcal{B}$ has as input a random and uniformly distributed instance $(P, Q, P_a, Q_b, P_c, Q_c) = (P, Q, aP, bQ, cP, cQ)$ of the GBDH problem. For finding the solution $e(P,Q)^{abc}$ with $\mathcal{A}$'s assistance, $\mathcal{B}$ has control over the hash functions H and $\hat{H}$. Basically, when queried for a hash value of, say, H, it outputs a random group element, obtained as a random multiple of P or P_a. Thus $\mathcal{B}$ conforms to the random oracle model (to $\mathcal{A}$, the hash function appears as a random function) while at the same time keeping track of additional information (the discrete logarithms with respect to the bases P or P_a). Of course, as a is unknown to $\mathcal{B}$, it may control only one of the discrete logarithms. To be able to answer to extraction queries, $\mathcal{B}$ should attach multiples of P to the corresponding identities; to retrieve the solution to the GBDH problem, it should attach a multiple of P_a to the identity for which $\mathcal{A}$ finally emits its guess. These requirements put $\mathcal{B}$ into a dilemma, because $\mathcal{A}$ may request hash values *before* deciding to query the private key or to emit a guess for the corresponding identity. To solve the problem, $\mathcal{B}$ randomly goes for multiples of P or P_a and declares failure whenever it realises that it has made the wrong choice previously. The probabilities of selecting P or P_a must depend on q_E, since otherwise $\mathcal{B}$'s success probability becomes exponentially small for q_E tending to infinity. The more extraction queries $\mathcal{A}$ makes, the more often $\mathcal{B}$ has to return a multiple of P. This is the reason why $\mathcal{B}$ needs to know at least an upper bound on q_E, and furthermore its success probability decreases the more private keys $\mathcal{A}$ extracts. In detail, $\mathcal{B}$ implements the following routines:

H **queries:** $\mathcal{B}$ keeps an initially empty list L of tuples $(X, R, h, u) \in \{0,1\}^* \times G \times [1, \ell - 1] \times \{0,1\}$, sorted according to X. When $\mathcal{A}$ queries for the hash value of some bit string X, $\mathcal{B}$ checks if L contains a tuple (X, R, h, u). If this is not the case, then $\mathcal{B}$

- picks uniformly a random $h \in [1, \ell - 1]$
- picks $u \in \{0,1\}$ with $\text{Prob}(u = 0) = \delta$, where δ is a parameter to be determined later
- if $u = 0$, sets $R = hP$, otherwise sets $R = hP_a$
- appends (X, R, h, u) to L

Finally, it sends R to $\mathcal{A}$.

$\hat{H}$ **queries:** These are handled in the same way, $\mathcal{B}$ keeping a list $\hat{L}$ and returning a multiple of Q with probability δ and a multiple of Q_b with probability $1 - \delta$.

Extraction queries: To answer to a query issued by $\mathcal{A}$ upon the string ID, the algorithm $\mathcal{B}$:

- queries H and $\hat{H}$ as described above to make sure that L contains a tuple of the form (ID, R, h, u) and $\hat{L}$ a tuple of the form $(\text{ID}, S, \hat{h}, \hat{u})$
- checks if $u = 1$ or $\hat{u} = 1$, in which case it reports failure
- computes the tuple $(hP_c, \hat{h}Q_c)$ and sends it to $\mathcal{A}$

Guess: Upon receiving the guess $(\text{ID}_A, \text{ID}_B, \alpha, \beta)$ from $\mathcal{A}$, the algorithm $\mathcal{B}$

- proceeds as in the case of H and $\hat{H}$ queries to make sure that L contains tuples of the form $(\mathrm{ID}_i, R_i, h_i, u_i)$ and $\hat{L}$ tuples of the form $(\mathrm{ID}_i, S_i, \hat{h}_i, \hat{u}_i)$ for $i = A, B$
- uniformly picks a random $t \in \{0, 1\}$
- if $t = 0$, checks if $u_A = 1$ and $\hat{u}_B = 1$ (otherwise reports failure), then outputs $\alpha^{1/(h_A \hat{h}_B)}$ as a guess
- if $t = 1$, checks if $u_B = 1$ and $\hat{u}_A = 1$ (otherwise reports failure), then outputs $\beta^{1/(h_B \hat{h}_A)}$ as a guess

Now, suppose that $\mathcal{B}$ does not abort and let γ be its output. With probability $1/2$, we have $t = 0$, whence $u_A = 1$, $\hat{u}_B = 1$, $H(\mathrm{ID}_A) = h_A P_a = a\, h_A\, P$, $\hat{H}(\mathrm{ID}_B) = \hat{h}_B Q_b = b\, \hat{h}_B\, Q$ and $\gamma = \alpha^{(h_A \hat{h}_B)^{-1}}$, where the inverse is taken in $\mathbb{F}_\ell^\times$. Independently, with probability $p_{\mathcal{A},1}$, we have $\alpha = e(H(\mathrm{ID}_A), \hat{H}(\mathrm{ID}_B))^c$. Thus, the following event happens with overall probability $p_{\mathcal{A},1}/2$:

$$\begin{aligned} \gamma &= \alpha^{(h_A \hat{h}_B)^{-1}} = e\left(H(\mathrm{ID}_A), \hat{H}(\mathrm{ID}_B)\right)^{c \cdot (h_A \hat{h}_B)^{-1}} \\ &= e\left(a\, h_A\, P, b\, \hat{h}_B\, Q\right)^{c \cdot (h_A \hat{h}_B)^{-1}} = e(P, Q)^{abc}, \end{aligned}$$

where the last equality follows from the bilinearity of the pairing.

A similar analysis for $t = 1$ shows that $\mathcal{B}$ guesses correctly with an additional probability of $p_{\mathcal{A},2}/2$. Since these two events are disjoint, $\mathcal{B}$'s guess is correct with a total probability of $(p_{\mathcal{A},1} + p_{\mathcal{A},2})/2 \geq \varepsilon/2$ whenever it does not abort.

We now compute the probability for $\mathcal{B}$ to abort. Let q_E be the number of extraction queries issued by $\mathcal{A}$. Then the probability of non-abortion during each extraction query being δ^2 and the probability of non-abortion during the guess phase being $(1 - \delta)^2$, the overall probability of non-abortion is at least (as q_E has been taken to be an upper bound on the actual number of extraction queries) $\delta^{2q_E}(1 - \delta)^2$. Minimising this function, we find the optimal value $\delta = q_E/(1 + q_E)$ and an overall probability of non-abortion of at least $1/(\exp(1)(1 + q_E))^2$. Hence, the probability that $\mathcal{B}$ outputs the correct solution to the GBDH instance is at least $\varepsilon/\left(2 \exp(1)^2 (1 + q_E)^2\right)$.

The running time analysis of $\mathcal{B}$ is straightforward. □

Proposition 1 and Theorem 2 show that the GBDH problem and the key distribution protocol are polynomially equivalent, and describe accurately how the running times and success probabilities are transformed during the reductions. Assuming that the GBDH problem is hard, the security of the protocol is thus established.

It is possible to furthermore formalise the security notion from a complexity theoretic point of view. To do so, it is necessary to introduce infinite families of problem instances. Let thus $\mathcal{F} = \left((G_k, \hat{G}_k, V_k, \ell_k, e_k)\right)_{k \in \mathbb{N}}$ be a family of GBDH parameters as above. We say that $\mathcal{F}$ satisfies the polynomial GBDH assumption if, for any polynomials P and Q in $\mathbb{Z}[X]$, there is no randomised algorithm $\mathcal{A}$ that $(P(k), 1/Q(k))$-solves the GBDH problem for $(G_k, \hat{G}_k, V_k, \ell_k, e_k)$ for all $k \in \mathbb{N}$. The above proof shows that under the random oracle model, if $\mathcal{F}$ satisfies the polynomial GBDH assumption, then the protocol with parameters from $\mathcal{F}$ is secure in the sense that no polynomial time algorithm achieves a polynomial advantage in breaking the protocol.

Similarly, one might admit adversaries with subexponential computing power and define in the same way the subexponential GBDH assumption. Then our security analysis shows that under the subexponential GBDH assumption, no algorithm of subexponential complexity can break the protocol with a subexponential advantage.

6 Conclusion

We have defined a notion of security for non-interactive key distribution protocols. Slightly generalising the pairing based protocol of [20], we have shown that the scheme satisfies this security property in the random oracle model if the GBDH assumption holds for the involved pairing. In particular, the protocol is secure against an arbitrary number of colluding attackers.

Recently, the concept of hierarchical identity based system has been introduced, and such schemes have been proposed [10, 8]. Using similar ideas, it is easy to see that the protocol can also be transformed into a hierarchical system.

Acknowledgements: We thank François Morain for valuable discussions concerning this work. The second author gratefully acknowledges being supported by a fellowship within the postdoctoral programme of the German Academic Exchange Service (DAAD). This research was partially supported by the French Ministry of Research — ACI Cryptologie.

References

[1] S. Al-Riyami and K. Paterson. Authenticated three party key agreement protocols from pairings. Preprint, available at `http://www.isg.rhul.ac.uk/~kp/`, 2002.

[2] P. Barreto, B. Lynn, and M. Scott. Constructing elliptic curves with prescribed embedding degrees. In S. Cimato, C. Galdi, and G. Persiano, editors, *Security in Communication Networks — Third International Conference, SCN 2002, Amalfi, Italy, September 2002*, volume 2576 of *Lect. Notes Comput. Sci.*, pages 263–273 (?), Berlin, 2003. Springer-Verlag. To appear.

[3] D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In J. Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lect. Notes Comput. Sci.*, pages 213–229. Springer-Verlag, 2001.

[4] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In C. Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lect. Notes Comput. Sci.*, pages 514–532. Springer-Verlag, 2001.

[5] J. Cha and J. Cheon. Identity-based signature from the Weil pairing. Preprint, available at `http://vega.icu.ac.kr/~jhcheon/publications.html`, 2001.

[6] Régis Dupont, Andreas Enge, and François Morain. Building curves with arbitrary small MOV degree over finite prime fields. Cryptology ePrint Archive, Report 2002/094, available at `http://eprint.iacr.org/2002/094/`, 2002.

[7] G. Frey and H.-G. Rück. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Math. Comp.*, 62(206):865–874, April 1994.

[8] C. Gentry and A. Silverberg. Hierarchical ID-based cryptography. In Yuliang Zheng, editor, *Advances in Cryptology — ASIACRYPT 2002*, volume 2501 of *Lect. Notes Comput. Sci.*, pages 548–566, Berlin, 2002. Springer-Verlag.

[9] F. Hess. Efficient identity based signature schemes based on pairings. In K. Nyberg and H. Heys, editors, *Selected Areas in Cryptography — 9th Annual International Workshop, SAC 2002, St. Johns, Canada, August 2002*, volume 2595 of *Lect. Notes Comput. Sci.*, Berlin, 2003. Springer-Verlag. To appear.

[10] J. Horwitz and B. Lynn. Toward hierarchical identity-based encryption. In L. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lect. Notes Comput. Sci.*, pages 466–481. Springer-Verlag, 2002.

[11] D. Hühnlein, M. J. Jacobson Jr., and D. Weber. Towards practical non-interactive public-key cryptosystems using non-maximal imaginary quadratic orders. In D. R. Stinson and S. Tavares, editors, *Selected Areas in Cryptography 2000*, volume 2012 of *Lect. Notes Comput. Sci.*, pages 275–287. Springer-Verlag, 2000.

[12] A. Joux. A one round protocol for tripartite Diffie-Hellman. In W. Bosma, editor, *Algorithmic Number Theory — ANTS-IV*, volume 1838 of *Lect. Notes Comput. Sci.*, pages 358–394. Springer-Verlag, 2000.

[13] D. Kügler. Eine Aufwandsanalyse für identitätsbasierte Kryptosyteme. Diplomarbeit, Technische Universität Darmstadt, Deutschland, 1998. Available at `ftp://ftp.informatik.tu-darmstadt.de/pub/TI/reports/-kuegler.IDCS.diplom.ps.gz`.

[14] D. Kügler and M. Maurer. A note on the weakness of the Maurer–Yacobi squaring method. Technical Report TI-15/99, Fachbereich Informatik, Technische Universität Darmstadt, 1999. Available at `ftp://ftp.informatik.tu-darmstadt.de/pub/TI/TR/-TI-99-15.weaksquaring.ps.gz`.

[15] P. J. Lee and C. H. Lim. Modified Maurer-Yacobi's scheme and its applications. In J. Seberry and Y. Zheng, editors, *Advances in Cryptology – AUSCRYPT'92*, volume 718 of *Lect. Notes Comput. Sci.*, pages 308–323, 1992.

[16] U. Maurer and Y. Yacobi. Non-interactive public-key cryptography. In D. Davies, editor, *Advances in Cryptology – EUROCRYPT '91*, volume 547 of *Lect. Notes Comput. Sci.*, pages 498–507. Springer–Verlag, 1992.

[17] U. Maurer and Y. Yacobi. A non-interactive public-key distribution system. *Des. Codes Cryptogr.*, 9(3):305–316, 1996.

[18] A. Menezes, T. Okamoto, and S. A. Vanstone. Reducing elliptic curves logarithms to logarithms in a finite field. *IEEE Trans. Inform. Theory*, IT–39(5):1639–1646, 1993.

[19] A. Miyaji, M. Nakabayashi, and S. Takano. New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Trans. Fundamentals*, E84-A(5):1234–1243, 2001.

[20] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing, 2000. SCIS 2000, The 2000 Symposium on Cryptography and Information Security, Okinawa, Japan, January 26–28.

[21] A. Shamir. Identity-based cryptosystems and signature schemes. In G. Goos and J. Hartmanis, editors, *Advances in Cryptology – CRYPTO'84*, volume 196 of *Lect. Notes Comput. Sci.*, pages 47–53. Springer-Verlag, 1985.

[22] N. Smart. An identity based authenticated key agreement protocol based on the Weil pairing. *Electronics Letters*, 38:630–632, 2002.