Trading Time for Space: Towards an Efficient IBE Scheme with Short(er) Public Parameters in the Standard Model

Sanjit Chatterjee and Palash Sarkar

Applied Statistics Unit,
Indian Statistical Institute,
203, B.T. Road, Kolkata
700108, India
{sanjit_t, palash}@isical.ac.in

Abstract. At Eurocrypt 2005, Brent Waters proposed an efficient Identity Based Encryption scheme which is secure in the standard model. One drawback of this scheme is that the number of elements in the public parameter is rather large. Here we propose a generalisation of Waters scheme. In particular, we show that there is an interesting trade-off between the tightness of the security reduction and smallness of the public parameter. For a given security level, this implies that if one reduces the number of elements in public parameter then there is a corresponding increase in the computational cost due to the increase in group size. This introduces a flexibility in choosing the public parameter size without compromising in security. In concrete terms, to achieve 80-bit security for 160-bit identities we show that compared to Waters protocol the public parameter size can be reduced by almost 90% while increasing the computation cost by 30%. Our construction is proven secure in the standard model without random oracles. Additionally, we show that CCA security can also be achieved through the reduction to oracle decision bilinear Diffie-Hellman problem (OBDH).

Keywords: identity based encryption, standard model, security, parameter size.

1 Introduction

The area of public key cryptography called Identity Based Encryption (IBE) has witnessed a rapid progress in recent times. Initially proposed by Shamir [23], it was as well a challenge to the crypto community to come out with a practical IBE scheme. Boneh and Franklin [6, 7] were first to define a security model for IBE and gave an implementable solution based on the Bilinear Diffie-Hellman (BDH) problem. There is another construction due to Cocks [13] that uses quadratic residues modulo a composite. The security of these encryption schemes were proved in the random oracle model [11], i.e., the security of these schemes requires cryptographic hash functions that are modelled as random oracles. However,

D. Won and S. Kim (Eds.): ICISC 2005, LNCS 3935, pp. 424–440, 2006.

such hash functions do not exist in reality. Consequently, there were several works such as [14, 2] to construct IBE schemes secure without the random oracle model. They used a weaker notion of security called *selective*-ID model in which an adversary has to commit in advance which identity it wants to attack.

Finally, Boneh and Boyen came out with a scheme for IBE [3] that is secure in the standard model without random oracles. Their work was more of a feasibility study. It solved the open problem but was not practical to be implemented. This work was soon supplemented by that of Waters [24]. Using a method from [2] and introducing a new trick, it provided an improved IBE scheme that is secure in the standard model without random oracle.

However, one disadvantage of the scheme in [24] is the requirement of a rather large public parameter file. If identities are represented by a bit string of length n, then the scheme requires a vector of length n to be maintained as part of public parameter, where each element of the vector is a point on a suitable elliptic curve group.

OUR CONTRIBUTION: We provide a generalisation of the identity based encryption scheme of Waters [24]. This generalisation shows that if one tries to reduce the number of elements in the public parameter then there is a corresponding degradation in the security reduction. In other words, a trade-off is involved in the tightness of security reduction and smallness of public parameter. The trade-off between tightness and smallness can be converted to a trade-off between group size and smallness of public parameter. When desiring a specific security level, the loss of security due to loss of tightness in the security reduction can be compensated by working in a larger group. This increses the bit length of representation of the elements in the public parameter but the number of elements in the public parameters decreases so drastically that there is a significant reduction in the overall size of the public parameter. The increse in group size in turn affects the efficiency of the protocol. Thus, the trade-off is actually between the space required to store the parameters and the time required to execute the protocol. For example, if identities are represented by 160-bit strings, then Waters protocol require to store 160 extra elements (EC points) as part of the public parameter. Alternatively, using our generalisation if one wants to store 16 elements, then to achieve 80-bit security, compared to Waters protocol the space requirement reduces by around 90% while the computation cost increases by around 30%.

– Like Waters, applying Naor's technique, our scheme can also be easily converted to a signature scheme where the underlying security assumption is the computational Diffie-Hellman problem.
– Our construction resembles closely the construction of Waters [24] and security against the chosen ciphertext attack (i.e., the CCA security) of the former follows from that of the later by constructing a 2 level hierarchical identity based encryption scheme (HIBE) and applying the technique of [15]. As an alternative, we show that CCA security can also be achieved by assuming the hardness of the oracle bilinear decision Diffie-Hellman assumption (OBDH).

2 Waters Construction

Waters has recently proposed an efficient identity based encryption scheme without random oracle [24]. We first briefly describe his construction. The relevant definitions of bilinear map, IBE protocol, its security model and hardness assumption are given in Appendix A

Waters IBE: Let $G_1 = \langle P \rangle$, G_2 and $e()$ be as defined in Section A.1. Here, identities are represented as bitstrings of length n.

Setup: Randomly choose a secret $x \in Z_p$. Set $P_1 = xP$, then choose $P_2 \in G_1$ at random. Further, choose a random element $U' \in G_1$ and a random n-length vector $\overrightarrow{U} = \{U_1, \ldots, U_n\}$, whose elements are from G_1. The master secret is xP_2 whereas the public parameters are $\langle P, P_1, P_2, U', \overrightarrow{U} \rangle$. Also $e()$ is publicly known.

Key Generation: Let $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_n) \in \{0,1\}^n$ be any identity. A secret key for v is generated as follows. Choose a random $r \in Z_p^*$, then the private key for v is

$$D_{\mathsf{v}} = (xP_2 + rV, rP).$$

where

$$V = U' + \sum_{\{i:\mathsf{v}_i=1\}} U_i.$$

Encryption: Any message $M \in G_2$ is encrypted for an identity v as

$$C = (e(P_1, P_2)^t M, tP, tV),$$

where t is a random element of Z_p and V is as defined in key generation algorithm.

Decryption: Let $C = (C_1, C_2, C_3)$ be a ciphertext and v be the corresponding identity. Then we decrypt C using secret key $D_{\mathsf{v}} = (D_1, D_2)$ by computing $C_1 e(D_2, C_3)/e(D_1, C_2)$.

3 Our Generalisation

Here we describe our generalisation of Waters scheme. The groups $G_1 = \langle P \rangle$, G_2 and the map $e()$ are as already defined in Section A.1. In the following, we assume the message space $\mathcal{M}$ is G_2, the cipher space $\mathcal{C}$ is $G_2 \times G_1 \times G_1$.

Note that, in Waters scheme identities are represented as n-bit strings. Because of this representation, Waters requires to store n elements of G_1 i.e., $\overrightarrow{U}$ in the public parameter. Depending upon the choice of representation of the identities we can change the size of the public parameter.

Let $N = 2^n$, then we can consider the identities as elements of Z_N and one extreme case would be to consider the identities just as elements of Z_N. A more moderate approach, however, is to fix *a-priori* a size parameter ℓ, where

$1 < \ell \leq n$. In this case, an identity v is represented as $\mathsf{v} = (\mathsf{v}_1, \mathsf{v}_2, \ldots, \mathsf{v}_\ell)$, where $\mathsf{v}_i \in Z_{N^{1/\ell}}$ i.e., each v_i is an n/ℓ bit string. (If identities are considered to be bit strings of arbitrary length, then as in Waters protocol we hash them into Z_N using a collision resistant hash function.)

In this case the protocol is changed to the following, which we call $\mathsf{IBE\text{-}SPP}(\ell)$.

$\mathsf{IBE\text{-}SPP}(\ell)$ **with** $1 < \ell \leq n$

Setup: Randomly choose a secret $x \in Z_p$. Set $P_1 = xP$, then choose $P_2 \in G_1$ at random. Further, choose random elements $U', U_1, U_2, \ldots, U_\ell \in G_1$. The master secret is xP_2 whereas the public parameters are $\langle P, P_1, P_2, U', U_1, U_2, \ldots, U_\ell \rangle$. Also $e()$ is publicly known.

Key Generation: Let v be any identity, a secret key for v is generated as follows. Choose a random $r \in Z_p^*$, then the private key for v is

$$D_{\mathsf{v}} = (xP_2 + rV, rP).$$

where $V = U' + \sum_{i=1}^{\ell} \mathsf{v}_i U_i$.

Encryption, Decryption: As in Waters IBE with the modified definition of V.

Note that, for $\ell = n$ this is exactly Waters protocol. For $\ell = 1$, some minor modifications in the above scheme give a protocol where the additional requirement in the public parameter is just a single element of G_1 as described below.

$\mathsf{IBE\text{-}SPP}$*(1)*

Setup: Randomly choose a secret $x \in Z_N$. Set $P_1 = xP$, then choose $P_2 \in G_1$ at random. Further, choose a random element $U' \in G_1$. The master secret is xP_2 whereas the public parameters are $\langle P, P_1, P_2, U' \rangle$. Also $e()$ is publicly known.

Key Generation: Let v be any identity. A secret key for v is generated as follows. Choose a random $r \in Z_p^*$, then the private key for v is

$$D_{\mathsf{v}} = (xP_2 + rV, rP).$$

where $V = U' + \mathsf{v}P_2$.

Here also the Encryption and Decryption algorithms remain unaltered and this is essentially the Boneh-Boyen scheme of [2] in the *adaptive*-ID model.

Efficiency: Consider $\mathsf{IBE\text{-}SPP}(\ell)$ with $1 < \ell \leq n$. Let $\mathsf{cost}(V)$ be the cost of computing V. The cost of key generation is two scalar multiplications over G_1 plus $\mathsf{cost}(V)$. By including $e(P_1, P_2)$ instead of P_1, P_2 in the public parameter, we can avoid the pairing computation during encryption. So the cost of encryption is one exponentiation over G_2, two scalar multiplications over G_1 plus $\mathsf{cost}(V)$. The cost of decryption is two pairings, one multiplication and one inversion over

G_2. The effect of ℓ is in $\mathsf{cost}(V)$ and affects key generation and encryption costs but does not affect decryption cost.

We first consider the costs of scalar multiplication over G_1 and exponentiation over G_2. As mentioned earlier, G_1 is an elliptic curve group. Let $\mathbb{F}_a$ denote the base field over which G_1 is defined. Then G_2 is a subgroup of $\mathbb{F}_a^k$, where k is the MOV degree. Additions and doublings over G_1 translate into a constant number of multiplications over $\mathbb{F}_a$. The actual number is slightly different for addition and doubling, but we will ignore this difference. Let $|\mathbb{F}_a|$ be the size of the representation of an element of $\mathbb{F}_a$. Assuming the cost of multiplication over G_1 is approximately equal to $|\mathbb{F}_a|^2$, the cost of a scalar multiplication over G_1 is equal to $c_1|\mathbb{F}_a|^3$ for some constant c_1. One can also show that the cost of exponentiation over G_2 is equal to $c_2|\mathbb{F}_a|^3$. Thus, the total cost of scalar multiplication and exponentiation is equal to $c|\mathbb{F}_a|^3$.

The cost of computing V amounts to computing ℓ scalar multiplications where each multiplier is an (n/ℓ)-bit string. On an average, the cost of each such multiplication will be $n/2\ell$ additions and $(n/\ell - 1)$ doublings over G_1. Hence, the total cost of computing V is $n/2$ additions and $(n - \ell)$ doublings over G_1. This cost is equal to $d(3/2 - \ell/n)n|\mathbb{F}_a|^2$ for some constant d.

We consider the cost of encryption. The total cost is

$$c|\mathbb{F}_a|^3 + d(3/2 - \ell/n)n|\mathbb{F}_a|^2 = \left(c + d \times \frac{n}{|\mathbb{F}_a|}\left(\frac{3}{2} - \frac{\ell}{n}\right)\right)|\mathbb{F}_a|^3. \qquad (1)$$

This cost is minimum when $\ell = n$ (as in Waters protocol). The maximum value of the coefficient of $|\mathbb{F}_a|^3$ is $(c + (3nd)/(2|\mathbb{F}_a|))$ whereas the minimum value is $(c + (nd)/(2|\mathbb{F}_a|))$. The value of $|\mathbb{F}_a|$ is usually greater than n and hence the value of $(nd)/(2|\mathbb{F}_a|)$ will be a small constant and hence there is not much effect of ℓ on the total cost of encryption. A similar analysis shows that the effect of ℓ is also not very significant on the cost of key generation. We note, however, that key generation is essentially a one-time offline activity.

3.1 Security Reduction

In this section, we only consider the security of $\mathsf{IBE} - \mathsf{SPP}(\ell)$ against chosen plaintext attacks (IND-ID-CPA). (The extension to chosen ciphertext attack is considered later.) The security (in the sense of IND-ID-CPA) of the identity based encryption scheme (IBE-SPP(ℓ)) developed above can be reduced from the hardness of the DBDH problem as stated in the following theorem.

Theorem 1. *For $t \geq 1$, $q \geq 1$ let $\epsilon = \mathsf{Adv}^{\mathsf{IBE-SPP}}(\ell)(t, q)$. Then,*

$$\epsilon \leq 16q(\mu_\ell + 1)\mathsf{Adv}^{\mathsf{DBDH}}(t + O(\tau q) + \chi),$$

where identities are chosen from Z_N; ℓ is a size parameter with $1 < \ell \leq \lg N$; $\mu_\ell = \ell(N^{1/\ell} - 1)$; $\chi = O(\epsilon^{-2}\ln(\epsilon^{-1})\lambda^{-1}\ln(\lambda^{-1}))$; $\lambda = \frac{1}{4q(\mu_\ell+1)}$; and τ is the time for a scalar multiplication in G_1.

Note that, for $\ell = n$ we have $\mu_n = n$ and one gets the corresponding relationship for Waters protocol. The component χ in the time comes due to the so-called “artificial abort” technique. The proof of Theorem 1 essentially follows the technique already developed by Boneh-Boyen [3] and Waters [24] and we defer it to Section 5.

3.2 Signature

It is an observation of Naor that any identity based encryption scheme can be converted to a signature scheme. Waters in his paper [24] has given a construction of a signature scheme based on his IBE scheme. A similar construction is possible for the generalised scheme $\mathsf{IBE\text{-}SPP}(\ell)$ which we detail here. The sketch of the security reduction is provided in Appendix C.

Let $G_1 = \langle P \rangle,\ G_2$ and $e()$ be as defined in Section A.1. Messages are assumed to be elements of Z_N. Alternatively, if messages are assumed to be bit strings of arbitrary length, then we use a collision resistant hash function to map the messages into Z_N.

Setup: Choose a random x in Z_p. Let $P_1 = xP$. Next, choose random points $P_2, U', U_1, \ldots, U_l$ from G_1. The public key is $\langle P, P_1, P_2, U', U_1, \ldots, U_\ell \rangle$ and the secret key is xP_2.

Signing: Let $M = (m_1, m_2, \ldots, m_\ell)$ is the message to be signed, where each $m_i, 1 \le i \le \ell$ belongs to $Z_{N^{1/\ell}}$. To generate a signature on M, first choose a random $r \in Z_P^*$. Then the signature is

$$\sigma_M = (xP_2 + rV, rP),$$

where $V = U' + \sum_{i=1}^{\ell} m_i U_i$

Verification: Given a message $M = (m_1, m_2, \ldots, m_\ell)$ and a signature $\sigma = (\sigma_1, \sigma_2)$ on M, one accepts σ as a valid siganture on M if

$$e(\sigma_1, P) = e(P_1, P_2)e(\sigma_2, V)$$

where $V = U' + \sum_{i=1}^{\ell} m_i U_i$.

4 Concrete Security

From the security reduction of previous section we observe that any (t, q, ϵ) adversary $\mathcal{A}$ against $\mathsf{IBE\text{-}SPP}(\ell)$ can actually be used to build an algorithm $\mathcal{B}$ to solve the DBDH problem over (G_1, G_2) which runs in time t' and has a probability of success ϵ'. Then $t' = t + O(\tau q) + \chi \approx t + c\tau q + \chi$ for some constant c and $\epsilon' \approx \epsilon/\delta$ where τ is the time for a group operation in G_1 and δ is the corresponding degradation in the security reduction. Resistance of $\mathsf{IBE\text{-}SPP}(\ell)$ against $\mathcal{A}$ can be quantified as $\rho_{|\mathcal{A}}^{(\ell)} = \lg(t/\epsilon)$. To assert that $\mathsf{IBE\text{-}SPP}(\ell)$ has at least 80-bit security, we must have $\rho_{|\mathcal{A}}^{(\ell)} \ge 80$ for all possible $\mathcal{A}$. Similarly, the resistance of DBDH against $\mathcal{B}$ can be quantified as

$$\rho_{|\mathcal{B}} = \lg\left(\frac{t'}{\epsilon'}\right) \approx \lg\left(\delta \times \frac{t + c\tau q + \chi}{\epsilon}\right) = \lg(\delta(A_1 + A_2))$$

where $A_1 = t/\epsilon$ and $A_2 = (c\tau q + \chi)/\epsilon$. We now use $\max(A_1, A_2) \leq A_1 + A_2 \leq 2\max(A_1, A_2)$. Since a factor of two does not significantly affect the analysis we put $\rho_{|\mathcal{B}} = \lg(\delta \times \max(A_1, A_2))$. By our assumption, $A_1 = t/\epsilon \geq 2^{80}$ and hence $\max(A_1, A_2) \geq A_1 \geq 2^{80}$. This results in the condition $\rho_{|\mathcal{B}} \geq 80 + \lg\delta$.

Thus, if we want $\mathsf{IBE\text{-}SPP}(\ell)$ to have 80-bit security, then we must choose the group sizes of G_1, G_2 in such a way that the best possible algorithm for solving DBDH in these groups takes time at least $2^{80+\lg\delta}$. Hence, in particular the currently best known algorithm for solving the DBDH should also take this time. Currently the only method to solve the DBDH problem over (G_1, G_2) is to solve the discrete log problem (DLP) over either G_1 or G_2. The best known algorithm for the former is the Pollard's rho method while that for the later is number/function field sieve. Thus, if we want $\mathsf{IBE\text{-}SPP}(\ell)$ to have 80-bit security, then we must choose the group sizes such that, $2^{80+\lg\delta} \leq \mathsf{min}(t_{G_1}, t_{G_2})$, where t_{G_i} stands for the time to solve DLP in G_i for $i \in \{1, 2\}$.

We have assumed that G_1 is a group of elliptic curve points of order p defined over a finite field $\mathbb{F}_a$ (a is a prime power). Suppose G_2 is a subgroup of order p of the finite field $\mathbb{F}_{a^k}$ where k is the MOV degree. The Pollard's rho algorithm to solve ECDLP takes time $t_{G_1} = O(\sqrt{p})$, while the number/function field seive method to solve the DLP in $\mathbb{F}_{a^k}$ takes time $t_{G_2} = O(e^{c^{1/3}\ln^{1/3} a^k \ln^{2/3}(\ln a^k)})$ where $c = 64/9$ (resp. $32/9$) in large characteristic fields (resp. small characteristic fields).

4.1 Space/Time Trade-Off

In this section we parametrize the quantities by ℓ wherever necessary. Let, $\delta^{(\ell)}$ denote the degradation factor in $\mathsf{IBE\text{-}SPP}(\ell)$. We have already noted in Section 3 that $\ell = n$ stands for Waters protcol. $\delta^{(\ell)}$ and hence $\rho^{(\ell)}$ is minimum when $\ell = n$ and we use this as a bench mark to compare with other values of ℓ. Suppose $\Delta\rho^{(\ell)} = \rho^{(\ell)} - \rho^{(n)} = \lg(\delta^{(\ell)}/\delta^{(n)}) = (n/\ell) - \lg(n/\ell)$. This parameter $\Delta\rho^{(\ell)}$ gives us an estimate of the extra bits required in case of $\mathsf{IBE\text{-}SPP}(\ell)$, to achieve the same security level as that of $\mathsf{IBE\text{-}SPP}(n)$ i.e., Waters protocol.

Suppose, $|p^{(\ell)}|$ (resp. $|G_2^{(\ell)}|$) denotes the bit length of representation of $p^{(\ell)}$ (resp. an element of $G_2^{(\ell)}$). Like [16], we assume that the adversary $\mathcal{A}$ is allowed to make a maximum of $q = 2^{30}$ number of queries. For a given security level, we can now find the values of $|p^{(\ell)}|$ and $|G_2^{(\ell)}|$ for $\mathsf{IBE\text{-}SPP}(\ell)$ based on the bit length of the identities (i.e., n), q and ℓ. Note that, the value of $|p^{(\ell)}|$ (resp. $|G_2^{(\ell)}|$) thus obtained is the *minimum* required to avoid the Pollards rho (resp. number/function field seive) attack. In our comparison, the MOV degree k is taken to be same for different values of ℓ and $|G_2^{(\ell)}| = k\lg a$ ($G_2^{(\ell)}$ is a multiplicative subgroup of order $p^{(\ell)}$ of the finite field $\mathbb{F}_a^k$). As already noted, the value of $p^{(\ell)}$ is given by Pollard's rho. On the other hand, the logarithm of the size of $G_1^{(\ell)}$ is equal to

Table 1. *Approximate group sizes for attaining 80-bit security for* IBE-SPP(ℓ) *for different values of* ℓ *and relative space and time requirement. The first part corresponds to* $n = 160$ *and the second to* $n = 256$.

ℓ	$\Delta\rho^{(\ell)}$	$\lvert p^{(\ell)}\rvert$	$\lvert G_2^{(\ell)}\rvert$		$\alpha^{(\ell)}$		$\beta^{(\ell)}$	
			(a)	(b)	(a)	(b)	(a)	(b)
160	–	246	1891(2225)	3284(3872)	–	–	–	–
4	34	314	3269(3730)	5721(6538)	4.3(4.2)	4.4(4.2)	5.17(4.71)	5.46(4.81)
8	15	276	2443(2831)	4258(4944)	6.5(6.4)	6.5(6.4)	2.16(2.06)	2.18(2.08)
16	6	258	2102(2457)	3655(4288)	11.1(11.0)	11.1(11.1)	1.37(1.35)	1.38(1.35)
32	2	250	1960(2300)	3405(4006)	20.7(20.7)	20.7(20.7)	1.11(1.11)	1.12(1.11)
80	1	248	1924(2262)	3344(3939)	50.9(50.8)	50.9(50.9)	1.05(1.05)	1.06(1.05)
256	–	246	1891(2225)	3284(3872)	–	–	-	–
4	58	362	4530(5090)	7959(8954)	3.7(3.6)	3.8(3.6)	13.75(11.97)	14.24(12.37)
8	27	300	2948(3381)	5151(5919)	4.9(4.7)	4.9(4.8)	3.79(3.51)	3.86(3.57)
16	12	270	2326(2703)	4051(4717)	7.7(7.6)	7.7(7.6)	1.86(1.79)	1.88(1.81)
32	5	256	2066(2417)	3592(4212)	13.7(13.6)	13.7(13.6)	1.30(1.28)	1.31(1.29)
64	2	250	1960(2300)	3405(4006)	25.9(25.8)	25.9(25.9)	1.11(1.11)	1.11(1.11)
128	1	248	1924(2262)	3344(3939)	50.9(50.8)	50.9(50.9)	1.05(1.05)	1.06(1.05)

$\mathsf{max}(p^{(\ell)}, |G_2^{(\ell)}|/k)$. For relatively small MOV degree (i.e., $k \leq 6$), $|G_2^{(\ell)}|/k > |p^{(\ell)}|$ and so the logarithm of the size of $G_1^{(\ell)}$ is equal to $|G_2^{(\ell)}|/k = |\mathbb{F}_a^{(\ell)}|$. For a given ℓ, we have to store ℓ elements of $G_1^{(\ell)}$ in the public parameter file and a scalar multiplication in $G_1^{(\ell)}$ takes time proportional to $(|\mathbb{F}_a^{(\ell)}|)^3$.

Now, we are in a position to compare the space requirement in the public parameter file and the time requirement for a scalar multiplication in $G_1^{(\ell)}$ for different values of ℓ. Let $\alpha^{(\ell)} = \frac{\ell\times|G_1^{(\ell)}|}{n\times|G_1^{(n)}|} \times 100$ i.e., the relative amount of space (expressed in percentage) required to store the public parameters in case of IBE-SPP(ℓ) with respect to IBE-SPP(n) and $\beta^{(\ell)} = |\mathbb{F}_a^{(\ell)}|^3/|\mathbb{F}_a^{(n)}|^3$, i.e., the relative increase in time for scalar multiplication in $G_1^{(\ell)}$in the case of IBE-SPP(ℓ) with respect to IBE-SPP(n). Note that, $\beta^{(\ell)}$ can be computed from $|G_2^{(\ell)}|$ and $|G_2^{(n)}|$ since k cancels out from both numerator and denominator. An analysis similar to the efficiency consideration in Section 3 shows that pairing computation is also of order $|\mathbb{F}_a^{(\ell)}|^3$ (but with a larger constant factor). So, the ratio $\beta^{(\ell)}$ also holds for pairing computation and exponentiation in case of IBE-SPP(ℓ) with respect to Waters protocol.

In Table 1 we sum-up these results for $n = 160$ and 256 for different values of ℓ ranging from 4 to n for 80-bit security. The subcolumns (a) and (b) under $\alpha^{(\ell)}$ and $\beta^{(\ell)}$ stand for the values obtained for general characteristic field and field of characteristic three respectively. The values of $|G_2^{(\ell)}|, \alpha^{(\ell)}, \beta^{(\ell)}$ are computed using the formula as suggested in [16] (see Section 3); while in parenthesis we give the corresponding values as computed from the formula obtained from [21] (as given in Section 3 of [16]). Note that, the values of $\alpha^{(\ell)}$ and $\beta^{(\ell)}$ being the ratio of two quantities remain more or less invariant whether the underlying field is a general characteristic field or a field of characteristic three or which formula (of [16] or of [21]) is used.

Public parameter consists of $(\ell + 4)$ elements of G_1. From Table 1, for 80-bit security in general characteristic fields using EC with MOV degree 2, the public parameter size for Waters protocol will be around 37 kilobyte (kb) for 160-bit identities and 59 kb for 256-bit identities. The corresponding values in case of $\mathsf{IBE\text{-}SPP}(\ell)$ with $\ell = 16$ will be around 4 kb and 4.5 kb respectively. Similarly, in characteristic three field EC with MOV degree 6, the corresponding values are respectively 21.5 kb and 34.2 kb and for $\mathsf{IBE\text{-}SPP}(\ell)$ with $\ell = 16$ these are respectively 2.4 kb and 2.64 kb. There is an associated increase in computation cost by 30%. In typical applications, the protocol will be used in a key encapsulation mechanism (KEM). Thus the encryption and decryption algorithms will be invoked once for a message irrespective of its length. Also the key generation procedure is essentially an one-time offline activity. In view of this, the increase in computation cost will not substantially affect the throughput. On the other hand, the significant reduction in space requirement will be an advantage in implementing the protocol and also in reducing the time for downloading or transmitting the public parameter file over the net. Overall, we suggest $\ell = 16$ to be a good choice for implementing the protocol.

5 Security Proof

We prove Theorem 1 through a reductionist security argument. The proof is very similar to that of Waters and we describe only the essential features.

Proof : Suppose $\mathcal{A}$ is a (t, q)-CPA adversary for $\mathsf{IBE} - \mathsf{SPP}(\ell)$. Then we construct an algorithm $\mathcal{S}$ for DBDH running in time $(t + O(\tau q + \chi)$ such that, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{IBE-SPP}(\ell)} \leq 16q(\mu_\ell + 1)\mathsf{Adv}_{\mathcal{S}}^{\mathsf{DBDH}}$, where $\mu_\ell = \ell(N^{1/\ell} - 1)$. $\mathcal{S}$ will take as input a 5-tuple $\langle P, aP, bP, cP, Z\rangle$ where P is a generator of G_1, $aP, bP, cP \in G_1$ and $Z \in G_2$. We define the following game between $\mathcal{S}$ and $\mathcal{A}$.

Setup: $\mathcal{S}$ first chooses random $x, x_1, \ldots, x_\ell \in Z_m$ where $m = 4q$ (justified later); random
$y, y_1, \ldots, y_\ell \in Z_p$ and a random $k \in \{0, \ldots, \mu_\ell\}$. It then defines three functions: $F(\mathsf{v}) = p - mk + x + \sum_{i=1}^{\ell} x_i \mathsf{v}_i$, $J(\mathsf{v}) = y + \sum_{i=1}^{\ell} y_i \mathsf{v}_i$ and

$$K(\mathsf{v}) = \begin{cases} 0 \text{ if } x + \sum_{i=1}^{\ell} x_i \mathsf{v}_i \equiv 0 \bmod m \\ 1 \text{ otherwise} \end{cases}$$

Here, $F(\mathsf{v})$ and $K(\mathsf{v})$ are defined in such a way that $K(\mathsf{v}) \neq 0$ implies $F(\mathsf{v}) \not\equiv 0 \bmod p$. Next, $\mathcal{S}$ assigns $P_1 = aP$, $P_2 = bP$, $U' = (p - mk + x)P_2 + yP$ and $U_i = x_i P_2 + y_i P$ for $1 \leq i \leq \ell$. It provides $\mathcal{A}$ the public parameters $\langle P, P_1, P_2, U', U_1, \ldots, U_\ell\rangle$. Everything else is internal to $\mathcal{S}$. Note that from $\mathcal{A}$'s point of view the distribution of the public parameters is identical to the distribution of the public parameters in an actual setup.

Phase 1: The adversary $\mathcal{A}$ issues key extraction queries. Suppose, the adversary asks for the private key corresponding to an identity v. $\mathcal{S}$ first checks whether $K(\mathsf{v}) = 0$ and aborts in that situation and outputs a random bit. Otherwise, it gives $\mathcal{A}$ the pair

$$(D_1, D_2) = \left(-\frac{J(\mathsf{v})}{F(\mathsf{v})}P_1 + r(F(\mathsf{v})P_2 + J(\mathsf{v})P), \frac{-1}{F(\mathsf{v})}P_1 + rP\right)$$

where r is chosen at random from Z_p. As in Waters proof it is possible to show that (D_1, D_2) is a valid private key for v following the proper distribution. $\mathcal{S}$ will be able to generate this pair (D_1, D_2) if and only if $F(\mathsf{v}) \not\equiv 0$, for which it suffices to have $K(\mathsf{v}) \neq 0$.

Challenge: At this stage the adversary $\mathcal{A}$ submits two messages $M_0, M_1 \in G_2$ and an identity v^* with the constraint that it has not asked for the private key of v^* in Phase 1. $\mathcal{S}$ aborts if $F(\mathsf{v}^*) \neq 0$ and outputs a random bit. Otherwise, $\mathcal{S}$ chooses a random bit $\gamma \in \{0,1\}$ and gives $\mathcal{A}$ the tuple $C' = \langle ZM_\gamma, cP, J(\mathsf{v}^*)cP\rangle$.

If $\langle P, aP, bP, cP, Z\rangle$ given to $\mathcal{S}$ is a valid DBDH tuple, i.e., $Z = e(P,P)^{abc}$ then C' is a valid encryption for M_γ. Since,

$$e(P,P)^{abc} = e(aP, bP)^c = e(P_1, P_2)^c$$

and using $F(\mathsf{v}^*) = p - mk + x + \sum_{i=1}^{\ell} x_i \mathsf{v}_i^* \equiv 0 \bmod p$ it is possible to show that $J(\mathsf{v}^*)cP = cV$. Note that, this condition is satisfied as long as $F(\mathsf{v}^*) \equiv 0 \bmod p$, which holds if $x + \sum_{j=1}^{\ell} x_j \mathsf{v}_j^* = km$.

Otherwise, Z is a random element of G_2 and C' gives no information about $\mathcal{S}$'s choice of γ.

Phase 2: This phase is similar to Phase 1, with the obvious restriction that $\mathcal{A}$ cannot ask for the private key of v^*. We note that the total number of key extraction queries together in Phase 1 and 2 should not exceed q.

Guess: $\mathcal{A}$ outputs a guess γ' of γ. Then $\mathcal{S}$ outputs $1 \oplus \gamma \oplus \gamma'$.
Suppose the adversary has not aborted upto this point. Waters introduces a technique whereby the simulator is allowed to abort under certain condition. The simulator samples the transcript it received from the adversary during the attack phase. Based on the sample, it decided whether to abort and output a random string. The rationale for such "artificial abort" is the following: The probability of abort during the attack phase depends on the adversarial transcript and can be different for different transcripts. The purpose of artificial abort is to ensure that the simulator aborts with (almost) the same probability for all adversarial queries. This ensures that the adversary's success is independent of whether the simulator aborts or not. The probability analysis performed by Waters in [24] requires this independence. For details of this method see [24]. Here we just note that the artificial abort stage requires an additional

$\chi = O(\epsilon^{-2}\ln(\epsilon^{-1})\lambda^{-1}\ln(\lambda^{-1}))$ time. Further, it is independent of the parameter ℓ which defines the generalisation over Waters [24] that we introduce here.

Let abort be the probability of the simulator aborting during the actual attack (as opposed to artificial abort) and let $\lambda = Pr[\overline{\mathsf{abort}}]$. In Appendix B, we calculate the lower bound of λ to be $\frac{1}{m(\mu_\ell+1)}(1-2\frac{q}{m})$. Using $m = 4q$ gives $\lambda \geq \frac{1}{4q(\mu_\ell+1)}$. Now using the analysis performed by Waters [24], we obtain

$$\epsilon \leq 16q(\mu_\ell+1)\ \mathsf{Adv}^{\mathsf{DBDH}}(t + O(\tau q) + \chi).$$

The time component of $O(\tau q)$ comes because of the scalar multiplications performed in Phase 1 and 2 of Key Generation (these scalar multiplications are the only computationally intensive part in the simulation). This completes the proof. □

Remark 1: Note that, in the simulation, only the computation of $F(\mathsf{v}), J(\mathsf{v}) \in Z_p$ depends on the size parameter ℓ. Once $F(\mathsf{v})$ and $J(\mathsf{v})$ are obtained, the key generation in Phase 1 and 2 and cipher text generation in Challenge is done through some scalar multiplications involving $F(\mathsf{v})$ and $J(\mathsf{v})$. Cost of computation of $F(\mathsf{v})$ and $J(\mathsf{v})$ are insignificant compared to the cost of a scalar multiplication. So the simulation time is independent of the size parameter ℓ.

Remark 2: The technique of "artificial abort" is new to security proofs and was introduced by Waters [24]. (It is not present in the security proof of Boneh and Boyen [3] which is also an identity based encryption protocol which is secure in the full model.) We feel that the technique of artificial abort can be avoided. This technique only lowers the probability of not aborting. Hence, it should be possible to directly work with the lower bound λ of not aborting, without actually going through the artificial abort step. Avoiding the artificial abort step will require performing a new probability analysis. We hope to do that in the future.

6 CCA Security

Recent works of Boneh, Canetti, Halevi and Katz [5, 8, 15] show how to build CCA secure encryption scheme from identity based encryption. One way to achieve CCA-security for our scheme is to follow the strategy suggested in [24]. As our scheme closely resembles that of [24] it is possible to build a hybrid 2-level HIBE [19, 18] in essentially the same way and the reduction follows.

We show that it is possible to take a different approach based on the oracle bilinear decision Diffie-Hellman (OBDH) assumption which is a variation of the ODH assumption used in [1]. The OBDH assumption is as follows [22].

- Instance : $\langle P, aP, bP, cP, \mathsf{str}\rangle$ where $a, b, c \in Z_p$ and $\mathsf{str} \in \{0,1\}^k$.
- Oracle : $\mathcal{H}_a(X, Y)$, with $X, Y \in G_1$. When invoked with (a_1P, b_1P) returns $H(a_1P, e(a_1P, a_2P)^a)$, where $H : G_1 \times G_2 \to \{0,1\}^k$ is a hash function.
- Restriction : Cannot query $\mathcal{H}_a(,)$ on (cP, bP).
- Task : Determine whether $\mathsf{str} = H(cP, e(cP, bP)^a)$ or str is random.

Any algorithm $\mathcal{A}$ for OBDH takes as input an instance $(P, aP, bP, cP, \mathsf{str})$ of OBDH and produces as output either zero or one. The advantage of an algorithm $\mathcal{A}$ in solving OBDH is formally defined in the following manner.

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{OBDH}} = |\mathsf{Pr}[\mathcal{A} \text{ outputs } 1|E_1] - \mathsf{Pr}[\mathcal{A} \text{ outputs } 1|E_2]|$$

where E_1 is the event that $\mathsf{str} = H(cP, e(cP, bP)^a)$ and E_2 is the event that str is random. The quantity $\mathsf{Adv}^{\mathsf{OBDH}}(t, q)$ denotes the maximum of $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{OBDH}}$ where the maximum is taken over all adversaries running in time at most t and making at most q queries to the oracle $\mathcal{H}_a(,)$.

To suit into the OBDH assumption we modify our constructions of Section 3 as follows: **Setup** and **Key Generation** remain unaltered. To encrypt a message, we first generate a symmetric key $\mathsf{sym.key} = H(tP, e(P_1, P_2)^t)$. Then the cipher is $C = \langle tP, tV, y) \rangle$, where y is the encryption of the message using the symmetric key $\mathsf{sym.key}$. To decrypt, all that we need is $e(P_1, P_2)^t = e(D_1, tP)/e(D_2, tV)$ and then find $\mathsf{sym.key}$ using H.

Security: Breaking the (modified) IBE implies either solving OBDH or breaking the symmetric encryption scheme. The later we assume to be unbreakable under chosen ciphertext attack. CCA security under the OBDH assumption is expressed in the following theorem proof of which will be provided in the full version of the paper.

Theorem 2. *For $t \geq 1$, $q \geq 1$; $\mathsf{Adv}^{\mathsf{IBE}}(t, q) \leq 16q(\mu_\ell + 1)\mathsf{Adv}^{\mathsf{OBDH}}(t + O(\tau q) + \chi)$, where identities are chosen from Z_N, $1 < \ell \leq \lg N$ is a size parameter, $\mu_\ell = \ell(N^{1/\ell} - 1)$.*

Note: Subsequent to the acceptance notification of this submission to ICISC 2005, we came to know that a paper describing a similar construction as ours has been posted on the eprint archive by David Naccache. (We also note that an earlier version of the present paper was submitted to Asiacrypt 2005, whose submission deadline was May 30, 2005.) Though the construction is similar, the paper by Naccache does not perform any concrete security analysis. In fact, the paper mentions that the loss of security due to the generalisation is "insignificant". As discussed in Section 4, this is not correct. In fact, the conversion of security degradation into a trade-off between time and space is original to our paper and is the most important feature of the generalisation of Waters scheme. On the other hand, we would like to mention that Naccache's paper presents a better exposition of the security proof than that given in Waters.

References

1. M. Abdalla, M. Bellare and P. Rogaway. DHIES : An encryption scheme based on the Diffie-Hellman problem, *Proceedings of CT-RSA 2001*, Lecture Notes in Computer Science, Springer-Verlag, pages 143–158.
2. D. Boneh, X. Boyen. Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles, EUROCRYPT 2004, LNCS, pp 223–238.

3. D. Boneh, X. Boyen. Secure Identity Based Encryption without Random Oracles. In *Proceedings of the Advances in Cryptology* – (CRYPTO'04), 2004.
4. D. Boneh, X. Boyen, E. Goh, Hierarchical Identity Based Encryption with Constant Size Ciphertext, EUROCRYPT 2005, Vol. 3494 of LNCS, pp 440-456.
5. D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-Ciphertext Security from Identity-Based Encryption. Journal Submission. Available from D. Boneh's website.
6. D. Boneh, M. Franklin. Identity Based Encryption from the Weil Pairing. CRYPTO – 2001, volume 2139 of LNCS, pp. 213–229, 2001.
7. D. Boneh, M. Franklin. Identity Based Encryption from the Weil Pairing. SIAM J. of Computing, Vol. 32, No. 3, pp. 586–615, 2003.
8. D. Boneh and J. Katz. Improved Efficiency for CCA-Secure Cryptosystems Built Using Identity Based Encryption. In *Proceedings of RSA-CT '05*, LNCS 3376, pp. 87-103, 2005.
9. P. S. L. M. Barreto, H. Y. Kim, B. Lynn and M. Scott. Efficient Algorithms for Pairing-Based Cryptosystems. CRYPTO 2002, LNCS 2442, pp. 354–368.
10. P. S. L. M. Barreto and M. Naehrig. Pairing-Friendly Elliptic Curves of Prime Order. Cryptology ePrint Archive, Report 2005/133. Available from http://eprint.iacr.org/2005/133/. Accepted for presentation at SAC 2005.
11. M. Bellare and P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In ACM Conference on Computer and Communications Security - CCS 1993, pp 62-73, 1993.
12. X. Boyen, Q. Mei and B. Waters. Direct Chosen Ciphertext Security from Identity-Based Techniques. In 12th ACM Conference on Computer and Communication Security – CCS 2005, To appear. This version is available from Cryptology ePrint Archive, Report 2005/288.
13. C. Cocks. An Identity Based Encryption Scheme Based on Quadratic Residue. In Proceedings of the 8th IMA International Conference on Cryptography and Coding, pp 26–28, 2001.
14. R. Canetti, S. Halevi and J. Katz. A Forward-Secure Public-Key Encryption Scheme. In EUROCRYPT 2003, Volume 2656 of LNCS, pp 255-271. 2003.
15. R. Canetti, S. Halevi and J. Katz. Chosen-ciphertext Security from Identity Based Encryption. In *Proceedings of Eurocrypt 2004*. LNCS, 2004.
16. D. Galindo. The Exact Security of Pairing Based Encryption and Signature Schemes. Talk at Workshop on Provable Security, INRIA, Paris. November 3-5, 2004. Available from author's website.
17. S. Galbraith, K. Harrison and D. Soldera. Implementing the Tate Pairing. ANTS V, LNCS 2369, pp. 324-337, 2002.
18. C. Gentry and A. Silverberg, Hierarchical ID-Based Cryptography, ASIACRYPT 2002, LNCS, 2002.
19. J. Horwitz and B. Lynn. Towards Hierarchical Identity-Based Encryption. In EUROCRYPT 2002, pp 466–481, 2002.
20. N. Koblitz and A. Menezes, Another look at "provable security", Cryptology ePrint Archive, Report 2004/152, http://eprint.iacr.org/2004/152/, final version (to appear in Journal of Cryptology).
21. A. K. Lenstra and E. R. Verheul, Selecting Cryptographic Key Sizes, Jr. Cryptology 14(4), pp. 255-293 (2001)
22. P. Sarkar, HEAD: Hybrid Encryption with Delegated Decryption Capability, Proceedings of Indocrypt 2004, LNCS, pp 230-244.
23. A. Shamir. Identity-based Cryptosystems and Signature Schemes. CRYPTO 84, LNCS, pp 47-53, 1985.

24. B. Waters. Efficient Identity-Based Encryption Without Random Oracles. *Eurocrypt 2005*, Eurocrypt 2005, Vol. 3494 of LNCS, Also available from Cryptology ePrint Archive, Report 2004/180, http://eprint.iacr.org/2004/180/.

A Definitions

A.1 Cryptographic Bilinear Map

Let G_1 and G_2 be cyclic groups of same prime order p and $G_1 = \langle P \rangle$, where we write G_1 additively and G_2 multiplicatively. A mapping $e : G_1 \times G_1 \rightarrow G_2$ is called a cryptographic bilinear map if it satisfies the following properties:

- Bilinearity : $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1$ and $a, b \in Z_p$.
- Non-degeneracy : If $G_1 = \langle P \rangle$, then $G_2 = \langle e(P, P) \rangle$.
- Computability : There exists an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

Since $e(aP, bP) = e(P, P)^{ab} = e(bP, aP)$, $e()$ also satisfies the symmetry property. Modified Weil pairing [6] and Tate pairing [9, 17] are examples of cryptographic bilinear maps.

A.2 IBE Protocol

Following [6] an identity based encryption scheme is specified by four algorithms: Setup, Key Generation, Encryption and Decryption.

Setup: It takes input a security parameter and returns the system parameters together with the master key. The system parameters include a description of the message space, the ciphertext space and the identity space. They are publicly known while the master key is known only to the private key generator (PKG).

Key Generation: It takes as input an identity v and returns a private key D_{v}, using the master key. The identity v is used as the public key while D_{v} is the corresponding private key.

Encryption: It takes as input the identity v and a message from the message space and produces a ciphertext in the cipher space.

Decryption: It takes as input the ciphertext and the private key of the corresponding identity v and returns the message or bad if the ciphertext is not valid.

A.3 Security Model

Here we define indistinguishability under chosen ciphertext attack for identity based encryption schemes under a chosen identity. In this model, an adversary is allowed to choose adaptively the public key it wishes to attack. In concrete terms, security of an IBE scheme can be defined using the following game.

An adversary (whom we denote by $\mathcal{A}$) is allowed to query two oracles – a decryption oracle and a key-extraction oracle. At the initiation it is provided with the system public parameters.

Phase 1: Adversary $\mathcal{A}$ makes a finite number of queries where each query is addressed either to the decryption oracle or to the key-extraction oracle. In a query to the decryption oracle it provides the ciphertext as well as the identity under which it wants the decryption. Similarly, in a query to the key-extraction oracle, it asks for the private key of the identity it provides. Further, $\mathcal{A}$ is allowed to make these queries adaptively, i.e., any query may depend on the previous queries as well as their answers.

Challenge: At this stage $\mathcal{A}$ fixes an identity, v^* and two equal length messages M_0, M_1 under the (obvious) constraint that it has not asked for the private key of v^* and gets a ciphertext (C^*) corresponding to M_γ, where γ is chosen uniformly at random from $\{0, 1\}$.

Phase 2: $\mathcal{A}$ now issues additional queries just like Phase 1, with the (obvious) restriction that it cannot ask the decryption oracle for the decryption of C^* under v^* nor the key-extraction oracle for the private key of v^*.

Guess: $\mathcal{A}$ outputs a guess γ' of γ.

The advantage of the adversary $\mathcal{A}$ in attacking the IBE scheme is defined as:

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{IBE}} = 2|Pr[(\gamma = \gamma')] - 1/2|$$

The quantity $\mathsf{Adv}^{\mathsf{IBE}}(t, q_{\mathsf{ID}}, q_{\mathsf{C}})$ denotes the maximum of $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{IBE}}$ where the maximum is taken over all adversaries running in time at most t and making at most q_{C} queries to the decryption oracle and q_{ID} queries to the key-extraction oracle. Any IBE scheme secure against such an adversary is said to be secure against chosen ciphertext attack (CCA).

In our security reduction of Theorem 1, we restrict the adversary $\mathcal{A}$ from making any query to the decryption oracle. An IBE scheme secure against such an adversary is said to be secure against chosen plaintext attack (CPA). $\mathsf{Adv}^{\mathsf{IBE}}(t, q)$ in this context denotes the maximum advantage where the maximum is taken over all adversaries running in time at most t and making at most q queries to the key-extraction oracle.

A.4 Hardness Assumption

We define the security of our identity based encryption scheme in terms of the *decision bilinear Diffie-Hellman* problem (DBDH). The DBDH problem [7] in G_1 is as follows: given a tuple $\langle P, aP, bP, cP, Z\rangle$, where $Z \in G_2$, decide whether $Z = e(P, P)^{abc}$ which we denote as Z is real or Z is random. The advantage of a probabilistic algorithm $\mathcal{B}$, which takes as input a tuple $\langle P, aP, bP, cP, Z\rangle$ and outputs a bit, in solving the DBDH problem is defined as

$$\begin{aligned}\mathsf{Adv}_{\mathcal{B}}^{\mathsf{DBDH}} = |&Pr[\mathcal{B}(P, aP, bP, cP, Z) = 1|\mathsf{Z\ is\ real}] \\ &- Pr[\mathcal{B}(P, aP, bP, cP, Z) = 1|\ \mathsf{Z\ is\ random}]|\end{aligned}$$

where the probability is calculated over the random choice of $a, b, c \in Z_p$ as well as the random bits used by $\mathcal{B}$. The quantity $\mathsf{Adv}^{\mathsf{DBDH}}(t)$ denotes the maximum of $\mathsf{Adv}_{\mathcal{B}}^{\mathsf{DBDH}}$ where the maximum is taken over all adversaries running in time at most t.

B Lower Bound of $\boldsymbol{\lambda}$

We calculate a lower bound on λ for any set of q queries $\mathsf{v}^{(1)}, \ldots, \mathsf{v}^{(q)}$ and a challenge identity v^* as:

$$\begin{aligned}
\lambda &= Pr[\bigwedge_{i=1}^{q} \Big(K(\mathsf{v}^{(i)}) = 1\Big) \wedge (x + \sum_{j=1}^{\ell} x_j \mathsf{v}_j^* = km)] \\
&= Pr[\bigwedge_{i=1}^{q} \Big(K(\mathsf{v}^{(i)}) = 1\Big)] Pr[(x + \sum_{j=1}^{\ell} x_j \mathsf{v}_j^* = km)| \bigwedge_{i=1}^{q} \Big(K(\mathsf{v}^{(i)}) = 1\Big)] \\
&= (1 - Pr[\bigvee_{i=1}^{q} \Big(K(\mathsf{v}^{(i)}) = 0\Big)] Pr[(x + \sum_{j=1}^{\ell} x_j \mathsf{v}_j^* = km)| \bigwedge_{i=1}^{q} \Big(K(\mathsf{v}^{(i)}) = 1\Big)] \\
&\geq (1 - \sum_{i=1}^{q} Pr[\Big(K(\mathsf{v}^{(i)}) = 0\Big)] Pr[(x + \sum_{j=1}^{\ell} x_j \mathsf{v}_j^* = km)| \bigwedge_{i=1}^{q} \Big(K(\mathsf{v}^{(i)}) = 1\Big)] \\
&= (1 - \frac{q}{m}) Pr[(x + \sum_{j=1}^{\ell} x_j \mathsf{v}_j^* = km)| \bigwedge_{i=1}^{q} \Big(K(\mathsf{v}^{(i)}) = 1\Big)] \\
&= \frac{1}{\mu_\ell + 1}(1 - \frac{q}{m}) Pr[K(\mathsf{v}^*) = 0| \bigwedge_{i=1}^{q} \Big(K(\mathsf{v}^{(i)}) = 1\Big)] \\
&= \frac{1}{\mu_\ell + 1}(1 - \frac{q}{m}) \frac{Pr[K(\mathsf{v}^*) = 0]}{Pr[\bigwedge_{i=1}^{q} K(\mathsf{v}^{(i)} = 1)]} Pr[\bigwedge_{i=1}^{q} \Big(K(\mathsf{v}^{(i)}) = 1\Big) |K(\mathsf{v}^*) = 0] \\
&\geq \frac{1}{m(\mu_\ell + 1)}(1 - \frac{q}{m}) Pr[\bigwedge_{i=1}^{q} \Big(K(\mathsf{v}^{(i)}) = 1\Big) |K(\mathsf{v}^*) = 0] \\
&= \frac{1}{m(\mu_\ell + 1)}(1 - \frac{q}{m})(1 - Pr[\bigvee_{i=1}^{q} \Big(K(\mathsf{v}^{(i)}) = 0\Big) |K(\mathsf{v}^*) = 0] \\
&\geq \frac{1}{m(\mu_\ell + 1)}(1 - \frac{q}{m})(1 - \sum_{i=1}^{q} Pr[K(\mathsf{v}^{(i)}) = 0|K(\mathsf{v}^*) = 0] \\
&= \frac{1}{m(\mu_\ell + 1)}(1 - \frac{q}{m})^2 \\
&\geq \frac{1}{m(\mu_\ell + 1)}(1 - 2\frac{q}{m})
\end{aligned}$$

In the above derivation the equality in the last but one step comes from the fact that

$$Pr[K(\mathsf{v}^{(i)}) = 0 | K(\mathsf{v}^{*}) = 0] = Pr[K(\mathsf{v}^{(i)}) = 0] = 1/m$$

since $K(\mathsf{v}^{(i)}) = 0$ for $1 \leq i \leq q$ and $K(\mathsf{v}^{*}) = 0$ are mutually independent events.

C Security of the Signature Scheme

Brief sketch: This proof also is a reduction. Suppose $\mathcal{A}$ is a CPA adversary for the signature scheme. Then we construct an algorithm $\mathcal{S}$ for Comutational Diffie-Hellman problem (CDH). $\mathcal{S}$ will take as input a 3-tuple $\langle P, aP, bP \rangle$ where P is a generator of G_1 and $aP, bP \in G_1$. We define the following game between $\mathcal{S}$ and $\mathcal{A}$.

The Setup and Signature Generation steps of this game is exactly same as the Setup and Phase 1 of Section 5.

Forge: At this stage the adversary $\mathcal{A}$ submits a message $M^{*} \in Z_N$ and a signature $\sigma^{*} = (\sigma_1^{*}, \sigma_2^{*})$ with the constraint that it has not asked for the signature of M^{*} in the Signature Generation phase. $\mathcal{A}$ wins if σ^{*} is a valid signature on M^{*}.

If $\mathcal{A}$ is successful in forging the signature, $\mathcal{S}$ first checks whether $F(M^{*}) \neq 0$ and aborts in that situation. Otherwise, $\mathcal{S}$ first computes $J(M^{*})\sigma_2^{*}$ and then adds the inverse of this product with σ_1^{*}. It returns the end result as the value of abP.

Since $F(M^{*}) = 0$, then as in the Challenge part of proof of Theorem 1, we have

$$J(M^{*})\sigma_2^{*} = rV.$$

Note that, this condition is satisfied as long as $F(M^{*}) \equiv 0 \bmod p$, which holds if $x + \sum_{j=1}^{\ell} x_j m_j^{*} = km$.

Now, $\sigma_1^{*} = abP + rV$ and hence $abP = \sigma_1^{*} - rV$.

Note that, the conditions under which $\mathcal{S}$ aborts this game is exacly the same under which $\mathcal{S}$ aborts the game in Theorem 1. So the lower bound on the probability of not aborting remains exactly the same.