基于身份的加密 IBE 双线性映射

关键词：身份认证、双线性映射、椭圆曲线

背景

shamir 寻求一个公钥密码加密体制，其中的公钥密码是一个任意的随机数。
在这个体制中，有四步算法：

1. **安装时**，将生成全局参数和一个主密钥；
2. **提取时**，使用主密钥生成与任意公钥字符串相对应的私钥；
3. **加密** 功能，使用公钥ID加密邮件；
4. **解密** 功能，使用相应的私钥解密消息；

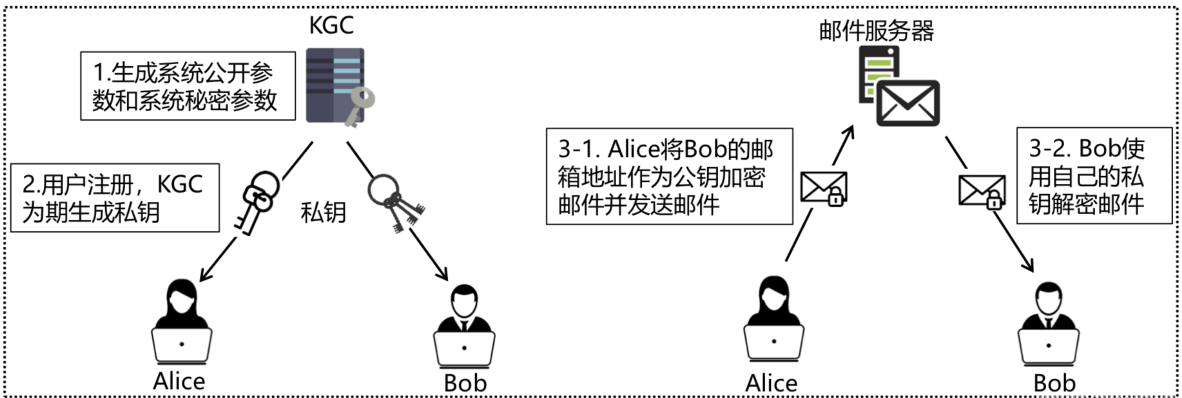
产生的原因是为了简化电子邮件系统中的证书管理。
以一个例子为例，Alice要把邮件发送给Bob，她简单的使用Bob的邮箱账号作为公钥；Bob在接受信息的时候，向密钥分发中心PKG验证自己的身份，从PKG中获得私钥，然后解密邮件。

与**现有的安全的电子邮件**的基本架构不同的是，即使B没有申请他的公钥，但仍然能够收到别人发给他的使用公钥加密的邮件。
注意：

密钥托管是介于身份得到电子邮件系统中固有的，
PKG知道Bob的私钥

简单基于BF-IBE邮件系统

- 初始化阶段
  - 密钥生成中心（KGC）运行Setup算法（输入：安全参数），生成系统公开参数和系统秘密参数
- 用户加入阶段
  - 令新加入用户的邮箱地址为ID，KGC运行Extract算法（输入：系统秘密参数，ID），生成并授予该用户私钥
- 邮件安全传输阶段
  - 令接收方的邮箱地址为ID，发送方运行Enc算法加密邮件（输入：系统公开参数，ID，邮件内容），生成密文C并发送给接收方
  - 接收方用私钥解密出邮件内容



完整功能的IBE方案

数学难题：CBDH（Computational Bilinear Diffie-Hellman）

证书撤销：

1. Bob的**公钥证书**中包含一个预设的过期时间；
2. **在IBE系统中**，可以将公钥密码设置为“ bob@company.com || current year”；密钥更新之后，Bob向PKG询问新的私钥。但对于信息发送者来说，不需要从Bob中获取新的公钥
3. 甚至可以时间设置精确到**每一天**，密钥分发中心由Bob来维护。
4. 管理用户证书，可以将公钥密码设置为“ bob@company.com || current year || clearance=secret”；Bob得到sercret授权码之后，才可以阅读信件。

解密授权：

此处将Bob视为PKG，在初始阶段，Bob运行Setup算法获得系统公开参数和主密钥。Bob从CA那获取包含公钥参数的证书，Alice从B的证书中获取公钥

1. 授权笔记本，Alice使用公钥加日期的方式，Bob按照天数由主密钥生成解密子密钥保存在笔记本上。 保护主密钥不泄露
2. 授权责任，Alice使用公钥加主题的方式，Bob按照不同的任务生成解密子密钥分发给不同的助理。 每一个任务助理能够获取与自己任务相关的信息

$\forall M \in \mathrm{M} : Decrypt(params, C, d) = M, whereC = Encrypt(params, ID, M)$

选择密文安全性：

1. **允许攻击者**攻击她所选择的任意公钥ID；
2. **选择密文攻击时**，攻击者从PKG获取她选择的任何公钥的私钥，而不是ID的公钥；攻击者获得了一些与她所选择的某些身份对应的私人密钥，然后试图攻击她所选择的其他公钥；即使在这种询问下，攻击者获得的成功的机会仍然微不足道。

Q1：攻击者在第一阶段如何进行询问？

1. 提取私钥的询问 *Extraction query* ，攻击者使用 *Extract* 算法生成与IDi对应的di；
2. 解密密文的询问 *Decryption query* ，攻击者使用 *Decrypt* 算法解密密文Ci，获得明文；

Q2:什么是IND-ID-CCA安全？

1. 首先我们需要了解什么是IND-CCA安全？
   转载链接：**什么是公钥密码学的IND-CCA安全定义**.

   1.生成公钥和私钥(p_k, s_k)。攻击者A能够获得公钥p_k。

   2.私密的指定$b \leftarrow \{0, 1\}$。

   3.攻击者A可以进行解密问询Dec_{s_k}，和加密问询Enc_{p_k}。

   4.A输出一对消息(m_0, m_1)。

   5.我们输出加密$c = Enc_{p_k}(m_b)$。

   6.攻击者被允许使用更多的加密和解密，例如在第三步中，但是我们不被允许要求解密c。

   7.A输出$b^{‘}$。如果$b = b^{‘}$，A就获胜了。

2. 运用类比思想理解IND-ID-CCA安全？

   > 1. 生成$(params, master - key, ID, d)$,攻击者能够获得$parsms$。
   > 2. 攻击者输出一对明文(M_0, M_1)。
   > 3. 挑战者私密的指定$b \in \{0, 1\}$。
   > 4. 挑战者进行加密，输出$C = Encrypt(params, ID, M_b)$。

5. 攻击者可以进行私钥询问、解密询问、 加密询问 。
6. 攻击者输出b'，如果$b=b'$,攻击者就获胜.

   定义攻击者A对方案ε的优势为与安全参数k有关的函数:
   $Adv_{\varepsilon,A}(k)=\left|Pr[b=b']-\dfrac{1}{2}\right|$。

3. IND-CCA与IND-ID-CCA的区别在于?

   1）后者多了私钥询问。

   2）前者攻击者任意选取公钥进行挑战，后者由挑战者选取的公钥

IBE方案的选择密文安全性： 在多项式时间内，IND-ID-CCA问题中，攻击者A对方案ε的优势可忽略。

语义安全： （选择明文攻击的语义安全）

1. 语义安全中攻击者不能发出解密查询。
2. 我们将语义安全的公钥系统称为IND-CPA安全。

   【补充】语义安全的含义是给定一个密文，攻击者对对应的明文一无所知。

IBE方案的语义安全:

1. 允许攻击者进行私钥提取查询。
2. 同样的攻击者不能发出解密查询。
3. 在IBE方案中的语义安全称为IND-ID-CPA安全。游戏规则和IND-ID-CCA类似

单向加密(OWE): 通过对随机明文的加密，对手不能完全生成整个明文。

双线性映射和双线性DH假设

双线性映射

符号定义： 对于素数阶的群G,我们使用G^*表示集合$G^*=G\backslash\{O\}$，

IBE利用的双线性映射： $\hat{e}:G_1\times G_1\to G_2$, G_1,G_2是两个素数阶群，素数q为大素数。

映射关系满足以下三个条件的称为可接受的映射:

1. 双线性。如果有下式对于所有$P,Q\in G_1$，$a,b\in Z$都成立。

$$\hat{e}(aP,bQ)=\hat{e}(P,Q)^{ab}$$

2. 非退化性。该映射不会将$G_1\times G_1$的所有二元组都映射到G_2中。如果P是G_1的生成元，那么$\hat{e}(P,P)$是G_2的生成元。
3. 可计算性。存在有效算法可以计算$\hat{e}(P,Q)$

DDH问题:

在Diffie-Hellman密钥协议中，G，g，g^a，g^b 都是公共的， g^{ab}是密钥。DDH问题就是对手是否能够从随机的G中的元素区分出Alice和Bob的密钥g^{ab}。

具体的说，给定

$$G,g,g^a,g^b\text{和}T_X\text{使得}T_0\text{是}G\text{中随机的一个元素，}T_1=g^{ab}\text{同时}x\text{被随机均匀的从}\{0,\ 1\}\text{中选择找出}x\text{。}$$

如果对手输出正确x的概率大于1/2，则说明对手能够解决DDH问题，表明了G，g^a，g^b泄露了关于g^{ab}的信息

CDH问题：

在Diffie-Hellman密钥协议中，攻击者已知G, g^a, g^b，找到g^{ab}，是困难的。

双线性映射对这两个群的影响：

1. MOV规约。将G1上的离散对数问题，规约到G2上的离散对数问题。

$$\alpha \in Z_q, Q = \alpha P, g = \hat{e}(P, P), h = \hat{e}(Q, P), h = g^a$$

   为了使离散对数在G1中是困难的，我们必须保证离散对数在G2中是困难的。

2. DDH问题是简单的。在G1上的DDH问题是区分$< P, aP, bP, abP >$和$< P, aP, bP, cP >$

   因为我们有

$$c = ab \; mod \; q \Longleftrightarrow \hat{e}(P, cP) = \hat{e}(aP, bP)$$

【补充】G1上的CDH问题仍然是困难的，即已知$< G, P, aP, bP >$计算出 abP是困难的。

双线性DH假设（BDH）

由于G1上的DDH问题是容易的，所以我们不能使用DDH来构建密码体系，在IBE系统的安全性是基于CDH假设的一个变体，称为双线性DH假设（BDH）

BDH问题：

已知$< P, aP, bP, cP >$，其中$a, b, c \in Z_q^*$，计算$W = \hat{e}(P, P)^{abc} \in G_2$>。算法A具有$\varepsilon$的优势求解BDH问题。

$$Pr[A(P, aP, bP, abP) = \hat{e}(P, P)^{abc}] \geq \varepsilon$$

BDH参数生成器：

$\mathcal{G}$是一个随机算法，作为BDH参数生成器。

输入：安全参数$k \in Z^+$。

输出：大素数q, G_1, G_2以素数q为阶的两个群，可接受的双线性映射关系$\hat{e} : G_1 \times G_1 \rightarrow G_2$ 。

用下面的式子表示：

$$\mathcal{G}(1^k) =< q, G_1, G_2, \hat{e} >$$

符号再次定义：

我们在描述群G时，包含了能够在多项式时间内计算出群G的算法和群G的生成元。群G的生成元能够在群G中生成均匀的随机元素；我们在描述$\hat{e}$时，包含了能够在多项式时间内计算出可接受的双线性映射算法。

BDH假设：

首先定义算法A对于解决由$\mathcal{G}$产生的BDH问题的优势ε

$$Adv_{\mathcal{G},A}(k) = Pr\left[A\begin{pmatrix} q, G_1, G_2, \hat{e}, \\ P, aP, bP, cP \end{pmatrix} = \hat{e}(P, P)^{abc} \middle| \begin{matrix} < q, G_1, G_2, \hat{e} > \leftarrow \mathcal{G}(1^k), \\ P \leftarrow G_1^*, a, b, c \leftarrow Z_q^* \end{matrix} \right] \geq \epsilon(k)$$

当任意的算法A对解决$\mathcal{G}$产生的BDH问题的优势，小于1/f(k)时，则说明$\mathcal{G}$时满足BDH假设的，可以说BDH在$\mathcal{G}$产生的群G上困难的。

论文中的IBE方案

一个基本的IBE方案（BasicIdent）方便展示：

Setup:给定一个安全参数k

1. 运行$\mathcal{G}$，$\mathcal{G}(1^k) \to < q, G_1, G_2, \hat{e} >, P$是G1的生成元。
2. 选择随机数$s \in Z_q^*$，设置公钥$P_{pub} = sP$。
3. 选择加密哈希函数H1和H2。$H_1 : \{0,1\}^* \to G_1^*, H_2 : G_2 \to \{0,1\}^n$

   密文空间$C = G_1^* \times \{0,1\}^n$。系统参数有$params =< q, G_1, G_2, \hat{e}, n, P, P_{pub}, H_1, H_2 >$

Extract:给定字符串ID

1. $ID \in \{0,1\}^*$，计算$Q_{ID} = H_1(ID) \in G_1^*$
2. $d_{ID} = sQ_{ID}$，其中s就是主密钥

Encrypt:选择随机数r

$$C =< rP, M \oplus H_2(g_{ID}^r) >, where\ g_{ID} = \hat{e}(Q_{ID}, P_{pub}) \in G_2^*$$

Decrypt:已知私钥d

$$C = \langle U, V \rangle,\ compute\ M = V \oplus H_2(\hat{e}(d_{ID}, U))$$
$$\hat{e}(d_{ID}, U) = \hat{e}(sQ_{ID}, rP) = \hat{e}(Q_{ID}, P)^{sr} = \hat{e}(Q_{ID}, P_{pub})^r = g_{ID}^r$$

安全性证明：证明基本的IBE方案是语义安全（IND-ID-CPA），假设BDH在由$\mathcal{G}$生成的群上是困难的。

一个定理：

1. 假设哈希函数H1，H2是随机预言机
2. BasicIdent是语义安全的，基于BDH在$\mathcal{G}$生成的群上是安全的。

   假设在ND-ID-CPA上有攻击者A对BasicIdent方案具有优势$\epsilon(k)$，如果A能够进行q_E次私钥查询和q_{H_2}次对H2的哈希查询，那么存在攻击者B对BDH问题有优势

$$Adv_{\mathcal{G},B}(k) \geq \frac{2\epsilon}{e(1+q_E) \cdot q_{H_2}}$$

一个公钥加密方案BasicPub：与基本的IBE方案类似，但是Q_{ID}是随机选取的随机数。

引理1：

1. H1是一个从$\{0,1\}\ to\ G_1^*$的随机预言机。
2. A可以看作攻击者，也可以看作算法。在IND-ID-CPA上攻击者A对于BasicIdent有优势$\epsilon(k)$。
3. 假设A最多进行q_E次的私钥提取查询。

   由前两条可以推出，在IND-CPA上有攻击者B对于BasicPub有优势$\epsilon(k)/e(1+q_E)$

引理证明：

如何构建一个IND-CPA的对手B ：

Setup阶段：B给A一些基本参数$q, G_1, G_2, \hat{e}, n, P, P_{pub}, H_1, H_2$其中$q, G_1, G_2, \hat{e}, n, P, P_{pub}, H_2$都来自$K_{pub}$，$H_1$是由B控制的随机预言机。

H1-queries:由于A会询问随机预言机H1，所以B需要一个方法来响应这些查询。B需要维护一个列表$H_1^{list}\langle ID_i, Q_i, b_i, c_i \rangle$

> 1. 如果ID_i在H_1^{list}中，算法B直接返回$H_1(ID_i) = Q_i \in G_1^*$
> 2. 否则B，会生成一个随机硬币$coin \in \{0,1\}$
> 3. 算法B选择一个随机数$b \in Z_q^*$，

$$if\ coin = 0, \ compute\ Q_i = bP \in G_1^*. \ if\ coin = 1, compute\ Q_i = bQ_{ID} \in G_1^*$$

4. 算法B将$\langle ID_i, Q_i, b_i, c_i \rangle$增加到$H_1^{list}$，并将结果返回。

第一阶段：

1. 运行上述算法响应H1-queries，如果coin=1的话，则B对BasicPub的攻击失败。
2. 定义$d_i = b_i P_{pub} \in G_1^*$，所以私钥与公钥IDi相关联、

当A结束私钥查询时，A输出一个公钥ID_{ch}和两个消息M_0, M_1

挑战：

1. B将M_0, M_1分别使用BasicPub加密，用M_c表示选择的0或1
2. B运行算法获取$H_1(ID_{ch}) = Q$，由于ID_{ch}不在H_1^{list}中，生成随机硬币。

如果coin =0 时，B对BasicPub的攻击失败。

3. coin=1，所以$Q = bQ_{ID}$

$$\hat{e}(b^{-1}U, d_{ch}) = \hat{e}(b^{-1}U, sQ) = \hat{e}(U, sb^{-1}Q) = \hat{e}(U, sQ_{ID}) = \hat{e}(U, d_{ID}))$$

第二阶段：B继续响应A的私钥提取查询。

猜测阶段：

1. A输出猜测c' 作为c
2. B将A的输出c'作为自己的猜测答案

具体证明:

算法B在不中止的情况下,仿真效果实际攻击相同,此时有$Pr[c = c'] - 1/2 \geq \epsilon$，由于B在仿真的情况下不中止的概率为$\delta^{q_E}(1-\delta)$，由于$\delta$的取值为（0，1），当$\delta_{opt} = 1 - 1/q_E + 1$时，我们可以计算出B不重质的最小概率为$1/e(1+q_E)$。所以B的优势至少为$\epsilon(k)/e(1+q_E)$

引理2:

1. H_2是一个从$G_2\ to\ \{0,1\}^n$的随机预言机。
2. 在IND-CPA问题上，A对BasicPub有优势$\epsilon(k)$
3. 假设A能够对H2进行q_{H_2}次查询

   由上述3条可以推出存在算法B对由$\mathcal{G}$生成的BDH问题有优势$2\epsilon(k)/q_{H_2}$

引理证明:

B收到由$\mathcal{G}$生成的BDH的参数$\langle q, G_1, G_2, \hat{e} \rangle$和一组随机的实例$\langle P, aP, bP, cP \rangle = \langle P, P_1, P_2, P_3 \rangle$

Setup阶段：B将$P_{pub} = P_1\ and\ Q_{ID} = P_2$，构造了BasicPub的公钥$K_{pub} = \langle q, G_1, G_2, \hat{e}, n, P, P_{pub}, Q_{ID}, H_2 \rangle$。其中H2由B控制。

H2-queries: 面对A的询问，B需要H_2^{list}来存放一系列的二元组$\langle X_i, H_i \rangle$

1. 如果查询Xi已经出现过，那么返回$H_2(X_i) = H_i$
2. 否则，B将返回一个随机串$H_i \in \{0,1\}^n$，并将$\langle X_i, H_i \rangle$添加到H_2^{list}，并将随机串返回。

挑战：

A输出两个消息串M_0, M_1，B随机选取一个字符串$R \in \{0,1\}^n$，定义密文$C = \langle P_3, R \rangle$，C的解密为$R \oplus H_2(\hat{e}(P_3, d_{ID})) = R \oplus H_2(D))$

$$P_3 = cP = abP, d_{ID} = sQ_{ID}, Q_{ID} = P_2 = bP, \hat{e}(P_3, d_{ID}) = \hat{e}(P,P)^{abc} = D$$

猜测：

A输出它的猜测$c' \in \{0,1\}$。B在H_2^{list}里面选择一个随机的元组$\langle X_j, H_j \rangle$,并将X作为结果。

具体证明：

算法B模拟了攻击者A在真实攻击情况下的环境，下面证明算法B输出正确答案D的概率至少为$2\epsilon / q_{H_2} \Leftrightarrow$

证明$Pr[\mathcal{H}] \geq 2\epsilon$，$\mathcal{H}$表示A向B发出H2查询请求的事件。

> 1. 用递归方法证明在仿真情况和真实攻击中的$Pr[\mathcal{H}]$是一样的，假设前l次的查询，都有$Pr[\mathcal{H}_{l-1}]$是一样的。
>
> $$Pr[\mathcal{H}_{l-1}] = Pr[\mathcal{H}_l | \mathcal{H}_{l-1}] Pr[\mathcal{H}_{l-1}] + Pr[\mathcal{H}_l | \neg \mathcal{H}_{l-1}] Pr[\neg \mathcal{H}_{l-1}]$$
>
> 2. 在真实的攻击中，我们有$Pr[\mathcal{H}] \geq 2\epsilon$
>
>   我们知道如果A从来没有发出H2请求，那么C解密与A独立。有，$Pr[c = c' | \neg \mathcal{H}] = 1/2$
>
>   根据A的定义，我们知道有，$|Pr[c = c'] - 1/2| \geq \epsilon$
>
> $$Pr[c = c'] = Pr[c = c' | \neg \mathcal{H}] Pr[\neg \mathcal{H}] + Pr[c = c' | \mathcal{H}] + Pr[\mathcal{H}]$$
> $$\leq Pr[c = c' | \neg \mathcal{H}] Pr[\neg \mathcal{H}] + Pr[\mathcal{H}] = \frac{1}{2} Pr[\neg \mathcal{H}] + Pr[\mathcal{H}] = \frac{1}{2} + \frac{1}{2} Pr[\mathcal{H}]$$
> $$Pr[c = c'] \geq Pr[c = c' | \neg \mathcal{H}] Pr[\neg \mathcal{H}] = \frac{1}{2} - \frac{1}{2} Pr[\mathcal{H}]$$
>
>   所以有$\epsilon \leq |Pr[c = c'] - \frac{1}{2}| \leq \frac{1}{2} Pr[\mathcal{H}]$

具有选择密文安全的IBE

Fujisaki-Okamoto Transformation（藤崎-冈本转化法）：

1. 定义E为一个概率性的公钥加密方案，用$E_{pk}(M; r)$表示用随机数r和公钥pk，加密M。

$$E_{pk}^{hy}(M) = \langle E_{pk}(\sigma; H_3(\sigma, M)), H_4(\sigma) \oplus M \rangle$$

2. 如果E是一个单项加密方案，那么有E^{hy}是选择密文安全的。

我们可以将BasicIdet应用FO Transformation转化到FullIdent得到IND-ID-CCA安全的方案。

FullIdent：

Setup:给定一个安全参数k

1. 运行$\mathcal{G}$，$\mathcal{G}(1^k) \rightarrow < q, G_1, G_2, \hat{e} >, P$是G1的生成元。
2. 选择随机数$s \in Z_q^*$，设置公钥$P_{pub} = sP$。
3. 选择加密哈希函数H1、H2、H3和H4。
   $H_1 : \{0,1\}^* \rightarrow G_1^*, H_2 : G_2 \rightarrow \{0,1\}^n, H_3 : \{0,1\}^n \times \{0,1\}^n \rightarrow Z_q^*, H_4 : \{0,1\}^n \rightarrow \{0,1\}^n$
   密文空间$C = G_1^* \times \{0,1\}^n$。系统参数有$params =< q, G_1, G_2, \hat{e}, n, P, P_{pub}, H_1, H_2, H_3, H_4 >$

Extract:给定字符串ID

1. $ID \in \{0,1\}^*$，计算$Q_{ID} = H_1(ID) \in G_1^*$
2. $d_{ID} = sQ_{ID}$，其中s就是主密钥

Encrypt:选择随机数$\sigma \in \{0,1\}^n$,$set\ r = H_3(\sigma, M)$

$$C =< rP, \sigma \oplus H_2(g_{ID}^r), M \oplus H_4(\sigma) >, where\ g_{ID} = \hat{e}(Q_{ID}, P_{pub}) \in G_2^*$$

Decrypt:已知私钥d

解密明文:

$$C = \langle U, V, W \rangle,\ compute\ \sigma = V \oplus H_2(\hat{e}(d_{ID}, U))$$
$$\hat{e}(d_{ID}, U) = \hat{e}(sQ_{ID}, rP) = \hat{e}(Q_{ID}, P)^{sr} = \hat{e}(Q_{ID}, P_{pub})^r = g_{ID}^r$$
$$compute\ M = W \oplus H_4(\sigma)$$

身份验证:

$$Compute\ r = H_3(\sigma, M)$$
$$Test\ U = rP$$

身份验证结果如果不符合，则不会输出M。

安全性证明：证明FullIdent方案是选择密文安全的（IND-ID-CCA），假设BDH在由$\mathcal{G}$生成的群上是困难的。

一个定理：

1. 假设一个IND-ID-CCA的攻击者A对于方案FullIdent有优势$\epsilon(k)$
2. 假设A能够进行q_E次的私钥提取询问和q_D次的解密询问，以及q_{H_2}、q_{H_3}、q_{H_4}次的哈希函数询问。

   那么存在一个算法B对BDH具有优势:

$$Adv_{\mathcal{G},B}(k) \geq 2FO_{adv}(\frac{\epsilon(k)}{e(1+q_E+q_D)}, q_{H_3}, q_{H_4}, q_D)/q_{H_2},$$
$$t_1(k) \leq FO_{time}(t(k), q_{H_4}, q_{H_3})$$

3.

另一个定理（藤崎-冈本）：

1. 假设在IND-CCA上攻击者A对于$BasicPub^{hy}$有优势$\epsilon(k)$
2. A算法运行$t(k)$时间，能够够进行q_D次的解密询问，以及q_{H_3}、q_{H_4}次的哈希函数询问

   那么在IND-CPA上攻击者B对BasicPub有优势:

$$\epsilon_1(k) \geq FO_{adv}(\epsilon(k), q_{H_A}, q_{H_3}, q_D) = \frac{1}{2(q_{H_4}+q_{H_3})}[(\epsilon(k)+1)(1-2/q)^{q_D} - 1],$$
$$t_1(k) \leq FO_{time}(t(k), q_{H_4}, q_{H_3}) = t(k) + O((q_{H_4}+q_{H_3}) \cdot n)$$

引理6：

1. 假设一个IND-ID-CCA的攻击者A对于方案FullIdent有优势$\epsilon(k)$
2. 假设A能够进行q_E次的私钥提取询问和q_D次的解密询问

   那么在IND-CCA上攻击者B对于$BasicPub^{hy}$至少有优势$\frac{\epsilon(k)}{e(1+q_E+q_D)}$，运行时间和算法A的运行时间同级。

引理证明:

Setup阶段：B给A一些基本参数$q, G_1, G_2, \hat{e}, n, P, P_{pub}, H_1, H_2, H_3, H_4$其中$q, G_1, G_2, \hat{e}, n, P, P_{pub}, H_2, H_3, H_4$都来自$K_{pub}$，$H_1$是由B控制的随机预言机。

H1-queries:由于A会询问随机预言机H1，所以B需要一个方法来响应这些查询。B需要维护一个列表$H_1^{list}\langle ID_i, Q_i, b_i, c_i \rangle$

1. 如果ID_i在H_1^{list}中，算法B直接返回$H_1(ID_i)=Q_i\in G_1^*$
2. 否则B，会生成一个随机硬币$coin\in\{0,1\}$
3. 算法B选择一个随机数$b\in Z_q^*$，

$$if\ coin=0,\ compute\ Q_i=bP\in G_1^*.\ if\ coin=1, compute\ Q_i=bQ_{ID}\in G_1^*$$

4. 算法B将$\langle ID_i,Q_i,b_i,c_i\rangle$增加到$H_1^{list}$，并将结果返回。

第一阶段：

1. 运行上述算法响应H1-queries，如果coin=1的话，则B对$BasicPub^{hy}$的攻击失败。
2. 定义$d_i=b_iP_{pub}=sQ_i\in G_1^*$，所以私钥与公钥IDi相关联、
3. 将密文设置为$C_i'=\langle b_iU_i,V_i,W_i\rangle$，并使用下面的方法进行解密

$$\hat{e}(b_iU_i,d_{ID})=\hat{e}(b_iU_i,sQ_{ID})=\hat{e}(U_i,b_isQ_{ID})=\hat{e}(U_i,sQ_i)=\hat{e}(U_i,d_i)$$

4.

当A结束私钥查询和解密查询时，A输出一个公钥ID_{ch}和两个消息M_0,M_1

挑战：

1. B将M_0,M_1分别使用$BasicPub^{hy}$加密，用M_c表示选择的0或1
2. B运行算法获取$H_1(ID_{ch})=Q$，由于ID_{ch}不在H_1^{list}中，生成随机硬币。

如果coin =0 时，B对$BasicPub^{hy}$的攻击失败。

3. coin=1，所以$Q=bQ_{ID}$

$$\hat{e}(b^{-1}U,d_{ch})=\hat{e}(b^{-1}U,sQ)=\hat{e}(U,sb^{-1}Q)=\hat{e}(U,sQ_{ID})=\hat{e}(U,d_{ID}))$$

4. $C'=\langle b^{-1}U,V,W\rangle$是$M_c$在FullIdent方案下用$ID_{ch}$加密的结果，

第二阶段：B继续响应A的私钥提取查询和解密查询。

猜测阶段：

1. A输出猜测c' 作为c
2. B将A的输出c'作为自己的猜测答案

具体证明:

算法B在不中止的情况下,仿真效果实际攻击相同,此时有$Pr[c=c']-1/2\geq\epsilon$，#由于B在仿真的情况下不中止的概率为$\delta^{q_E+q_D}(1-\delta)$，由于$\delta$的取值为（0，1），当$\delta_{opt}=1-1/q_E+q_D+1$时，我们可以计算出B不重质的最小概率为$1/e(1+q_E+q_D)$。所以B的优势至少为$\epsilon(k)/e(1+q_E+q_D)$#

算法B在仿真实验中可能中止的三个原因（定义为事件）：

1. E_1 :A在阶段1或者阶段2发出私钥请求查询，coin=0的情况导致算法B中止。
2. E_2 :A选择公钥ID_{ch}，coin=1的情况导致算法B中止。
3. E_3 :阶段2中A发出解密查询时，如果Ci等于C的话，算法B会中止。

放宽哈希函数要求：

将$\{0,1\}^*$先散列到一些集合A上，然后使用确定的编码函数，将A映射到G_1^*

合适的编码函数L：$A \rightarrow G_1^*$的要求：

> 1. 可计算的，存在有效的算法计算$L(x)$
> 2. $l-to-1$，对于$y \in G_1^*$，y在L中的原像有固定的尺寸l，有$|L^{-1}(y)|=l$。
> 3. 可作为样本的，存在有效的随机函数$L_S(y)$

IBE using the Weil pairing

Weil pairing 的性质:

1. p>3 满足p=2 mod 3；q是p+1的素因子
2. E是由F_P上的公式$y^2=x^3+1$定义的椭圆曲线

> Fact 1 . $E(F_p)$包含p+1个点。O表示无穷远点，P是一个G1的生成元。
>
> Fact 2 . x与y一一对应
>
> Fact 3 . $Q \in E(F_p), \phi(Q)$线性独立，$\phi(Q) \in E(F_{P^2}), \phi(Q) \notin E(F_{p^2})$
>
> Fact 4 . 生成一个$Z_q \times Z_q$的群，定为$E[q]$

$e: E[q] \times E[q] \rightarrow G_2$是退化的，我们定义Weil Pairing为

$$\hat{e}: G_1 \times G_1 \rightarrow G_2$$
$$\hat{e}(P,Q)=e(P,\phi(Q))$$

BDH参数生成器$\mathcal{G}$：

1. q,p满足一下条件

> p=2 mod 3
>
> q 整除 p+1
>
> q^2 不整除p+1。记作$p=lq+1$

2. G1,G2和e，如Weil pairing定义

编码函数：

1. 找到一个$H_1:\{0,1\}^n \rightarrow A$和一个编码函数$L: A \rightarrow G_1^*$
2. 其中集合A就是Fp