[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
*H04L 9/30* (2006.01)
*H04L 9/32* (2006.01)

# [12] 发明专利申请公布说明书

[21] 申请号 200710122566.1

[43] 公开日 2008 年 4 月 23 日

[11] 公开号 CN 101166088A

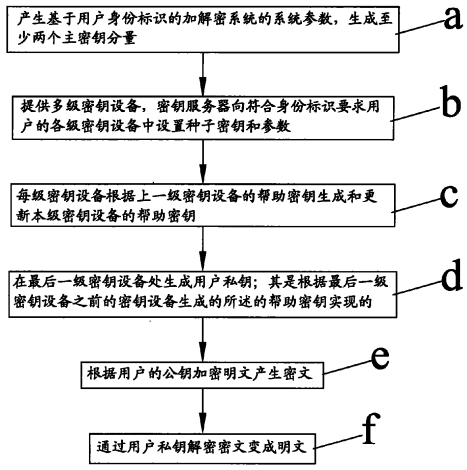
[22] 申请日 2007.9.27
[21] 申请号 200710122566.1
[71] 申请人 航天信息股份有限公司
地址 100097 北京市海淀区杏石口路甲 18 号
[72] 发明人 张庆胜 程登峰 丁 瑶 王 磊
[74] 专利代理机构 北京科龙寰宇知识产权代理有限责任公司
代理人 孙皓晨
权利要求书 3 页 说明书 11 页 附图 3 页

[54] 发明名称

基于用户身份标识的加解密方法

[57] 摘要

本发明为一种基于用户身份标识的加解密方法，其包括的步骤为：步骤 a：产生基于用户身份标识的加解密系统的系统参数，生成至少两个主密钥分量；步骤 b：提供多级密钥设备，密钥服务器向符合身份标识要求用户的各级密钥设备中设置种子密钥和参数；步骤 c：每级密钥设备根据上一级密钥设备的帮助密钥生成和更新本级密钥设备的帮助密钥；步骤 d：在最后一级密钥设备处生成用户私钥；其是根据最后一级密钥设备之前的密钥设备生成的所述的帮助密钥实现的；步骤 e：根据用户的公钥加密明文产生密文；步骤 f：通过用户私钥解密密文变成明文。 实现了避免单一主密钥被破解，使整个系统被破解的风险；保证了用户私钥的频繁更新，减轻了密钥服务器的负担；计算量小，存贮空间也小。



1、一种基于用户身份标识的加解密方法，其特征在于，其包括的步骤为：

步骤 a：产生基于用户身份标识的加解密系统的系统参数，生成至少两个主密钥分量；

步骤 b：提供多级密钥设备，密钥服务器向符合身份标识要求用户的各级密钥设备中设置种子密钥和参数；

步骤 c：每级密钥设备根据上一级密钥设备的帮助密钥生成和更新本级密钥设备的帮助密钥；

步骤 d：在最后一级密钥设备处生成用户私钥；其是根据最后一级密钥设备之前的密钥设备生成的所述的帮助密钥实现的；

步骤 e：根据用户的公钥加密明文产生密文；

步骤 f：通过用户私钥解密密文变成明文。

2、根据权利要求 1 所述的基于用户身份标识的加解密方法，其特征在于，所述的步骤 a：产生基于用户身份标识的加解密系统的系统参数，生成至少两个主密钥分量，其包括的步骤为：

步骤 a1：选择一至少 512 比特长的大素数 p 和满足 BDH 安全假设的超奇异椭圆曲线 E/GF(p)，其中，P 是曲线 E 的基点，基点的阶是大素数 q，q 的长度至少为 160 比特，定义 q 阶加法循环群 G1、q 阶乘法循环群 G2，以及双线性配对 $\hat{e}$: $G1\times G1\rightarrow G2$;

步骤 a2：定义 hash 函数 H2：$GF(p^2)\rightarrow\{0，1\}^n$，及一用于将用户身份 ID 映射到 $G1^*$上元素的函数 H1，其中，$G1^*$表示 G1 去除 *O* 元素；

步骤 a3：确定明文空间 M、密文空间是 C，其中，$M=\{0，1\}^n$、$C=G1^*\times\{0，1\}^n$;

步骤 a4：根据实际需要，确定用户私钥更新的级数 m，随机选择 m 个主密分量：$s_1$，$s_2$，…$s_m\in Z_q^*$，并令 $P_{pub}=(s_1+s_2+\ldots+s_m)P$;

步骤 a5：保密各个主密钥分量 $s_i$，其中 i=1，2，…，m，公开公共参数 param=<q，G1，G2，$\hat{e}$，n，P，$P_{pub}$，H1，H2>;

步骤 a6：执行步骤 b。

3、根据权利要求 1 或 2 所述的基于用户身份标识的加解密方法，其特征在于，所述的步骤 b：提供多级密钥设备，密钥服务器向符合身份标识要求用户的各级

密钥设备中设置种子密钥和参数，其包括的步骤为：

步骤 b1：计算用户公钥 $Q_{ID}$，其中 $Q_{ID}$=H1(ID)；

步骤 b2：各级密钥设备分别设置种子密钥：$s_i Q_{ID}$，其中 i=1，2，…，m；

步骤 b3：将系统参数 p 设置到各级密钥设备中。

4、根据权利要求 1 所述的基于用户身份标识的加解密方法，其特征在于，所述的步骤 c：每级密钥设备根据上一级密钥设备的帮助密钥生成和更新本级密钥设备的帮助密钥，是在与密钥服务器脱线的情况下完成的，第 i 级中间设备的帮助密钥为：

$$HK = (\sum_{j=1}^{i} s_j) Q_{ID} - \sum_{j=1}^{i} r_j P_{t_j}$$

相关的中间计算结果为：$r_i P$

其中，j=1，2，…，m-1，$r_j$ 为各级密钥设备产生的随机数，$P_{tj}$ 为各级密钥设备根据相应的时间段，经过 hash 函数 H1 运算得到的 G1*中的元素，

$$P_{t_1} = H1(ID \| T_1(time))$$

$$P_{t_2} = H1(ID \| T_1(time) \| T_2(time))$$

…

$$P_{t_{m-1}} = H1(ID \| T_1(time) \| T_2(time) \| \mathrm{L}\ T_{m-1}(time))$$

其中，$T_j$(time)表示对时间进行相应的运算，取出其中相应的时间段的信息，标识本级密钥更新一次的时间，其中，第 i 级密钥设备帮助密钥更新的时间大于第 i－1 级密钥设备帮助密钥更新的时间。

5、根据权利要求 1 或 4 所述的基于用户身份标识的加解密方法，其特征在于，所述的密钥设备为硬件智能卡或 UsbKey。

6、根据权利要求 1 或 4 所述的基于用户身份标识的加解密方法，其特征在于，所述的步骤 d：在最后一级密钥设备处生成用户私钥；其是根据最后一级密钥设备之前的密钥设备生成的所述的帮助密钥实现的；用户私钥为：

$$d_{ID} = (\sum_{j=1}^{m} s_j) Q_{ID} - \sum_{j=1}^{m} r_j P_{t_j}$$

相关的中间计算结果为：

$$r_i P, (i = 1, 2, \mathrm{L}\ , m)$$

其中：

$$P_{t_1} = H1(ID \| T_1(time))$$

$$P_{t_2} = H1(ID \| T_1(time) \| T_2(time))$$

…

$$P_{t_{m-1}} = H1(ID \| T_1(time) \| T_2(time) \| \mathrm{L}\ T_{m-1}(time))$$

$$P_{t_m} = H1(ID \| T_1(time) \| T_2(time) \| \mathrm{L}\ T_{m-1}(time) \| T_m(time));$$

$T_j(time)$表示对时间进行相应的运算，取出其中相应的时间段的信息，标识本级密钥更新一次的时间。

7、根据权利要求 1 所述的基于用户身份标识的加解密方法，其特征在于，所述的步骤 e：根据用户的公钥加密明文产生密文，其包括的步骤为：

步骤 e1：令 $g_{ID}=\hat{e}(Q_{ID}, P_{pub})=\hat{e}(Q_{ID}, (s_1+s_2+\ldots+s_m)P)\in G2$;

步骤 e2：$C=<rP, rP_{t_1}, rP_{t_2}, \mathrm{L}\ , rP_{t_m}, M \oplus H2(g_{ID}{}^r)>$，其中 $i=1,2,\mathrm{L}\ ,m$，

$$P_{t_1} = H1(ID \| T_1(time))$$

$$P_{t_2} = H1(ID \| T_1(time) \| T_2(time))$$

…

$$P_{t_{m-1}} = H1(ID \| T_1(time) \| T_2(time) \| \mathrm{L}\ T_{m-1}(time))$$

$$P_{t_m} = H1(ID \| T_1(time) \| T_2(time) \| \mathrm{L}\ T_{m-1}(time) \| T_m(time))$$

$T_j(time)$表示对时间进行相应的运算，取出其中相应的时间段的信息，标识本级密钥更新一次的时间。

8、根据权利要求 1 所述的基于用户身份标识的加解密方法，其特征在于，所述的步骤 f：通过用户私钥解密密文变成明文，其包括的步骤为：

步骤 f1：检查 U 是否是 E/GF(p)中的点，如果不是就拒绝该密文；

步骤 f2：计算得出明文 $M=W \oplus H2(\hat{e}(d_{ID}, U)\ \hat{e}(rP_{t_1}, r_1P)\ldots\hat{e}(rP_{t_m}, r_mP))$;

其中，加解密的一致性是由下面的等式保证的：

$\hat{e}(d_{ID}, U)\ \hat{e}(rP_{t_1}, r_1P)\ldots\hat{e}(rP_{t_m}, r_mP)=\hat{e}((s_1+s_2+\ldots+s_m)Q_{ID}, rP)=g_{ID}{}^r$

其中密文 $C=<U, V_1, V_2, \ldots, V_m, W>$。

# 基于用户身份标识的加解密方法

## 技术领域

本发明涉及的是一种加解密方法，特别涉及的是一种基于用户身份标识的加解密方法。

## 背景技术

PKI技术是一种成熟的公钥密码技术，近10年来获得了广泛的应用，如现在的网上银行，网上证券、电子商务等等都基于PKI技术，来保证数据传输的安全性。我国2004颁布的《电子签名法》也是基于PKI技术。在公钥密码技术中，用户有两把密钥，一把密钥为用户独有，称为用户私钥；一把密钥公开给大家，称为公钥，利用用户公钥就可以给该用户发送加密信息，但在PKI技术中用户公钥是一串没有意义的随机数字，因而要将公钥和标志用户身份标识的信息绑定起来，形成数字证书，才方便大家查询，一旦用户数量过多的情况，用户繁琐的数字证书管理问题成了PKI系统运行的瓶颈。

为了解决繁琐的数字证书管理问题，早在1984年，RSA公钥密码技术的发明者之一Adi Shamir教授就提出了基于身份加密（Identity-Based Encryption）的思想，IBE是基于身份加密的缩写，它的最大特点是利用标志用户身份标识的信息（如：用户的身份证号、电子邮件地址、QQ号、手机号等等）直接作为用户公钥，不采用数字证书的概念，因而避免了繁琐的数字证书管理问题。但在那时还没有具体方法在实际中实现这一思想，IBE技术成为密码学界未解决的主要问题之一。

2001年，基于椭圆曲线密码和Weil配对数学理论，斯坦福大学计算机科学技术系的教授Dan Boneh和加州大学戴维斯分院的教授Matt Franklin分别发明了具体可实施的IBE算法，该算法又简称为D.B/M.F算法。

D.B/M.F算法方案的安全性建立在CDH（Computation Diffie-Hellman）困难问题的一个变形之上，称为BDH（Bilinear Diffie-Hellman）困难问题。D.B/M.F算法的核心是使用了超奇异椭圆曲线上的一个双线性映射Weil Pairing。描述如

下：

1、设 p 是一个大素数，$p \equiv 2 \bmod 3$，并且存在大素数 q，使得 p+1 能被 q 整除，但不能被 $q^2$ 整除，记为 p=lq-1；

2、E/GF(p)是在有限域 GF(p)上构造的椭圆曲线：$y^2=x^3+1$，P 是该曲线上阶为 q 的点，也称为基点，定义加法循环群 G1 利用 P 的点积方法生成；定义乘法循环群 G2 利用 P 的乘幂的方法生成；

3、BDH 问题：对于随机的 a，b，$c \in Z_q^*$，已知(P，aP，bP，cP)来计算 $\hat{e}(P, P)^{abc} \in G2$。注意到 E/GF(p)是超奇异椭圆曲线。“$\hat{e}$”是由修改的 Weil Pairing 变来的映射，$\hat{e}$：$G1 \times G1 \rightarrow G2$，满足以下三条性质：

1）双线性性：

对于所有 P，$Q \in G1$，和所有的 a，$b \in Z$ 有：$\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$，其中 Z 是整数集；

2）非退化性：如果 P 是 G1 的生成元，则 $\hat{e}(P, P) \in GF(p^2)^*$ 是 G2 的生成元。在群 G 中如果存在有 $P \in G$ 使得 $G=\{P^k \mid k \in Z\}$，则称 G 为循环群，称 P 为 G 的生成元；

3）可计算性：对于任何 P，$Q \in G1$，存在一个有效的算法来计算 $\hat{e}(P, Q) \in G2$。

Weil Pairing 的存在本来是对超奇异椭圆曲线上的密码体制的威胁，也就是说，G 中的离散对数问题可轻易地简化成 $GF(p^2)^*$中的离散对数问题。E/GF(p)是超奇异椭圆曲线，所以为使 G 中的离散对数问题难解，必须要求 p 的长度至少为 512 比特。

D.B/M.F 算法方案分为四个执行阶段：系统参数建立阶段(Setup)、用户私钥生成阶段(Extract)、加密阶段(Encrypt)，以及解密阶段(Decrypt)，其中，

所述的系统参数建立阶段，包括的内容是：

可信第三方密钥服务器进行如下步骤产生 D.B/M.F 的系统参数：

1）首先选择一个至少 512 比特长的大素数 p，找一条满足 BDH 安全假设的超奇异椭圆曲线 E/GF(p)，P 是曲线 E 的基点，基点的阶是大素数 q，q 的长度至少为 160 比特，定义 q 阶加法循环群 G1、q 阶乘法循环群 G2，以及双线性配对 $\hat{e}$：$G1 \times G1 \rightarrow G2$；

2）定义 hash 函数 H2：$GF(p^2) \rightarrow \{0, 1\}^n$，及一个用于将用户身份 ID 映射到

$G1^*$上元素的函数 H1，这里的 $G1^*$表示 G1 去除 $O$ 元素；

3）明文空间是：$M=\{0, 1\}^n$，密文空间是 $C= G1^* \times \{0, 1\}^n$；

4）随机选择 $s \in Z_q^*$作为系统主密钥(master key)，并令 $P_{pub}=sP$；

5）保密主密钥 s，公开公共参数 param=<q，G1，G2，ê，n，P，$P_{pub}$，H1，H2>。

所述用户私钥生成阶段，其包括的步骤为：

身份标识为 $ID\in\{0, 1\}^n$ 的用户向密码服务器申请自己的解密私钥，密码服务器需要做以下工作：

1）计算用户公钥 $Q_{ID}$：$Q_{ID}=H1(ID)$

2）产生用户私钥 $d_{ID}$：$d_{ID}=sQ_{ID}$。

所述的加密阶段，其包括的步骤为：

随机选取 $r\in Z_q^*$，用 $Q_{ID}$ 加密明文 M，产生密文 C：

1）令 $g_{ID}=\hat{e}(Q_{ID}, P_{pub}) \in G2$；

2）$C=<rP, M \oplus H2(g_{ID}^r)>$

所述的解密阶段，其包括的步骤为：

用 $d_{ID}$ 解密密文 C=<U，V>，获取明文 M：

1）检查 U 是否是 E/GF(p)中的点，如果不是就拒绝该密文；

2）计算得出 $M=V \oplus H2(\hat{e}(d_{ID}, U))$；

其中，加解密的一致性是由下面的等式保证的：

$\hat{e}(d_{ID}, U)= \hat{e}(sQ_{ID}, rP)= \hat{e}(Q_{ID}, sP)^r=\hat{e}(Q_{ID}, P_{pub})^r=g_{ID}^r$。

该方案根据椭圆曲线密码和双线性映射理论实现具体可实施的 IBE 方案，解决了密码学的一大难题。但这个系统中也存在一些问题：（1）主密钥一旦被破解，黑客可计算出任何人的私钥，则系统即被攻破；（2）密钥管理的问题没有很好的解决。IBE 技术和 PKI 技术相比的优点是无证书管理问题，但缺点是密钥管理没有 PKI 那样方便、安全。PKI 技术的私钥可以在客户端自行随机产生，为 CA 服务器端所不知，然后计算生成公钥，结合着自己的身份等相关信息到 CA 申请数字证书；而 D.B/M.F 的 IBE 方案的用户私钥由密钥服务器统一计算并分发，D.B/M.F 方案在密钥更新过程中要求用户身份信息不被改变，采用将用户身份信息和时间信息连接，诸如 ID||T 的形式，作为用户公钥，这样经过一定的时间密钥就要更新一次，如果设定 T 是年，则用户私钥要每年更新一次，为了减少用户私

钥被破解的风险，时间段 T 要尽量短一些，比如可设置 T 为天，用户私钥每天更新一次，这样即使黑客盗取了用户私钥，他也只能解密当天的加密信息，但这样频繁地通过密码服务器更新用户私钥，又极大地增加了密码服务器的负担。

以手机为例：保密手机可以通过 IBE 技术来实现，每个手机号都是唯一的，可作为标识用户的身份信息，在密码系统中，作为公开密钥，用户的手机号不会轻易的变化。出于密码安全需要，要求用户的私钥每天都更新一次，如果采用 D.B/M.F 的 IBE 算法，用户每天都要和密码服务器交互，这样势必密码服务器的负荷过重，就不能体现 IBE 技术公钥管理方便，计算量小的优越性。

为解决上述问题，本发明创作者经过长时间的研究和实践终于获得了本创作。

发明内容

本发明的目的在于，提供一种基于用户身份标识的加解密方法，用以克服上述缺陷。

为实现上述目的，本发明采用的技术方案在于，提供一种基于用户身份标识的加解密方法，其包括的步骤为：

步骤 a：产生基于用户身份标识的加解密系统的系统参数，生成至少两个主密钥分量；

步骤 b：提供多级密钥设备，密钥服务器向符合身份标识要求用户的各级密钥设备中设置种子密钥和参数；

步骤 c：每级密钥设备根据上一级密钥设备的帮助密钥生成和更新本级密钥设备的帮助密钥；

步骤 d：在最后一级密钥设备处生成用户私钥；其是根据最后一级密钥设备之前的密钥设备生成的所述的帮助密钥实现的；

步骤 e：根据用户的公钥加密明文产生密文；

步骤 f：通过用户私钥解密密文变成明文；

较佳的，所述的步骤 a：产生基于用户身份标识的加解密系统的系统参数，生成至少两个主密钥分量，其包括的步骤为：

步骤 a1：选择一至少 512 比特长的大素数 p 和满足 BDH 安全假设的超奇异椭圆曲线 E/GF(p)，其中，P 是曲线 E 的基点，基点的阶是大素数 q，q 的长度至少为 160 比特，定义 q 阶加法循环群 G1、q 阶乘法循环群 G2，以及双线性配对 ê:

$G1\times G1\rightarrow G2$;

步骤 a2：定义 hash 函数 H2：$GF(p^2)\rightarrow\{0,1\}^n$，及一用于将用户身份 ID 映射到 $G1^*$上元素的函数 H1，其中，$G1^*$表示 G1 去除 $O$ 元素；

步骤 a3：确定明文空间 M、密文空间是 C，其中，$M=\{0,1\}^n$、$C=G1^*\times\{0,1\}^n$；

步骤 a4：根据实际需要，确定用户私钥更新的级数 m，随机选择 m 个主密分量：$s_1$，$s_2$，…$s_m\in Z_q^*$，并令 $P_{pub}=(s_1+s_2+\ldots+s_m)P$；

步骤 a5：保密各个主密钥分量 $s_i$，其中 i=1，2，…，m，公开公共参数 param=<q，G1，G2，ê，n，P，$P_{pub}$，H1，H2>；

步骤 a6：执行步骤 b；

较佳的，所述的步骤 b：提供多级密钥设备，密钥服务器向符合身份标识要求用户的各级密钥设备中设置种子密钥和参数，其包括的步骤为：

步骤 b1：计算用户公钥 $Q_{ID}$，其中 $Q_{ID}=H1(ID)$；

步骤 b2：各级密钥设备分别设置种子密钥：$s_i Q_{ID}$，其中 i=1，2，…，m；

步骤 b3：将系统参数 p 设置到各级密钥设备中；

步骤 c：每级密钥设备根据上一级密钥设备的帮助密钥生成和更新本级密钥设备的帮助密钥，是在与密钥服务器脱线的情况下完成的，第 i 级中间设备的帮助密钥为：

$$HK=(\sum_{j=1}^{i}s_j)Q_{ID}-\sum_{j=1}^{i}r_jP_{t_j}$$

相关的中间计算结果为：$r_iP$

其中，j=1，2，…，m-1，$r_j$ 为各级密钥设备产生的随机数，$P_{tj}$ 为各级密钥设备根据相应的时间段，经过 hash 函数 H1 运算得到的 G1*中的元素，

$$P_{t_1}=H1(ID\|T_1(time))$$

$$P_{t_2}=H1(ID\|T_1(time)\|T_2(time))$$

…

$$P_{t_{m-1}}=H1(ID\|T_1(time)\|T_2(time)\|\mathrm{L}\ T_{m-1}(time))$$

其中，$T_j(time)$表示对时间进行相应的运算，取出其中相应的时间段的信息，标识本级密钥更新一次的时间，其中，第 i 密钥设备帮助密钥更新的时间大于第 i－1 密钥设备帮助密钥更新的时间；

较佳的，所述的密钥设备为硬件智能卡或 UsbKey;

步骤 d: 在最后一级密钥设备处生成用户私钥；其是根据最后一级密钥设备之前的密钥设备生成的所述的帮助密钥实现的；用户私钥为:

$$d_{ID} = (\sum_{j=1}^{m} s_j) Q_{ID} - \sum_{j=1}^{m} r_j P_{t_j}$$

相关的计算结果为:

$$r_i P, (i = 1, 2, \mathrm{L}\ , m)$$

其中:

$$P_{t_1} = H1(ID \| T_1(time))$$

$$P_{t_2} = H1(ID \| T_1(time) \| T_2(time))$$

…

$$P_{t_{m-1}} = H1(ID \| T_1(time) \| T_2(time) \| \mathrm{L}\ T_{m-1}(time))$$

$P_{t_m} = H1(ID \| T_1(time) \| T_2(time) \| \mathrm{L}\ T_{m-1}(time) \| T_m(time))$; $T_j$(time)表示对时间进行相应的运算，取出其中相应的时间段的信息，标识本级密钥更新一次的时间，用户私钥更新相对频繁;

步骤 e: 根据用户的公钥加密明文产生密文，其包括的步骤为:

步骤 e1: 令 $g_{ID}$=ê($Q_{ID}$， $P_{pub}$)= ê($Q_{ID}$， $(s_1+s_2+\ldots+s_m)P$)∈G2;

步骤 e2: $C = < rP, rP_{t_1}, rP_{t_2}, \mathrm{L}\ , rP_{t_m}, M \oplus H2(g_{ID}^{\ r}) >$，其中 $i = 1, 2, \mathrm{L}\ , m$，

$$P_{t_1} = H1(ID \| T_1(time))$$

$$P_{t_2} = H1(ID \| T_1(time) \| T_2(time))$$

…

$$P_{t_{m-1}} = H1(ID \| T_1(time) \| T_2(time) \| \mathrm{L}\ T_{m-1}(time))$$

$$P_{t_m} = H1(ID \| T_1(time) \| T_2(time) \| \mathrm{L}\ T_{m-1}(time) \| T_m(time))$$

$T_j$(time)表示对时间进行相应的运算，取出其中相应的时间段的信息，标识本级密钥更新一次的时间;

步骤 f: 通过用户私钥解密密文变成明文，其包括的步骤为:

步骤 f1: 检查 U 是否是 E/GF(p)中的点，如果不是就拒绝该密文;

步骤 f2: 计算得出明文 M=W ⊕ H2(ê($d_{ID}$， U) ê(r $P_{t_1}$， $r_1$P)…ê(r $P_{t_m}$， $r_m$P));

其中，加解密的一致性是由下面的等式保证的：

$\hat{e}(d_{ID}, U)\ \hat{e}(rP_{t_1}, r_1P)\ldots\hat{e}(rP_{t_m}, r_mP)= \hat{e}((s_1+s_2+\ldots+s_m)Q_{ID}, rP)= g_{ID}^{r}$

其中密文 $C=<U, V_1, V_2, \ldots, V_m, W>$。

与现有技术比较本发明的有益效果在于，首先引入了多个主密钥分量，避免单一主密钥被破解，从而导致整个系统被破解的风险；

其次采用多级密钥更新装置，既保证了用户私钥的频繁更新，又极大地减轻了密码服务器的负担；

最后本发明还具有计算量小，需要的存贮空间也小的优点。

附图说明

图 1 为本发明基于用户身份标识的加解密方法的流程图；

图 2 为本发明基于用户身份标识的加解密方法中步骤 a 的流程图；

图 3 为本发明基于用户身份标识的加解密方法中步骤 b 的流程图。

具体实施方式

以下结合附图，对本发明上述的和另外的技术特征和优点作更详细的说明。

请参阅图 1 所示，其为本发明基于用户身份标识的加解密方法的流程图；其包括得步骤为：

步骤 a：产生基于用户身份标识的加解密系统的系统参数，生成至少两个主密钥分量；

步骤 b：提供多级密钥设备，密钥服务器向符合身份标识要求用户的各级密钥设备中设置种子密钥和参数；

步骤 c：每级密钥设备根据上一级密钥设备的帮助密钥生成和更新本级密钥设备的帮助密钥；

步骤 d：在最后一级密钥设备处生成用户私钥；其是根据最后一级密钥设备之前的密钥设备生成的所述的帮助密钥实现的；

步骤 e：根据用户的公钥加密明文产生密文；

步骤 f：通过用户私钥解密密文变成明文；

其中，所述的步骤 a：产生基于用户身份标识的加解密系统的系统参数，生成至少两个主密钥分量，请参阅图 2 所示，其包括的步骤为：

步骤 a1：选择一至少 512 比特长的大素数 p 和满足 BDH 安全假设的超奇异椭圆曲线 E/GF(p)，其中，P 是曲线 E 的基点，基点的阶是大素数 q，q 的长度至少为 160 比特，定义 q 阶加法循环群 G1、q 阶乘法循环群 G2，以及双线性配对 ê: G1×G1→ G2;

步骤 a1：定义 hash 函数 H2：$GF(p^2)\rightarrow \{0,1\}^n$，及一用于将用户身份 ID 映射到 $G1^*$上元素的函数 H1，其中，$G1^*$表示 G1 去除 $O$ 元素;

步骤 a2：确定明文空间 M、密文空间是 C，其中，$M=\{0,1\}^n$、$C=G1^*\times\{0,1\}^n$;

步骤 a3：根据实际需要，确定用户私钥更新的级数 m，随机选择 m 个主密分量：$s_1$，$s_2$，…$s_m\in Z_q^*$，并令 $P_{pub}=(s_1+s_2+\ldots+s_m)P$;

步骤 a4：保密各个主密钥分量 $s_i$，其中 i=1，2，…，m，公开公共参数 param=<q，G1，G2，ê，n，P，$P_{pub}$，H1，H2>;

步骤 a5：执行步骤 b;

对于本发明的步骤 a 而言，其与现有的 D.B/M.F 算法技术对比，主要是增加了主密钥分量。

对于所述的步骤 b：提供多级密钥设备，密钥服务器向符合身份标识要求用户的各级密钥设备中设置种子密钥和参数，请参阅图 3 所示，其包括的步骤为：

步骤 b1：计算用户公钥 $Q_{ID}$，其中 $Q_{ID}=H1(ID)$;

步骤 b2：各级密钥设备分别设置种子密钥：$s_i Q_{ID}$，其中 i=1，2，…，m;

步骤 b3：将系统参数 p 设置到各级密钥设备中;

对于本发明所述的步骤 c：每级密钥设备根据上一级密钥设备的帮助密钥生成和更新本级密钥设备的帮助密钥，是在与密钥服务器脱线的情况下完成的，第 i 级中间设备的帮助密钥为：

$$HK=(\sum_{j=1}^{i}s_j)Q_{ID}-\sum_{j=1}^{i}r_jP_{t_j}$$

相关的中间计算结果为：$r_iP$

其中，j=1，2，…，m-1，$r_j$ 为各级密钥设备产生的随机数，$P_{tj}$ 为各级密钥设备根据相应的时间段，经过 hash 函数 H1 运算得到的 G1*中的元素，

$$P_{t_1}=H1(ID\,||\,T_1(time))$$

$$P_{t_2}=H1(ID\,||\,T_1(time)\,||\,T_2(time))$$

…

$$P_{t_{m-1}} = H1(ID \| T_1(time) \| T_2(time) \| \mathrm{L}\ T_{m-1}(time))$$

这里的 $T_j$(time)表示对时间进行相应的运算，取出其中相应的时间段的信息，标识本级密钥更新一次的时间，如果 $T_1$(time)=year，表示经过运算取出时间中的年份信息，表明第一级密钥设备每年更新一次，以此类推，级数越靠前的密钥设备帮助密钥更新的时间段越长，即第 i 级密钥设备帮助密钥更新的时间小于第 i－1 级密钥设备帮助密钥更新的时间；所需要的安全性也更高，所述的密钥设备采用硬件智能卡或 UsbKey，这类比较安全的硬件介质。

对于所述的步骤 d：在最后一级密钥设备处生成用户私钥；其是根据最后一级密钥设备之前的密钥设备生成的所述的帮助密钥实现的；其是出于方便使用的考虑，用户私钥可以导入到比如 PC、终端等比较不安全的设备上，因为用户私钥更新比较频繁，比如规定每天对用户私钥更新一次，如果黑客截获了用户私钥，他也只能在一天内破解用户的机密信息，第二天他所截获的用户私钥就再也没什么用处，其中所述的用户私钥为：

$$d_{ID} = (\sum_{j=1}^{m} s_j) Q_{ID} - \sum_{j=1}^{m} r_j P_{t_j}$$

相关的计算结果为：

$$r_i P, (i = 1, 2, \mathrm{L}\ , m)$$

其中：

$$P_{t_1} = H1(ID \| T_1(time))$$

$$P_{t_2} = H1(ID \| T_1(time) \| T_2(time))$$

…

$$P_{t_{m-1}} = H1(ID \| T_1(time) \| T_2(time) \| \mathrm{L}\ T_{m-1}(time))$$

$$P_{t_m} = H1(ID \| T_1(time) \| T_2(time) \| \mathrm{L}\ T_{m-1}(time) \| T_m(time));$$

$T_j$(time)表示对时间进行相应的运算，取出其中相应的时间段的信息，标识本级密钥更新一次的时间，用户私钥更新相对频繁；

对于所述的步骤 e 与步骤 f 而言，其与现有的 D.B/M.F 算法技术对比，区别主要是由于步骤 a 以及步骤 b 中增加了主密钥分量以及在密钥设备上设置种子密钥和参数的技术特征所引起的后续处理的改变，所述的步骤 e：根据用户的公钥加密明文产生密文，其包括的步骤为：

步骤 e1：令 $g_{ID}=\hat{e}(Q_{ID}, P_{pub})=\hat{e}(Q_{ID}, (s_1+s_2+\ldots+s_m)P)\in G2$；

步骤 e2：$C=<rP, rP_{t_1}, rP_{t_2}, \mathrm{L}, rP_{t_m}, M\oplus H2(g_{ID}^{r})>$，其中 $i=1,2,\mathrm{L},m$，

$$P_{t_1}=H1(ID\,\|\,T_1(time))$$

$$P_{t_2}=H1(ID\,\|\,T_1(time)\,\|\,T_2(time))$$

…

$$P_{t_{m-1}}=H1(ID\,\|\,T_1(time)\,\|\,T_2(time)\,\|\mathrm{L}\;T_{m-1}(time))$$

$$P_{t_m}=H1(ID\,\|\,T_1(time)\,\|\,T_2(time)\,\|\mathrm{L}\;T_{m-1}(time)\,\|\,T_m(time))$$

$T_j(time)$表示对时间进行相应的运算，取出其中相应的时间段的信息，标识本级密钥更新一次的时间；

对于所述的步骤 f：通过用户私钥解密密文变成明文，其包括的步骤为：

步骤 f1：检查 U 是否是 E/GF(p)中的点，如果不是就拒绝该密文；

步骤 f2：计算得出明文 $M=W\oplus H2(\hat{e}(d_{ID}, U)\,\hat{e}(rP_{t_1}, r_1P)\ldots\hat{e}(rP_{t_m}, r_mP))$；

其中，加解密的一致性是由下面的等式保证的：

$\hat{e}(d_{ID}, U)\,\hat{e}(rP_{t_1}, r_1P)\ldots\hat{e}(rP_{t_m}, r_mP)=\hat{e}((s_1+s_2+\ldots+s_m)Q_{ID}, rP)=g_{ID}^{r}$

其中密文 C=<U，$V_1$，$V_2$，…，$V_m$，W>。每一级的密钥更新设备只储存一个帮助密钥，第 i 级密钥更新设备储存 i 个有关运算结果。

为了使本领域技术人员更清楚，我们仍以保密手机为例，采用本发明的方法，通过三个主密钥分量($s_1$、$s_2$ 和 $s_3$)，结合发明内容步骤 a 中的参数 P （P 是曲线 E 的基点），将 $s_1P$ 设置到第一级密钥设备，$s_2P$ 设置在到第二级密钥设备，将 $s_3P$ 设置到用户手机，这里的密钥设备采用硬件智能卡或 Usb Key 的硬件介质形式，设置后的密钥生成和更新都是在和密钥服务器脱线的情况下进行的，按时间分三级，其中：

$T_1(time)$=year，$T_2(time)$=month，$T_3(time)$=day。

也就是说，第一级密钥设备的帮助密钥自行每年更新一次，第二级密钥设备的帮助密钥依赖于第一级帮助密钥每月更新一次，用户私钥依赖于第二级帮助密钥每天更新一次。

综上，本发明引入多个主密钥分量，避免单一主密钥被破解，从而导致整个系统被破解的风险；在用户密钥管理中，采用按时间分级的模式，既保持了用户私钥在较短时间得以更新，从而最大限度地避免了密钥被破解的风险，而且只有

种子密钥和参数设置需要和密钥服务器交互，其他的密钥更新都是在密钥服务器脱线的情况下完成的，所以又极大地减轻了密钥服务器的负担；另外，本发明的每一级帮助密钥的更新只需要进行一次关键的点积运算，每一级的中间结果也只需要进行一次关键的点积运算；因此具有计算量小，需要的储存空间也小的特点。

以上所述仅为本发明的较佳实施例，对本发明而言仅仅是说明性的，而非限制性的。本专业技术人员理解，在本发明权利要求所限定的精神和范围内可对其进行许多改变，修改，甚至等效，但都将落入本发明的保护范围内。

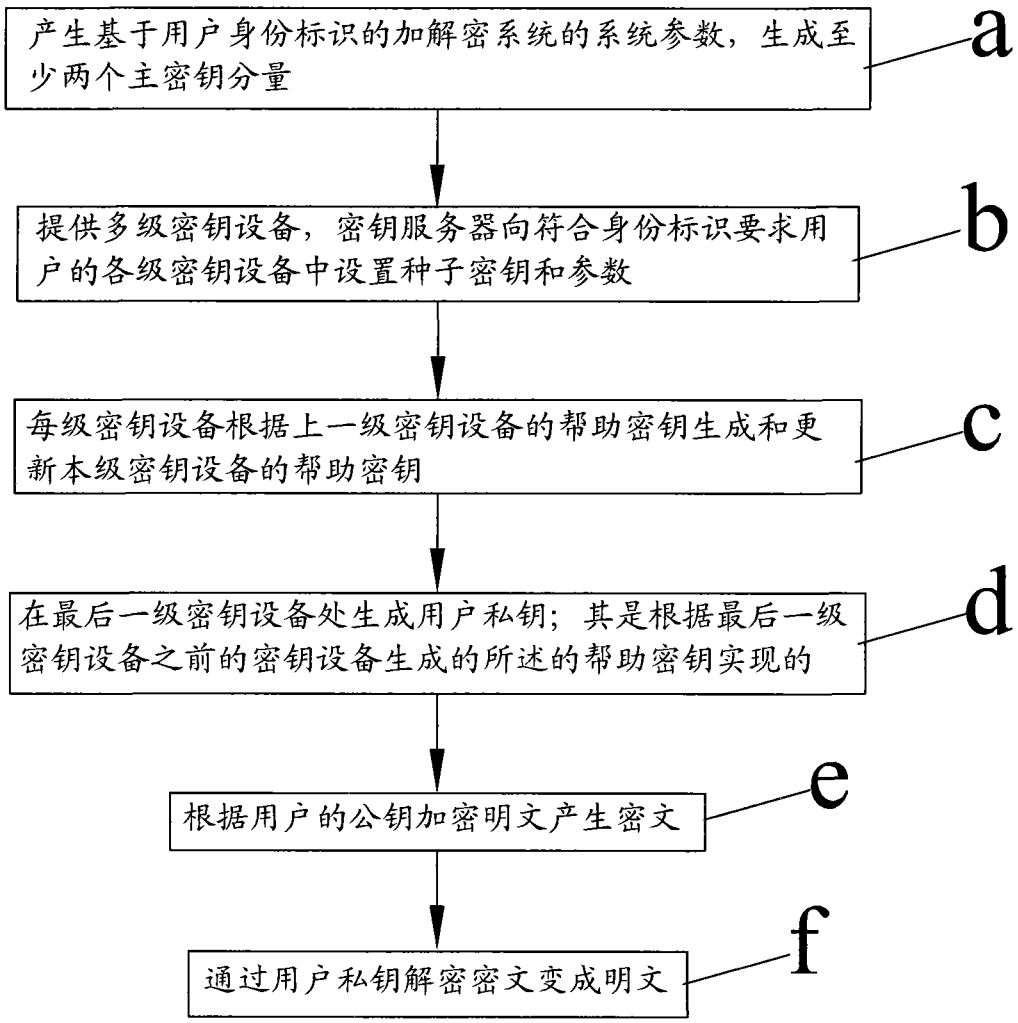

产生基于用户身份标识的加解密系统的系统参数，生成至少两个主密钥分量
a
提供多级密钥设备，密钥服务器向符合身份标识要求用户的各级密钥设备中设置种子密钥和参数
b
每级密钥设备根据上一级密钥设备的帮助密钥生成和更新本级密钥设备的帮助密钥
c
在最后一级密钥设备处生成用户私钥；其是根据最后一级密钥设备之前的密钥设备生成的所述的帮助密钥实现的
d
根据用户的公钥加密明文产生密文
e
通过用户私钥解密密文变成明文
f

图1

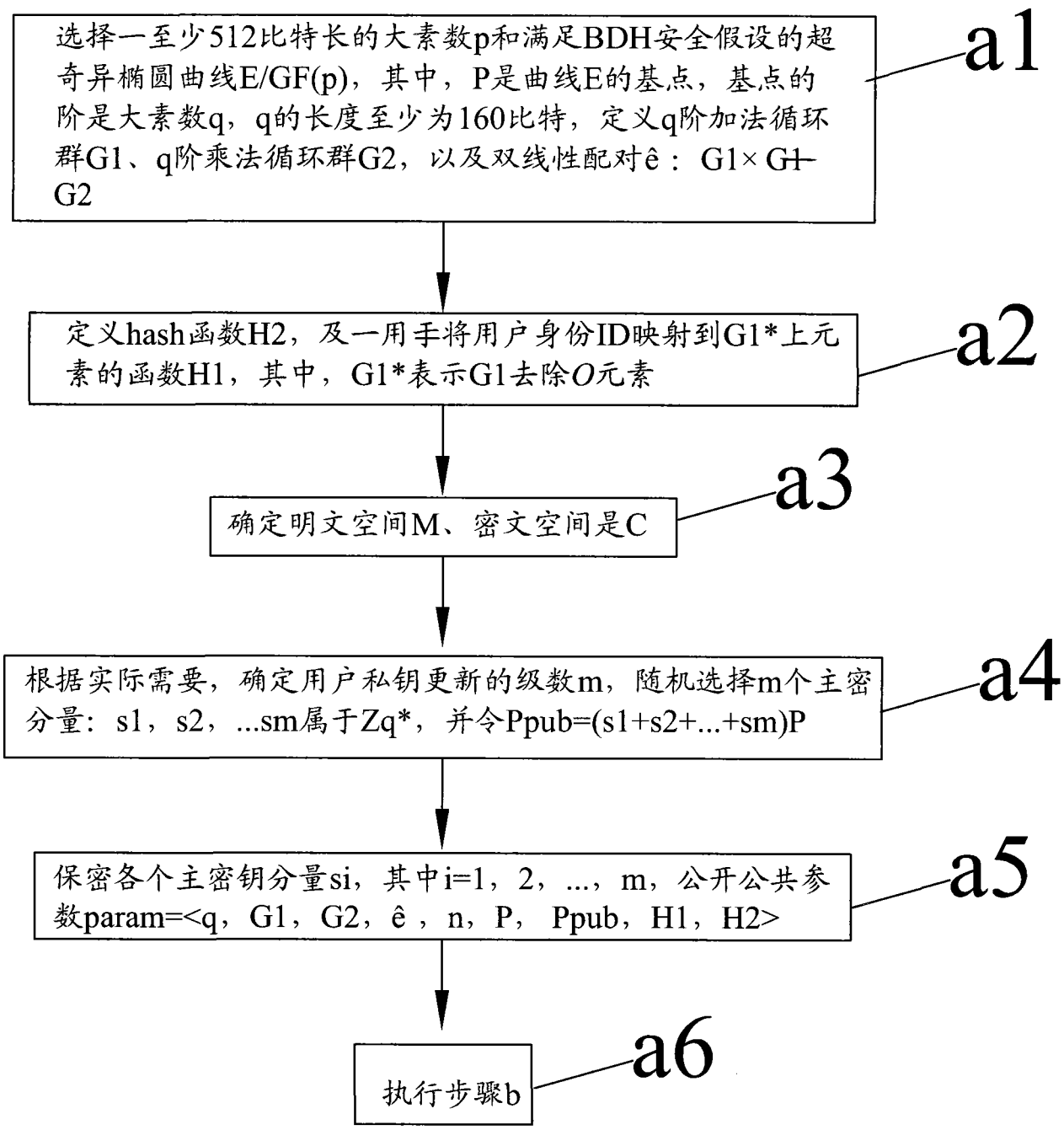

选择一至少512比特长的大素数p和满足BDH安全假设的超奇异椭圆曲线E/GF(p)，其中，P是曲线E的基点，基点的阶是大素数q，q的长度至少为160比特，定义q阶加法循环群G1、q阶乘法循环群G2，以及双线性配对ê：G1×G1→G2
a1
定义hash函数H2，及一用于将用户身份ID映射到G1*上元素的函数H1，其中，G1*表示G1去除O元素
a2
确定明文空间M、密文空间是C
a3
根据实际需要，确定用户私钥更新的级数m，随机选择m个主密分量：s1，s2，...sm属于Zq*，并令Ppub=(s1+s2+...+sm)P
a4
保密各个主密钥分量si，其中i=1，2，...，m，公开公共参数param=<q，G1，G2，ê，n，P，Ppub，H1，H2>
a5
执行步骤b
a6

图2

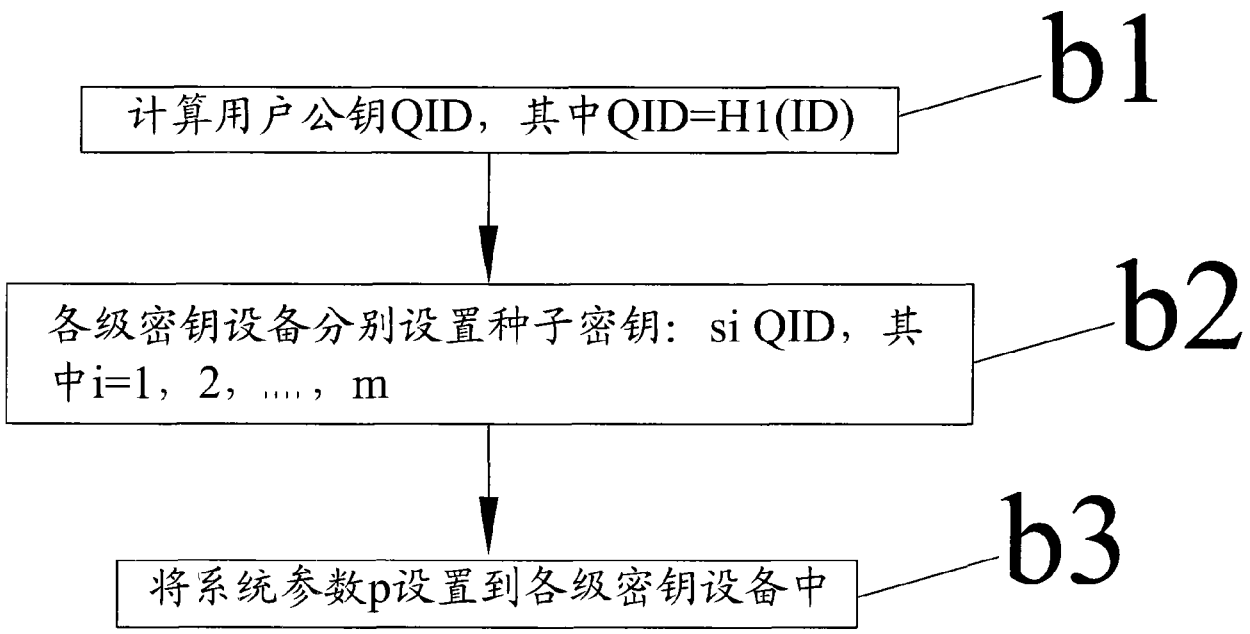

计算用户公钥QID，其中QID=H1(ID)
b1
各级密钥设备分别设置种子密钥：si QID，其中i=1，2，....，m
b2
将系统参数p设置到各级密钥设备中
b3

图 3