(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102946310 A
(43)申请公布日 2013.02.27

(21)申请号 201210322278.1

(22)申请日 2012.09.03

(71)申请人 杭州电子科技大学
地址 310018 浙江省杭州市下沙高教园区 2 号大街

(72)发明人 游林 范萌生 林刚 王升国 陆捷

(74)专利代理机构 杭州求是专利事务所有限公司 33200
代理人 杜军

(51)Int.Cl.
H04L 9/08(2006.01)
G06K 9/00(2006.01)

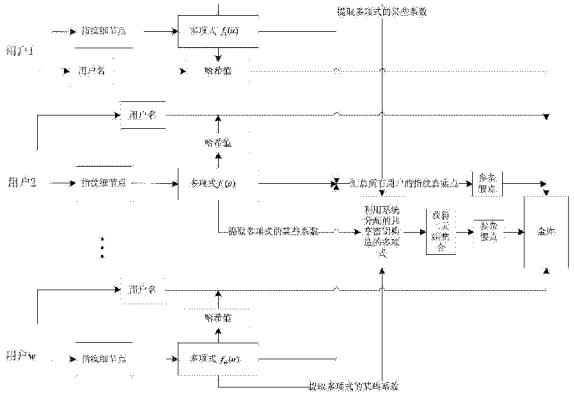
权利要求书 3 页 说明书 7 页 附图 5 页

(54)发明名称

一种基于(k,w)门限秘密共享方案的指纹模糊金库方法

(57)摘要

本发明涉及一种基于(k,w)门限秘密共享方案的指纹模糊金库方法。本发明包括共享密钥分发阶段和共享密钥重构阶段;共享密钥分发阶段又包含指纹模糊金库与用户子密钥的绑定过程和共享密钥绑定过程;共享密钥重构阶段又包含部分用户子密钥的释放过程和共享密钥释放过程。本发明利用了指纹模糊金库方法保护共享密钥的同时,通过用户的指纹特征安全方便地共享密钥的释放,密钥的释放过程相当简单,使密钥共享方案具有更好的实用性。



CN 102946310 A

1. 一种基于 (k,w) 门限秘密共享方案的指纹模糊金库方法，包括共享密钥分发阶段和共享密钥重构阶段；共享密钥分发阶段又包含指纹模糊金库与用户子密钥的绑定过程和共享密钥绑定过程；共享密钥重构阶段又包含部分用户子密钥的释放过程和共享密钥释放过程，其特征在于：

所述的共享密钥分发阶段具体如下：

（1）. 指纹模糊金库与用户子密钥的绑定过程；

步骤（1）. w 个用户分别输入个人的注册用户名和提取个人的指纹特征；将指纹特征的平面坐标和方向均线性映射到 $[0,255]$，分别用 8 比特表示；x，y 表示指纹特征点的平面坐标，θ 表示指纹特征点的脊线方向，t 表示指纹特征点的类型；其中指纹特征点的类型只采用端点和叉点，当其类型为端点时，$t=0$；其类型为叉点时，$t=1$；各用户的指纹特征分别表示为：$(x_{1,i_1},y_{1,i_1},\theta_{1,i_1},t_{1,i_1})(i_1=1,\ldots,n_1)$，$(x_{2,i_2},y_{2,i_2},\theta_{2,i_2},t_{2,i_2})(i_2=1,\ldots,n_2)$，…，$(x_{w,i_w},y_{w,i_w},\theta_{w,i_w},t_{w,i_w})(i_w=1,\ldots,n_w)$；

步骤（2）. w 个用户分别构造互不相同的多项式 $f_1(u)$，$f_2(u)$，…，$f_w(u)$：

$$\begin{aligned}
f_1(u)&=b_{1,0}+b_{1,1}u+\cdots+b_{1,\varepsilon}u^{\varepsilon} \bmod p\\
f_2(u)&=b_{2,0}+b_{2,1}u+\cdots+b_{2,\varepsilon}u^{\varepsilon} \bmod p\\
&\vdots\\
f_w(u)&=b_{w,0}+b_{w,1}u+\cdots+b_{w,\varepsilon}u^{\varepsilon} \bmod p
\end{aligned},$$

多项式的系数 $b_{y,j}$ 都是 16-bit 的随机数，$y=1,2,\ldots,w$，$j=0,1,\ldots,\varepsilon$，$p=65537$ 为一个素数；$b_{1,0},b_{1,1},\ldots,b_{1,\varepsilon}$ 被视为用户 1 的子密钥，$b_{w,0},b_{w,1},\ldots,b_{w,\varepsilon}$ 被视为用户 w 的子密钥；ls 为共享密钥的比特串长度，$l=\lceil ls/16\rceil$，$\varepsilon=\lceil (l+1)/k\rceil$，而 $\lceil\ \rceil$ 为向上取整运算；

步骤（3）. 计算各用户所对应的多项式系数比特串的哈希值；每个用户都以“注册用户名：哈希值”形式存储；

步骤（4）. 用户 y 将指纹每个特征点的平面坐标 $x_{y,i}$、$y_{y,i}$ 串联起来构成一个 16-bit 的数 $u_{y,i_y}=[x_{y,i_y}\|y_{y,i_y}]$，然后计算 $f_y(u_{y,i_y})$；用户 y 获得的指纹真实点集合记作 $G_y=\{(u_{y,i_y},\theta_{y,i_y},f_y(u_{y,i_y}),t_{y,i_y})\}$；汇集所有用户的指纹真实点集合记作 $G=\{G_1,G_2,\ldots,G_w\}$；

步骤（5）. 添加 ζ 个由随机数组成的元组 $(c_{j_2},e_{j_2},l_{j_2},d_{j_2})$ 作为杂凑点，c_{j_2} 为 16-bit 的随机数，e_{j_2} 为 8-bit 的随机数，l_{j_2} 为 16-bit 的随机数，d_{j_2} 只能随机地取值 0 和 1，$j_2=1,2,\ldots,\zeta$；将杂凑点集合记作 $C=\{(c_{j_2},e_{j_2},l_{j_2},d_{j_2})\,|\,c_{j_2},e_{j_2},l_{j_2},d_{j_2}\in \mathbb{F}_p, l_{j_2}\neq f_y(c_{j_2})\}$；将集合 G 和 C 混合置乱得到金库集合 V 并存储，其中 $V=\{v_j=(gy_j'',\theta_j'',\psi_j'',t_j'')\}$，$j=1,\ldots,\rho$，$\rho=n_1+n_2+\cdots+n_w+\zeta$，$gy_j''=c_{j_2}$ 或 u_{y,i_y}，$\psi_j''=l_{j_2}$ 或 $f_y(u_{y,i_y})$；

共享密钥绑定过程

步骤（1）. 利用共享密钥 s 构造多项式 $F(x)$；将 s 的二进制串分块组成在 $\mathbb{F}_p[x]$ 上的

$m-1$ 次多项式的部分系数，其余的 $m-l-1$ 个系数是 16-bit 的随机整数，其中 $m=\varepsilon\times k$ ；多项式的常数项为一个 16 比特的校验码；

步骤(2). 计算 $P(b_{y,j_k})$，$j_1=1,2,\ldots,\varepsilon$；得到集合 $GS=\{(b_{y,j_k},P(b_{y,j_k}))\}$；参杂假点集合 $GC=\{(g_{j_k},h_{j_k})\mid j_k=1,2,\ldots,\eta\}$，其中 g_{j_k}，h_{j_k} 都是 16-bit 的随机整数且 $h_{j_k}\neq P(g_{j_k})$；

步骤(3). 将集合 GS 和 GC 混合置乱，得到集合 GV 并将其存储；

所述的共享密钥重构阶段具体如下：

k 个共享密钥持有者恢复共享密钥 s，他们将做如下工作：

(1). 部分用户子密钥的释放过程：

步骤(1). 共享密钥持有者 y 输入指纹，将提取到的查询指纹图像每个特征点的平面坐标和方向均线性映射到 $[0,255]$，分别用 8 比特表示；查询指纹的特征点集合 $Q_y=\{r_{y,i_y}=(x'_{y,i_y},y'_{y,i_y},\theta'_{y,i_y},t'_{y,i_y})\mid i_y=1,2,\ldots,n'_y\}$；

步骤(2). 将金库集合 V 中元组的第一个元素分解可以得到 $V=\{r'_j=(x''_j,y''_j,\theta''_j,\psi''_j,t''_j)\}$，$gv''_j=x''_j\|y''_j$；

步骤(3). 从 Q_y 中选取一个查询指纹特征点 $r_{y,i_y}=(x'_{y,i_y},y'_{y,i_y},\theta'_{y,i_y},t'_{y,i_y})$ 作为参考点，计算 V 中一个点 $r''_j=(x''_j,y''_j,\theta''_j,t''_j)$ 与该参考点的旋转角度与位置偏移量；

$$\begin{cases}\Delta x=|x'_{y,i_y}-x''_j|\\ \Delta y=|y'_{y,i_y}-y''_j|\\ \Delta\theta=|\theta'_{y,i_y}-\theta''_j|\end{cases}\qquad(1)$$

步骤(4). 根据 (1) 式计算的变换量，对查询指纹所有剩下的特征点进行校准；令校准后的指纹特征点特征如下：

$$\begin{cases}\tilde{x}_{y,\tau}=x'_{y,i_y}+(x'_{y,\tau}-x'_{y,i_y})\cos\Delta\theta-(y'_{y,\tau}-y'_{y,i_y})\sin\Delta\theta+\Delta x\\ \tilde{y}_{y,\tau}=y'_{y,i_y}+(x'_{y,\tau}-x'_{y,i_y})\sin\Delta\theta+(y'_{y,\tau}-y'_{y,i_y})\cos\Delta\theta+\Delta y\\ \tilde{\theta}_{y,\tau}=(\theta'_{y,\tau}+\Delta\theta)\bmod 360\\ \tilde{t}_{y,\tau}=t'_{y,i_y}\end{cases}\qquad(2)$$

其中 $\tau=1,\ldots,n'_y$，校准后的特征点特征的平面坐标分别为 $\tilde{x}_{y,\tau}$，$\tilde{y}_{y,\tau}$，方向为 $\tilde{\theta}_{y,\tau}$，类型为 $\tilde{t}_{y,\tau}$；

步骤(5). 将校准后的特征点特征集合 $QX_y=\{\tilde{r}_{y,\tau}=(\tilde{x}_{y,\tau},\tilde{y}_{y,\tau},\tilde{\theta}_{y,\tau},\tilde{t}_{y,\tau})\}$ 与集合 $V'=\{r'_j=(x''_j,y''_j,\theta''_j,t''_j)\}$ 进行匹配，如果满足 (3) 式，那就认为是一个匹配点；

$$\begin{cases}\sqrt{(\tilde{x}_{y,\varepsilon}-x_j'')^2+(\tilde{y}_{y,\varepsilon}-y_j'')^2}<\delta \\ |\tilde{\theta}_{y,\varepsilon}-\theta_j''|<\sigma \\ \tilde{t}_{y,\varepsilon}=t_j''\end{cases} \quad (3)$$

其中 δ σ 为设定的阈值；根据匹配点的个数得到以第 i_y 个查询指纹特征点和第 j 个金库点作为一对参考点的一个匹配个数 $sc_{i_y,j}$ ；

步骤(6). 遍历完 V' 中剩下的点依次计算(1)、(2)、(3)分别得到对应的匹配个数 $sc_{i_y,j}$ ；选取出其中一个最大的匹配个数 $sc_{y,max}$ ；

步骤(7). 重复步骤(3)、(4)、(5)、(6)，将每次得到的 $sc_{y,max}$ 进行比较，保留较大的匹配个数，若匹配分数大于阈值说明该查询指纹与注册指纹匹配；同时将得到匹配点集合 $TC_y=\{(\alpha_{y,\varphi},\beta_{y,\varphi},\sigma_{y,\varphi},\gamma_{y,\varphi})|\varphi=1,2,\ldots,sc_{y,max}\}$ ；

步骤(8). 利用牛顿内插值法重构出多项式 $f_y^*(u)$，此时要求共享密钥持有者输入用户名；计算多项式 $f_y^*(u)$ 的系数比特串的哈希值 hs_y'，与通过用户名索引到的哈希值 hs_y 比较；若相等则说明多项式重构正确，否则，要求用户重新输入指纹；若用户被要求重新输入指纹的次数超过 3 次，该用户视为非法用户；

步骤(9). 当 k 个共享密钥持有者正确地重构出对应的多项式后，从对应的多项式提取出 k 个共享密钥持有者的子密钥；

(2). 共享密钥释放过程：

从集合 GV 匹配出对应的二元组集合，利用牛顿内插值法重构多项式 $P^*(\lambda)$ ；同时计算除常数项外的多项式系数比特串的校验码，比较校验码是否等于多项式的常数项；若相等，则释放的共享密钥是正确的。

一种基于 (k,w) 门限秘密共享方案的指纹模糊金库方法

技术领域

[0001] 本发明属于模式识别和密码学技术领域，具体涉及一种(k,w)门限秘密共享方案与自动对齐的指纹模糊金库方案。

背景技术

[0002] 秘密共享是现代密码学领域中一个非常重要的分支，也是信息安全方向一个重要研究内容。1979 年，Shamir 和 Blakley 独立地提出了密钥分散管理的概念，实现这一思想的机制称为(k,w)-门限方案。该方案是将一个密钥(称为共享密钥)分成w个部分(称为w个子密钥或影子，分别交给w个人保管，使得对确定的整$k<w$）满足：(1) 在这w个人中，任意r（$r\geq k$）个人协作利用它们的子密钥能够恢复出共享密钥；(2) 任意$k-1$个人协作对恢复共享密钥没有任何帮助。这种密钥分散管理的思想使密钥管理更加安全灵活，然而每个成员的子密钥存在安全隐患。各成员采用指纹模糊金库方法保护各自的子密钥。

[0003] 在 2002 年 A. Juels 和 M. Sudan 提出了“A fuzzy vault scheme”。在他们提出的模糊金库方法中，将用户惟一的集合 A 混合用户的密钥进入基于 Reed-Solomon 的金库中。用户可以利用与集合 A 有绝大部分元素相同的集合 B 恢复出密钥。

[0004] 基于全局域配准的指纹模糊金库方案的思想，模糊金库方案可以用于保护各成员的子密钥。此时这种密钥分散管理的安全是基于多项式重构的困难性和用户生物特征未泄漏。

发明内容

[0005] 在真实可靠的实验条件下，本发明提供了一套实用化的基于(k,w)门限秘密共享方案的指纹模糊金库方法。这是一套既有效地保护了用户的指纹数据，又确保了共享密钥的安全的解决方案。

[0006] 一种基于(k,w)门限秘密共享方案的指纹模糊金库方法包括共享密钥分发阶段和共享密钥重构阶段：共享密钥分发阶段又包含指纹模糊金库与用户子密钥的绑定过程和共享密钥绑定过程；共享密钥重构阶段又包含部分用户子密钥的释放过程和共享密钥释放过程。

[0007] 所述的共享密钥分发阶段具体如下：

1. 指纹模糊金库与用户子密钥的绑定过程

步骤 1. w个用户分别输入个人的注册用户名和提取个人的指纹特征。将指纹特征的平面坐标和方向均线性映射到$[0,255]$，分别用 8 比特表示。x，y表示指纹特征点的平面坐标，θ表示指纹特征点的脊线方向，t表示指纹特征点的类型。其中指纹特征点的类型只采用端点和叉点，当其类型为端点时，$t=0$；其类型为叉点时，$t=1$。各用户的指纹特征分别表示为：$(x_{1,i_1},y_{1,i_1},\theta_{1,i_1},t_{1,i_1})(i_1=1,\ldots,n_1)$，$(x_{2,i_2},y_{2,i_2},\theta_{2,i_2},t_{2,i_2})(i_2=1,\ldots,n_2)$，…，

$(x_{w,i_w}, y_{w,i_w}, \theta_{w,i_w}, t_{w,i_w})(i_w = 1,\ldots,n_w)$。

[0008] 步骤 2. w 个用户分别构造互不相同的多项式 $f_1(u)$，$f_2(u)$，…，$f_w(u)$：

$$f_1(u) = b_{1,0} + b_{1,1}u + \cdots + b_{1,\varepsilon}u^{\varepsilon} \bmod p$$
$$f_2(u) = b_{2,0} + b_{2,1}u + \cdots + b_{2,\varepsilon}u^{\varepsilon} \bmod p$$
$$\vdots$$
$$f_w(u) = b_{w,0} + b_{w,1}u + \cdots + b_{w,\varepsilon}u^{\varepsilon} \bmod p$$

多项式的系数 b_{y,j_1} 都是 16-bit 的随机数，$y = 1,2,\ldots,w$，$j_1 = 0,1,\ldots,8$，$p = 65537$ 为一个素数。$b_{1,0}, b_{1,1}, \ldots, b_{1,\varepsilon}$ 被视为用户 1 的子密钥，…，$b_{w,0}, b_{w,1}, \ldots, b_{w,\varepsilon}$ 被视为用户 w 的子密钥。ls 为共享密钥的比特串长度，$l = \lceil ls/16 \rceil$，$\varepsilon = \lceil (l+1)/k \rceil$，而 $\lceil\ \rceil$ 为向上取整运算。

[0009] 步骤 3. 计算各用户所对应的多项式系数比特串的哈希值。每个用户都以“注册用户名 :哈希值”形式存储。

[0010] 步骤 4. 用户 y 将指纹每个特征点的平面坐标 x_{y,i_y}、y_{y,i_y} 串联起来构成一个 16-bit 的数 $u_{y,i_y} = [x_{y,i_y} \| y_{y,i_y}]$，然后计算 $f_y(u_{y,i_y})$。用户 y 获得的指纹真实点集合记作 $G_y = \{(u_{y,i_y}, \theta_{y,i_y}, f_y(u_{y,i_y}), t_{y,i_y})\}$。汇集所有用户的指纹真实点集合记作 $G = \{G_1, G_2, \ldots, G_w\}$。

[0011] 步骤 5. 添加 ζ 个由随机数组成的元组 $(c_{j_2}, e_{j_2}, l_{j_2}, d_{j_2})$ 作为杂凑点，c_{j_2} 为 16-bit 的随机数，e_{j_2} 为 8-bit 的随机数，l_{j_2} 为 16-bit 的随机数，d_{j_2} 只能随机地取值 0 和 1，$j_2 = 1,2,\ldots,\zeta$。将杂凑点集合记作 $C = \{(c_{j_2}, e_{j_2}, l_{j_2}, d_{j_2}) | c_{j_2}, e_{j_2}, l_{j_2}, d_{j_2} \in \mathrm{F}_p, l_{j_2} \neq f_y(c_{j_2})\}$。将集合 G 和 C 混合置乱得到金库集合 V 并存储，其中 $V = \{v_j' = (gy_j'', \theta_j'', \psi_j'', t_j'')\}$，$j' = 1,\ldots,\rho$，$\rho = n_1 + n_2 + \cdots + n_w + \zeta$，$gy_j'' = c_{j_2}$ 或 u_{y,i_y}，$\psi_j'' = l_{j_2}$ 或 $f_y(u_{y,i_y})$。

[0012] 2. 共享密钥绑定过程

步骤 1. 利用共享密钥 s 构造多项式 $P(x)$。将 s 的二进制串分块组成在 $\mathbb{F}_p[\lambda]$ 上的 $m-1$ 次多项式的部分系数，其余的 $m-l-1$ 个系数是 16-bit 的随机整数，其中 $m = \varepsilon \times k$。多项式的常数项为一个 16 比特的校验码。

[0013] 步骤 2. 计算 $P(b_{y,j_3})$，$j_3 = 1,2,\ldots,\varepsilon$。得到集合 $GS = \{(b_{y,j_3}, P(b_{y,j_3}))\}$。参杂假点集合 $GC = \{(g_{j_4}, h_{j_4}) | j_4 = 1,2,\ldots,\eta\}$，其中 g_{j_4}，h_{j_4} 都是 16-bit 的随机整数且 $h_{j_4} \neq P(g_{j_4})$。

[0014] 步骤 3. 将集合 GS 和 GC 混合置乱，得到集合 GV 并将其存储。

[0015] 所述的共享密钥重构阶段具体如下：

k 个共享密钥持有者恢复共享密钥 s，他们将做如下工作：

1. 部分用户子密钥的释放过程

步骤 1. 共享密钥持有者 y' 输入指纹，将提取到的查询指纹图像每个特征点的平面坐标和方向均线性映射到 $[0,255]$，分别用 8 比特表示。查询指纹的特征点集合 $Q_{y'} = \{(y'_{y',i'_y} = (x'_{y',i'_y}, y'_{y',i'_y}, \theta_{y',i'_y}, t'_{y',i'_y}) | i'_y = 1,2,\ldots,n'_y\}$。

[0016] 步骤2. 将金库集合 V 中元组的第一个元素分解可以得到 $V=(r_j^{'}=(x_j^{''},y_j^{'},\theta_j^{''},\psi_j^{''},t_j^{''}))$，$gy_j^{''}=x_j^{''}\|y_j^{'}$。

[0017] 步骤3. 从 Q_y 中选取一个查询指纹特征点 $r_{y,i_y}^{'}=(x_{y,i_y}^{'},y_{y,i_y}^{'},\theta_{y,i_y}^{'},t_{y,i_y}^{'})$ 作为参考点，计算 V 中一个点 $r_j^{''}=(x_j^{''},y_j^{''},\theta_j^{''},t_j^{''})$ 与该参考点的旋转角度与位置偏移量。

[0018]
$$\begin{cases}\Delta x=|x_{y,i_y}^{'}-x_j^{''}|\\ \Delta y=|y_{y,i_y}^{'}-y_j^{''}|\\ \Delta\theta=|\theta_{y,i_y}^{'}-\theta_j^{''}|\end{cases}\qquad(1)$$

步骤4. 根据(1)式计算的变换量，对查询指纹所有剩下的特征点进行校准。令校准后的指纹特征点特征如下：

$$\begin{cases}\tilde{x}_{y,\varepsilon}=x_{y,i_y}^{'}+(x_{y,\varepsilon}^{'}-x_{y,i_y}^{'})\cos\Delta\theta-(y_{y,\varepsilon}^{'}-y_{y,i_y}^{'})\sin\Delta\theta+\Delta x\\ \tilde{y}_{y,\varepsilon}=y_{y,i_y}^{'}+(x_{y,\varepsilon}^{'}-x_{y,i_y}^{'})\sin\Delta\theta+(y_{y,\varepsilon}^{'}-y_{y,i_y}^{'})\cos\Delta\theta+\Delta y\\ \tilde{\theta}_{y,\varepsilon}=(\theta_{y,\varepsilon}^{'}+\Delta\theta)\bmod 360\\ \tilde{t}_{y,\varepsilon}=t_{y,i_y}^{'}\end{cases}\qquad(2)$$

其中 $\varepsilon=1,\ldots,n_y^{'}$，校准后的特征点的平面坐标分别为 $\tilde{x}_{y,\varepsilon}$，$\tilde{y}_{y,\varepsilon}$，方向为 $\tilde{\theta}_{y,\varepsilon}$，类型为 $\tilde{t}_{y,\varepsilon}$。

[0019] 步骤5. 将校准后的特征点特征集合 $QX_y=(\tilde{r}_{y,\varepsilon}=(\tilde{x}_{y,\varepsilon},\tilde{y}_{y,\varepsilon},\tilde{\theta}_{y,\varepsilon},\tilde{t}_{y,\varepsilon}))$ 与集合 $V^{'}=(r_j^{'}=(x_j^{''},y_j^{''},\theta_j^{''},t_j^{''}))$ 进行匹配，如果满足(3)式，那就认为是一个匹配点。

[0020]
$$\begin{cases}\sqrt{(\tilde{x}_{y,\varepsilon}-x_j^{''})^2+(\tilde{y}_{y,\varepsilon}-y_j^{''})^2}<\delta\\ |\tilde{\theta}_{y,\varepsilon}-\theta_j^{''}|<\sigma\\ \tilde{t}_{y,\varepsilon}=t_j^{''}\end{cases}\qquad(3)$$

其中 δ σ 为设定的阈值。根据匹配点的个数得到以第 i_y 个查询指纹特征点和第 $j^{'}$ 个金库点作为一对参考点的一个匹配个数 $sc_{i_y,j}$。

[0021] 步骤6. 遍历完 $V^{'}$ 中剩下的点依次计算(1)、(2)、(3)分别得到对应的匹配个数 $sc_{i_y,j}$。选取出其中一个最大的匹配个数 $sc_{y,max}$。

[0022] 步骤7. 重复步骤3、4、5、6，将每次得到的 $sc_{y,max}$ 进行比较，保留较大的匹配个数，若匹配分数大于阈值说明该查询指纹与注册指纹匹配。同时将得到匹配点集合 $TC_y=\{(\alpha_{y,\varphi},\beta_{y,\varphi},\sigma_{y,\varphi},\gamma_{y,\varphi})|\varphi=1,2,\ldots,sc_{y,max}\}$。

[0023] 步骤8. 利用牛顿内插值法重构出多项式 $f_y^{*}(u)$，此时要求共享密钥持有者输入用户名。计算多项式 $f_y^{*}(u)$ 的系数比特串的哈希值 $hs_y^{'}$，与通过用户名索引到的哈希值 hs_y 比较。若相等则说明多项式重构正确，否则，要求用户重新输入指纹。若用户被要求重新输入指纹的次数超过3次，该用户视为非法用户。

[0024] 步骤 9. 当 k 个共享密钥持有者正确地重构出对应的多项式后，从对应的多项式提取出对应的 k 个共享密钥持有者的子密钥。

[0025] 2. 共享密钥释放过程

从集合 GV 匹配出对应的二元组集合，利用牛顿内插值法重构多项式 $F^*(x)$。同时计算除常数项外的多项式系数比特串的校验码，比较校验码是否等于多项式的常数项。若相等，则释放的共享密钥是正确的。

[0026] 这种 (k,w) 门限秘密共享方案使密钥管理更加安全灵活，然而每个成员的子密钥存在安全隐患。本发明的特点是利用了指纹模糊金库方法保护共享密钥的同时，通过用户的指纹特征安全方便地共享密钥的释放，密钥的释放过程相当简单，使密钥共享方案具有更好的实用性。

附图说明

[0027] 图 1 是共享密钥绑定过程的流程图；

图 2 是共享密钥释放过程的流程图；

图 3 是进行试验的指纹数据库中的部分指纹图像；

图 4 是注册指纹图像中提取的特征点图；

图 5 是查询指纹图像中提取的特征点图。

具体实施方式

[0028] 以下结合附图对本发明作进一步说明。

[0029] 所述的共享密钥分发阶段具体如下(如图 1 所示)：

1. 指纹模糊金库与用户子密钥的绑定过程

步骤 1. w 个用户分别输入个人的注册用户名和指纹。进行试验的指纹数据库中的部分指纹图像如图3。对该指纹图像进行分割操作，方向场和梯度的计算，均衡，收敛，平滑，增强，二值化，细化等一系列预处理操作得到一幅清晰的保持了指纹特征信息二值图像。然后提取该图像中的所有特征点，并过滤和去除其中的伪特征点，保留原始图像的真实特征点，如图 4 所示。

[0030] 步骤 2. 将指纹每个特征点的平面坐标和方向均线性映射到 $[0,255]$，分别用 8 比特表示。x，y 表示指纹特征点的平面坐标，θ 表示指纹特征点的脊线方向，t 表示指纹特征点的类型。其中指纹特征点的类型只采用端点和叉点，当其类型为端点时，$t=0$；其类型为叉点时，$t=1$。各用户的指纹特征分别表示为：$(x_{1,i_1},y_{1,i_1},\theta_{1,i_1},t_{1,i_1})(i_1=1,\ldots,n_1)$，$(x_{2,i_2},y_{2,i_2},\theta_{2,i_2},t_{2,i_2})(i_2=1,\ldots,n_2)$，…，$(x_{w,i_w},y_{w,i_w},\theta_{w,i_w},t_{w,i_w})(i_w=1,\ldots,n_w)$。

[0031] 步骤 3. w 个用户分别构造互不相同的多项式 $f_1(u)$，$f_2(u)$，…，$f_w(u)$：

$$\begin{aligned} f_1(u) &= b_{1,0}+b_{1,1}u+\cdots+b_{1,n}u^n \bmod p \\ f_2(u) &= b_{2,0}+b_{2,1}u+\cdots+b_{2,n}u^n \bmod p \\ &\vdots \\ f_w(u) &= b_{w,0}+b_{w,1}u+\cdots+b_{w,n}u^n \bmod p \end{aligned},$$

多项式的系数 $b_{y,j}$ 都是 16-bit 的随机数，$y=1,2,\ldots,w$，$j_1=0,1,\ldots,8$，$p=65537$ 为一个素数。$b_{1,0},b_{1,1},\ldots,b_{1,\varepsilon}$ 被视为用户 1 的子密钥，…，$b_{w,0},b_{w,1},\ldots,b_{w,\varepsilon}$ 被视为用户 w 的子密钥。ls 为共享密钥的比特串长度，$l=\lceil ls/16\rceil$，$\varepsilon=\lceil (l+1)/k\rceil$，而 $\lceil\ \rceil$ 为向上取整运算。

[0032] 步骤 4. 计算各用户所对应的多项式系数比特串的哈希值。用户 y 计算 $hs_y=H(b_{y,0}\|b_{y,1}\|\cdots\|b_{y,8})$，$H(\cdot)$ 为一个生成 32-bit 数的单向哈希函数。每个用户都以“注册用户名 :哈希值”形式存储。

[0033] 步骤 5. 用户 y 将指纹每个特征点的平面坐标 x_{y,t_y}、y_{y,t_y} 串联起来构成一个 16-bit 的数 $u_{y,t_y}=[x_{y,t_y}\|y_{y,t_y}]$，然后计算 $f_y(u_{y,t_y})$。用户 y 获得的指纹真实点集合记作 $G_y=\{(u_{y,t_y},\theta_{y,t_y},f_y(u_{y,t_y}),t_{y,t_y})\}$。汇集所有用户的指纹真实点集合记作 $G=\{G_1,G_2,\ldots,G_w\}$。

[0034] 步骤 6. 添加 ζ 个由随机数组成的元组 $(c_{j_2},e_{j_2},l_{j_2},d_{j_2})$ 作为杂凑点，c_{j_2} 为 16-bit 的随机数，e_{j_2} 为 8-bit 的随机数，l_{j_2} 为 16-bit 的随机数，d_{j_2} 只能随机地取值 0 和 1，$j_2=1,2,\ldots,\zeta$。将杂凑点集合记作 $C=\{(c_{j_2},e_{j_2},l_{j_2},d_{j_2})|c_{j_2},e_{j_2},l_{j_2},d_{j_2}\in F_p,l_{j_2}\neq f_y(c_{j_2})\}$。将集合 G 和 C 混合置乱得到金库集合 V 并存储，其中 $V=\{v_j'=(gy_j'',\theta_j'',\psi_j'',t_j'')\}$，$j'=1,\ldots,\rho$，$\rho=n_1+n_2+\cdots+n_w+\zeta$，$gy_j''=c_{j_2}$ 或 u_{y,t_y}，$\psi_j''=l_{j_2}$ 或 $f_y(u_{y,t_y})$。

[0035] 2. 共享密钥绑定过程

步骤 1. 利用共享密钥 s 构造多项式 $P(x)$。将 s 的二进制串分块组成在 $F_p[x]$ 上的 $m-1$ 次多项式的部分系数，其余的 $m-l-1$ 个系数是 16-bit 的随机整数，其中 $m=\varepsilon\times k$。

[0036] $P(x)=\mu_0+\mu_1x+\cdots+\mu_{m-1}x^{m-1}$，

多项式的常数项 μ_0 为一个 16 比特的校验码，即 $\mu_0=crc_{16}(\mu_1\|\cdots\|\mu_{m-1})$。其中共享密钥 $s=u_{m-1}\|u_{m-2}\|\cdots\|u_{m-l}$。

[0037] 步骤 2. 计算 $P(b_{y,j_3})$，$j_3=1,2,\ldots,\varepsilon$。得到集合 $GS=\{(b_{y,j_3},P(b_{y,j_3}))\}$。参杂假点集合 $GC=\{(g_{j_4},h_{j_4})|j_4=1,2,\ldots,\eta\}$，其中 g_{j_4}，h_{j_4} 都是 16-bit 的随机整数且 $h_{j_4}\neq P(g_{j_4})$。

[0038] 步骤 3. 将集合 GS 和 GC 混合置乱，得到集合 GV 并将其存储。

[0039] 所述的共享密钥重构阶段具体如下(如图 2 所示)：

k 个共享密钥持有者恢复共享密钥 s，他们将做如下工作：

1. 部分用户子密钥的释放过程

步骤 1. 共享密钥持有者 y 输入指纹，对该输入的查询指纹图像进行分割操作，方向场和梯度的计算，均衡，收敛，平滑，增强，二值化，细化等一系列预处理操作得到一幅清晰的保持了指纹特征信息二值图像。然后提取该图像中的所有特征点，并过滤和去除其中的伪特征点。最终提取得到查询指纹的真实特征点，如图 5 所示。将提取到的查询指纹图像每个特征点的平面坐标和方向均线性映射到 $[0,255]$，分别用 8 比特表示。查询指纹的特征点

集合$Q_y=\{r_{y,i_y}=(x_{y,i_y},y_{y,i_y},\theta_{y,i_y},t_{y,i_y})\mid i_y=1,2,\ldots,n_y\}$。

[0040] 步骤2. 将金库集合V中元组的第一个元素分解可以得到$V=\{r_j=(x_j,y_j,\theta_j,\psi_j,t_j)\}$，$gr_j=x_j\|y_j$。

[0041] 步骤3. 从Q_y中选取一个查询指纹特征点$r_{y,i_y}=(x_{y,i_y},y_{y,i_y},\theta_{y,i_y},t_{y,i_y})$作为参考点，计算$V$中一个点$r_j=(x_j,y_j,\theta_j,t_j)$与该参考点的旋转角度与位置偏移量。

[0042]
$$\begin{cases}\Delta x=|x_{y,i_y}-x_j|\\ \Delta y=|y_{y,i_y}-y_j|\\ \Delta\theta=|\theta_{y,i_y}-\theta_j|\end{cases}\qquad(1)$$

步骤4. 根据(1)式计算的变换量，对查询指纹所有剩下的特征点进行校准。令校准后的指纹特征点特征如下：

$$\begin{cases}\tilde{x}_{y,\tau}=x_{y,i_y}+(x_{y,\tau}-x_{y,i_y})\cos\Delta\theta-(y_{y,\tau}-y_{y,i_y})\sin\Delta\theta+\Delta x\\ \tilde{y}_{y,\tau}=y_{y,i_y}+(x_{y,\tau}-x_{y,i_y})\sin\Delta\theta+(y_{y,\tau}-y_{y,i_y})\cos\Delta\theta+\Delta y\\ \tilde{\theta}_{y,\tau}=(\theta_{y,\tau}+\Delta\theta)\bmod 360\\ \tilde{t}_{y,\tau}=t_{y,i_y}\end{cases}\qquad(2)$$

其中$\tau=1,\ldots,n_y$，校准后的特征点的平面坐标分别为$\tilde{x}_{y,\tau}$，$\tilde{y}_{y,\tau}$，方向为$\tilde{\theta}_{y,\tau}$，类型为$\tilde{t}_{y,\tau}$。

[0043] 步骤5. 将校准后的特征点特征集合为$QX_y=\{\tilde{r}_{y,\tau}=(\tilde{x}_{y,\tau},\tilde{y}_{y,\tau},\tilde{\theta}_{y,\tau},\tilde{t}_{y,\tau})\}$与集合$V=\{r_j=(x_j,y_j,\theta_j,t_j)\}$进行匹配，如果满足(3)式，那就认为是一个匹配点。

[0044]
$$\begin{cases}\sqrt{(\tilde{x}_{y,\tau}-x_j)^2+(\tilde{y}_{y,\tau}-y_j)^2}<\delta\\ |\tilde{\theta}_{y,\tau}-\theta_j|<\sigma\\ \tilde{t}_{y,\tau}=t_j\end{cases}\qquad(3)$$

其中δ、σ为设定的阈值。遍历完QX_y和V金库中的点后得到匹配点的个数，即以第i_y个查询指纹特征点和第j个金库点作为一对参考点的一个匹配分数$sc_{i_y,j}$。

[0045] 步骤6. 遍历完V中剩下的点依次计算(1)、(2)、(3)分别得到对应的匹配分数$sc_{i_y,j}$。选取出其中一个最大的匹配分数$sc_{y,max}$。

[0046] 步骤7. 重复步骤3、4、5、6，将每次得到的$sc_{y,max}$进行比较，保留较大的匹配分数，若匹配分数大于阈值m说明该查询指纹与注册指纹匹配。同时根据$sc_{y,max}=sc_{i_y,j}$得到以第i_y个查询指纹特征点和第j个金库点作为一对参考点进行匹配时，查询指纹预注册指纹匹配点的个数最多。根据公式(1)、(2)、(3)再次匹配查询指纹和注册指纹，将得到一个匹配点集合$TC_y=\{(\alpha_{y,\varphi},\beta_{y,\varphi},\sigma_{y,\varphi},\gamma_{y,\varphi})\mid\varphi=1,2,\ldots,sc_{y,max}\}$。

[0047] 步骤 8. 利用牛顿内插值法重构多项式 $f_y^*(u)=b'_{y,0}+b'_{y,1}u+\cdots+b'_{y,k-1}u^{k-1}$，此时要求共享密钥持有者输入用户名。计算多项式 $f_y^*(u)$ 的系数比特串的哈希值 $hs'_y=H(b'_{y,0}\|\cdots\|b'_{y,k-1})$，与通过用户名索引到的哈希值 hs_y 比较。若相等则说明多项式重构正确，否则，要求用户重新输入指纹。若用户被要求重新输入指纹的次数超过 3 次，该用户视为非法用户。

[0048] 步骤 9. 当 k 个共享密钥持有者全部正确地重构出对应的多项式后，从对应的多项式提取出对应的 k 个共享密钥持有者的子密钥。

[0049] 2. 共享密钥释放过程

从集合 GV 匹配出对应的二元组集合，利用牛顿内插值法重构出的多项式为 $P^*(\lambda)$。

[0050] $P^*(\lambda)=\mu'_0+\mu'_1\lambda+\cdots+\mu'_{m-1}\lambda^{m-1}$，

同时计算除常数项外的多项式系数比特串的校验码，比较校验码是否等于多项式的常数项。即若 $\mu'_0=crc_{16}(\mu'_1\|\cdots\|\mu'_m)$，则释放的共享密钥是正确的，否则提示共享密钥无法正确地释放。

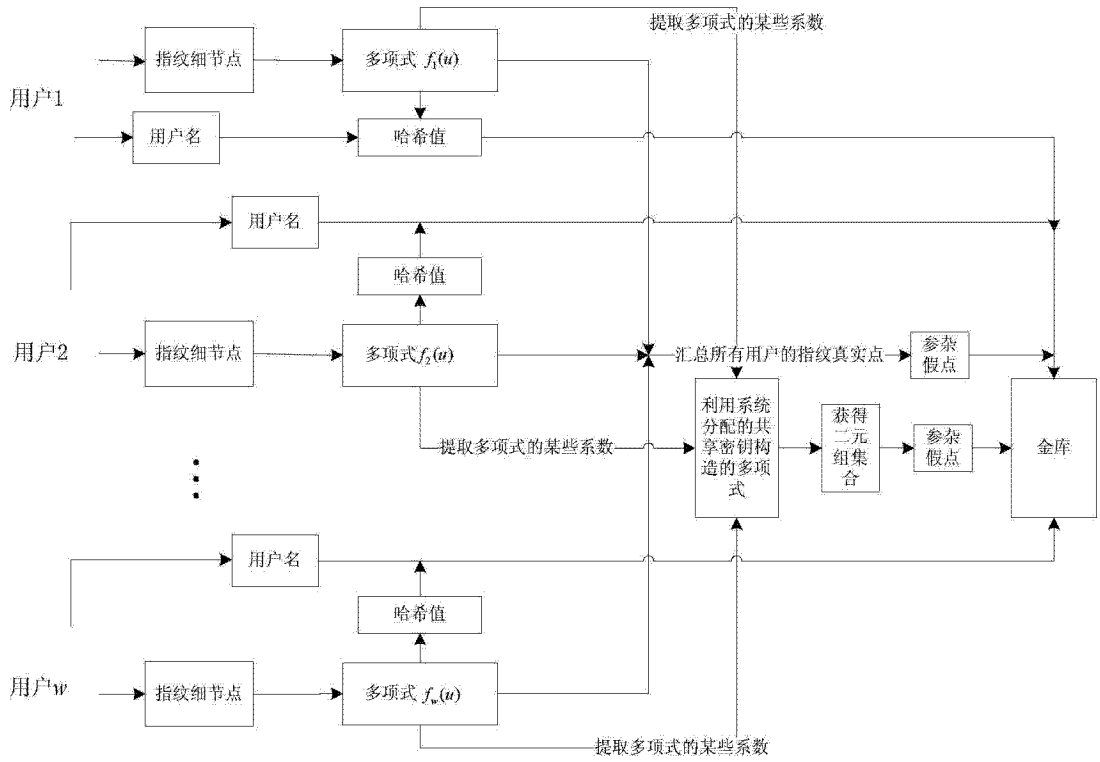

提取多项式的某些系数
指纹细节点
多项式 $f_1(u)$
用户1
用户名
哈希值
用户名
哈希值
用户2
指纹细节点
多项式 $f_2(u)$
汇总所有用户的指纹真实点
参杂假点
提取多项式的某些系数
利用系统分配的共享密钥构造的多项式
获得二元组集合
参杂假点
金库
用户名
哈希值
用户w
指纹细节点
多项式 $f_w(u)$
提取多项式的某些系数

图 1

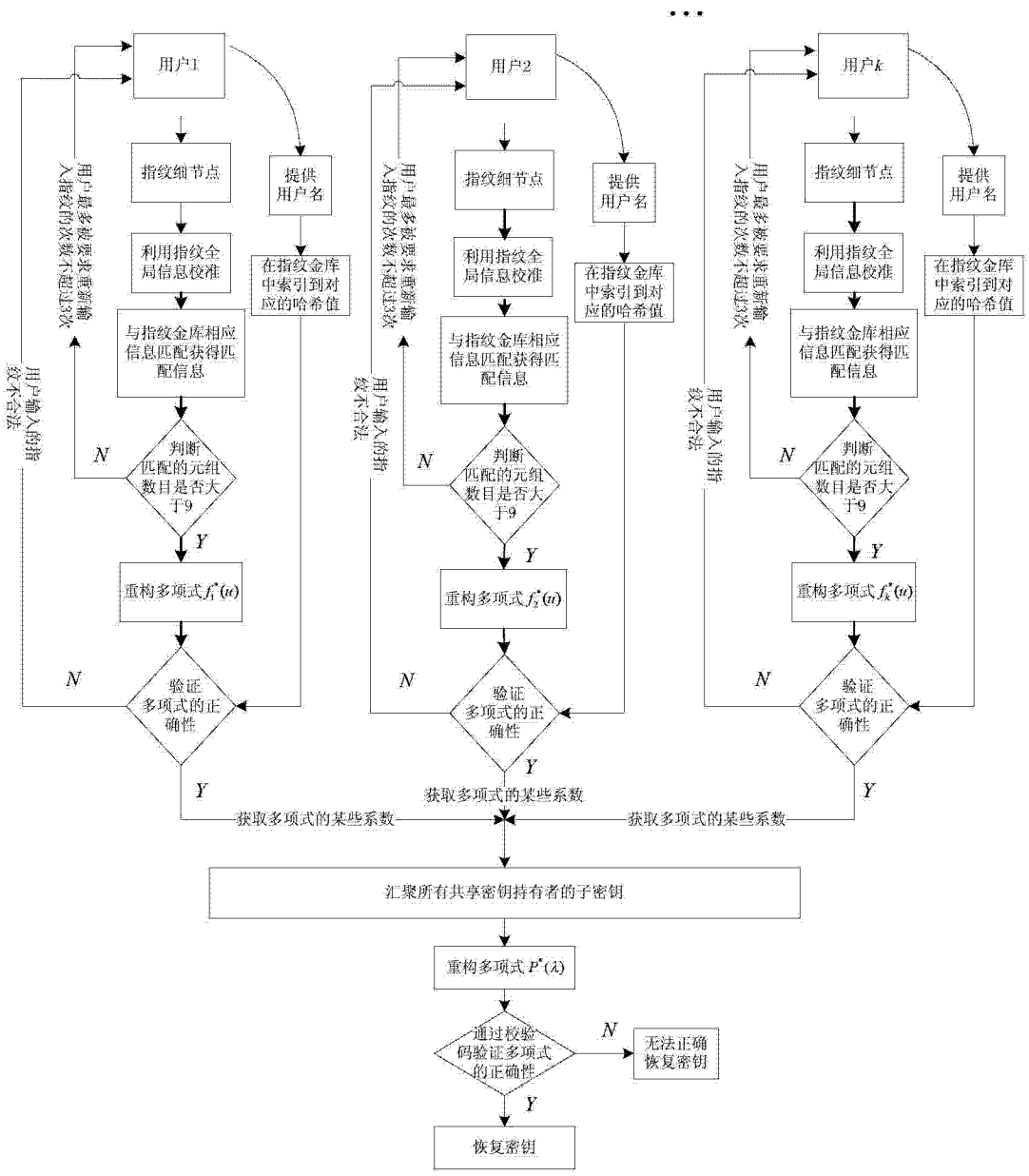
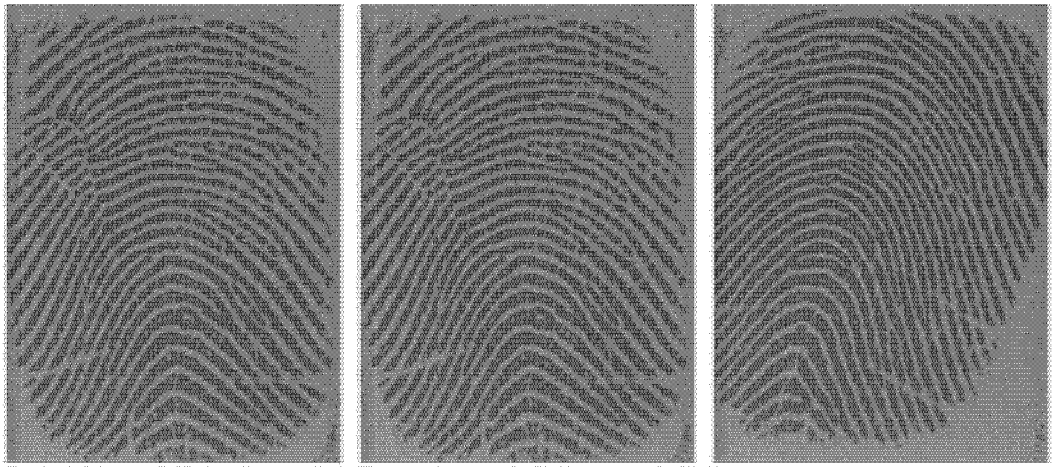
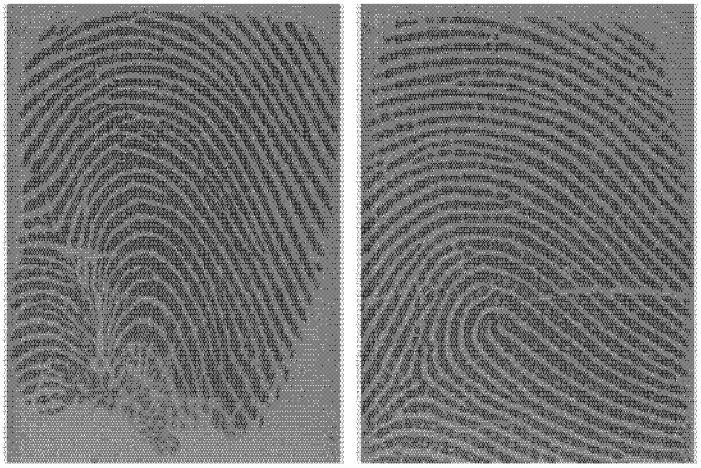

用户1
用户2
用户k
指纹细节点
利用指纹全局信息校准
与指纹金库相应信息匹配获得匹配信息
判断匹配的元组数目是否大于9
重构多项式 $f_1^*(u)$
重构多项式 $f_2^*(u)$
重构多项式 $f_k^*(u)$
验证多项式的正确性
提供用户名
在指纹金库中索引到对应的哈希值
用户最多被要求重新输入指纹的次数不超过3次
用户输入的指纹不合法
N
Y
获取多项式的某些系数
汇聚所有共享密钥持有者的子密钥
重构多项式 $P^*(\lambda)$
通过校验码验证多项式的正确性
无法正确恢复密钥
恢复密钥

图 2



用户 1 用户 2 用户 3



用户 4 用户 5

图 3

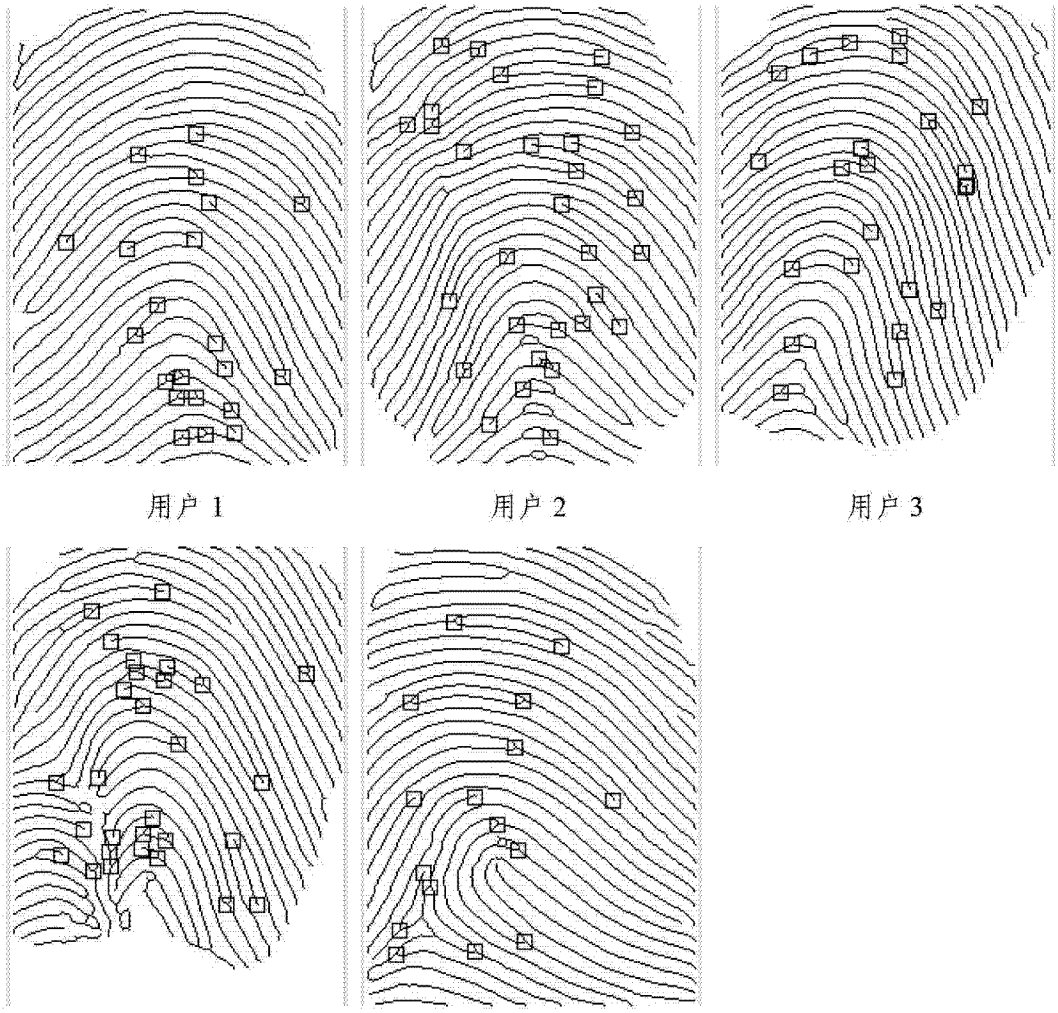


图 4

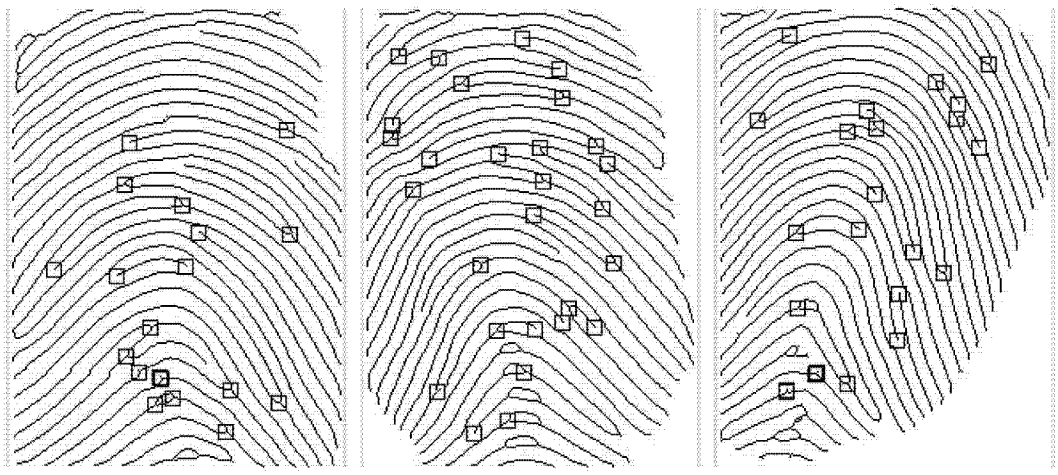


图 5