

文章编号:1007-5321(2014)06-0081-05

DOI:10.13190/j.jbupt.2014.06.017

基于二进制加密电路的无预对齐指纹匹配

李梦醒¹, 冯全², 杨梅², 赵建², 贺康²

(1. 湖南城市学院 通信与电子工程学院, 湖南 益阳 41300; 2. 甘肃农业大学 工学院, 兰州 730070)

摘要: 针对在加密域中进行指纹匹配时指纹模板和现场样本整体对齐的困难性,提出了一种免对齐的指纹匹配方案. 采用具有旋转和平移不变性的细节点纹线方向特征和细节点局部结构,设计实现了相应的加密二进制电路,使得服务器能验证用户指纹,而不会泄露各自的数据. 实验结果表明,所提方案在 FVC2002-DB2 指纹库上具有较高的匹配精度.

关键词: 加密二进制电路; 隐私保护; 无对齐; 指纹匹配

中图分类号: TP391.4; TN911.22

文献标志码: A

Garbled Circuits Based Alignment-Free Fingerprint Matching

LI Meng-xing¹, FENG Quan², YANG Mei², ZHAO Jian², HE Kang²

(1. School of Communication and Electronic Engineering, Hunan City University, Hunan Yiyang 41300, China;

2. Engineering College, Gansu Agricultural University, Lanzhou 730070, China)

Abstract: When a server authenticates users based on fingerprint over open network, the process of fingerprint matching is usually handled in the encrypted domain with purpose of protecting the privacy and security of both parties. However, an accurate alignment of the template and the query sample is rather difficult in this situation. An alignment-free fingerprint-matching scheme was presented, which extracts two kinds of local features around the minutiae, ridge orientation and minutiae local structure. Both features have the properties of invariant to rotation and translation. Garbled circuits were designed to implement the corresponding matching algorithm, which allow the server to verify the user without leaking the respective data. Experiment shows that the scheme achieves a relatively higher accuracy on FVC2002-DB2 public database.

Key words: garbled circuits; privacy protection; alignment-free; fingerprint matching

在开放网络中进行基于生物特征识别的身份认证时,面临的一个重要挑战是,认证的双方——服务器和用户往往相互不信任,双方只要认证结果,但并不想让对方知道自己私有的生物特征数据. 而传统方法中,运行在服务器或客户端的识别程序需要将预先存储的模板和用户的现场生物特征放在一起进行匹配,从而得到认证结果,这就需要知道双方的数据. 为了解决数据隐私性和开放网络环境下认证的

矛盾,一些解决方案采用了密码术中的同态加密^[1]和加密二进制电路(GC, garbled circuit)^[2]. 一些研究者将这些技术结合指纹^[3,4]设计了匹配协议,以实现远程身份认证.

上述文献均采用全局指纹特征,这样需要将模板和现场指纹特征进行预对齐. 但目前的预对齐方案对图像质量要求很高,对齐精度则难以保证^[5]. 为了避免进行预对齐,采用具有旋转和平移不变性

收稿日期: 2014-01-01

基金项目: 国家自然科学基金项目(61062012); 湖南省自然科学基金项目(12JJ3065)

作者简介: 李梦醒(1972—), 男, 副教授, E-mail: mengxingli@hnu.edu.cn.

的局部指纹特征,利用 GC 设计具有隐私保护的认证方案,无须预对齐. 但单一局部特征存在识别精度不高的缺点,故采用了基于细节点局部纹线方向特征和细节点局部结构互相补充,以解决精度和隐私保护的矛盾.

1 特征选取与匹配算法

采用的两种特征均以细节点为中心:细节点纹线方向特征描述了细节点周围纹理特征^[6],而细节点局部结构则反映了局部的细节点分布特点^[7],两者在描述指纹的局部特征方面可以互为补充. 记细节点模板 $M^T = \{(x_i^T, y_i^T, \theta_i^T) | 1 \leq i \leq N_T\}$, 现场样本细节点集合 $M^Q = \{(x_i^Q, y_i^Q, \theta_i^Q) | 1 \leq i \leq N_Q\}$.

1.1 细节点纹线方向特征

Tico 等^[6]提出细节点纹线方向描述子,它以细节点平面坐标为圆心,在其周围画 L 个半径 r_l ($1 \leq l \leq L$) 同心圆,在每个圆周上均匀取 K_l 个采样点 $p_{k,l}$ ($1 \leq k \leq K_l$). 以细节点方向为参考,这些点逆时针方向排列,每个圆上的第 1 个点为细节点方向延长线和该圆的交点. 估计出 $p_{k,l}$ 的局部纹线方向,可以定义细节点纹线特征为: $\mathbf{o} = \{\{\lambda(\theta_{k,l}, \theta)\}_{k=1}^{K_l}\}_{l=1}^L$, 其中 $\lambda(\theta_{k,l}, \theta)$ 为 $\theta_{k,l}$ 相对于 θ 的角度.

记 m_i^T 和 m_j^Q 为 M^T 和 M^Q 中的两个细节点,它们的纹线特征分别记为 $\mathbf{o}(m_i^T) = \{\alpha_{k,l}^i\}$ 和 $\mathbf{o}(m_j^Q) = \{\beta_{k,l}^j\}$, 则它们之间的相似性函数定义为

$$S_0(\mathbf{o}(m_i^T), \mathbf{o}(m_j^Q)) = \sum_{l=1}^L \sum_{k=1}^{K_l} s(x_{k,l}(i, j)) \quad (1)$$

其中 $x()$ 为相对角度:

$$x_{k,l}(i, j) = \min\{360 - |\alpha_{k,l}^i - \beta_{k,l}^j|, |\alpha_{k,l}^i - \beta_{k,l}^j|\} \quad (2)$$

$s(x)$ 为相似值, Tico 认为其最优形式是 $\exp()$ 函数, 但它无法用 GC 实现, 取如下便于 GC 实现的符号函数形式

$$s(x) = \begin{cases} 0, & \text{if } x \leq \tau_0 \\ 1, & \text{其他} \end{cases} \quad (3)$$

其中 τ_0 为一预设阈值. 显然, 若两个纹线方向特征相似, 则 S_0 取值就较小.

1.2 细节点局部结构

考虑 GC 实现的复杂性, 采用尽可能少的细节点构造局部结构, 具体特征构造如图 1 所示. 图中 m_1 为参考细节点, 计算其邻域内的细节点 m_2 与 m_1 之间的长度 l , m_2 与 m_1 的相对方向 φ 以及 m_2 的方向

与两个细节点连线的夹角 ϕ , 可构造细节点局部特征向量 $\mathbf{d} = [l, \varphi, \phi]$.

对于 $m_i^T \in M^T$ 和 $m_j^Q \in M^Q$, 记它们的局部特征分别为 $\mathbf{d}(m_i^T) = [l_i^T, \varphi_i^T, \phi_i^T]$, $\mathbf{d}(m_j^Q) = [l_j^Q, \varphi_j^Q, \phi_j^Q]$. 则作者定义的两者的相似性函数为

$$S_D(\mathbf{d}(m_i^T), \mathbf{d}(m_j^Q)) = |l_i^T - l_j^Q| + x(\varphi_i^T - \varphi_j^Q) + x(\phi_i^T - \phi_j^Q) \quad (4)$$

其中 $x()$ 为相对角度

$$x(\varphi_i^T - \varphi_j^Q) = \min(|\varphi_i^T - \varphi_j^Q|, 360 - |\varphi_i^T - \varphi_j^Q|) \quad (5)$$

$$x(\phi_i^T - \phi_j^Q) = \min(|\phi_i^T - \phi_j^Q|, 360 - |\phi_i^T - \phi_j^Q|) \quad (6)$$

显然, 两个细节点局部结构特征相似, 则 S_D 取值就较小.

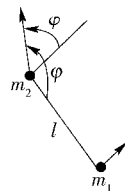


图 1 细节点局部结构

1.3 匹配算法

设由 M^T 构造出的纹线特征和局部结构特征集合分别为 $O^T = \{\mathbf{o}(m_i^T) | 1 \leq i \leq N_T\}$ 和 $D^T = \{\mathbf{d}(m_i^T) | 1 \leq i \leq N_T\}$, 由 M^Q 构造出的纹线特征和局部细节点结构特征集合分别为 $O^Q = \{\mathbf{o}(m_i^Q) | 1 \leq i \leq N_Q\}$ 和 $D^Q = \{\mathbf{d}(m_j^Q) | 1 \leq j \leq N_Q\}$. 为判断模板和现场指纹的匹配程度, 作者对 O^T 和 O^Q 以及 D^T 和 D^Q 中的元素进行逐点比较, 形成两个 $N_T \times N_Q$ 的矩阵 \mathbf{U}_O 和 \mathbf{U}_D , 它们的索引 (i, j) 分别代表模板和现场指纹的局部特征, $U_O(i, j)$ 的值由式 (1) 计算得到, $U_D(i, j)$ 的值由式 (4) 计算得到. \mathbf{U}_O 中的第 i 行表示模板中 $\mathbf{o}(m_i^T)$ 与 O^Q 中每个元素的相似性, 值越小相似程度越高, 可取第 i 行的最小值与预设阈值 τ_{s0} 进行比较, 若小于则认为在 O^Q 中存在一个与 $\mathbf{o}(m_i^T)$ 匹配的特征. O^T 和 O^Q 的相似度得分可按下式计算

$$S_{s0} = N_O / \min(N_T, N_Q) \quad (7)$$

其中 N_O 是 O^T 和 O^Q 中匹配的特征总数量.

同理, 取 \mathbf{U}_D 第 i 行的最小值与预设阈值 τ_{sd} 进行比较, 若小于则认为在 D^Q 中存在一个与 $\mathbf{d}(m_i^T)$ 相似的特征. 采用式 (8) 计算 D^T 和 D^Q 的相似度得分

$$S_{SD} = N_D / \min(N_T, N_Q) \quad (8)$$

其中 N_D 是 D^T 和 D^Q 中相似特征的总数量.

最后, 计算模板和现场指纹的相似度总得分为

$$S_s = \omega_0 S_{S_0} + \omega_D S_{SD} \quad (9)$$

其中 ω_0 和 ω_D 为权重因子, 可由实验确定. 由 S_s 可以得到模板和现场指纹的匹配程度, 其值越高, 两者越匹配, 由 S_s 可以计算出接受者操作特性 (ROC, receiver operating characteristic) 曲线.

2 匹配算法的 GC 实现

在认证模型中, 服务器持有 $\{O^T, D^T\}$, 而用户输入为 $\{O^Q, D^Q\}$, 双方交互计算 S_s , 但各自的数据不能泄露给对方, 且 S_s 的结果只能由服务器知道, 用户不能获知. 安全模型采用半诚实模型 (semi-honest model), 即双方依照认证协议执行各自任务, 但可保留交互过程中的中间数据, 以此分析对方数据. 采用 GC 实现上节中匹配算法. GC 的特点是首先设计完成特定功能的二进制门电路, 服务器对于电路的每一条线 (位) 分配一个二进制加密值 (garbled value), 电路输入输出的关系由 garbled 表描述而非真值表. 服务器将自己输入对应的 garbled 值以及 garbled 表发送给用户. 用户的输入则通过不经意传输协议 (OT, oblivious transfer)^[8] 获得对应的 garble 值, 而服务器不知道用户真实数据. 用户根据双方的 garbled 值和 garbled 表, 计算电路输出的 garble 值, 该值回送服务器后, 由服务器解密即可得到真实数值.

2.1 基本 GC 模块

Kolesnikov 等^[9]给出了一些基本的 GC 模块, 如加法器 (ADD)、减法器 (SUB)、选择器 (MUX)、比较器 (CMP). 在此基础上, 设计了能实现匹配算法的电路. 图 2 给出的电路 ABS 以 μ 位 w_1 和 w_2 二进制数为输入, 输出 o 为 $|w_1 - w_2|$. 图中 SUB 为减法器, 其输入为有符号数, 以补码形式表示^[9], 最高位 (MSB) 为符号位. 图 3 给出了相对角度的计算电路 ROD, α 和 β 为输入的角度, 其输出为 $|\alpha - \beta|$ 和 $360 - |\alpha - \beta|$ 中较小数.

2.2 匹配算法的 GC 实现

图 4 给出了根据式 (1) 计算两个纹线特征相似函数电路 SOC 的结构, 它的输入是两个纹线特征 $o(m_i^T) = \{\alpha_{k,l}^i\}$ 和 $o(m_j^Q) = \{\beta_{k,l}^j\}$. 图中每个 CMP 计算式 (3) 的符号函数, 由于它的输出只有 1 位, 为减少电路复杂度, 不采用普通加法器计算式 (1) 中连加, 而采用文献 [4] 中的计数器电路 COUNTER 对

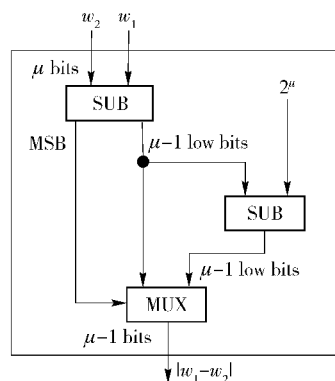


图 2 绝对值电路 ABS

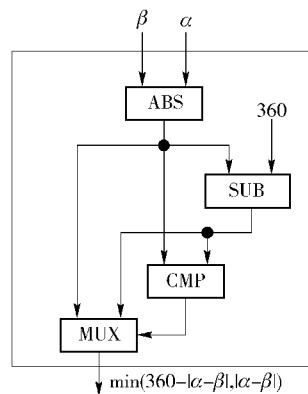


图 3 相对角度计算电路 ROD

“1”的数量进行累加。

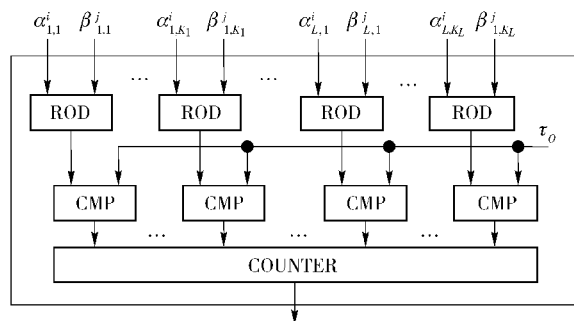


图 4 纹线方向特征相似函数计算电路 SOC

图 5 给出了模板和现场指纹的相似性得分计算电路 SSOC, 其输入是模板纹线特征集合 O^T 和现场指纹纹线特征集合 O^Q , 由于 GC 的除法电路非常复杂, 故该电路只输出两个集合中匹配上的纹线特征的总数量 N_0 , 用户将 N_0 的 garbled 值传给服务器后, 由后者解密得到真实值后根据式 (7) 计算得分 S_{S_0} . 为简单起见, 图 5 只画出了计算矩阵 U_0 的第 1 行和最后一行的电路, 其他省略, 用 \dots 表示. 此外, 因为求 N_0 个数最小值的电路复杂度高于 N_0 个比较

电路,且实际上无需知道 $o(m_i^T)$ 在 O^Q 中确切的匹配特征的索引,为降低电路的复杂度,图 5 的电路没有采用 1.3 节中的先搜索矩阵 U_0 每行最小值,然后与

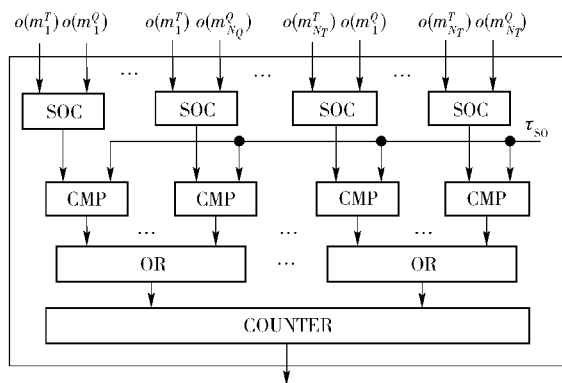


图 5 匹配上的纹线方向特征数量计算电路 SSOC

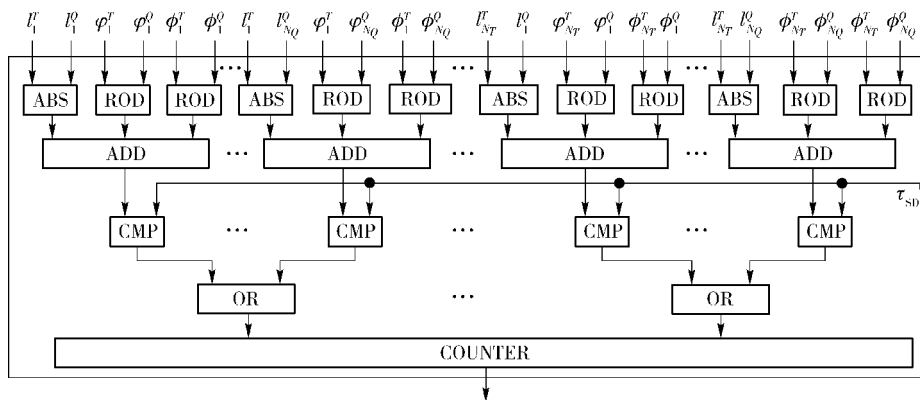


图 6 匹配上的细节点局部结构数量计算电路 SSDC

3 实验结果与分析

在 MyEclipse 10 环境中开发了相应的 Java 程序实现上述匹配电路所对应的 GC, 使用了 Huang 等^[10]开发的 GC Java 库, 并在 FVC2002-DB2 指纹库上按照 FVC2002 竞赛规则测试了真实接受率 GAR (genuine accept rate) 和认假率 FAR (false accept rate)。该库有 800 幅指纹图像, 图像大小为 296×560 像素。由于 GC 只能进行整数计算, 故需对各特征数据作量化处理, 用 7 位表示一个数据, 将两种特征的每个分量线性映射成 $[0, 127]$ 的整数。一些主要参数选择如下: 1) 对于纹线特征, $L = 3, r_1 = 30, r_2 = 48, r_3 = 66, K_1 = 10, K_2 = 16, K_3 = 22, \tau_0 = 6$; 2) 对于局部细节点结构, $l_{\min} = 20, l_{\max} = 90, \tau_{SD} = 22$ 。这些参数由经验选择。GC 的参数选择如下: 每个门的每根导线采用 80 位 garbled 值, 统计安全参数为 80 位。计算环境为 1 台 Dell 商用机 (CPU 为英特尔酷睿 i3-2120 处理器 3.30 GHz, 内存 8 G DDR3) 和 1

阈值 τ_{SO} 比较的方法, 而是将每行的 N_Q 个 SOC 的输出直接与 τ_{SO} 比较, 所有结果进行或运算, 对 N_T 行的结果进行计数就可以得到 N_0 。

图 6 给出了计算模板和现场样本的细节局部结构特征匹配数量的电路 SSDC, 图中只画出了计算矩阵 U_D 的第 1 行和最后一行的电路, 其他省略, 用 \dots 表示。矩阵元素代表相似值, 由计算式 (4) 而得到, 图 6 中由一个 ABS, 两个 ROD 和一个 ADD 电路计算。与纹线匹配类似, 对于矩阵每一行相似值, 不搜索最小值, 而是直接与阈值 τ_{SD} 比较后, 经 OR 电路, 最后送计数器 COUNTER 确定匹配上的细节点局部结构数量 N_D 。该值的 garbled 值被用户送到服务器后, 由后者根据式 (8) 计算得分 S_{SD} 。最后模板和现场指纹的总得分由服务器根据式 (9) 计得到。

台 Dell 服务器, 100 M 局域网。

图 7 给出了由 GC 实现的提出算法在 FVC2002-DB2 上的 ROC 曲线, 作为对比, 作者还分别测试了纹线特征和细节点结构单独匹配时的 ROC 曲线。从图中可以看出细节点结构的匹配性能优于纹线特征, 而提出算法将两种特征在得分层进行融合, 提高了匹配性能, 3 种方法的等错误率分别为: 9.1%

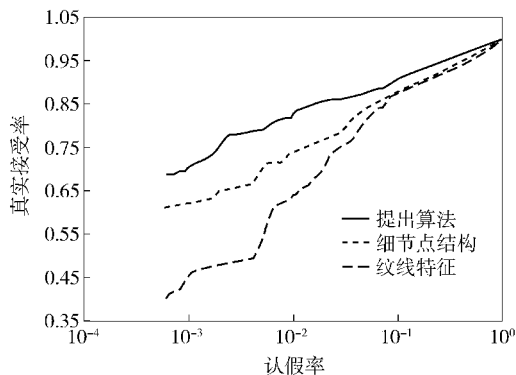


图 7 提出方法在 FVC2002-DB2 上的 ROC 曲线

(提出算法), 11.5% (细节点结构), 12.9% (纹线特征). 由 GC 实现的提出算法运行时间比较长, 在 FVC2002-DB2 上两枚指纹进行匹配的平均时间是 217.4 s, 该时间包括 OT 协议和 GC 的执行的总时间, 但不包括 OT 协议和 GC 的初始化的时间.

4 结束语

几乎所有基于全局特征的指纹匹配均需要将两枚指纹对齐, 而这需要复杂的算法. 采用两种细节点周围的局部特征: 纹线特征和局部结构作为匹配依据, 这些特征具有旋转和平移不变性, 对图像质量不高的指纹也能适用. 作者用 GC 实现了两种特征的匹配算法, 并在公开指纹数据库 FVC2002-DB2 上进行了性能测试, 实验结果表明提出方法有较高的精确度, 但运行时间较长. 由于提出方法中, 很多电路都是并行工作的, 可以考虑采用现场可编程门阵列 (FPGA, field programmable gate array) 实现 GC, 以减少运行时间.

参考文献:

- [1] Rappe D K. Homomorphic cryptosystems and their applications [D]. Dortmund, Germany: University of Dortmund, 2004.
- [2] Lindell Y, Pinkas B. A proof of Yao's protocol for secure two-party computation [J]. *Journal of Cryptology*, 2009, 22(2): 161-188.
- [3] Huang Yan, Malka L, Evans D, et al. Efficient privacy-preserving biometric identification [C] // 18th Network and Distributed System Security Conference, San Diego, California: Internet Society, 2011: 6-9.
- [4] 冯全, 杨梅, 康立军, 等. 基于二进制加密电路的指纹细节点匹配 [J]. *四川大学学报: 工程科学版*, 2013, 45(2): 75-80.
Feng Quan, Yang Mei, Kang Lijun, et al. Minutiae matching based on garbled circuits [J]. *Journal of Sichuan University: Engineering Science Edition*, 2013, 45(2): 75-80.
- [5] Nandakumar K, Jain A K, Pankanti S. Fingerprint-based fuzzy vault: implementation and performance [J]. *IEEE Trans. on Information Forensics and Security*, 2007, 2(4): 744-757.
- [6] Tico M, Kuosmanen P. Fingerprint matching using an orientation-based minutia descriptor [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003, 25(8): 1009-1014.
- [7] Kisel A, Kochetkov A, Kranauskas J. Fingerprint minutiae matching without global alignment using local structures [J]. *INFORMATICA*, 2008, 19(1): 31-44.
- [8] Naor M, Pinkas B. Efficient oblivious transfer protocols [C] // Twelfth Annual ACM-SIAM Symposium On Discrete Algorithms (SODA), Washinton DC, Association for Computing Machinery, Inc. , 2001: 448-457.
- [9] Kolesnikov V, Sadeghi A, Schneider T. Improved garbled circuit building blocks and applications to auctions and computing minima [C] // 8th International Conference on CANS'09, Kanazawa, Japan: Springer, 2009: 1-20.
- [10] Huang Yan, Evans D, Katz J, et al. Faster secure two-party computation using garbled circuits [C] // 20th USENIX Security Symposium, San Francisco, California: USENIX Association, 2011: 1-16.