第 35 卷 第 3 期 计 算 机 工 程 2009 年 2 月
Vol.35 No.3 Computer Engineering February 2009
·软件技术与数据库· 文章编号：1000—3428(2009)03—0063—03 文献标识码：A 中图分类号：TP311

基于 PayWord 的自更新 Hash 链微支付协议

孟 健，杨 阳
(武汉大学信息管理学院，武汉 430072)

摘 要：针对基于 Hash 链的小额支付协议 PayWord 在支付效率和安全性方面的不足，以及传统 Hash 链的应用存在长度限制的问题，提出一种新的适用于移动电子商务认证与微支付的协议，包括“多面额”Hash 链思想、可自更新的 Hash 链机制和基于令牌的快速认证方法。分析结果证明新协议可实现安全性、公平性和效率的统一，适用于移动用户与同一网络信息服务提供商进行频繁小额交易的移动商务环境。
关键词：移动商务；微支付协议；Hash 链；认证

Self-renewal Hash Chains Micro-payment Protocol Based on PayWord

MENG Jian, YANG Yang
(School of Information Management, Wuhan University, Wuhan 430072)

【Abstract】In accordance with the shortcomings in efficiency and security of PayWord, which is a mobile micro-payment protocol based on conventional Hash chains that has a length limit, this paper brings up a new protocol suitable for mobile commerce certification and micro-payment, involving the idea of “multiple denominations” Hash chains, a self-renewal Hash chains scheme and a token-based authentication method. Analysis results show that the protocol has better security, efficiency and fairness, and it is fit for frequent micro-payment between a mobile user and a fixed information service provider.
【Key words】mobile commerce; micro-payment protocol; Hash chains; authentication

1 概述

随着互联网和无线通信的发展，在移动互联网基础上进行商务活动成为一种全新的服务体验。作为移动商务的核心，移动支付占有重要地位。移动支付按交易额可分为宏支付和微支付，现有的移动支付大多是微支付。微支付是由移动运营商和金融机构等共同推出的实现小额支付的移动增值业务，其交易费用从用户话费中扣除，不涉及到银行的直接参与。微支付的出现，使手机由通信工具变成具有信用卡功能的支付工具，促进了移动商务的发展。然而，由于无线网络和移动终端设备数据处理功能的限制，移动商务交易主体的可靠性认证及支付效率和安全性等问题成为当前移动支付的障碍。因此，本文结合多面额[1]、可自更新的 Hash 链机制[2]和基于令牌的快速认证思想，设计了一种新的基于 PayWord[3]的移动商务微支付协议。

2 移动微支付协议

2.1 协议抽象模型

协议主要包括 3 个参与实体：

(1)顾客 *C*(Client)，即移动用户。*C* 购买商品或服务并进行支付。

(2)商家 *M*(Merchant)，即网络信息服务提供商。*M* 提供商品或服务并接受支付。

(3)*C* 和 *M* 信任的第三方经纪人 *B*(Broker)。*B* 为 *C* 和 *M* 建立、维护一个与银行相关联的账户和共享密钥，并进行信用担保，解决争端和支付转账。

协议抽象模型如图 1 所示。

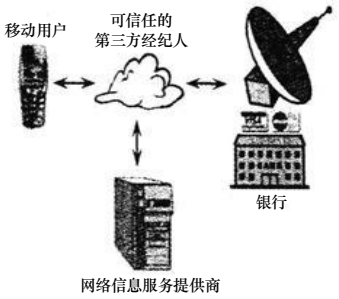

图 1 简单的协议抽象模型

协议参数说明如下：

(1)ID_C, ID_M 和 ID_B 分别代表 *C*, *M*, *B* 的身份标识；

(2)PK_C, PK_M 和 PK_B 分别代表 *C*, *M*, *B* 的公钥；

(3)SK_C, SK_M 和 SK_B 分别代表 *C*, *M*, *B* 的私钥；

(4)$SessionKey_{BC}$ 代表 *B* 和 *C* 共享的会话密钥，$SessionKey_{CM}$ 代表 *B* 给 *C*, *M* 分配的一次交易会话密钥；

(5)$(X)K_A$ 表示实体 *A* 用其密钥 *K* 对 *X* 进行加密；

(6)$H(X)$表示对消息 *X* 进行单向 Hash 运算，生成 *H* 函数摘要；

(7)$(H(X))SK_A$ 表示实体 *A* 用其私钥 SK_A 加密 *X* 的 *H* 函数摘要，形成数字签名；

(8)$A \rightarrow B$：$\{X\}$表示 *A* 向 *B* 发送消息 *X*；

作者简介：孟 健(1967－)，女，副教授，主研方向：信息系统，电子商务；杨 阳，硕士研究生
收稿日期：2008-06-23 **E-mail：**yangyang_2897@yahoo.com.cn

(9)TS_C, TS_M 和 TS_B 分别表示 C, M 和 B 产生的时戳；

(10)R_C, R_M 和 R_B 分别表示 C, M 和 B 产生的随机数。

2.2 开户子协议

C 和 M 分别选择 1 个匿名标识 ID_C 和 ID_M 在 B 处开户，B 将 ID_C, ID_M 与各自对应的真实身份信息进行绑定，并存储到账户管理数据库中。具体过程如下：B 为 C 和 M 各建一个账户，颁发 PayWord 证书 C_C 和 C_M(C_c=H(ID_B, ID_C, PK_C, *Expire*)SK_B，其中，*Expire* 为证书有效期)，并授权 C 生成不同面额的 Hash 链(即电子货币或支付凭证)进行购买。这些 Hash 链和现实生活中不同面额的货币类似，当涉及交易额相对较大的购买时，采用多面额的 Hash 链可以提高支付效率。B 和 C 还共享一个会话密钥 $SessionKey_{BC}$。同时，C 在银联系统进行手机与银行账号的绑定，一旦手机话费余额不足可快速充值。

2.3 认证子协议

当 C 和 M 首次交易时，须协商一个安全 Hash 函数 h，其安全参数为 k。C 执行：

(1)随机选取秘密种子 $s \in_R \{0,1\}^k$，计算 n+3 个 Hash 链节点：ω_0, ω_1,…, ω_{n+1}, ω，其中，$\omega_0=h(s)$, $\omega_i=h(\omega_{i-1})(1 \leqslant i \leqslant n+1)$，链尾$\omega=h(\omega_{n+1})$为验证锚。前 n 个链节点用作支付字据，ω_n 和 ω_{n+1} 用作认证令牌。

(2)生成首对一次签名(One-Time Signature, OTS)[4]私钥/公钥。随机选取 SK_{OTS1}, $SK_{OTS1}{}' \in_R \{0,1\}^k$ 作为私钥，计算对应公钥 $PK_{OTS1}=h(SK_{OTS1})$和 $PK_{OTS1}{}'=h(SK_{OTS1}{}')$。

(3)计算消息完整性验证码 $MIC0=h(h^{n+2}(s), PK_{OTS1}, PK_{OTS1}{}')$。

(4)$C \rightarrow B$：{($H(ID_M, \omega)$)SK_C}$SessionKey_{BC}$。

C 和 M 在交易前须完成基于令牌的身份认证。具体流程如下：

(1)$C \rightarrow M$：{ID$_C$, ID_M, TS_C, $Token_c{}^n$, (TS_C, $Token_c{}^{n+1}$, PTK)PK_B, $MIC1$}。其中，令牌 $Token_c{}^n=h^{n+1}(s)$, $Token_c{}^{n+1}=h^{n+2}(s)$；$PTK$ 为临时密钥对(Pairwise Transient Key)，PTK=PRF-256(ID_C, ID_M, R_C)，PRF-256 表示输出长度为 256 bit 的伪随机数生成函数；C 与 M 交易的会话密钥 $SessionKey_{CM}$ = *RightStr*(PTK, 128)，即通过右取 PTK 的 128 bit 获得其值；消息完整性认证码 $MIC1=HMAC(ID_C, ID_M, TS_C, Token_c{}^n, (TS_C, Token_c{}^{n+1}, PTK)PK_B)$，$HMAC$ 表示带密钥的 Hash 函数。

(2)$M \rightarrow B$：{ID_C, ID_M, TS_C, $Token_c{}^n$, (TS_C, $Token_c{}^{n+1}$, PTK)PK_B, $MIC2$}。其中，消息完整性认证码 $MIC2=HMAC(SK2_{MB}, ID_C, ID_M, TS_C, Token_c{}^n, (TS_C, Token_c{}^{n+1}, PTK)PK_B, MIC1)$，$SK2_{MB}$ 是 M 和 B 的 2 个共享子密钥之一，用于构造两者传递消息的完整性认证码 MIC。

(3)$B \rightarrow M$：{ID_C, ID_M, TS_B, SUCC, (PTK, ω)$SK1_{MB}$, $MIC3$}。其中，SUCC 是认证成功的标识；消息完整性认证码 $MIC3=HMAC(SK2_{MB}, ID_C, ID_M, TS_B$, SUCC, ($PTK$, ω)$SK1_{MB}$)，$SK1_{MB}$ 是 M 和 B 的 2 个共享子密钥之一，用于加密两者之间传递的数据。

(4)$M \rightarrow C$：{ID_C, ID_M, TS_C+1, SUCC, $MIC4$}。其中，消息完整性认证码 $MIC4=HMAC(PTK\_KCK, ID_C, ID_M, TS_C$+1, SUCC)，$PTK\_KCK$ 用于确认会话密钥以及构建保护密钥协商完整性的 $MAC4$，值为 PTK 最左端的 128 bit，即 PTK_KCK = *LeftStr*(PTK,128)。

2.4 服务请求子协议

(1)C 利用移动终端查看 M 的网站后，根据要购买的商品价格和数量，选择几种面额的 Hash 链 W_i，并计算出相应的长度 L_i，对每条 Hash 链选取 1 个随机数 R_i，计算该链的各节点值，即 $W_i=(\omega_{i0}, \omega_{i1},\cdots, \omega_{i,n+1}, \omega_i)$。其中，$i$ 为 Hash 链的面额，n 为 Hash 链的支付长度，$\omega_{i0}= h(R_i)$, $\omega_{ij} =h(\omega_{i,j-1})$ $(1 \leqslant j \leqslant n+1)$，链尾$\omega_i =h(\omega_{i,n+1})$为验证锚，$\omega_{in}$ 和$\omega_{i,n+1}$ 用作认证令牌，前 n 个链节点用作支付字据，ω_{i0} 为 Hash 链 W_i 的根。

(2)电子货币生成完毕后，C 向 B 发出服务请求：{ID_T, ID_M, $\omega_{10}, \omega_{20},\cdots,\omega_{i0}$, ($C_C$, I)PK_B, ($H(ID_T, ID_M, \omega_{10}, \omega_{20},\cdots,\omega_{i0}, C_C, I)$)$SK_C$}，其中，此次交易标识 ID_T 是一个随机数；I 为附加信息，包括各 Hash 链面额的说明及其长度、商品的简单描述信息；($H(ID_T, ID_M, \omega_{10}, \omega_{20},\cdots, \omega_{i0}, C_C, I)$)$SK_C$ 为支付承诺。

(3)B 收到 C 的服务请求后，利用 PK_C 验证 C 对支付承诺的签名，再用 SK_B 解密得到 C_C 和 I，通过 ID_C 检索到 C 的账户信息，检查 C 的存款余额是否足够支付 C 所需商品。如果 C 的请求不满足条件，则给 C 发送拒绝信息(如果 C 的现有话费不足，则向其发送提示信息。C 看到话费不够，就将银行卡号、密码、转账命令以及个人证书使用私钥签名，再用银行公钥加密，发送转账短信到银行。银行根据此短信，将钱转到 B 的顾客账号中，然后向 C 发送转账成功信息。C 收到此信息后，再次向 B 发出服务请求)；反之，B 就从 C 的账户中预扣 C 所要购买商品的金额 C_PrePay，同时在数据库中存储(ID_T, ID_M, ω_{10}, $\omega_{20},\cdots$, ω_{i0}, I, C_PrePay)的值来标识此次交易。然后 B 给 C 发送请求允许信息：{(ID_T, $SessionKey_{CM}$, OK)$SessionKey_{BC}$}，其中，$SessionKey_{CM}$ 为 B 给 C 和 M 分配的此次交易会话密钥，OK 为请求允许标识；同时，B 将 C 的服务请求发送给 M，即 $B \rightarrow M$：{ID_T, ID_M, ω_{10}, $\omega_{20},\cdots,\omega_{i0}$, ($C_C$, I, $SessionKey_{CM}$)PK_M, ($H(ID_T, ID_M, \omega_{10}, \omega_{20},\cdots,\omega_{i0}, C_C, I)$)$SK_C$, ($H(u)$)$SK_B$}，其中，$u$={$ID_T$, ID_M,ω_{10},ω_{20},…,ω_{i0}, (C_C, I, $SessionKey_{CM}$)PK_M, ($H(ID_T, ID_M, \omega_{10}, \omega_{20},\cdots, \omega_{i0}, C_C, I)$) SK_C}。服务请求过程如图 2 所示。

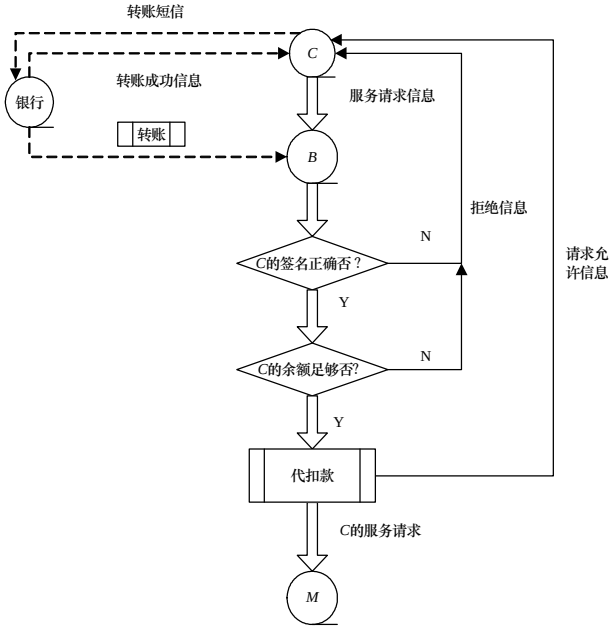

图 2 服务请求步骤

(4)M 收到 B 的服务请求后，先验证 B 的签名，然后使用 SK_M 解密得到(C_C, I, $SessionKey_{CM}$)，根据 C_C 和ω_{10}, ω_{20}, …, ω_{i0} 来验证此次交易的有效性，并验证 I 中有关商品信息的正确性，以及 C 对支付承诺的签名。验证通过后 M 将(ID_T, ID_M, ω_{10}, $\omega_{20},\cdots$, ω_{i0}, $SessionKey_{CM}$)保存到自己的数据库中。

2.5 交易子协议

M 根据每条 Hash 链的面额 i 和其长度 L_i 将商品分成 m 个价格不等的单位(m 为全部 Hash 链长度之和)，逐单位为 C 提供商品，C 则将链节点值ω_{i0}, ω_{i1}, …, $\omega_{i,\ n-1}$作为支付字据逐一发送给 M。该协议流程中 C 和 M 的第 r 次交互过程如下：

(1)$C\rightarrow M$：$\{(TS_C,\ t_C,\ \omega_r,\ PK_{OTSr},\ PK_{OTSr'},\ MIC_r)\ Session\ Key_{CM}\}$，其中，$\omega_r=(\omega_{ij},\ i,\ j)$，即 C 请求 M 提供相应的商品单位 P_r。

(2)M收到支付信息并解密得到ω_{ij}后，验证等式$\omega_{ij}=h(\omega_{i,j-1})$是否成立。若不成立，$M$ 拒绝提供服务；反之，$M\rightarrow C$：$\{(TS_M,\ t_M,\ P_r)SessionKey_{CM}\}$，其中，$t_C$, t_M 分别为 C 和 M 等候对方回复的有效时间段。

(3)C 验证 P_r 是否是所需的。若不是，则要求 M 重发 P_r；若是，则继续向 M 发送下一条请求。

交易结束时，M 只要把每条 Hash 链最后的支付信息(ω_{ij}, i, n)添加到数据库中即可。

2.6 清算子协议

(1)M 向 B 发送转账请求：$\{(ID_T,\ ID_M,\ \omega_{10},\ \omega_{20},\cdots,\omega_{i0},\ (C_C,\ I)PK_B,\ (H(ID_T,\ ID_M,\ \omega_{10},\ \omega_{20},\cdots,\omega_{i0},\ C_C,\ I))SK_C,\ TS_M,\ (\omega_{1p},\ 1,\ p),\ (\omega_{2q},\ 2,\ q),\cdots,(\omega_{in},\ i,\ n),\ (H(v))SK_M)SK1_{MB}\}$，其中，$v=\{ID_T,\ ID_M,\ \omega_{10},\ \omega_{20},\cdots,\omega_{i0},\ (CC,\ I)PK_B,\ (H(ID_T,\ ID_M,\ \omega_{10},\ \omega_{20},\ \cdots,\ \omega_{i0},C_C,\ I))SK_C,\ TS_M,\ (\omega_{1p},\ 1,\ p),\ (\omega_{2q},\ 2,\ q),\cdots,(\omega_{in},\ i,\ n)\}$。

(2)B 收到 M 的转账请求后，先验证 M 的签名，然后通过(ω_{1p}, 1, p), (ω_{2q}, 2, q),…,(ω_{in}, i, n)和 C 提交的验证锚$\omega(1)$, $\omega(2)$,…,$\omega(i)$验证等式$\omega_{1p}=h^p(\omega_{10}),\omega_{2q}=h^q(\omega_{20}),\cdots,\omega_{in}=h^n(\omega_{i0})$是否成立(其中，$\omega(i)$是面额为 i 的 Hash 链的验证锚)。若成立，B 再验证 C 对支付承诺的签名。验证通过后 B 计算出 C 应该支付的金额数，并从 C_PrePay 中扣除相应的金额存入到 M 的账户中。

(3)清算成功后，B 分别给 C 和 M 发送转账成功的确认信息：$\{ID_T,\ RD,\ (H(ID_T,\ RD))SK_B\}$，其中，$RD$ 表示本次交易清算成功的确认标识，包括交易总金额、清算时间等信息。这些处理过程都可以采用离线、定期集中结算方式的进行，不会成为瓶颈。至此完成全部交易过程。

3 协议性能分析

3.1 安全性分析

(1)机密性：C 和 B 之间采用会话密钥 $SessionKey_{BC}$ 加密信息，因此，可形成 C 与 B 之间的安全信道；而 M 和 B 在 Internet 环境下通信，也能保证他们之间的数据机密性。其次，C 使用 PK_B 对 C 和 M 之间的会话密钥 $SessionKey_{CM}$ 进行加密传输，即使攻击者截获了消息，但其不拥有 SK_B，也无法解密获得 $SessionKey_{CM}$，从而保证了 C 和 M 之间会话密钥的安全性。在交易子协议中，C, M 双方采用 $SessionKey_{CM}$ 对其间的交互信息进行加密，防止被非法用户截获、篡改。通过以上手段，确保了在整个交易过程中，只有交易参与者 C, B, M 知道交易的内容，移动终端在传送各种敏感信息时不会被轻易破译或盗用。

(2)完整性：在认证子协议中，每条认证消息都带有完整性验证码，实现了消息的完整性保护和消息源认证；在交易子协议中，又使用了 OTS 及消息完整性验证码 MIC_i，确保了传输位的完整性。

(3)可认证性：在认证子协议中，Hash 函数的单向性保证了令牌不可伪造，且令牌每次释放都是新鲜的，确保了令牌不能重放，非法用户没有正确的令牌就无法通过 B 的认证。同时非法用户没有合法公钥证书对应的私钥，从而保证了 C 的可认证性。而在清算子协议中，依据 Hash 函数的抗碰撞性以及运算不可逆等特点，可以确保支付字据的可认证性。

(4)不可否认性：该协议每条消息摘要都经过了主体的数字签名，它们可以有效地防范事后抵赖行为。

(5)抵抗拒绝服务攻击：B 通过 Hash 函数可以快速验证 C 的令牌，而 M 则可以采取地址过滤/阻塞等方法，降低非法(攻击)消息占用资源。

(6)有限的匿名性：C 与 M 采用匿名方式进行交易，只有 B 知道 C 的真实身份，在一定程度上保护了 C 的隐私权。

3.2 公平性分析

公平性是指在协议执行的任何阶段，协议中的诚实主体相对于其他主体不处于劣势[5]。该协议遵循公平性原则，最大程度地实现了支付字据和信息服务的公平交换：

(1)M 与 C 交易前，B 先核对 C 的账户，避免 C 恶意透支，保障了 M 的利益。

(2)M 与 C 进行交易时，M 对 C 信息商品的提供和 C 对 M 的支付都是逐步交互进行的，无论谁先中断双方都不可能得到好处。此外，交易子协议中的每条消息都添加了接收回复的有效时间段，促使双方按规定时间完成，也确保了协议的公平性。

(3)由 B 来指定交易会话密钥 $SessionKey_{CM}$ 进一步保证了交易的公平性。

3.3 效率分析

(1)交易过程采用多面额 Hash 链，支付所需散列值减少，验证电子货币合法性的散列运算也相应减少，提高了支付效率。另外，Hash 链更新过程中，存储和通信载荷的增加非常小，对移动设备和移动通信均不会造成负担。

(2)该协议充分利用了对称密码系统和 HMAC 计算速度快的特点，减少了计算数据的复杂性，且密钥分配可在离线的方式下解决，打破了移动终端处理数据能力受限的束缚，使支付效率大大提高，最终实现了安全、方便、快捷的移动支付。

4 结束语

本文设计的移动微支付协议可有效兼顾安全性、公平性和效率的统一，非常适用于移动用户与同一网络信息服务提供商长期、频繁地进行小额交易的移动商务场合。实用的移动微支付系统在技术上涉及到许多方面，本文的协议方案着重考虑了无线环境下的性能，但有线环境下的性能同样需要谨慎设计。另外，如何提高系统的吞吐量，当用户数量增加时对用户的响应仍然相对及时，从而使用户感觉不到系统的延时，有关这方面的问题仍有待进一步研究。

参考文献

[1] 李凌春. 基于 PayWord 的移动微支付模型的研究与设计[J]. 福建电脑, 2007, (7): 123-124.

[2] 陈　莉, 张浩军, 祝跃飞. 一种新的移动商务小额支付协议[J]. 计算机应用, 2007, 27(8): 2059-2061.

[3] Rivest R, Shamir A. Payword and Micromint: Two Simple Micropayment Protocols[C]//Proceedings of International Workshop on Security Protocols. Berlin, Germany: Springer-Verlag, 1996.

[4] Haller N, Metz C, Nesser P, et al. A One-time Password System[S]. RFC 2289, 2005.

[5] Asokan N. Fairness in Electronic Commerce[D]. Waterloo, Canada: University of Waterloo, 1998.