使用签密的认证邮件协议

王彩芬[1,2],葛建华[1],杜欣军[1],赵铁山[1],秦利卿[1]
(1. 西安电子科技大学综合务网国家重点实验室,陕西西安 710071;2. 西北师范大学数学与信息科学学院,甘肃兰州 730070)

摘 要: 认证邮件协议,指一个发送方要将消息 M 与收方收到的证据进行交换的协议. 在交换中最重要的一个性质就是要保证公平性. 通过对原有的签密方案修改得到一种适应于认证邮件协议的新签密方案,并且在该方案的基础上设计了新的认证邮件协议,新的协议效率高且弥补了其他认证邮件协议中的缺陷.

关键词: 电子商务; 公平交换协议; 认证邮件协议; 签密方案
中图分类号: TP309 **文献标识码**: A **文章编号**: 0372-2112 (2003) 03-0475-03

Certified Mail Protocols by Using Signcryption System

WANG Cai-fen[1,2], GE Jian-hua[1], DU Xin-jun[1], ZHAO Tie-shan[1], QIN Li-qing[1]
(1. *National Key Laboratory of ISN, Xiadian University, Xi'an, Shaanxi* 710071, *China*;
2. *College of Mathematics and Information Science, Northwest Normal University, Lanzhou, Gansu* 730070, *China*)

Abstract: Certified mail protocol is fair exchange of a message M for a receipt. Fairness is the most important property in exchange. We modify existing signcryption system so that it can be used by certified mail protocols, and based on the system we propose new certified mail protocols to guarantee fairness and low computation cost.

Key words: electronic commerce; fair exchange protocols; certified mail protocols; signcryption system

1 引言

公平交换中认证邮件协议(certified mail),指一个发送方要将消息 m 与收方收到的证据进行交换的协议. 在公平交换中为了保证公平性使用了可验证加密作为基础. 可验证加密在公平交换中起着重要作用,然而其效率不高. 最近提出的签密方案[1,2],是在一个逻辑步骤内同时实现签名和加密,可以有效地减少运算,如使用 DSA 签名方案,加密方案为 ELgamal 的可验证加密需要 6 个模的指数运算,而使用文[1]中的签密只需要 3 个模的指数运算. 该方案或其变型方案已被应用于 SET 协议中[3,4],而用于公平交换协议的签密方案还没有.

本文对签密方案作了修改,使得可用于认证邮件协议,并在此基础上设计出同步和异步情况下的认证邮件协议,新的协议保证了公平性,提高协议的执行效率并弥补了已有的认证邮件协议中的缺陷.

2 修改的签密方案

设发送方 A 和收方 B 的私/公钥对与文[1]中相同. 协议中可信第三方(TTP)的私/公钥对为 x_T/y_T,将要传递的邮件为 m, $M = H(m)$, H 是单向 *hash* 函数. 首先 A 随机选择 $x \in Z_q^*$,并计算 $k = hash_1(y_T^x \bmod p)$, $y = g^x \bmod p$, $c = E_k(m)$, $r = hash_2(y, c, M)$, $s = x/(r + x_a) \bmod q$,其中 $hash_1$, $hash_2$ 是单向 *hash* 函数. A 将 (c, r, s, M) 发送给 B, B 由此计算: $y = (y_a g^r)^s \bmod p$,验证 $r = hash_2(y, c, M)$ 从而确定签名是否可接受.

修改的方案中使用可信第三方的公钥计算 k,因此 k 只能由 TTP 计算. 这样 B 拥有 M,但不能直接解密得到 m. 这与认证邮件协议的性质是相符合的. 为了使 c 与 M 联系起来,使 $r = hash_2(y, c, M)$. 修改后的方案只适应认证邮件协议.

3 新的协议

3.1 异步情况下的认证邮件协议

异步情况下的协议在文[6]被提出,文[5]指出其中的缺陷并进行了修改,然而修改后的协议仍然存在一个缺陷即:协议中第 3 步 O 向 R 传递加密的密钥 K,则 R 得到 K 可以解密 C,从而获得消息 m,若解密后得的 m 与 $L = H(m, K)$ 不一致,则 R 调用 *Resolve* 子协议, R 向 TTP 发送用 TTP 公钥加密的 K,TTP 解密得到 K,若协议没有执行 *abort* 子协议,则 TTP 向 R 发送解密后的 K 及 *CON-K*, *EOR-K*. 但若 R 从 $C = eK(m)$ 中解密得到的 m 与 $L = H(m, K)$ 仍不一致时,协议仍然需要执行第 4 步. 改进的协议将会弥补这一点,并且因为使用签密方案作为基础使得效率也比较高. 新协议如下:

协议中的记号: O 表示协议中消息 m 的发送方; R 表示消息接收方; $O \rightarrow R$; m 表示 O 向 R 发送消息 m; $R \leftarrow$TTP: m 表示通过 ftp, R 从 TTP 处读取消息 m; $sig_U(\cdot)$ 表示 U 的签名 $U \in \{O, R, \text{TTP}\}$; h 是单向 *hash* 函数; (c, r, s, M) 与修改后签

收稿日期:2001-09-10;修回日期:2002-03-17
基金项目:国家教育部高校骨干教师资助项目

密方案中的记号相同.

$O \rightarrow R: m1 = (c, r, s, M)$,若 R 放弃,则终止

$R \rightarrow O: m2 = sig_R(m1, h(key_R))$, $h(key_R)$ 若 O 放弃,则执行 *abort* 子协议

$O \rightarrow R: m$ 若 R 放弃,则执行 *resolve-R* 子协议

$R \rightarrow O: key_R$ 若 O 放弃,则执行 *resolve-O* 子协议

Abort 子协议

$O \rightarrow$TTP: $m1, sig_O(m1)$

若 *resolved* 则 TTP $\rightarrow O: m2, h(key_R)$

否则 TTP $\rightarrow O: aborted = sig_{TTP}(m1)$

resolve $\rightarrow R$ 子协议

$R \rightarrow$TTP: $m1, sig_R(m1, h(key_R))$, $h(key_R)$, key_R

若已经 *abort* 则 TTP $\rightarrow R: aborted = sig_{TTP}(m1)$ 否则

TTP 首先检查 $sig_R(m1, h(key_R))$ 的正确性,若正确则从 $m1$ 计算 k,解密得到 m,

若 $M = H(m)$,则 TTP 执行

TTP $\rightarrow O: key_R, sig_{TTP}(k, sig_R(m1, h(key_R)))$

TTP $\rightarrow R: k, sig_{TTP}(k, sig_R(m1, h(key_R)))$

否则$(M \neq H(m))$

$R \leftarrow$TTP: $k, c, r, s, M, R, O, sig_{TTP}(k, c, r, s, M, R, O)$

$O \leftarrow$TTP: $k, c, r, s, M, R, O, sig_{TTP}(k, c, r, s, M, R, O)$

resolve-O 子协议

$O \rightarrow$TTP: $m1, sig_O(m1)$

若已经 *abort* 则 TTP $\rightarrow O: aborted = sig_{TTP}(m1)$ 否则

TTP 首先检查 $sig_O(m1)$ 的正确性,若正确则从 $m1$ 计算 k,解密得到 m,

若 $M = H(m)$ 则 TTP 执行

TTP $\rightarrow R: k, sig_{TTP}(k, sig_O(m1))$

TTP $\rightarrow O: k, sig_{TTP}(k, sig_O(m1))$

否则$(M \neq H(m))$

$R \leftarrow$TTP: $k, c, r, s, M, R, O, sig_{TTP}(k, c, r, s, M, R, O)$

$O \leftarrow$TTP: $k, c, r, s, M, R, O, sig_{TTP}(k, c, r, s, M, R, O)$

协议的执行:O 首先使用签密方案计算出 $m1 = (c, r, s, M)$ 并将其发送给 R,$m1$ 可以作为发送方发送 M 的证据;收到 $m1$ 后,R 计算 y 并验证 $r = hash(y, c, M)$ 成立与否,若不成立或 R 没有收到,R 可以终止协议的执行;若成立,则 R 把签名 $sig_R(m1)$ 及 $h(key_R)$ 发送给 O,作为收到 M 的证据;O 检验 R 签名的正确性,若不成立,则 O 执行 *abort* 子协议,否则 O 将消息 m 发送给 R;收到 m 后 R 检验 $H(m) = M$ 是否成立,若 $M \neq H(m)$ 则执行 *resolve-R* 子协议,否则,R 将收到消息 m 的证据 key_R 发送给 O;O 检验 key_R 与 $h(key_R)$ 是否一致,若不一致,则 O 执行 *resolve-O* 子协议,若一致则协议正常结束. O 得到 R 收到消息的证据 $m2$ 和 key_R,R 收到消息 m.

abort 子协议:该子协议只能由 O 执行. O 向 TTP 发出终止协议的请求,TTP 首先检验 $sig_O(m1)$ 的真伪,若签名正确且协议已经执行 *resolve* 子协议,则 TTP 将 $m2, h(key_R)$ 发送给 O,以便协议继续执行. 若协议没有执行 *resolve* 子协议,则 TTP 将 $aborted = sig_{TTP}(m1)$ 传给 O,表示协议终止.

Resolve-R 子协议:当需要时 R 将 $m1, sig_R(m1, h(key_R))$, key_R 发送给 TTP,若协议已经执行 *abort* 子协议,则 TTP 向 R 发送 $sig_{TTP}(m1)$ 表示协议的终止,否则 TTP 首先检查 $sig_R(m1, h(key_R))$ 的正确性,若正确则从 $m1$ 计算 k,解密得到 m,若 $M = H(m)$ 则 TTP 将密钥 k 及发送密钥的证据 $sig_{TTP}(k, sig_R(m1, h(key_R)))$ 传递给 R,将 $key_R, sig_{TTP}(k, sig_R(m1, h(key_R)))$ 传递给 O;否则,若 $M \neq H(m)$,则 TTP 将 k 和证据 $sig_{TTP}(k, c, r, s, M, R, O)$ 公布在一个公共路径中,以表示协议不成功.

Resolve-O 子协议:当需要时 O 将 $m1, sig_O(m1)$ 发送给 TTP,若协议已经执行 *abort* 子协议,则 *TTP* 向 O 发送 $sig_{TTP}(m1)$ 表示协议的终止,否则 TTP 首先检查 $sig_O(m1)$ 的正确性,若正确则从 $m1$ 计算 k,解密得到 m,若 $M = H(m)$,则 TTP 执行将密钥 k 及发送密钥的证据 $sig_{TTP}(k, sig_o(c, r, s, M))$ 传递给 O 和 R;否则若 $M \neq H(m)$ 则 TTP 将 k 和证据 $sig_{TTP}(k, c, s, M, R, O)$ 公布在一个公共路径中,以表示协议交换不成功.

下面仅分析协议的公平性. 协议的其他性质与文[6]中同.

协议的公平性:当协议正常结束时 O 有$(m2, key_R)$; R 有$(m1, m)$即 O 得到 R 收到 m 的证据,R 收到想要的邮件 m,所以协议的公平性成立;当 O 收到 m_2 若放弃,则 R 只有 m_1,不能得到 m,O 也只有 R 收到密文的证据. 所以公平性仍然成立;若 R 执行 *resolve-R* 子协议,则终止时

O 有$(m2, key_R, sig_{TTP}(k, sig_R(m1, h(key_R))))$

R 有$(m1, k, sig_{TTP}(k, sig_R(c, r, s, M)))$ R 由 k 可以解出消息 m,在这种情况下仍然有 O 得到 R 收到 m 的证据,R 收到想要的邮件 m,所以协议的公平性成立,或者

O 有$(m2, k, sig_{TTP}(k, c, r, s, M, R, O))$

R 有$(m1, k, sig_{TTP}(k, c, r, s, M, R, O))$,此种情况下 O 有 R 收到密文的证据,R 有 O 提供密文的证据;$k, sig_{TTP}(k, c, r, s, M, R, O)$ 用于说明 O 提供的消息 m 为假的证据,此时 O 和 R 都没有得到所要的,也没有泄露自己有用的信息,所以协议的公平性仍然成立;若 O 执行 *resolve-O* 子协议,则终止时

O 有$(m2, k, sig_{TTP}(k, sig_O(m1)))$;

R 有$(m1, k, sig_{TTP}(k, sig_O(m1)))$,分别表示 O 有 R 收到密文和解密密钥的证据,R 有 O 提供密文和 TTP 提供解密密钥的证据,所以协议的公平性成立;或者

O 有$(m2, k, sig_{TTP}(k, c, r, s, M, R, O))$;

R 有$(m1, k, sig_{TTP}(k, c, r, s, M, R, O))$ 分别说明 O 有 R 收到密文的证据,R 有 O 提供密文的证据;$k, sig_{TTP}(k, c, r, s, M, R, O)$ 用于说明 O 提供的消息 m 为假的证据,此时 O 和 R 都没有得到所要的,也没有泄露自己有用的信息,所以协议的公平性仍然成立. 综上可得出协议在执行的每个阶段

都保证了公平性.

新协议与文[5]中协议比较有如下优点:

(1) 效率高:新的协议中使用签密要比文[5]中协议的运算量小

(2) 当执行 *resolve* 子协议时,一旦出现 $M \neq H(m)$ 的情况,协议能够立即终止,并给出明确的说明. 这既提高了协议的效率,同时又弥补了文[5]中协议的缺陷,使 O 和 R 处于相同的地位.

3.2 同步情况下的认证邮件协议

文[7]中使用 CEMBS 的可验证加密给出了一种协议,该协议需要发送 4 条消息才能完成同步认证邮件协议. 使用上述签密方案后,本文的协议只要 3 条消息,并且关于模的指数运算也比原来减少. 协议如下:

(1) $O \rightarrow R: (c, r, s, M)$

(2) $R \rightarrow O: sig_R(c, r, s, M)$

(3) $O \rightarrow R: m$　若 R 没有得到或得到的 m 使 $M \neq H(m)$,则 R 执行

$R \rightarrow TTP: (c, r, s, M)$, $sig_R(c, r, s, M)$

TTP 首先检查 $sig_R(c, r, s, M)$ 的正确性,若正确则从 (c, r, s, M) 计算 k,解密得到 m,若 $M = H(m)$ 则 TTP 执行

$TTP \rightarrow R: k, sig_{TTP}(k, c, r, s, M)$

$TTP \rightarrow O: sig_R(c, r, s, M), sig_{TTP}(k, c, r, s, M)$

否则$(M \neq H(m))$

$R \leftrightarrow TTP: k, c, M, sig_{TTP}(k, c, r, s, M, R, O)$

$O \leftrightarrow TTP: k, c, M, sig_{TTP}(k, c, r, s, M, R, O)$

协议的执行:O 首先使用签密方案计算出 (c, r, s, M) 并将其发送给 R,(c, r, s, M) 可以作为发送方发送 M 的证据;收到 (c, r, s, M) 后,R 计算 y 并验证 $r = hash(y, c, M)$ 成立与否,若成立,则 R 将签名 $sig_R(c, r, s, M)$ 发送给 O,作为收到 M 的证据;O 检验 R 签名的正确性,若确为 R 的签名,则 O 将消息 m 发送给 R;收到 m 后 R 检验 $H(m) = M$ 是否成立,若成立,则协议结束. 这时 O 有 R 收到消息的证据,R 得到 O 发送的消息. 若 $M \neq H(m)$ 则 R 请求 TTP 的帮助,TTP 首先检验 $sig_R(c, r, s, M)$ 的真伪,若签名为真则 TTP 从 (c, r, s, M) 计算 k 和 y,然后解密 c 得到将要传递的消息 m,若 $H(m) = M$,则 TTP 分别将 k, $sig_{TTP}(k, c, r, s, M)$ 和 $sig_R(c, r, s, M)$, $sig_{TTP}(k, c, r, s, M)$ 传递给 R 和 O(其中 $sig_{TTP}(k, c, r, s, m)$ 表示 TTP 提供密钥 k 的证据);若 $M \neq H(m)$ 则 TTP 将 k 和证据 $sig_{TTP}(k, c, r, s, M, R, O)$ 公布在一个公共路径中,以表示协议交换不成功.

协议的比较:新协议在计算 c, r, s, M 时要比使用可验证加密需要的运算少;并且新协议中避免将消息 m 传递给 TTP,这样当消息 m 很大时可以有效地减轻 TTP 的负担.

协议的其他性质与文[7]中同.

4 结束语

公平交换协议在电子商务中有着重要作用,而认证邮件协议是比较常用的,设计实用高效的认证邮件协议是一项非常有意义的工作. 本文从改变协议使用的密码基础入手,将签密技术引入代替可验证加密,设计了两个认证邮件协议,这两个协议执行效率比较高,并且弥补了以前相应协议中的缺陷.

参考文献:

[1] Y Zheng. Digital signcryption or how to achieve cost(Signature and encryption) ≪Cost(Signature) + Cost(Encryption) [A]. Advance Cryptology-CRYPTO '97 [C]. LNCS1294, Berlin: Springer-Verlag, 1997. 169 - 179.

[2] Y Zheng. Signcryption and its application in efficient public key solutions [A]. Proc of Information Security Workshoup (ISW '97) [C]. LNCS1396, Japan: Springer-Verlag, 1998. 215 - 226.

[3] G Hanaoka, Y Zheng, H Imai. LITESET: A light-weight secure electronic transaction [A]. Proc of ACISP '98 [C]. LNCS1438, Australia: Springer-Verlag, 1998, 215 - 226.

[4] Moonseog Kwangjo Kim. Electronic funds transfer protocol using domain-verifiable signcryption scheme [A]. Information Security and Cryptology (ICISC '99) [C]. Springer-Verlag, 2000. 269 - 277.

[5] Jianyingzhou, Robert. Deng, Feng Bao. Some remarks on a fair exchange protocol [A]. Proc of PKC '2000 [C]. Australia: Springer-Verlag, 2000. 46 - 57.

[6] N Asokan, V Shoup, M Waider. Asynchronous protocols for optimistic fair exchange [A]. Proceeding of 1998 IEEE Symposium On Security And Privacy [C]. Oakland, USA, 1998. 86 - 99.

[7] Feng Bao, R H Deng, Wenbo Mao. Efficient and practical fair exchange protocols with off-line TTP [A]. In 1998 IEEE symposium on Security and Privacy [C]. Oakland, USA, 1998. 77 - 85.

作者简介:

王彩芬　女,1963 年出生于河北省安国市,1983 年获兰州大学数学系学士学位,1998 年获兰州大学计算机系硕士学位. 现为西北师范大学数学与信息学院副教授,西安电子科技大学通信学院博士生,主要从事密钥托管和电子商务中协议的设计与形式化分析方面的研究.

葛建华　男,1961 年出生于江苏南通,西安电子科技大学通信学院教授,博士导师,主要从事网络安全和 HDTV 方面的研究. 承担国家级项目若干项,发表论文多篇.