

一个公平的多方不可否认协议

韩志耕 罗军舟

(东南大学计算机科学与工程学院 南京 210096)

摘 要 实用的多方不可否认协议必须具备存活性、公平性、时限性、无排斥性和保密性. 文中指出现有典型协议, 如 KM, OZCL 和 OZL 均无法提供时限性和无排斥性, 而且易遭受服务失效等攻击, 致使它们不能成为实用的协议. 为此, 给出一个新协议 NKM, 其基于无需全局时钟同步机制支持的时间段概念实现时限性, 借助双重群加密技术确保具备保密性的同时不丢失无排斥性, 利用证据链技术既可高效维护协议证据, 又能避开服务失效和重放攻击; 同时还形式化验证了该协议的安全性, 并对协议部署时将牵涉到的安全问题进行了考虑. 与现有协议相比, NKM 在安全性和性能方面均存在优势, 可成为实用的协议.

关键词 多方不可否认; 时限性; 无排斥性; 服务失效攻击; 重放攻击

中图法分类号 TP393

A Fair Multi-Party Non-Repudiation Protocol

HAN Zhi-Geng LUO Jun-Zhou

(School of Computer Science and Engineering, Southeast University, Nanjing 210096)

Abstract Practical multi-party non-repudiation protocols must respect viability, fairness, timeliness, exclusion-freeness and confidentiality. In this paper, the authors point out that most of the existing representative multi-party non-repudiation protocols with online trusted third party, such as KM and its extensional version OZCL and OZL, lack the supports for the properties of timeliness and exclusion-freeness, and are vulnerable to denial of the non-repudiation service attack and so on. Bearing these issues in mind, the authors present a new protocol NKM, which respects timeliness with time-span notion, which does not need any global clock synchronism mechanism, and respects exclusion-freeness and confidentiality with double group encryption notion, and makes evidence managed efficiently and avoids potential denial of non-repudiation service attack and replay attack with evidence chain notion. Subsequently, the authors give a formal analysis of its security and put some consideration on some security issues of protocol deployment. Compared with existing protocols, NKM have advantage over them in terms of security and performance and can be a practical protocol.

Keywords multi-party non-repudiation; timeliness; exclusion-freeness; denial of the non-repudiation service attack; replay attack

1 引 言

不可否认服务是 ISO 定义的五项安全服务之

一, 用于收集、维护、验证与一个事件或行为相关的不可抵赖的证据, 以便解决有关该事件或行为是否发生的纠纷. 与其它安全服务不同, 不可否认服务不是用来防止非法用户的攻击, 而是为了防止通信对

方的欺骗行为. 单纯的数字签名等不可否认机制只能防止发送方否认, 更一般意义上的不可否认要通过专门的不可否认协议来实现. 由于不可否认提供的电子证据的法律效力必须得到社会法规的承认才有意义, 使得对不可否认协议的研究一直落后于其它密码协议. 通常认为实用的不可否认协议至少应具备不可否认性、公平性、时限性、保密性等基本性质. 实现不可否认协议的关键在于如何确保协议公平, 目前通常的做法是在协议中引入 TTP (Trusted Third Party)^[1]. 当前的研究主要集中在确保协议公平的同时, 进一步降低协议对 TTP 的依赖^[2-4].

多方涉入电子商务导致多方不可否认服务成为当前研究热点, 但至今仍无一个成熟通用的多方模型. 多方协议与两方协议相比, 除了实体数量存在差别, 协议的基本性质也不尽相同^[5], 通常多方协议还应具备无排斥性^[6]. 虽然目前在多方公平交换协议上存在一些研究, 但针对多方不可否认协议的研究却很少. 现有多方不可否认协议多数采用一对多协议拓扑结构, 协议的构造通常借助于扩展两方不可否认协议得到^[5,7-8].

文献[9]提出了一个著名的两方不可否认协议 (ZG), 协议提出后得到了广泛讨论, 主要包括: 基于多种方法验证安全性^[10]、多个协议变体^[11]、如何避免重放攻击和服务失效攻击^[12-14]. ZG 的主要缺陷在于不具备时限性, 文献[15]借助让收发双方都可限制消息发送的办法改进过该协议, 但文献[16]指出该方法高度依赖于全局时钟同步, 难于部署, 为此提出使用基于相对时钟概念的时间段技术重新改进了 ZG (新协议简称 NZG). 为使 ZG 可用于多方场合, 文献[5]通过多方扩展 ZG, 首次提出多方不可否认协议 (KM). 此后, 对 KM 的研究包括: 文献[7-8]考虑到其在单轮执行中只能组播单条消息, 并易遭受重放攻击, 通过扩充 KM 提出了 OZCL 和 OZL 协议; 文献[17]为使 KM 能高效部署, 基于面向事件的仿真模型对各实体引入的超时进行了量化评估. 然而, 本文指出 KM, OZCL 和 OZL 均不具备时限性和无排斥性, 也易遭受服务失效等攻击, 无法真正实用; 为此在现有协议的基础上, 通过多方扩展 NZG, 提出新协议 NKM.

本文第 2 节指出现有典型协议 KM, OZCL 和 OZL 存在的诸多缺陷; 第 3 节给出本文所采用的信道模型和双重群加密方案; 第 4 节提出 NKM 协议, 并讨论了协议执行及纠纷解决; 第 5 节形式化验证 NKM 安全性; 第 6 节将 NKM 与现有协议进行比

较, 并给出部署中相关安全问题的对策; 最后一节总结全文, 并指出下一步工作.

2 现有典型协议的缺陷

2.1 KM 协议

KM 是 ZG 的多方扩展^[5]. T 用于标识实体 A 及集合 B' 中成员 B'_j 获取 K 和 Con_K 的最终期限. 协议执行后, A 收集到证据 $\{EOO, Con_K\}$, B'_j 收集到证据 $\{EOR_i, Con_K\}$. 协议交互步骤如下 (见图 1 左):

1. $A \Rightarrow B: f_{EOO}, B, L, T, C, EOO;$
where $L = h(M, K)$ and
 $EOO = S_A(f_{EOO}, B, L, T, h(C));$
2. $B_i \rightarrow A: f_{EOR}, A, B_i, L, EOR_i;$
where $B_i \in B$ and $i \in \{1, 2, \dots, |B|\}$ and
 $EOR_i = S_{B_i}(f_{EOR}, A, L, T, C);$
3. $A \rightarrow TTP: f_{Sub}, B', L, T, E_{B'}(K), Sub_K;$
where $Sub_K = S_A(f_{Sub}, B', L, T, E_{B'}(K));$
4. $A \leftrightarrow TTP: f_{Con}, A, B', L, E_{B'}(K), Con_K;$
where $Con_K = S_{TTP}(f_{Con}, A, B', L, T, E_{B'}(K));$
5. $B'_j \leftrightarrow TTP: f_{Con}, A, B', L, E_{B'}(K), Con_K;$
where $B'_j \in B'$ and $\forall j: 1 \leq j \leq |B'|.$

KM 存在如下诸多缺陷:

(1) 时限性缺陷, 对接收者不公平. A 在期限 T 快到时将 Sub_K 提交给 TTP , 同时加强干扰 $B'_j (1 \leq j \leq |B'|)$ 所在网络或系统, 导致后者因无法检索 Con_K 而不能正常终止协议轮.

(2) 无排斥性缺陷, 对接收者不公平. 合法接收者集合 B' 由 A 确定, A 有可能利用优势地位将某些合法接收者排除在 B' 之外, 导致它们无法获得最终明文.

(3) 单一消息组播. 在要求组播不同消息的特殊场合, KM 需要多次执行.

(4) 不可否认服务失效攻击. 恶意 A 可故意选择 B 中成员不会赞同的 T , 并先执行协议步 3 和步 4, 再执行步 1, 导致 B 中成员提前终止协议轮. 若 A 频繁执行上述协议轮, TTP 将无法参与正常的不可否认服务.

(5) 信道安全时的重放攻击. 标签 $L = h(M, K)$ 无法唯一标识协议轮, 恶意 A 可重放 Con_K , 然而 B_i 却因无法关联自己所收集到的证据, 最终处于不利地位.

2.2 OZCL 和 OZL 协议

OZCL 和 OZL 协议是 KM 协议的扩充, 借助不同子密钥加密不同消息的办法, 使得协议在单轮执

行中组播多条消息^[7-8]. 以 OZCL 协议为例, 描述如下(见图 1 左):

1. $A \rightarrow IN: f_{EOO}, IN, B, T, request, C'$;
where $C' = l_1 c_1 x_1 u B_1 EOO_{c_1} \dots l_n c_n x_n u B_n EOO_{c_n}$ and $l_i = h(M_i, k_i)$ and $EOO_{c_i} = S_A(f_{EOO}, IN, B_i, l_i, x_i, u B_i, T, h(request), h(c_i))$;
2. $IN \rightarrow B_i: f_{EOOI}, B_i, A, T, l_i, c_i, x_i, u B_i, EOOI_i$;
where $EOOI_i = S_{IN}(f_{EOOI}, B_i, O, l_i, x_i, u B_i, T, h(c_i))$;
3. $B_i \rightarrow IN: f_{EOR}, IN, A, l_i, EOR_{c_i}$;
where $EOR_{c_i} = S_{B_i}(f_{EOR}, IN, A, l_i, x_i, u B_i, T, c_i)$;
4. $IN \rightarrow A: f_{EORI}, A, B', L', EORI$;
where $EORI = S_{IN}(f_{EORI}, A, B', L', T, h(request), h(C'))$;
5. $A \rightarrow TTP: f_{Sub}, TTP, IN, B', L', T, E_{B'}(K), EORI, h(request), h(C'), Sub_K$;
where $Sub_K = S_A(f_{Sub}, TTP, IN, B', L', T, E_{B'}(K), EORI)$;
6. $All \rightarrow TTP: f_{Con}, TTP, All, L', E_{B'}(K), EORI, Con_K$;
where $Con_K = S_{TTP}(f_{Con}, All, L', T, E_{B'}(K), EORI)$.

OZCL 仍存在如下诸多缺陷:

(1) 时限性缺陷. 与 KM 类似, A 仍可拖延密钥发送, 并通过干扰来阻止诚实 B'_i 接收 k_i , 使其无法正常终止协议轮.

(2) 无排斥性缺陷. 代理(IN)作为一种非完全可信实体, 可能会与 A 共谋将合法接收者 B_i 排除在集合 B' 之外.

(3) 不可否认服务失效攻击. 恶意 A 与 IN 串通, 故意选择 B 中成员不会赞同的 T , 并先执行步 1, 步 4~步 6, 再执行步 2, 导致 B 中成员提前终止协议轮. A 与 IN 频繁执行上述协议轮, TTP 同样无法参与正常的不可否认服务.

3 通信信道和群加密方案

文献[1,5]将公平不可否认协议使用的信道分为三类: 不可靠信道(unreliable channel)、弹性信道(resilient channel)和可用信道. 为简单起见, 下面仅对 NKM 使用的不可靠信道和弹性信道作如下定义.

定义信道之前先介绍 4 个元组 $\tau_S \langle S, M_S, T_S, Ch_S \rangle$, $\tau_R \langle R, M_R, T_R, Ch_R \rangle$, $\tau_C \langle S, C, R \rangle$, $\tau_F \langle S, F, Ch, R \rangle$. τ_S 为发送四元组, 在 T_S 时刻实体 S 向信道 Ch_S 载入信息 M_S . τ_R 为接收四元组, 在 T_R 时刻实体 R 从信道 Ch_R 收到信息 M_R . τ_C 为信道三元组, 实体 S 和 R 间的信道为 C . τ_F 为信道行为四元组, 行为

函数 F 可描述实体 S 和 R 间信道 C 的通信行为. 为与协议描述中信息发送符“ \rightarrow ”相区分, 信道定义中使用符号“ ∞ ”表示信息传递. 假设已存在元组 $\tau_C \langle A, Ch, B \rangle$, $\tau_S \langle A, M, T, Ch \rangle \vee \langle B, M, T, Ch \rangle$, $\tau_R \langle B, M', T+t, Ch \rangle \vee \langle A, M', T+t, Ch \rangle$, $\tau_F \langle A, f, Ch, B \rangle \vee \langle B, g, Ch, A \rangle$, 则有如下定义.

定义 1(不可靠信道). Ch 为不可靠信道(unreliable channel), 仅当至少满足下列命题之一:

(1) $A \infty B: f(\langle A, M, T, Ch \rangle) = \langle B, M', T+t, Ch \rangle$, where $M \neq \emptyset \wedge \{((M \equiv M' \vee (M \neq M' \wedge M' \neq \emptyset)) \wedge 0 < t < +\infty) \vee (M' = \emptyset \wedge t = +\infty)\}$.

(2) $B \infty A: g(\langle B, M, T, Ch \rangle) = \langle A, M', T+t, Ch \rangle$, where $M \neq \emptyset \wedge \{((M \equiv M' \vee (M \neq M' \wedge M' \neq \emptyset)) \wedge 0 < t < +\infty) \vee (M' = \emptyset \wedge t = +\infty)\}$.

定义 2(弹性信道). Ch 为弹性信道(resilient channel), 仅当如下命题同时满足:

(1) $A \infty B: f(\langle A, M, T, Ch \rangle) = \langle B, M', T+t, Ch \rangle$, where $M \neq \emptyset \wedge M \equiv M' \wedge 0 < t < +\infty$.

(2) $B \infty A: g(\langle B, M, T, Ch \rangle) = \langle A, M', T+t, Ch \rangle$, where $M \neq \emptyset \wedge M \equiv M' \wedge 0 < t < +\infty$.

不可否认服务中信息的安全级别要高于普通信息^[14-15], 多方不可否认协议保密性通常依靠群加密来实现^[5]. KM 协议中, A 将 K 提交给 TTP 前采用了群加密, 故只有 B' 中成员才可获得 K . 然而, A 在群加密时可故意排斥某些合法的接收者. 为确保提供保密性时不丢失无排斥性, NKM 采用了双重群加密, 包括一次群加密和二次群加密. 具体做法如下: A 先将 K 群加密(一次群加密, 潜在接收者为 B 中成员)后提交给 TTP , 在 TTP 确定了集合 B' 后, 对收到的密文再次群加密(二次群加密, 潜在接收者为 B' 中成员), 并将最终密文公布给 B' 中成员. 这样在协议内部只有 B' 中成员才能获得 K , 集合 B/B' 中成员和 TTP 都无法解密出最终明文. 一次群加密采用文献[5]中的定义 14, 在此基础上定义二次群加密方案和有效双重群操作(若无特殊说明, 本文均使用有效操作).

定义 3(二次群加密). 存在加解密函数分别为 E 和 D 的一次群加密 \mathfrak{R} ; 则 \wp 为二次群加密, 仅当同时满足:

(1) \wp 中存在加密函数 $\varphi: E(M) \times K^n \rightarrow C'$, 其中 $E(M)$ 为采取一次群加密方案 \mathfrak{R} 后形成的密文, 且 $E(M)$ 必须属于函数 φ 的明文有限集, K 为加密密钥有限集, C' 为密文有限集;

(2) \wp 中存在解密函数 $\psi: C' \times K \rightarrow E(M)$;

(3) $E(m) = \phi_k(\varphi_{k_1 \dots k_n}(E(m))) \wedge m \in M \wedge k_1, k_2, \dots, k_n \in K \wedge \exists k'$ 为与 $\{k_1, k_2, \dots, k_n\}$ 中 k_i (其中 $1 \leq i \leq n$) 相对应的解密密钥。

定义 4(双重群加解密). 一次群加密方案 \mathfrak{R} , 加解密函数分别为 E 和 D , 明密文有限集分别为 M 和 C ; 二次群加密方案 \wp , 加解密函数分别为 φ 和 ψ , 明密文有限集分别为 M' 和 C' , 则 $\varphi(E(M)), D(\psi(C'))$ 为有效双重群加解密操作, 当且仅当 $E(M) \subseteq C \wedge D(C) \subseteq M \wedge \varphi(M') \subseteq C' \wedge \psi(C') \subseteq M' \wedge E(M) \subseteq M' \wedge \psi(C') \subseteq C = \text{TRUE}$.

4 NKM 协议

4.1 协议描述

NKM 假设 A, B_i 都事先知道 TTP 的标识, 且 A, B_i 和 TTP 都有自己的签名私钥, 并知道彼此签名的验证公钥. 同时, TTP 与 A 或 B_i 间都采用弹性信道(定义 2), A 与 B_i 间采用不可靠信道(定义 1). 此外, 各实体时钟速度差异不是太大, 在可接收的范围内, 该假设在现有条件下较容易满足。

NKM 交互步骤类似于两方 NZG, 但前者侧重于多方不可否认协议的时限性和服务失效攻击以及后者所不具备的无排斥性. NKM 基于以下技术: (1) 运用时间段概念确保协议在无需全局时钟同步的情况下也能提供时限性; (2) 采用双重群加密保证了协议提供保密性的同时不丢失无排斥性; (3) 借助不同密钥加密不同消息的办法使得单轮协议可交互多条消息; (4) 采用证据链技术^[18] 既能高效维护证据, 又可避开服务失效攻击和信道安全时的重放攻击. 其中时间段概念来自文献[16]: 时间点(T)表示时间轴上的一点; 如果 T_x 和 T_y 是时间点, 那么 $t_{xy} = |T_x - T_y|$ 就是时间段(t). 时间段只表示时间长短, 而不管这段时间的起始时刻. 协议描述中的基本符号同文献[5, 7-8], 交互步骤如下(见图 1 右):

1. $A \rightarrow B_i: f_{EOO}, B_i, t_A, l_i, c_i, x_i, uB_i, EOO_i$;
where $l_i = h(A, B_i, TTP, h(c_i), h(K))$ and $EOO_i = S_A(f_{EOO}, B_i, l_i, x_i, uB_i, t_A, h(c_i))$;
2. $A \rightarrow TTP: f_{Sub}, B, t_A, L, E_B(K), EOO, Sub_K$;
where $L = \{l_i | B_i \in B \wedge 1 \leq i \leq |B|\}$ and $EOO = \{EOO_i | B_i \in B \wedge 1 \leq i \leq |B|\}$ and $Sub_K = S_A(f_{Sub}, B, L, t_A, E_B(K), EOO)$;
3. $B_i \rightarrow TTP: f_{EOR}, A, l_i, x_i, uB_i, t_{B_i}, EOO_i, EOR_i$;
where $EOR_i = S_{B_i}(f_{EOR}, A, l_i, x_i, uB_i, t_{B_i}, c_i, EOO)$;
4. $A \leftrightarrow TTP: f_{Con}, A, B', L', T, t_A, tSet_{B'}, \varphi_{B'}(E_B(K)), EOR, Con_K$;

where $L' = \{l_i | B_i \in B' \wedge 1 \leq i \leq |B'|\}$ and $EOR = \{EOR_i | B_i \in B' \wedge 1 \leq i \leq |B'|\}$ and $tSet_{B'} = \{t_{B_j} | 1 \leq j \leq |B'|\}$;

5. $B_i \leftrightarrow TTP: f_{Com}, A, B', L', T, t_A, tSet_{B'}, \varphi_{B'}(E_B(K)), EOR, Con_K$;
where $Con_K = S_{TTP}(f_{Com}, A, B', L', T, t_A, tSet_{B'}, \varphi_{B'}(E_B(K)), EOO, EOR)$.

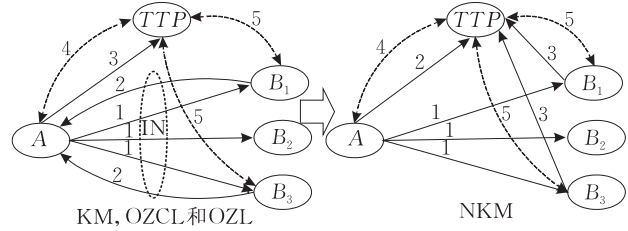


图 1 Online TTP 多方不可否认协议

针对信道安全时因协议轮标签不唯一而引发的重放攻击, 通常有两种解决方法: 让接收方在相应证据中添加现时(nonce)标签^[12]或在协议标签中包含 $h(c)$ ^[13]; 本文为 NKM 提供了一种新的解决方案: 由于采用了证据链技术, B_i 不直接将 EOR_i 提交给 A , 而且 EOR 又与 Con_K 一起由 TTP 同时发布, 故 A 无法借助重放 Con_K 实现攻击. NKM 中步 2 可在步 1 之前或步 3 之后执行; 同时步 2 与步 3 也可并发执行. 时间段 t_A 和 t_{B_i} 分别由 A 和 B_i 定义, 标识 TTP 存储 Sub_K 和 EOR_i 的最终期限; T 是 TTP 发布 Con_K 的时间点. 为防 A 或 B_i 发送假信息进行欺骗, 要求 A 将 EOO 及 B_i 将 EOO_i 发送到 TTP 进行验证, 故步 4 不可能在步 3 之前执行, TTP 不会为无效请求生成证据, 从而避免了不可否认服务失效攻击. 下面具体解释协议的各个交互步骤:

1. $A \rightarrow B_i$. 若 $B_i (1 \leq i \leq |B|)$ 没有从 A 处收到 EOO_i , 其不会产生 EOR_i , 故 A, B_i 都无法从 TTP 处获得证据.
2. $A \rightarrow TTP$. A 无需等待 B_i 确认就可向 TTP 提交 Sub_K . TTP 收到 Sub_K 后, 若私有目录中已有某些 B_i 发送的 EOR_i , 也不可立即产生 Con_K , 而须继续等待尽可能多的 B_i 发送 EOR_i , 等待的最长时间为 t_A (基于 TTP 时钟, 计时起点为 TTP 收到 Sub_K). 每当收到新的 EOR_i , TTP 就须将相应的实体标识 B_i 添加到集合 B' (初值为 t_A 计时前已提交 EOR_i , 且 EOR_i 仍保存在 TTP 处的那些实体标识中); 若在 t_A 内某些 $t_{B_j} (1 \leq j \leq |B'|)$ 已超时, 则 TTP 须将 B_j 从 B' 中移除. 若到 t_A 即将超时为止 B' 仍为 \emptyset , 则 TTP 可将 Sub_K 安全地删除, 不会引发纠纷.
3. $B_i \rightarrow TTP$. $B_i (1 \leq i \leq |B|)$ 收到 EOO_i 后, 若不同意 t_A 或不关心 M_i , 可不提交 EOR_i 给 TTP , 不会产生任何纠纷. t_A 并非限制 B_i 一定要在 t_A 内执行此协议步, 而是提醒 B_i 密钥 k_i 将在 t_A 超时后被删除. 即使 B_i 过了 t_A 时间单位后才提交 EOR_i , 协议也还是可能顺利完成的, 因为 A 有可能在

步 3 之后再执行步 2。

生成 Con_K ：上述 3 步完成后， TTP 收集到 Sub_K 和 $EOR_j (1 \leq j \leq |B'|)$ 。接着 TTP 将 EOR_j 中 EOO 和 Sub_K 中 EOO 逐一比较，若任意两个不等，则不会产生 Con_K 。为防止 TTP 无限期保存证据，其可预先定义时间段 t_0 ，表示证据发布了 t_0 个时间单位后将被删除。由于 TTP 与其它实体间采用弹性信道，故信道受到干扰的时间是有限的（设最长为 t_d ）；只要 TTP 将 t_0 定义为 $t_d + x (x > 0)$ ， A 和 B_j 总能从 TTP 处检索到 Con_K 。

4. $A \leftrightarrow TTP$ 。A 提交 Sub_K 后，就可以不断检索 TTP 是否发布了 Con_K ，若直到 $t_A + t_0$ 超时后还无法检索到 Con_K ，A 可确信 $B' = \emptyset$ ，并停止检索证据，不会产生纠纷。若其检索到 Con_K ，它能证明 $B_j (1 \leq j \leq |B'|)$ 必在时刻 T 与 $T + t_0$ 间从 TTP 处检索到 k_j 。同时 A 须保存 EOO 、 EOR 和 Con_K ，以防日后纠纷。

5. $B_i \leftrightarrow TTP$ 。 $B_i (1 \leq i \leq |B|)$ 提交 EOR_i 后，也需要不停地检索 Con_K 。若直到 $t_{B_i} + t_0$ 超时后还没检索到 Con_K ， B_i 可确信 TTP 未收到 Sub_K ，并可停止检索 TTP ，不会引发纠纷。若 B_i 检索到 Con_K ，其也未必能获得 k_i ，因为 B_i 可能由于 t_{B_i} 超时而被 TTP 从 B' 中移除。 B' 中成员需保存 EOO 、 EOR 和 Con_K 以便日后纠纷解决。

需要说明的是，以上针对 NKM 协议步骤的描述中并没有考虑在协议执行过程中， A 和/或 B_i 可能会通过拦截对方与 TTP 间的弹性信道（定义 2），并获取对方发送给 TTP 的协议消息所引发的重放攻击问题（该攻击会导致 NKM 协议无法实现公平）；与此攻击相关的描述与解决方案本文将在 6.2 节协议部署中详细讨论。

4.2 协议执行与纠纷解决

图 2 以描述 NKM 如何防止 B_{i+1} 试图阻止 A 从 TTP 处检索密钥确认证据（ Con_K ）为例，解释该协议如何严格控制各实体的操作时间。其中 TTP 时间轴上的粗线段表示该证据的生命期。协议执行中， $B_j (2 \leq j \leq i-1 \vee i+3 \leq j \leq n-1)$ 因不赞同 t_A 或不关心 M_j 放弃协议执行； B_i 因在 t_A 超时后才提交 EOR_i ， B_{i+2} 因 $t_{B_{i+2}}$ 定义太短，都被排除在 B' 之外；

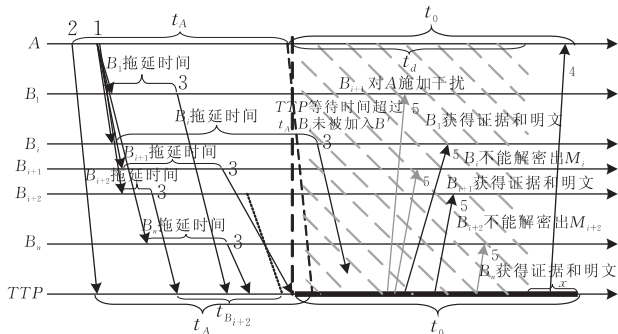


图 2 NKM 协议的特殊运行分析

B_{i+1} 在 t_A 快超时前才提交 EOR_{i+1} ，并持续地干扰 A 所在信道以阻止其检索 Con_K ，但由于最长干扰时间 t_d 不超过 t_0 ，故 A 总能在合适的时候检索到 Con_K 。

协议执行后引发的纠纷可分为两类：发送否认 (Repudiation of Origin) 与接收否认 (Repudiation of Receipt)。对纠纷的解决遵循如下假设：仲裁者 J 认为 TTP 有能力产生有效的证据，且不会与其它任何实体共谋欺骗；协议外的实体要想得到期望的信息，要么作为协议实体参与协议执行，要么设法破解协议所基于的加密原理。

(1) Non-repudiation of Origin. 若 $B_i (1 \leq i \leq |B|)$ 宣称收到 A 发送的信息 M_i ，但 A 否认向其发送过 M_i ， B_i 可将 $(A, B', L', T, t_A, tSet_{B'}, \varphi_{B'}(E_B(K)), l_i, x_i, uB_i, c_i, k_i)$ 和证据 EOO_i, EOR, Con_K 提交给 J ， J 做如下验证来判断 A 是否发送过 M_i 给 B_i ：

① 验证 Con_K 是 TTP 对 $(f_{Con}, A, B', L', T, t_A, tSet_{B'}, \varphi_{B'}(E_B(K)), EOR, Con_K)$ 的签名；

② 验证 EOO_i 是 A 对 $(f_{EOO}, B_i, l_i, x_i, uB_i, t_A, h(c_i))$ 的签名；

③ 验证 $B_i \in B'$ ；

④ 验证 $l_i = h(A, B_i, TTP, h(c_i), h(K))$ ；

⑤ 验证 $c_i = E_{k_i}(M_i)$ 。

若第①个验证正确， J 就认为 TTP 公布了 Con_K ，可推断 A 必发送过密钥 K （据此得子密钥 k_i ）给 TTP ；第②个验证正确说明 B_i 收到的 c_i 由 A 发送；第③个验证正确说明 B_i 是 TTP 确定的合法接收者；若最后两个也正确，则确信 A 发送过 M_i 给 B_i 。

(2) Non-repudiation of Receipt. 若 A 宣称已经发送信息 M_i 给 B_i ，但后者否认收到 M_i ， A 可将 Non-repudiation of Origin 中 B_i 提交给 J 的信息，另加 EOR_i 提交给 J ， J 做如下验证来判断 B_i 是否收到 M_i ：

① 验证 Con_K 是 TTP 对 $(f_{Con}, A, B', L', T, t_A, tSet_{B'}, \varphi_{B'}(E_B(K)), EOR, Con_K)$ 的签名；

② 验证 EOR_i 是 B_i 对 $(f_{EOR}, A, l_i, x_i, uB_i, t_{B_i}, c_i, EOR)$ 的签名；

③ 验证 EOO_i 是 A 对 $(f_{EOO}, B_i, l_i, x_i, uB_i, t_A, h(c_i))$ 的签名；

④ 验证 $B_i \in B'$ ；

⑤ 验证 $l_i = h(A, B_i, TTP, h(c_i), h(K))$ ；

⑥ 验证 $c_i = E_{k_i}(M_i)$ ；

若第①个验证正确， J 就认为 TTP 产生并公布

了 Con_K , 可推断 B_i 在 T 与 $T+t_0$ 间从 TTP 获得 k_i ; 第②个验证正确说明 B_i 收到 A 发送的 c_i 并愿从 TTP 处获得 k_i ; 第③个验证正确可断定 B_i 收到的 c_i 不是假的; 第④个验证正确说明 B_i 是 TTP 决定的合法接收者; 若最后两个也正确, 则确信 B_i 在 T 与 $T+t_0$ 之间收到了 M_i .

5 协议分析

多方不可否认协议分析的主要任务是考察参与协议的实体数量增加时, 一个实体的欺骗行为对其它实体造成的影响. 虽然目前安全协议的形式化分析方法有很多, 但还不存在专门用于多方不可否认协议的形式化分析方法, 已有的类似研究也仅是文献[19]使用有限状态工具 M_{OCHA} 分析过两个多方合同签名协议. NKM 协议具有无排斥性, 可发送多条消息, 且避免了服务失效和重放攻击是无争议的. 但同时 NKM 在 KM 的交互步骤中加入了时间控制信息, 还改变了协议交互步骤, 故须形式化验证 NKM 协议是否达到了 KM 协议目标并能提供时限性. 本文之所以选用文献[20]给出的扩展 SVO 逻辑方法验证 NKM 的时限性, 一方面是由于逻辑方法分析大系统时不会发生状态爆炸, 另一方面是由于文献[20]借助该方法从理论上证实了 ZG 无法提供时限性, 其也必然适用于验证具有一对多通信拓扑的 NKM 的时限性. 对 NKM 的分析仍按两步进行: 逻辑推理完全基于 SVO 逻辑的证明方法, 其正确性由基本的 SVO 逻辑保证; 时间演算基于初等代数和集合论, 其正确性在数学中已得到保证. 验证过程中需要使用的扩展后的公理包括 (其中 formula at $[T]$ 表示由公式 formula 描述的协议事件发生在由时间表达式 T 定义的时间点处):

Nec 规则: 由 $\vdash \varphi$ 有 $\vdash P$ believes φ .

A1. P believes $\varphi \wedge P$ believes $(\varphi \supset \psi) \supset P$ believes ψ ;

A4. $(PK_e(Q, K) \wedge R$ received $\{X\}_K^{-1}$ at $[T]) \supset Q$ said X at $[T]$;

A6. P received (X_1, X_2, \dots, X_n) at $[T] \supset P$ received X_i at $[T]$;

A7. $(P$ received $\{X\}_K$ at $[T_1] \wedge P$ has K' at $[T_2]) \supset P$ received X at $[\max(T_1, T_2)]$;

A13. P said (X_1, X_2, \dots, X_n) at $[T] \supset (P$ said X_i at $[T] \wedge P$ sees X_i at $[T])$.

首先根据 NKM 描述定义如下缩写, 以便保证

分析过程的可读性和简洁性:

$$c_i = \{M_i\}_{K_i};$$

$$EOO_{ip} = \{f_{EOO}, B_i, l_i, x_i, uB_i, t_A, h(c_i)\};$$

$$EOO_i = \{EOO_{ip}\}_{KA}^{-1};$$

$$EOR_{ip} = \{f_{EOR}, A, l_i, x_i, uB_i, t_{B_i}, c_i, EOO\};$$

$$EOR_i = \{EOR_{ip}\}_{KB_i}^{-1};$$

$$Sub_{Kp} = \{f_{Sub}, B, L, t_A, E_B(K), EOO\};$$

$$Sub_K = \{Sub_{Kp}\}_{KA}^{-1};$$

$$Con_{Kp} = \{f_{Con}, A, B', L', T, t_A, tSet_{B'},$$

$$\varphi_{B'}(E_B(K)), EOO, EOR\};$$

$$Con_K = \{Con_{Kp}\}_{KT}^{-1}.$$

分析过程中要用到的时间常元有 $T, t_A, t_{B'}, t_{B_i}, t_0$, 其中 t_0 表示网络不可用的最长时间 (NKM 假设网络非永久不可用), $t_A, t_{B'}$ 和 t_{B_i} 的取值都由协议信息中的相应字段确定. 时间变元有 $T_x, T_y, T_z, T_o, T_r, T_i, T_A, T_{B'}$. 设基本项集为 $\{\{A, B', TTP, J\}, \{f_{EOO}, f_{EOR}, f_{Sub}, f_{Con}, L_i, M_i\}, \{KA, KB'_j, KT, (KA)^{-1}, (KB'_j)^{-1}, (KT)^{-1}, K_i\}\}$. 其中 $i (1 \leq i \leq |B|)$ 与 $j (1 \leq j \leq |B'|)$ 的映射关系为: 令接收实体在 B 中标号为 i , 且其又被 TTP 加到 B' 中, 那么在 B' 中标号就为 j . $\{A, B'_j, TTP, J\}$ 是系统中的实体, $\{f_{EOO}, f_{EOR}, f_{Sub}, f_{Con}, L_i, M_i\}$ 是系统中的常量, $\{KA, KB'_j, KT, (KA)^{-1}, (KB'_j)^{-1}, (KT)^{-1}, K_i\}$ 是系统中的公/私钥, 共享密钥.

首先给出实体密钥假设. 每个实体拥有一个签名私钥, 且相应的公钥公开:

P1. (1) J believes $PK_e(A, KA)$, (2) J believes $PK_e(B_i, KB_i)$, (3) J believes $PK_e(TTP, KT)$;

P2. (1) J believes $(B_i$ has $KA)$, (2) J believes $(B_i$ has $KT)$.

协议运行完后有一个仲裁过程, A 和 B'_j 将自己收集的证据提交给仲裁者 J :

P3. J believes J received $\{EOO_i, EOR_i, Con_K\}$.

TTP 会将证据发布的时间 T 包含在发布的证据中:

P4. J believes $(TTP$ said $Con_{Kp} \supset TTP$ said Con_{Kp} at $[T])$.

协议中 TTP 必须称职, 即它必须在 A 和 B' 所有实体规定的时间内收到各方提交的信息后同时为 A 和 B'_j 产生证据, 而不只为其中一方产生证据, 结合定义 2 给出的弹性信道和时间常元 t_A, t_{B_i} 的含义, 有

P5. J believes $(TTP$ said Con_{Kp} at $[T_x] \supset$

TTP received Sub_K at $[T_y | \{x | T_x - t_A \leq x \leq T_x\}]$;

P6. J believes (TTP said Con_{K_p} at $[T_x] \supset TTP$ received EOR_i at $[T_y | \{x | T_x - t_{B'_i} \leq x \leq T_x\}]$).

TTP 与 A 和 B'_j 间采用弹性信道, 只要 TTP 发布了证据, A 和 B'_j 就一定能在此后一段时间内收到证据, 结合定义 2 给出的弹性信道以及时间常元 t_0 的含义, 有

P7. J believes (TTP said Con_{K_p} at $[T_x] \supset A$ received Con_K at $[T_y | \{x | T_x \leq T_y \leq T_x + t_0\}]$);

P8. J believes (TTP said Con_{K_p} at $[T_x] \supset B'_j$ received Con_K at $[T_y | \{x | T_x \leq T_y \leq T_x + t_0\}]$).

协议还假设 A 和 B_i 间的信道不可靠, 且 B_i 不会做对自己不利的事, 因此只有在收到 A 的 EOO_i 后才会发送 EOR_i , 结合定义 1 给出的不可靠信道, 有

P9. J believes (B_i said EOR_{ip} at $[T_x] \supset B_i$ received EOO_i at $[T_y | \{x | x \leq T_x\}]$).

最后假设关于 M_i 还原. 只要 B'_j 收到 (或 A 发送) 了 c_i 和 k_i , 它就收到 (发送) 了 M_i :

P10. J believes (A said c_i at $[T_x] \wedge A$ said k_i at $[T_y] \supset A$ said M_i at $[\max(T_x, T_y)]$);

P11. J believes (B'_j received c_i at $[T_x] \wedge B'_j$ received k_i at $[T_y] \supset B'_j$ received M_i at $[\max(T_x, T_y)]$).

NKM 的目标就是要保证协议运行完成后, 实体能通过所收集的证据让仲裁者相信通信行为已发生, 并能确定事件发生的时间满足约束条件:

G1. J believes (A said M_i at $[T_x] \wedge A$ received Con_K at $[T_y] \wedge (T_x \leq T_y \leq T_x + t_A + t_0)$);

G2. J believes (B'_j received M_i at $[T_x] \wedge B'_j$ said EOR_{ip} at $[T_y] \wedge ((T \leq T_x \leq T + t_0) \wedge (T_x - t_{B'_j} - t_0 \leq T_y \leq T_x))$).

G1 说明若 A 收到 TTP 发布的证据, 则它一定是在提交 Sub_K 到 TTP 后有限的时间内收到该证据的, 而不可能在证据被删除后收到; G2 也说明了这种时间约束关系. 故只要能证明上述两个目标, 就能说明 NKM 正确并具有时限性. 下面给出这两个目标的证明过程.

G1. J believes (A said M_i at $[T_x] \wedge A$ received Con_K at $[T_y] \wedge (T_x \leq T_y \leq T_x + t_A + t_0)$).

逻辑推理:

1. J believes (TTP said Con_{K_p} at $[T]$);
P1(3), P3, P4, Nec, A1, A4, A6.
2. J believes (A said k_i at $[T_s | \{x | T - t_A \leq x \leq T\}]$);
1, P1(1), P5, Nec, A1, A4, A13.

3. J believes (B'_j said EOR_{ip} at $[T_r | \{x | T - t_{B'_j} \leq x \leq T\}]$);
1, P1(2), P6, Nec, A1, A4.

4. J believes (B'_j received EOO_i at $[T_o | \{x | x \leq T_r\}]$);
3, P9, A1.

5. J believes (A said c_i at $[T_o]$);
4, P1(1), Nec, A1, A4, A13.

6. J believes (A said M_i at $[\max(T_o, T_s)]$);
2, 5, P10, A1.

7. J believes (A received Con_K at $[T_A | \{x | T \leq x \leq T + t_0\}]$);
1, P7, A1.

8. J believes (A said M_i at $[\max(T_o, T_s)] \wedge A$ received Con_K at $[T_A]$);
6, 7, Nec, A1.

时间演算:

令 $T_x = \max(T_o, T_s) \wedge T_y = T_A$, 现证明 $T_x \leq T_y \leq T_x + t_A + t_0$:

由步 7 有 $T_A \in \{x | T \leq x \leq T + t_0\}$, 结合 $T_y = T_A$ 得 $T \leq T_y \leq T + t_0$ (1). 由步 4 有 $T_o \leq T_r$ 以及由步 3 有 $T - t_{B'_j} \leq T_r \leq T$, 得 $T_o \leq T_r \leq T$. 由步 2 有 $T - t_A \leq T_s \leq T$, 结合 $T_x = \max(T_o, T_s)$ 得 $T - t_A \leq T_x \leq T$, 即 $T_x \leq T \leq T_x + t_A$ (2).

由 (1) 和 (2) 有 $T_x \leq T_y \leq T_x + t_A + t_0$, G1 得证.

G2. J believes (B'_j received M_i at $[T_x] \wedge B'_j$ said EOR_{ip} at $[T_y] \wedge ((T \leq T_x \leq T + t_0) \wedge (T_x - t_{B'_j} - t_0 \leq T_y \leq T_x))$).

逻辑推理:

1. J believes (TTP said Con_{K_p} at $[T]$);
P1(3), P3, P4, Nec, A1, A4, A6.
2. J believes (B'_j said EOR_{ip} at $[T_r | \{x | T - t_{B'_j} \leq x \leq T\}]$);
1, P1(2), P6, Nec, A1, A4.
3. J believes (B'_j received EOO_i at $[T_o | \{x | x \leq T_r\}]$);
2, P9, A1.
4. J believes (B'_j received c_i at $[T_o]$);
3, P2(1), A7, A6, Nec, A1.
5. J believes (B'_j received Con_K at $[T_{B'_j} | \{x | T \leq x \leq T + t_0\}]$);
1, P8, Nec, A1.
6. J believes (B'_j received k_i at $[T_{B'_j}]$);
5, P2(2), A7, A6, Nec, A1.
7. J believes (B'_j received M_i at $[\max(T_o, T_{B'_j})]$);
4, 6, P11, Nec, A1.
8. J believes (B'_j received M_i at $[\max(T_o, T_{B'_j})] \wedge B'_j$ said EOR_{ip} at $[T_r]$);
2, 7, Nec, A1.

时间演算:

令 $T_x = \max(T_o, T_{B'_j}) \wedge T_y = T_r$, 现证明 $(T \leq T_x \leq T + t_0) \wedge (T_x - t_{B'_j} - t_0 \leq T_y \leq T_x)$:

根据步 2, 可得 $T - t_{B'_j} \leq T_r \leq T$, 再由步 3, 可得

$T_o \leq T_r$, 故有 $T_o \leq T_r \leq T$. 根据步 5, 有 $T \leq T_{B'_j} \leq T + t_0$, 又因为 $T_x = \max(T_o, T_{B'_j})$, 所以必有 $T_x = T_{B'_j}$. 因此, 得 $T \leq T_x \leq T + t_0$ (1), 即 $T_x - t_0 \leq T \leq T_x$ (2). 由于 $T_y = T_r$, 由步 2 有 $T_r \in \{x \mid T - t_{B'_j} \leq x \leq T\}$, 即 $T - t_{B'_j} \leq T_y \leq T$. 结合等式 (2), 得 $T_x - t_{B'_j} - t_0 \leq T_y \leq T_x$ (3).

由等式 (1) 和 (3), G2 得证.

上述分析表明 A 总能保证在提交 Sub_K 之后的有限时间 ($t_A + t_0$) 单位内收到 TTP 发布的 Con_K . 同时 B'_j 也能在提交 EOR_i (i 与 j 之间遵循上述映射) 后的有限时间, 即在 TTP 发布证据 Con_K 的时间点 T 与时间点 ($T_y + t_{B'_j} + t_0$) 内收到证据 Con_K . 若 B'_j 在提交 EOR_i 后过了 $(t_{B'_j} + t_0)$ 个时间单位还没收到 Con_K , 就可以将 EOO_i 安全地删除, 不会产生纠纷. 因此 NKM 协议具有时限性.

6 进一步讨论

6.1 协议比较

NKM 满足公平的多方不可否认协议应具备的所有性质:

(1) 存活性. 若 $B_i (1 \leq i \leq |B|)$ 收到 A 发送的 EOO_i 后同意 t_A , 并愿意接收 M_i ; 同时 TTP 在 t_A 内收到所有 EOR_i , 且 TTP 在欲删除 Sub_K 前未有 t_{B_i} 超时, 则存在一个所有实体都参与的不可否认信息交换.

(2) 不可否认性. 协议成功执行后, A 和 $B'_j (1 \leq j \leq |B'| \wedge B' \subseteq B)$ 可分别获取证据 $\{EOR_j, Con_K\}$, $\{EOO_i, Con_K\}$ 用于证实通信行为的确发生过 (i 到 j 的映射同第 5 节).

(3) 强公平性. 协议执行后, A 和 $B'_j (1 \leq j \leq |B'| \wedge B' \subseteq B)$ 可分别在 $t_A + t_0$ 和 $t_{B'_j} + t_0$ 内收到证据.

(4) 时限性. 不考虑网络响应速度、 TTP 特性以及实体数量和自身能力带来的超时^[17], 协议最多将在开始后 $\max\{t_A, \max\{t \mid t \in tSet_B \wedge tSet_B = \{t_{B_i} \mid 1 \leq i \leq |B|\}\}\} + t_0$ 个时间单位内结束.

(5) 无排斥性和保密性. 协议由 TTP 确定最终合法接收者集合 B' , A 和/或 $B_i (1 \leq i \leq |B| \wedge B_i = B'_j \wedge 1 \leq j \leq |B'|)$ 无法通过串通排斥合法接收者 $B_{i'} (1 \leq i' \leq |B| \wedge B_{i'} = B'_{j'} \wedge 1 \leq j' \leq |B'|)$; 同时协议内部仅有 B' 中成员才可获得 K , $B \setminus B'$ 中成员和 TTP 都无法解密出最终明文.

从签名生成与验证、随机数生成、非对称加解密、模运算、数据存取等基本操作所产生的开销角度, 对 KM, OZCL, OZL 和 NKM 协议进行了比较. 在实体 A 和 B_i 上, NKM 居中 (表 1, 表 2); 在实体 TTP 上, NKM 中除多了 m 次非对称加密和 n 单位存储开销外, TTP 仍作验证中心使用 (表 3); 同时, 仅 OZCL 引入实体 IN , 故只有其存在相关方面的开销 (表 4); 在安全性 (表 5) 和事后部署 (表 6) 上, NKM 优于其它协议. 此外, NKM 与 n 个并行的两方 NZG 相比, 除实体 B_i 性能略为下降外 (表 9), A 和 TTP 的性能均有所提高 (表 7, 表 8).

表 1 实体 A 所需单位开销对比

| | KM 协议 | OZCL 协议 | OZL 协议 | 本文协议 |
|-------|-------|---------|--------|-------|
| 签名生成 | 2 | $n+1$ | $n+1$ | $n+1$ |
| 非对称加密 | m | $n+m$ | $n+m$ | $n+m$ |
| 存储开销 | $n+3$ | $n+4$ | $n+3$ | $n+2$ |

表 2 实体 B_i 所需单位开销对比

| | KM 协议 | OZCL 协议 | OZL 协议 | 本文协议 |
|-------|-------|---------|--------|------|
| 签名生成 | 1 | 1 | 1 | 1 |
| 非对称加密 | 1 | 2 | 2 | 2 |
| 存储开销 | 2 | 3 | 2 | 2 |

表 3 实体 TTP 所需单位开销对比

| | KM 协议 | OZCL 协议 | OZL 协议 | 本文协议 |
|--------------|-------|---------|--------|---------|
| 验证 A 的签名 | 1 | 1 | 1 | 1 |
| 验证 B_i 的签名 | 0 | 0 | 0 | 最多为 n |
| 签名生成 | 1 | 1 | 1 | 1 |
| 非对称加密 | 0 | 0 | 0 | m |
| 存储开销 | 2 | 2 | 2 | $n+2$ |

表 4 实体 IN 所需单位开销对比

| | KM 协议 | OZCL 协议 | OZL 协议 | 本文协议 |
|--------------|-------|----------|--------|------|
| 签名生成 | 0 | $n+1$ | 0 | 0 |
| 非对称加密 | 0 | 0 | 0 | 0 |
| 验证 A 的签名 | 0 | 最多为 n | 0 | 0 |
| 验证 B_i 的签名 | 0 | 最多为 n | 0 | 0 |
| 存储开销 | 0 | $2n+m+1$ | 0 | 0 |

表 5 部分性质满足情况对比

| | KM 协议 | OZCL 协议 | OZL 协议 | 本文协议 |
|------|-------|---------|--------|------|
| 公平性 | 不具备 | 不具备 | 不具备 | 具备 |
| 时限性 | 不具备 | 不具备 | 不具备 | 具备 |
| 无排斥性 | 不具备 | 不具备 | 不具备 | 具备 |

表 6 协议部署情况对比

| | KM 协议 | OZCL 协议 | OZL 协议 | 本文协议 |
|--------|-------|---------|--------|------|
| 发送不同消息 | 不能 | 能 | 能 | 能 |
| 不可否认服务 | 存在 | 存在 | 存在 | 避免 |
| 失效攻击 | 存在 | 避免 | 避免 | 避免 |
| 重放攻击 | 存在 | 避免 | 避免 | 避免 |
| 时钟同步 | 需要 | 需要 | 需要 | 无需 |
| 证据管理 | 复杂 | 复杂 | 复杂 | 容易 |

表 7 扩展前后实体 A 开销对比

| n 个并行的两方 NZG 协议 | 本文协议 |
|---------------------------------------|---|
| Evidence of origin EOO_i | $= EOO_i$ |
| n signatures | n signatures |
| Generation of $n k_i$ | \approx Generation of $n n_i$ plus one k |
| Evidence of submission Sub_{K_i} | $\gg Sub_K$ |
| n signatures | one signatures |
| Encrypted key $E_{uB_i}(k_i)$ | \ll Encrypted key $E_B(K)$ plus $E_{uB_i}(n_i)$ |
| n asymmetric encryptions | $n+m$ asymmetric encryptions |
| n fetches operations of Con_{K_i} | \gg one fetches operation of Con_K |

表 8 扩展前后实体 TTP 开销对比

| n 个并行的两方 NZG 协议 | 本文协议 |
|---|--|
| Verificate and store | \gg Verificate and store |
| n evidences Sub_{K_i} | one evidences Sub_K |
| n evidences EOR_i at most | n evidences EOR_i at most |
| Store m keys | \gg Store 1 key |
| Second group encryption | \ll Second group encryption |
| No asymmetric encryption | m asymmetric encryption |
| Generation of m evidences Con_{k_i} | \gg Generation of 1 evidence Con_k |

表 9 扩展前后实体 B_i 开销对比

| n 个并行的两方 NZG 协议 | 本文协议 |
|-----------------------------|---------------------------------|
| Evidence of receipt EOR_i | $= EOR_i$ |
| Fetch k_i and Con_{k_i} | $=$ Fetch k and Con_k |
| Obtain k_i | $<$ Obtain k plus n_i |
| Decrypts $E_{uB_i}(k_i)$ | Decrypt $E_{uB_i}(E_{uB_i}(K))$ |
| | Decrypts $E_{uB_i}(n_i)$ |

6.2 协议部署

具体部署 NKM 协议时需解决诸如证据管理、网络时延以及信道不安全所引发的相关问题。

NKM 采用证据链技术管理证据，不同的证据

按产生的先后关系联成一串，后产生的包含先产生的；只要能保证后产生的有效，先产生的自然就有效。NKM 产生证据的顺序为 $EOO_i \rightarrow EOR_i \rightarrow Con_K$ 。由于 Con_K 中包含时间点 T ，故只要 J 能验证 TTP 的证书在 T 之前没被撤销，就可确定 Con_K 有效。若 J 还能证明 EOO_i, EOR_i 在 Con_K 之前产生，也可认为它们有效；这是因为 TTP 接收 A 和 $B_i (1 \leq i \leq |B|)$ 提交的证据时，需要验证它们证书的有效性，若当时 A 的证书已被撤销， TTP 不会产生 Con_K ，同样若 B_i 的证书已被撤销， TTP 会将其排斥在 B' 之外。协议中 EOR_i 包含 EOO_i ， Con_K 包含 $EOR_i (1 \leq i \leq |B| \wedge \exists j: 1 \leq j \leq |B'| \wedge B_i = B'_j)$ ，故当 A 或 B_i 向 J 提交证据时， J 必能验证证据的有效性。此外， B_i 收到 EOO_i 时只需验证其签名是由 A 产生，而不用检查 A 的证书是否有效。由于 TTP 可能是 PKI 的一部分，故其可以高效地验证证书的有效性。最终 J 在验证证据时只需判断 Con_K 中 T 是否早于 TTP 证书撤销时间，而不必检查 A 和 B_i 证书撤销时间。

针对网络时延问题，有效的解决方法是兼顾网络响应速度、 TTP 特性以及实体数量和自身能力带来的超时，并延长实体检索 TTP 的时间为网络最大时延的两倍；此外，所有关心 M_i 的实体 B_i 在选择 t_{B_i} 时都需要兼顾 t_A 和网络的整体性能。

考虑到协议不可能部署到理想的网络环境，在信道不安全的情况下需要解决如下两个安全问题（见表 10）：

表 10 信道不安全情况下的安全威胁及对策

| 安全威胁 | | 对策 |
|------|--------|---|
| 信息保密 | | (2) $A \rightarrow TTP: f_{Sub}, B, t_A, L, eP_{TTP}(E_B(K)), EOO, Sub_K$; where $Sub_K = S_A(f_{Sub}, B, L, t_A, eP_{TTP}(E_B(K)), EOO)$; |
| 重放攻击 | 现时挑战 | (2 ₁) $A \rightarrow TTP: A$; (2 ₂) $TTP \rightarrow A: S_{TTP}(Nonce_A)$; (2 ₃) $A \rightarrow TTP: S_A((f_{Sub}, B, t_A, L, E_B(K), EOO, Sub_K), Nonce_A)$; (3 ₁) $B_i \rightarrow TTP: B_i$; (3 ₂) $TTP \rightarrow B_i: S_{TTP}(Nonce_{B_i})$; (3 ₃) $B_i \rightarrow TTP: S_{B_i}((f_{EOR}, A, l_i, x_i, uB_i, t_{B_i}, EOO_i, EOR_i), Nonce_{B_i})$. |
| | 协议运行计数 | (2) $A \rightarrow TTP: f_{Sub}, B, t_A, L, E_B(K), EOO, Count_A, Sub_K$; (3) $B_i \rightarrow TTP: f_{EOR}, A, l_i, x_i, uB_i, t_{B_i}, EOO_i, Count_{B_i}, EOR_i$; where $EOR_i = S_{B_i}(f_{EOR}, A, l_i, x_i, uB_i, t_{B_i}, c_i, EOO, Count_{B_i})$; $Sub_K = S_A(f_{Sub}, B, L, t_A, E_B(K), EOO, Count_A)$. |

- (1) 信息保密. 为防止 $B_i \in B \setminus B'$ 试图通过侦听 A 与 TTP 间信道截获 $E_B(K)$ ， A 必须借助保密渠道向 TTP 提交信息，如使用 $eP_{TTP}(E_B(K))$ 替换 $E_B(K)$ ，此处 $eP_{TTP}(X)$ 表示用 TTP 公钥对 X 加密。
- (2) 重放攻击. 步 2 和步 3 中的信息可能会被对方监听并在日后重放用来骗取 TTP 产生不可否认

证据。事实上， A 可以延迟执行协议步 2，拦截并存储协议步 3 中的某些 B_i 发送给 TTP 的协议消息，为描述方便，此处假设被 A 拦截了消息的 B_i 构成实体集合 B'' ；当过了 $\max\{t_{B_i} | B_i \in B''\} + t_0$ 个时间单位后， B'' 中成员还没有从 TTP 公共目录中检索到 Con_K ，它们就认为 A 没有发送 Sub_K 给 TTP （忽略网络传输时间）， B'' 中所有成员便不再检索 TTP 的

公共目录. 而此时 A 执行协议步 2、重放所拦截的协议步 3, 接着 A 从 TTP 处得到 EOR 和 Con_K . 以后, A 可以向仲裁者提交 $(A, B', L', T, t_A, tSet_{B'}, \varphi_{B'}(E_B(K)), l_i, x_i, uB_i, c_i, k_i)$ 和证据 $EOO_i, EOO, EOR_i, EOR, Con_K$, 仲裁者将作出 $B_i (B_i \in B'')$ 收到 M_i 的仲裁, 而实际情况是 B_i 根本没有收到. 同样, B_i 也可以延迟执行协议步 3、拦截并存储协议步 2 的相同方法, 达到欺诈 A 的目的. 针对如上所述的重放攻击问题, 有两种解决手段:

(1) 基于现时挑战, 加强认证待提交给 TTP 的信息. A 和 B_i 都使用一个提交子协议将请求提交给 TTP . 提交子协议很简单, 实体 P 要将信息 X 提交给 TTP 时, TTP 将一个现时 Nonce 发送给该实体. P 需要使用自己的签名私钥将待提交的信息 X 和 Nonce 一起签名后发送给 TTP . 这样, TTP 就能够鉴别 X 是 P 所发还是一次重放攻击. 在这种方案下, TTP 无须长期维护任何协议轮状态, 降低了 TTP 的负担, 同时也增强了协议的可靠性. 但改进后的协议从总体上增加了 4 步 (实际执行时最多会增加 $2+2|B|$ 个交互步骤, 因为每个接收者都要多执行 2 步), 降低了协议效率.

(2) 基于协议运行计数, A 和 B_i 各自维护协议计数器 $Count_A$ 和 $Count_{B_i}$, TTP 为每个协议实体也维护一个计数器. 当实体向 TTP 提交信息时, 相应的计数器会自加 1, 同时 TTP 会将所提交的信息中计数器与 TTP 维护的计数器不同值的信息丢弃. 这样, TTP 就可以鉴别信息的初次发送与重放了. 同时, 协议交互步骤没有增加. 虽然 TTP 需要为每个通信实体维护一个计数器, 但这些数据量是固定的, 不会随着时间递增. 另外, 如果 TTP 由于某种原因导致某个通信实体的计数器丢失, 可以通过其他的机制与通信实体重新协商新的计数器. 具体的协议步骤见表 10.

7 结束语

本文指出基于时间段概念、双重群加密技术以及证据链技术可有效解决现有使用 online TTP 的典型多方不可否认协议无法提供时限性、无排斥性, 并容易遭受服务失效攻击的问题; 具有 One-to-many 通信拓扑的多方不可否认协议时限性可借助于扩展 SVO 逻辑进行形式化验证; 多方不可否认协议中使用时间段概念和证据链技术可大大降低协议部署的复杂度. 下一步工作打算从两方面展开: (1) 针对在

确保新协议安全实用, 性能不低于现有协议的情形下, 一定程度上增加的对 TTP 的依赖, 考虑如何在 不丢失协议公平性的前提下, 进一步降低协议对 TTP 的依赖. (2) 针对本文采用的形式化方法还无法验证新协议的所有性质的问题, 考虑如何从多方不可否认协议通信模型和形式化方法语义出发, 给出适用于全面分析多方不可否认协议安全性的方法.

参 考 文 献

- [1] Kremer S, Markowitch O, Zhou J. An intensive survey of non-repudiation protocols. *Computer Communications*, 2002, 25(17): 1606-1621
- [2] Puigserver M, Gomila J L F, Rotger L H. Certified electronic mail protocol resistant to a minority of malicious third parties//*Proceedings of the IEEE InfoCom'00*. Tel Aviv, Israel, 2000: 1401-1405
- [3] Micali S. Simple and fast optimistic protocols for fair electronic exchange//*Proceedings of the PODC*. Boston, USA, 2003: 12-19
- [4] Zhou J, Feng B, Robert D. Minimizing TTP 's involvement in signature validation. *International Journal of Information Security*, 2006, 5(1): 37-47
- [5] Kremer S, Markowitch O. Fair multi-party non-repudiation protocols. *International Journal of Information Security*, 2003, 1(4): 223-235
- [6] González-Deleito N, Markowitch O. Exclusions and related trust relationships in multi-party fair exchange protocols. *Electronic Commerce Research and Applications*, 2007, 6(3): 343-357
- [7] Onieva J, Zhou J, Carbonell M, Lopez J. Intermediary non-repudiation protocols//*Proceedings of the CEC*. Newport Beach, CA, USA, 2003: 207-214
- [8] Onieva J, Zhou J, Lopez J. Non-repudiation protocols for multiple entities. *Computer Communications*, 2004, 27(16): 1608-1616
- [9] Zhou J, Gollmann D. A fair non-repudiation protocol//*Proceedings of the IEEE Symposium on Security and Privacy*. Oakland, CA, 1996: 55-61
- [10] Pancho-Festin S, Gollmann D. On the formal analyses of the Zhou-Gollmann non-repudiation protocol//*Proceedings of the FAST 2005*. San Francisco, California, USA, LNCS 3866. Berlin Heidelberg: Springer-Verlag, 2006: 5-15
- [11] Louridas P. Some guidelines for non-repudiation protocols. *ACM SIGCOMM Computer Communication Review*, 2000, 30(4): 29-38
- [12] Gürgens S, Rudolph C. Security analysis of (un-)fair non-repudiation protocols//*Proceedings of the 1st International Conference on Formal Aspects of Security*. Royal Holloway, University of London, UK, LNCS 2629. Springer-Verlag, 2002: 97-114

[13] Gürgens S, Rudolph C, Vogt H. On the security of fair non-repudiation protocols//Proceedings of the 6th International Conference on Information Security (ISC 2003). Bristol, UK, LNCS 2851. Springer-Verlag, 2003: 193-207

[14] Petropoulos D, Kotzanikolaou P. Some more improvements on a fair non-repudiation protocol. Journal of Internet Technology, 2003, 4(4): 255-259

[15] Kim K, Park S, Baek J. Improving fairness and privacy of Zhou-Gollmann's fair non-repudiation protocol//Proceedings of the ICPP Workshop on Security(IWSEC). The University of Aizu, Aizu-Wakamatsu, Fukushima, Japan, 1999: 140-145

[16] Li B, Luo J. On timeliness of a fair non-repudiation protocol//Proceedings of the 3rd International Conference on Information Security (InfoSecu'04). Shanghai, 2004: 99-107

[17] Carbonell M, Onieva J, Lopez J, Zhou J, Galpert D. Time-

out estimation using a simulation model for non-repudiation protocols//Laganà A et al eds. Proceedings of the ICCSA 2004. Assisi, Italy, LNCS 3043. Springer, 2004: 903-914

[18] You C H, Zhou J, Lam K Y. On the efficient implementation of fair non-repudiation. ACM Computer Communication Review, 1998, 28(5): 50-60

[19] Chadha R, Kremer S, Scedrov A. Formal analysis of multi-party contract signing//Proceedings of the 17th IEEE Computer Security Foundations Workshop (CSFW'04). Pacific Grove, California, USA, 2004: 266-279

[20] Li Bo-Tao, Luo Jun-Zhou. Formal analysis of timeliness in non-repudiation protocols. Journal of Software, 2006, 17(7): 1510-1516(in Chinese)
(黎波涛, 罗军舟. 不可否认协议时限性的形式化分析. 软件学报, 2006, 17(7): 1510-1516)



HAN Zhi-Geng, born in 1976, Ph.D. candidate. His research interests include network security and cryptographic protocols.

LUO Jun-Zhou, born in 1960, Ph.D., professor, Ph.D. supervisor. His research interests include next generation network architecture, protocol engineering, network security and management, and grid computing.

Background

The impressive growth of open networks during the last decade has given more importance to several security related problems. The non-repudiation problem is one of them. In comparison to other security issues, non-repudiation has not been studied intensively, meanwhile, most of researches on non-repudiation only toward two-party scenario, and even now, there is still no practical solution for design and verification and deployment of multi-party non-repudiation protocols.

Other than two-party non-repudiation protocols, multi-party non-repudiation protocols must respect some special properties. This work is mainly focused on some practical techniques for design and verification and deployment of

multi-party non-repudiation protocols.

The authors' early works were focused on design and verification of two-party non-repudiation protocols. With respect to protocol design, they proposed to provide timeliness based on relative time notion, which does not need the support for any global clock synchronism mechanism. And with respect to formal analysis methods, they enhanced SVO logic for time description and made it be able to analyze timeliness of non-repudiation protocols. And based on the techniques used in general security protocols, they also presented a method for modeling and analysis of non-repudiation protocols with Color Petri nets.