移动通信中可证安全的双向认证密钥协商协议*

邓红素, 左益强, 赵一鸣+, 鲍振东

(复旦大学 计算机科学与工程系,上海 200433)

A Security-Provable Mutually Authenticated Key Agreement Protocol in Mobile Communication

DENG Hong-Su, ZUO Yi-Qiang, ZHAO Yi-Ming+, BAO Zhen-Dong

(Department of Computer Science and Engineering, Fudan University, Shanghai 200433, China)

+ Corresponding author: Phn: 86-21-65642837, Fax: 86-21-65642821, E-mail: zhym@fudan.edu.cn

http://www.cs.fudan.edu.cn

Received 2002-07-30; Accepted 2002-11-05

Deng HS, Zuo YQ, Zhao YM, Bao ZD. A security-provable mutually authenticated key agreement protocol in mobile communication. *Journal of Software*, 2003,14(8):1489~1494.

http://www.jos.org.cn/1000-9825/14/1489.htm

Abstract: In the distributed computing environments such as wireless network, the key exchange protocol with mutual authentication is critical to the following secure sessions between communicators and more attentions are paid to the provability of the protocol security. A mutual authentication key agreement protocol, MAKAP (mutual authenticated key agreement protocol) is proposed. The security of the protocol is proved in Bellare-Rogaway's model and its computation cost is also analyzed. MAKAP has advantages over other protocols in its security provability and only spends moderate computation cost, so it is quite practical.

Key words: authenticated key generation protocol; mutual authentication protocol with provable security; MAKAP protocol

摘 要: 在基于无线网络的分布式环境中,带认证的密钥协商协议对通信双方是否能够建立安全的会话至关重要.同时,协议的可证安全也逐步得到重视.在借鉴以往无线通信密钥建立协议的基础上,提出了一个可相互认证的密钥协商协议 MAKAP(mutual authenticated key agreement protocol),并在 Bellare 和 Rogaway 的模型下证明了它的安全性,同时分析了其计算代价.与以往许多协议相比,MAKAP 协议不仅在安全证明上有较明显的优势,而且其计算量也不大,有较高的实用性.

关键词: 带认证的密钥协商协议;可证安全的相互认证协议;MAKAP 协议

中图法分类号: TP309 **文献标识码**: A

移动通信系统中的安全问题已受到人们越来越多的重视,而认证协议更是倍受关注.如果不能正确认证通信双方的身份,那么以后的会话都是不安全的.一旦确认用户和网络的身份,就可以为他们建立一个共享的会话

* Supported by the National Natural Science Foundation of China under Grant No.60003007 (国家自然科学基金)

第一作者简介: 邓红素(1976－),女,四川合江人,硕士,主要研究领域为密码与网络安全.

密钥,并利用选择的协议和算法为无线通信提供进一步的安全.这种认证通信双方身份并建立共享密钥的协议,就称为带认证的密钥建立协议.

基于对称密码体制的协议需要两个通信实体共享一个长期密钥,或者一个可信第三方介入协议的执行.当这些协议付诸实践时,密钥管理和可扩展性就成为主要问题,因此基于非对称技术的密钥建立协议成为人们关注的热点.最近,针对无线环境提出的两个方案是文献[1,2].在文献[2]中提出的协议通过预计算来提供有效性,但这个协议的扩展性也不好,而且怀疑存在 interleaving 攻击.在文献[1]中,Wong 等人的密钥建立协议可以在用户端有效实现,而且也提供了大多数系统下的可扩展性,但是这个方案的扩展性并不比纯粹的公钥体制下的可扩展性好.

在以往无线通信密钥建立协议的基础上,本文提出了一个可相互认证的密钥协商协议 MAKAP(mutual authenticated key agreement protocol),并在 Bellare 和 Rogaway 的模型[3,4]下证明是安全的.第 1 节给出了 MAKAP 协议.第 2 节在 Bellare-Rogaway 模型下证明了它的安全性.第 3 节简单分析了该协议的实用性.

1 MAKAP 协议

本节提出了一个新的可相互认证的密钥协商协议——MAKAP(mutual authenticated key agreement protocol)协议,该协议在 Bellare-Rogaway 模型下可以证明它的安全性.

在 MAKAP 协议中,每个网络端服务器 B 有一对密钥(PK_B,SK_B),其中 PK_B 是 B 的公钥,SK_B 是 B 的私钥;每个用户拥有一个长期的秘密 a.密钥的可信度由证书权威机构颁发的证书证明.为统一证书格式,证书权威机构把 g^a 作为 B 的公钥进行签名,即用户端证书的形式是

$$Cert(A)=\langle ID_A,g,p,q,g^a,\{ID_A,g,p,q,g^a\}_{sig_{CA}}\rangle .$$

服务器端证书的格式是

$$Cert(B)=\langle ID_B,PK_B,\{ID_B,PK_B\}_{sig_{CA}}\rangle .$$

为了避免攻击者同时冒充用户和网络对协议进行攻击,要在证书中使用一个特殊符号表示证书拥有者的身份.协议如图 1 所示.

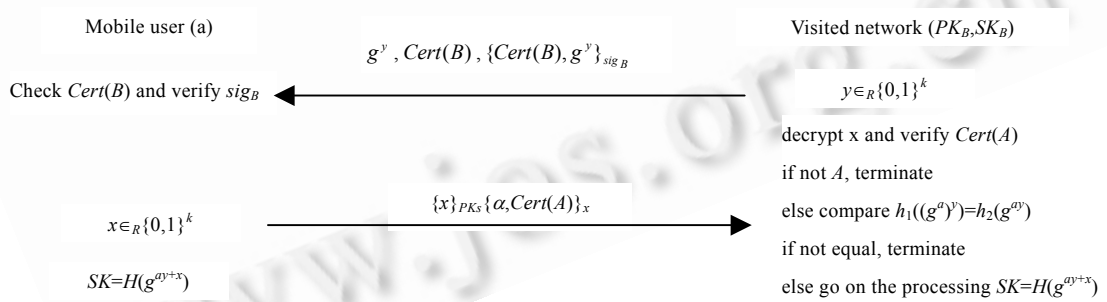

Fig.1 MAKAP protocol

图 1 MAKAP 协议

MAKAP 协议. 假设 A,B 是协议的两个忠实执行者,其中 A 是用户,B 是网络服务器.

(1) 网络端 B 随机选择一个 $y\leftarrow\{0,1\}^k$,计算 g^y,并连同 $Cert(B)$一起,用自己的私钥做一个数字签名 $\{Cert(B),g^y\}_{sig_B}$.B 把 g^y,$Cert(B)$和 $\{Cert(B),g^y\}_{sig_B}$传送给 A.

(2) A 检查 B 的数字签名,并检查 $1<g^y<p$ 是否成立.如果不成立,那么 A 终止协议执行,并报告网络端认证不成功.否则,协议转下一步.

(3) A 随机选择一个 $x\leftarrow\{0,1\}^k$,并用 B 的公钥 PK_B 加密. A 计算$(g^y)^a$ 和$\alpha=h_1(g^{ay})$,并用 x 加密α和 $Cert(A)$.A 将 $\{x\}_{PK_B}$ 和 $\{\alpha,Cert(A)\}_x$ 传送给 B.

(4) B 用 SK_B 解密 $\{x\}_{PK_B}$,并用 x 解密 $\{\alpha,Cert(A)\}_x$,获得$\alpha=h_1(g^{ay})$和 $Cert(A)$.然后 B 验证 $Cert(A)$,如果不成功,

那么 B 认为 A 未通过验证,终止协议执行.否则,转下一步.

(5) B 计算 $\beta = h_1((g^a)^y)$,并比较它是否和接收到的α相等.如果不相等,B 就认为这个消息不是 A 发送出来的,结束协议的执行并报告用户认证未通过.否则,继续执行下一步.

(6) B 计算 $h_2(g^x)$,并发送给 A 作为确认信息.

(7) A 计算 $h_2(g^x)$,并与 B 发送过来的消息比较.如果不相等,也认为消息遭到修改,终止协议,并报告消息传送有错.否则,它可以确定 B 已经可以计算出会话密钥.

(8) A 和 B 分别计算会话密钥 $SK=H(g^{ay+x})$.

$h_1(\cdot)$,$h_2(\cdot)$和 $H(\cdot)$都是$\{0,1\}^*\to\{0,1\}^k$上的散列函数,它们可以被看成实例化的随机 oracle.其中 $H(\cdot)$又称为会话密钥导出函数 SK,它可以尽量消除 SK 对 x 和 y 的依赖以及 g^{ay+x} 中的比较弱的位.另外,在第 2 个消息中传送 $h_1(g^{ay})$而不是 g^{ay},是为了防止 Hijacking 攻击.

2 MAKAP 协议是安全的

下面在 Bellare-Rogaway 模型下证明协议的安全性.在这个模型中,一个分布式系统包含一个用户集 I_c 和服务器集 I_s . I_c 和 I_s 中都是可参与协议执行的用户或服务器的符号.每个用户或服务器都是某个安全参数 k 的多项式函数.攻击者 E 不属于 I_c 或 I_s .

定义 1. MAKAP 是一个三元组 $P=(\Pi,\Psi,LL)$.其中Π,Ψ和 LL 都是它们第 1 个输入的多项式时间函数.Π定义一个诚实用户的行为;Ψ定义诚实的服务器的操作;LL 定义如何初始化用户与网络服务器长期密钥.

Π函数的输入/输出可以描述为

$$(m,\delta,\sigma)=\Pi(1^k,A,B,SK_A,PK_A,PK_B,conv,r),$$

其中,$1^k\in N$ 是一个安全参数.$A\in I_c$ 表示用户身份的符号.$B\in I_s$ 表示服务器的符号.$SK_A\in\{0,1\}^*$是用户 A 的长期私钥.$PK_A\in\{0,1\}^*\cup\{*\}$表示 A 的长期公钥,其中*表示 A 没有长期公钥.$PK_B\in\{0,1\}^*\cup\{*\}$表示 B 的长期公钥,其中*表示 B 没有长期公钥.$r\in\{0,1\}^\omega$是一个随机数,其长度不受限制.$m\in\{0,1\}^*\cup\{*\}$表示用户 A 下次要发送给服务器 B 的信息. $\delta\in\{A,R,*\}$ 表示用户 A 作出的决定.“A”表示接受,“R”表示拒绝,“*”表示到目前为止还没有做出决定.$\sigma\in\{0,1\}^*\cup\{*\}$表示用户 A 的私有输出(private output).

另外,令$\Pi^{m\delta}(\cdot)$表示$\Pi(\cdot)$的前两个输出值,$\Pi^{\delta}(\cdot)$表示$\Pi(\cdot)$的第 2 个输出值,$\Pi^{\delta}(\cdot)$表示$\Pi(\cdot)$的第 3 个输出值.

Ψ函数的定义是$(m,\delta,\sigma)=\Pi(1^k,B,A,SK_B,PK_B,PK_A,conv,r)$,其中每个符号的含义与$\Pi$函数中的类似.

LL 函数的定义是$(SK,PK)=LL(1^k,t,r)$,其中 $t\in\{$client,server$\}$,表示实体的身份类型.在协议中,如果 t 的值取是 client,那么 SK 就是一个四元组(g,p,q,a).其中 p 和 q 都是素数,且 p 的长度是 k 的多项式倍,$q|p-1$;g 是 Z^*_p 中的元素,其阶是 q;a 是从 $Z_q\backslash\{0\}$中随机选取的元素.PK 是 $g^a$$\{$把 g^a 看成 A 的 PK 是为了表述方便,其实更倾向于把*当成 A 的公钥$\}$.如果 t 的值是 server,那么 LL 函数返回的值就是服务器端的密钥对.

定义 2. “**良性攻击(benign adversary)**”. “良性攻击”用来描述一个可靠的网络,即这个网络是良定义的(在一次协议运行结束时,两个通信的 oracle 有完全相同的会话密钥).而对于每一个 $(i,j,s,t)\in I_c\times I_s\times N\times N$,存在一个确定的攻击者,在协议的每次运行时,能忠实地在 Π^s_{ij} 和 Ψ^t_{ji} 之间传送信息.

定义 3. **匹配的会话**. 固定奇数 $R=2\rho-1$,令 P 是一个 R 轮消息的协议.在攻击者存在的情况下执行 P.考虑两个会话的 oracle Π^s_{ij} 和 Ψ^t_{ji},其中 Π^s_{ij} 是会话的发起者,Ψ^t_{ji} 是会话的响应者,会话序列分别用 C 和 C'表示.

(1) C'被称为是 C 的匹配会话,如果存在 $\tau_0<\tau_1<\ldots<\tau_{R-1}$ 和 $\alpha_1,\beta_1,\alpha_2,\beta_2,\ldots,\beta_{\rho-1},\alpha_\rho$,使得 C 的前缀是 $(\tau_0,\lambda,\alpha_1),(\tau_2,\beta_1,\alpha_2),\ldots,(\tau_{2\rho-2},\beta_{\rho-1},\alpha_\rho)$,$C'$的前缀是$(\tau_1,\alpha_1,\beta_1),(\tau_3,\alpha_2,\beta_2),\ldots,(\tau_{2\rho-3},\alpha_{\rho-1},\beta_{\rho-1})$.

(2) C 被称为是 C'的匹配会话,如果存在 $\tau_0<\tau_1<\ldots<\tau_{R-1}$ 和 $\alpha_1,\beta_1,\alpha_2,\beta_2,\ldots,\beta_{\rho-1},\alpha_\rho$,使得 C'的前缀是 $(\tau_1,\alpha_1,\beta_1),(\tau_3,\alpha_2,\beta_2),\ldots,(\tau_{2\rho-3},\alpha_{\rho-1},\beta_{\rho-1}),(\tau_{2\rho-1},\alpha_\rho,*)$.

C 的前缀是$(\tau_0,\lambda,\alpha_1),(\tau_2,\beta_1,\alpha_2),\ldots,(\tau_{2\rho-2},\beta_{\rho-1},\alpha_\rho)$.如果 C 是 C'的匹配会话,C'也是 C 的匹配会话,就称这两个 oracle Π^s_{ij} 和 Ψ^t_{ji} 为有匹配的会话.匹配的会话定义了网络中 E 忠实转发信息的情况.

定义 4. 不匹配会话的概念 NoMatching$^E(k)$.当一个协议 P 在攻击者 E 的控制下执行时,如果存在一个没有

变坏的 oracle 呈接受状态,但没有一个与他有匹配的会话的 oracle 存在的时候,称这样的会话是不匹配的(non-matching).

由此,我们给出 Bellare-Rogaway 模型[3,4]下密钥建立协议的安全定义.

定义 5. 一个密钥建立协议 $P=(\Pi,\Psi,LL)$是安全的,如果满足下列要求:

(1) 良定义的协议(well-defined protocol).在一个良性攻击存在的情况下,oracle Π_{ij}^s 和 Ψ_{ji}^t 在协议结束时都处于接受状态,且拥有同样的会话密钥.

(2) 实时伙伴.对任意的攻击者 E,如果两个没有变坏的 oracle 有匹配的会话,那么他们都应处于接受状态,且拥有相同的会话密钥.

(3) 对消息的认证.$\mathrm{NoMatching}^E(k)$的概率是可忽略的.

(4) 保护新鲜的会话密钥.$\mathrm{advantage}^E(k)$是可忽略的.

这里利用了传统的可忽略函数的定义:称一个实函数$\varepsilon(k)$是可忽略的(neglectable),如果对每个 $c>0$,存在 $k_c>0$,使得对于所有的 $k>k_c,\varepsilon(k)<k^{-c}$.

模型中所有实体间的相互通信都受攻击者的控制.在 Bellare 和 Rogaway 的模型中,攻击者可以读、插入、修改、删除、延迟和重放任何消息,也可以在任何时候重新发起任何参与者的新会话.而且这个模型也模拟了攻击者发起 reflection 攻击、interleaving 攻击和 hijacking 攻击的可能.

定理 1. 假设安全的公钥加密方案、数字签名方案和抗冲突的单向散列函数存在,基于 DH 假设,MAKAP 是安全的.

要证明 MAKAP 协议是一个可认证的密钥协商协议,只需要证明它满足定义的每个条件就足够了.

引理 1. MAKAP 协议是良定义的.

证明:按照良定义协议的概念,在 oracle Π_{ij}^s 和 Ψ_{ji}^t 之间,存在一个良性攻击.那么攻击者只是忠实转发线路上的各种信息,而不会对线路中的信息进行破坏.按照 MAKAP 协议的描述,在通信结束的时候, Π_{ij}^s 和 Ψ_{ji}^t 能够获得同样的会话密钥 $SK=H(g^{ay+x})$,并且两个 oracle 都处于接受状态.所以 MAKAP 是一个良定义的协议. □

引理 2. MAKAP 协议为通信双方提供实时同伴.

证明:由定义的条件 2,假设 Π_{ij}^s 和 Ψ_{ji}^t 是两个没有变坏的 oracle,它们具有匹配的会话.按照协议的描述,会话的发起者 Ψ_{ji}^t 在传送第 1 个消息给 Π_{ij}^s 后, Π_{ij}^s 能根据这些消息计算出相应的内部输出,以及要发送给 Ψ_{ji}^t 的外部输出.oracle Ψ_{ji}^t 能根据他计算的信息判断协议处于的状态,并给 Π_{ij}^s 回答相应的值.即如果 Π_{ij}^s 和 Ψ_{ji}^t 忠实地执行协议,那么最后肯定可以正常终止,并且拥有相同的会话密钥 $SK=H(g^{ay+x})$. □

引理 1 和引理 2 说明,在可靠的网络环境中,如果协议参与者忠实地执行协议,那么 MAKAP 协议是正确的.下面证明协议具有可相互认证的性质.

引理 3. MAKAP 协议能够认证发送者的身份,即它是一个可相互认证的协议.

考察协议的构造,不难发现,在第 1 个消息中, B 作了一个数字签名来表明自己的身份;而在第 2 个消息中, A 通过计算$(g^y)^a$来表明自己的身份(因为按照 DH 假设,只有 A 才能计算出这个信息).引理 3 的证明将分成两个部分进行,先证明服务器向用户提供了正确的身份认证,然后再证明用户也向服务器提供认证.

命题 1. 如果存在安全的数字签名方案,那么 B 可以向 A 提供认证.

证明:令攻击者的一次实验为 E.假设 $NoMatching^E(k)$ 的概率是不可忽略的.称 E 攻击成功,如果它使得一个没有变坏的 oracle Π_{ij}^s 呈接受状态,而没有一个 Ψ_{ji}^t 与它有匹配的会话.这种情况也称为 E 成功攻击了 Π_{ij}^s.令 E 攻击成功的概率为 $\Pr\{E^{succ}\}$,那么,根据假设, $\Pr\{E^{succ}\}=n(k)$,其中 $n(k)$ 是一个不可忽略的函数.

从 E 构造数字签名方案的模仿者 F,使得其成功的概率是不可忽略的.首先,为算法 F,G 和 Sig 抛硬币.然后,利用算法 G 为 F 构造签名用的密钥对(PSK,SSK).最后用 PSK 作为输入来调用伪造者 F .

现在, F 开始调用 E ,从 I_s 中选择实体 $j\in I_s$,并猜测, E 能成功攻击 Π_{ij}^s (某个特定的 j 和 s). F 再抛硬币,并用 G 为除 j 外的所有实体构造密钥对;将自己的输入 PSK 作为 j 的公钥.此时 F 已获得 E 要运行的全部输入,

开始调用 E. F 利用签名 oracle *sig* 帮助实体 j 回答对它的查询,以完成协议中要求的对 E 的所有查询.

假设 E 成功攻击了 oracle Π_{ij}^{s},那么在某时刻 τ_0, E 用 λ(空信息)查询了 Ψ_{ji}^{t},故当 $\tau_1 > \tau_0$, Π_{ij}^{s} 必然会收到某个 g^y 的签名 $\{g^y\}_{sig_B}$.如果这个信息的数字签名以前没有被查询过,那么 F 可把它作为这次实验的输出.如果这个信息 F 以前代表 j 签过,则意味着 Ψ_{ji}^{t} 以前已经被查询过.如果产生信息的时刻 $\tau_1 < \tau_0$,那么 F 放弃,即不是这个攻击的有效 oracle 查询.同时,信息也不能在 $\tau_1 > \tau_0$ 时刻产生,因为按照假设, Π_{ij}^{s} 和 Ψ_{ji}^{t} 之间没有匹配的会话.若 E 没有成功攻击 Π_{ij}^{s} 的时候,则 F 放弃.

于是,F 利用 E 成功赢得自己的实验的概率至少是 $n(k)/q \times (1-\lambda(k))$,其中 $\lambda(k) = \Pr\{\tau_0 < \tau_1\}$ 是可忽略函数.这显然与数字签名是安全的假设矛盾,所以, $n(k)$ 是可忽略的,即 E 攻击 Π_{ij}^{s} 成功的概率不大. □

命题 2. 假设存在一个 E 可以在不破坏 B 向 A 提供认证的前提下,在时间 t 和 q 次 oracle 查询后,能以概率 π 违背 A 向 B 提供认证,那么 DH 问题可以在大致相同的时间内解决,且成功的概率为

$$\pi' \geq \pi - \left(\frac{q}{2^l} + \frac{q}{q'}\right).$$

证明:如果用 $NoAuth^{c2s}$ 表示在攻击者存在的情况下存在一个服务器实例 Ψ_{ji}^{t},在协议执行的最后,它处于接受状态而没有和它有匹配的会话的用户实例 Π_{ij}^{s} 存在.那么这个事件定义了破坏 A 向 B 提供认证的情况.同样可以用 $NoAuth^{s2c}$ 表示破坏 B 向 A 提供认证的情况,即在攻击者存在的情况下,存在一个用户实例 Π_{ij}^{s},在协议执行的最后处于接受状态,而没有一个和它有匹配的会话的服务器实例 Ψ_{ji}^{t} 存在.

下面讨论概率 $\Pr\{Noauth^{c2s} | \neg Noauth^{s2c}\}$.服务器发送了 g^y 和 $\{g^y\}_{sig_B}$ 之后接收到 $\alpha = h_1(g^{ay})$,如果这时服务器验证接受,但这个 α 实际上却不是 A 发送的,那么

- 这个信息可能是攻击者自己猜测出来的.假设一共进行了 q 次 oracle 查询,且 $h_1(\cdot)$ 的输出长度为 l_1,那么攻击者猜测正确的概率不超过 $\frac{q}{2^{l_1}}$.
- 或者 g^y 是以前某个 oracle 查询时已经问过的.假设以前一共查询了 q' 次,现在一共查询了 q 次,那么 g^y 被问过的概率不超过 $\frac{q}{q'}$.
- 或者是攻击者能计算出 g^{ay},并由此计算 $h_1(g^{ay})$ 产生正确的回答.令攻击者能产生 $h_1(g^{ay})$ 的事件为 $Event^{CDH}$,那么

$$\Pr\{NoAuth^{c2s} \mid \neg NoAuth^{s2c}\} \leq \Pr\left\{Event^{CDH} + \frac{q}{2^{l_1}} + \frac{q}{q'}\right\}.$$

注意,证明过程中没有考虑 x 的参与,因为如果攻击者能够计算 g^{ay},那么它就可以随机选择 x,并向 B 发送对应的信息.也就是说,攻击者获得 x 的事件是包含在他能计算 g^{ay} 事件之内的.由此,命题得证. □

根据命题 1 和命题 2,引理 3 成立,即 MAKAP 是一个可相互认证的协议. □

引理 4. MAKAP 协议可以保护新鲜的会话密钥.

证明:用反证法.假设攻击者可以在时间 t 内,以概率 ε 成功猜测 Test 查询选择的值.那么就能证明,可以在大致相当的时间内以不可忽略的概率解决 DH 问题.

散列函数 $H(\cdot)$是抗冲突的且具有一定伪随机性质的函数,因此如果要计算出会话密钥 $SK=H(g^{ay+x})$,就必然要获得 g^{ay+x} 的值.也就是说,在 Test 查询中需要获得 g^{ay} 和 g^x 的值.

按照协议的构造,要获得 g^x 的值,必须由 $\{x\}_{PK_B}$ 获得 x.对于这种情况,由于假设公钥加密体制是安全的,所以其概率可以忽略.根据协议,攻击者可以从 g^a 或 g^y 得到 g^{ay}.令 $AskH$ 表示查询 $H(g^{ay+x})$事件,概率 $\Pr\{AskH\}$ 表示此事件成功的概率.由假设, $\Pr\{AskH\} \geq \varepsilon/2$.按照 Test 查询的要求将查询分成两个部分: $Test(\Pi, i)$ 和 $Test(\Psi, j)$,并且有

$$\Pr\{AskH\} = \Pr\{AskH \wedge (\exists i) Test(\Pi, i)\} + \Pr\{AskH \wedge (\exists j) Test(\Psi, j)\}.$$

B 要成功计算会话密钥,且在第 1 条信息中通过认证, $\Pr\{AskH \cap (\exists j)Test(\Psi, j)\}$ 又可以分成

$$\Pr\{AskH \wedge (\exists j)Test(\Psi, j)\}$$
$$= \Pr\{AskH \wedge (\exists j)Test(\Psi, j) \wedge NoAuth^{s2c}\} + \Pr\{AksH \wedge (\exists j)Test(\Psi, j) \wedge Event^{NoMsg}\} +$$
$$\Pr\{AksH \wedge (\exists j)Test(\Psi, j) \wedge \neg(NoAuth^{s2c} \vee Event^{NoMsg})\}.$$

其中 $NoAuth^{s2c}$ 表示在协议执行以后,有一个用户实例处于接受状态,而没有一个服务器实例和它有匹配的会话的事件, $Event^{NoMsg}$ 表示没有成功发送第 3 条消息的事件.令 $p_{s2c}=\Pr\{Noauth^{s2c}, p_{NoMsg}\}=\Pr\{Event^{NoMsg}\}$,,那么有

$$\Pr\{AskH \wedge (\exists i)Test(\Pi, i)\} + \Pr\{AskH \wedge (\exists j)Test(\Psi, j) \wedge \neg(NoAuth^{s2c} \vee Event^{NoMsg})\} \geq \frac{\varepsilon}{2} - p_{s2c} - p_{NoMsg}.$$

现在来考察两个 oracle Ψ_{ji}^{t} 和 Π_{ij}^{s} .注意, $Test(\Pi, i)$ 查询仍然有失败的可能,因为在 $NoAuth^{s2c}$ 和 $Event^{NoMsg}$ 发生时,这个查询需要有一个服务器实例同它进行会话,那么这种情况下的 $Test(\Pi, i)$ 查询很可能失败.然而, $(\exists j)Test(\Psi, j) \wedge \neg(NoAuth^{s2c} \vee Event^{NoMsg})$ 事件的发生也就意味着进行 $Test(\Pi, i)$ 查询.于是有

$$\Pr\{AskH \wedge (\exists i)Test(\Pi, i)\} \geq \frac{\varepsilon}{2} - p_{s2c} - p_{NoMsg},$$

再考察 A 的运算,这时可以发现,Test(Π,i)在这里已经就是剩下如何从 g^a,g^y,计算 g^{ay},即

$$\Pr\{AskH : g^{ay} \leftarrow (g^a, g^y)\} \geq \frac{\varepsilon}{2} - p_{s2c} - p_{NoMsg},$$

也就是说,CDH 问题可以用 $\varepsilon/2-p_{s2c}-p_{NoMsg}$ 的概率来解决. □

定理 1 的证明:根据引理 1~引理 4,定理 1 成立. □

3 对 MAKAP 协议的实用性展望

本文提出一个可相互认证的密钥协商协议 MAKAP,并在 Bellare 和 Rogaway 的模型下,证明了 MAKAP 协议的安全性.通过简单分析可知,对于服务器端来说,该协议需要进行的计算包括:在第 1 个消息里进行一次模的指数运算 $b_1=g^y$ 和作一个数字签名;当接收到第 2 个消息后,进行一次公钥解密运算、一次非对称解密运算、一次证书验证、一次模指运算 $b_2=(g^a)^y$ 和一次散列运算 $h_1(\cdot)$;为获得会话密钥,要进行一次模指运算 $b_3=g^x$、一次模乘运算 $b_2 \cdot b_3$ 以及一次散列运算 $H(\cdot)$.对用户端来说,需要进行的运算有:当接收到第 1 个消息后,要验证证书,并进行一次模指运算 $a_1=(g^y)^a$、一次散列运算 $h_1(\cdot)$ 、一次公钥加密运算和一次私钥加密运算;为计算出会话密钥,要进行一次模指运算 $a_2=g^x$、一次模乘运算 $a_1 \cdot a_2$ 和一次散列运算 $H(\cdot)$.

其中,证书是可以事先告诉对方,移动用户可以利用通信的间隙对证书进行验证.数字签名的计算量比较大,但如果选用 Rabin 签名,那么数字签名就与一次模平方运算和求模平方根的运算相当.

另外,非对称加密可以采用一些利用服务器端的计算资源的办法来提高运算速度和减少运算量[5].而对称加密和散列运算,都是开销不大的运算.因此,MAKAP 协议有较高的实用价值.

References:

[1] Wong DS, Chan AH. Mutual authentication and key exchange for low power wireless communications. In: Edmonds A, Yenser G, Ferrari G, eds. Proceedings of the IEEE MILCOM 2001 Conference. Washington DC: IEEE Communication Society, 2001. 39~43.

[2] Jakobsson M, Pointcheval D. Mutual authentication for low-power mobile devices. In: Syverson PF, ed. Proceedings of the Financial Cryptography 2001. Heidelberg: Springer-Verlag, 2001. 178~195.

[3] Bellare M, Rogaway P. Entity authentication and key distribution. In: Stinson DR, ed. Proceedings of the CRYPTO'93. Lecture Notes in Computer Science Vol.773, Heidelberg: Springer-Verlag, 1994. 232~249.

[4] Bellare M, Rogaway P. Provably secure session key distribution—the three party case. In: Leighton FT, Borodin A, eds. Proceedings of the 27th ACM Symposium on Theory of Computing. Las Vegas: ACM, 1995. 57~66.

[5] Lee SW, Hong SM, Yoon HS, Cho YK. Accelerating key establishment protocols for mobile communication. In: Pieprzyk J, Safavi-Naini R, Seberry J, eds. Information Security and Privacy, Proceedings of the 4th Australasian Conference, ACISP'99 Proceedings. Lecture Notes in Computer Science, Vol.1587, Heidelberg: Springer-Verlag, 1999. 51~63.