软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2011,22(11):2820−2832 [doi: 10.3724/SP.J.1001.2011.03910]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

标准模型下通用可组合的口令认证密钥交换协议*

胡学先[1,2+], 张振峰[2], 刘文芬[1]

[1](解放军信息工程大学 信息工程学院,河南 郑州 450002)

[2](中国科学院 软件研究所 信息安全国家重点实验室,北京 100190)

Universal Composable Password Authenticated Key Exchange Protocol in the Standard Model

HU Xue-Xian[1,2+], ZHANG Zhen-Feng[2], LIU Wen-Fen[1]

[1](Institute of Information Engineering, PLA Information Engineering University, Zhengzhou 450002, China)

[2](State Key Laboratory of Information Security, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

+ Corresponding author: E-mail: xuexian_hu@yahoo.com.cn

Hu XX, Zhang ZF, Liu WF. Universal composable password authenticated key exchange protocol in the standard model. *Journal of Software*, 2011,22(11):2820−2832. http://www.jos.org.cn/1000-9825/3910.htm

Abstract: Through constructing and utilizing non-malleable, extractable, and weak simulation-sound trapdoor commitment schemes and corresponding smooth projective hash function familes, this paper proposes an efficient two-party password authenticated key exchange (PAKE) protocol within the universal composable (UC) framework, which is the optimal two-round PAKE protocol in this setting. Rigorous security proofs based on standard assumptions in the presence of static corruption adversary are then given out. Comparisons with previously proposed protocols show that, this protocol avoids the use of zero-knowledge protocols, and achieves a higher performance in terms of communication efficiency while attaining a comparable computational complexity.
Key words: password authenticated; key exchange protocol; universal composable; standard model

摘 要: 通过构造不可延展的、可提取的且是弱模拟可靠的陷门承诺体制,以及相应的平滑投射 Hash 函数簇,设计了一个高效的通用可组合(universal composable,简称 UC)安全的两方口令认证密钥交换(password authenticated key exchange,简称 PAKE)协议,并在静态腐化模型下给出了严格的安全性证明.该协议使得 PAKE 协议在 UC 框架下达到了最优的两轮.与已有的协议相比,该协议避免了零知识证明协议的使用,在保持计算复杂度相当的前提下有效地提高了通信效率.
关键词: 口令认证;密钥交换协议;通用可组合;标准模型
中图法分类号: TP309 **文献标识码**: A

口令认证密钥交换(password authenticated key exchange,简称 PAKE)协议使通信用户仅凭一个较短的、易于记住的口令就可以安全地生成高熵的会话密钥,避免了一般的认证密钥交换协议要求存在公钥基础设施或者是要求用户拥有存储长对称密钥的安全硬件等前提假设,极大地方便了使用,并降低了系统实施成本,实用性

* 基金项目: 国家自然科学基金(60873261); 国家高技术研究发展计划(863)(2009AA01Z417); 国家科技支撑计划(2008BAH37B02-2)

收稿时间: 2010-04-26; 定稿时间: 2010-06-29

较强,因而受到了广泛关注和重视[1].

为了使 PAKE 协议能够抵抗离线字典攻击,达到既定的安全性目标,采用可证明安全性理论[2]进行安全性证明是现在普遍采用的协议分析和设计方法.2000 年,Bellare 等人首次提出了两方 PAKE 协议的 BPR 模型[3],随后,BMP 模型[4]以及 ROR 模型[5]也相继被提出,但是这些模型没有考虑协议的组合性以及口令的相关性.2005 年,Canetti 等人在通用可组合(UC)框架[6]下提出了 PAKE 的一个新的安全性定义[7],克服了原有安全性模型的固有缺陷,通过 UC 组合定理保证了协议的通用可组合性,利用合适定义的理想功能保证了安全性分析可以基于任意的口令分布进行,提供了更强的安全性保证.

UC 框架下的 PAKE 安全性定义在提高安全性的同时,也为协议设计带来了更大的挑战.尤其是对于标准模型下 UC 安全的 PAKE 协议而言,目前的工作还相对较少,已有协议的效率与基于 BPR 等模型可证明安全的 PAKE 协议相比[8−11]还存在较大的差距.2005 年,Canetti 等人[7]在 KOY 协议[8]的基础上提出了一个在标准模型下 UC 安全的 PAKE 协议,安全性证明基于静态腐化模型.然而,该协议需要将模拟可靠的零知识证明协议作为子协议使用,使得最终的 PAKE 协议约需要 6 轮通信,效率相对较低.2009 年,Abdalla 等人[11]利用可提取的陷门承诺体制构造了一个安全性更强的 PAKE 协议,允许攻击者自适应腐化协议用户.不过,由于该协议采用了逐比特进行处理的承诺体制,使得用户每次发送消息至少需要进行$\Theta(mn)$次指数计算,其中,n 是安全参数,m 是存储口令所需的比特数,导致最终协议的效率比文献[7]中的协议还要低得多,因此并不实用.

本文考虑标准模型下高效的 UC 安全的 PAKE 协议的设计.注意到文献[12]中采用的陷门承诺体制实际上是模拟可靠的[13],即要求具有自适应腐化能力的攻击者即使得到承诺体制的模拟者生成的伪承诺以及相应的承诺打开信息,也不能以不可忽略的概率生成伪承诺,因此实现效率较低.如果考虑静态腐化模型,要求攻击者不能得到关于伪承诺体制的打开信息,则可以减弱对承诺体制的安全性需求,使其可以有效地构造.因此,我们首先给出了弱模拟可靠的陷门承诺体制的定义,其次,通过构造不可延展的、可提取的且是弱模拟可靠的陷门承诺体制以及相应的平滑投射 Hash 函数簇,设计了一个高效的 UC 安全的两方 PAKE 协议.该协议和 Canetti 等人的协议[7]一样,是基于静态腐化模型可证明 UC 安全的,还避免了零知识证明协议的使用,在保持计算复杂度相当的前提下有效地提高了通信效率,并使 PAKE 协议在 UC 框架下达到了最优的两轮.

1 预备知识

1.1 UC框架

粗略地说,UC 框架定义了两种协议运行模型:现实模型和理想模型.模型中的参与方都被抽象化为概率多项式时间(probabilistic polynomial time,简称 PPT)交互式图灵机(interactive turing machine,简称 ITM).现实模型主要涉及 3 类抽象的参与方:被分析的协议π;协议运行环境$\mathcal{Z}$,主要用于模型化系统中除被分析的协议之外的其他协议;现实攻击者$\mathcal{A}$,用来攻击协议消息和腐化协议参与方.理想模型中主要涉及的参与方包括环境$\mathcal{Z}$、理想攻击者$\mathcal{S}$以及理想功能$\mathcal{F}$.其中:理想功能$\mathcal{F}$模型化了密码学任务所应该实现的功能和可以允许的信息泄露,它可以通过虚拟用户与环境进行交互;理想攻击者$\mathcal{S}$模型化了针对理想功能的特殊攻击者,它只能与理想功能进行交互,而不能与虚拟用户进行交互,这从形式上保证了理想模型中协议的安全性.最后,通过协议模拟保证现实模型中的协议具有和理想模型中的协议相同的功能,给出现实协议安全性的定义.

定义 1(实现理想功能). 给定协议π以及理想功能$\mathcal{F}$,如果对任意的攻击者$\mathcal{A}$,都存在攻击者$\mathcal{S}$,使得对拥有任意输入的任何环境$\mathcal{Z}$,在和攻击者$\mathcal{A}$以及协议π交互后输出 1 的概率分布总体与在和攻击者$\mathcal{S}$以及理想功能$\mathcal{F}$交互后输出 1 的概率分布总体是计算不可区分的,则称协议π UC 实现了理想功能$\mathcal{F}$.

一般的组合定理仅仅适用于上层协议π调用的多个理想功能实例是相互独立的情形,如果多个实例共用一个共同参考串或者是随机谕示的情况,则需要采用 Canetti 和 Rabin[14]提出的具有共同状态的通用可组合(universal composition with joint state,简称 JUC)框架.JUC 框架通过定义理想功能的多会话扩展给出了一个新的组合算子,允许不同协议之间存在共有状态,并保证安全性.理想功能$\mathcal{F}$的一个多会话扩展 $\hat{\mathcal{F}}$ 定义如下: $\hat{\mathcal{F}}$ 本

质上运行着$\mathcal{F}$的多个实例,不同实例之间通过子会话标识 *SSID* 予以区分.当 $\hat{\mathcal{F}}$ 收到一个子会话标识为 *SSID=ssid* 的消息 *m* 时,将 *m* 转交给子会话标识为 *ssid* 的$\mathcal{F}$实例;如果这样的实例不存在,就激活一个新的实例.下文在不引起混淆的情况下将省略 *SID*,而只提到 *SSID*.

1.2 PAKE理想功能

2005 年,Canetti 等人[7]提出了 PAKE 的理想功能,该定义以一般情形下密钥交换[15]的理想功能为基础,并综合考虑了口令信息熵较低所带来的安全性缺陷.其主要设计思想是:如果参与协议的两个用户都没有被腐化,则他们将生成相同的均匀随机值作为会话密钥,并且攻击者得不到会话密钥的任何信息;然而,如果其中一个用户被腐化,或者是攻击者正确地猜到用户的口令,则让攻击者完全决定会话密钥.其中,攻击者对口令的猜测只能通过询问理想功能才能验证,模型化了攻击者只能进行在线字典攻击的情况.另外,用户的口令由环境确定并作为输入提供给用户实例,这使得相应的协议适用于任意口令分布的情形,甚至是相关的口令被用于不同的实例的情况.

1.3 承诺体制

一个承诺体制就是一个有两方参与的两阶段协议,这两方分别称为承诺者和接收者,第 1 阶段称为承诺阶段,第 2 阶段称为承诺打开阶段.通过这个协议承诺者能够将自己与一个消息绑定,且这种绑定满足下述隐藏(hiding)和绑定(binding)性质.

定义 2(承诺体制). 一个承诺体制 $CS=(\mathcal{K},C,Ver)$由 3 个 PPT 算法组成,其中,公共参考串生成算法$\mathcal{K}$在输入安全参数后输出承诺体制的公共参考串 $pk\in_R\mathcal{K}(1^n)$;承诺算法 *C* 在输入公共参考串 *pk* 和承诺消息 *m* 后输出 $(com,dec)=C_{pk}(m)$,其中,*com* 是承诺值,*dec* 是用于打开承诺的参数;承诺验证算法 *Ver* 在输入 *pk,com,m,dec* 时验证(*com,dec*)是否是承诺消息 *m* 的有效承诺值.并且,$(\mathcal{K},C,Ver)$需要满足下述性质:

(1) **绑定性质**. 存在可忽略函数 *negl*(*n*),使得对任意 PPT 攻击者$\mathcal{A}$,都有

$$Pr[pk\leftarrow\mathcal{K}(1^n),(com,m_1,m_2,dec_1,dec_2)=\mathcal{A}(pk):(Ver_{pk}(com,m_1,dec_1)=Ver_{pk}(com,m_2,dec_2)=1)\wedge(m_1\neq m_2)]\leqslant negl(n).$$

(2) **隐藏性质**. 对算法$\mathcal{K}$输出的任意 *pk* 以及任意两个等长度消息 m_1,m_2,随机变量总体$\{(com_1,dec_1)=C_{pk}(m_1):com_1\}$和$\{(com_2,dec_2)=C_{pk}(m_2):com_2\}$是计算上不可区分的.

除了绑定和隐藏这两个基本性质之外,根据实际使用需要,密码学者还定义了承诺体制的不可延展性(non-malleability)、可提取性(extractability)以及陷门性(equivocability)等不同的安全性质.不可延展性[16]是指攻击者即使能够访问承诺谕示得到关于某个承诺消息 *m* 的多项式个承诺值,也不能以不可忽略的概率生成一个新的承诺值 *com′*以及多项式时间可计算的非平凡关系 *R*,使得存在 *m′,dec′*满足 $Ver_{pk}(com',m',dec')=1$ 且 $R(m,m')=1$.可提取性[17]是指存在一种提取算法 *Extract*,使得对于任意由承诺算法$(com,dec)=C_{pk}(m)$生成的承诺值 *com*,除了可忽略的概率之外都成立,$Extract(sk,com)=m$,其中,*sk* 为提取算法所用到的私钥.

定义 3(陷门承诺体制)[18]. 一个陷门承诺体制 *TCS* 由 5 个 PPT 算法$(T\mathcal{K},TC,Tver,TfakeC,TfakeDecom)$组成,其中,$T\mathcal{K}$在输入安全参数 1^n 时生成公私钥对(*pk,sk*),记 $T\mathcal{K}_1(1^n)$为其中的公钥 *pk*,则$(T\mathcal{K}_1,TC,TVer)$是一个承诺体制,且 *TfakeC,TfakeDecom* 还满足:*TfakeC* 在输入公私钥对(*pk,sk*)时可以输出一个伪承诺$(com^*,\xi)=TfakeC(pk,sk)$,使这个伪承诺可以按照任意方式打开.

也就是说,对任意给定的消息 *m*,$TfakeDecom$ 算法可生成 $dec^*=TfakeDecom(\xi,com^*,m)$,满足 $TVer_{pk}(com^*,m,dec^*)=1$,并且,随机变量总体$\{(pk,sk)\leftarrow T\mathcal{K}(1^n);(com^*,\xi)=TfakeC(pk,sk);dec^*=TfakeDecom(\xi,com^*,m):(pk,com^*,m,dec^*)\}$和$\{pk\in_R T\mathcal{K}_1(1^n);(com,dec)=TC_{pk}(m):(pk,com,m,dec)\}$是计算不可区分的.

对于陷门承诺体制而言,攻击者看到了对伪承诺的多次打开可能会有助于其攻破体制的绑定性质.例如,Perdesen 承诺体制满足计算绑定性质,但如果攻击者看到了一个承诺值 *com* 以两种方式打开,则它利用 $com=g^rh^m=g^{r'}h^{m'}$可以得到 $\log_g h$,从而可以将每一个承诺以多种方式打开,破坏承诺体制的绑定性质.

为了限制针对陷门承诺体制的攻击者攻破体制的绑定性质的能力,MacKenzie 等人[13]给出了模拟可靠的(simulation-sound)陷门承诺体制的定义,要求攻击者即使看到了伪承诺的多次打开,也不能以不可忽略的概率

生成一个新的伪承诺.文献[11]中采用的陷门承诺体制实际上就是模拟可靠的,其中,协议的攻击者具有自适应腐化用户的能力,腐化诚实的用户相当于对伪承诺进行了打开.如果考虑静态腐化模型,则可以降低对 PAKE 协议设计中用到的承诺体制的安全性需求,因为此时协议的攻击者只能在协议开始运行之前腐化用户,不能得到关于伪承诺体制的打开信息.为此,我们对 MacKenzie 等人给出的定义进行了弱化,定义了弱模拟可靠(weak simulation-sound)的陷门承诺体制,其中,攻击者可以得到模拟器生成的伪承诺,但是不能得到关于伪承诺的任何打开信息.

定义 4(弱模拟可靠的陷门承诺体制). 称一个陷门承诺体制 $TCS=(T\mathcal{K},TC,Tver,TfakeC,TfakeDecom)$是弱模拟可靠的,如果存在可忽略函数 $negl(n)$,使得对能访问谕示$\mathcal{O}_{TfakeC}$的任意 PPT 攻击者$\mathcal{A}$都成立:

$$Pr[(pk,sk)\leftarrow T\mathcal{K}(1^n),com=\mathcal{A}^{\mathcal{O}_{TfakeC}}(pk):com\notin Q \text{ and } \exists m_1,m_2,dec_1,dec_2,$$
$$\text{s.t. } (TVer_{pk}(com,m_1,dec_1)=TVer_{pk}(com,m_2,dec_2)=1)\wedge(m_1\neq m_2)]\leqslant negl(n).$$

其中,$\mathcal{O}_{TfakeC}$谕示的定义如下:计算$(com^*,\xi)=TfakeC(pk,sk)$,将 com^*存入列表 Q,然后输出 com^*.

1.4 平滑投射Hash函数簇

平滑投射 Hash 函数簇的定义由 Cramer 和 Shoup 在文献[19]中首次提出,并用于构造 CCA2 安全的加密体制.随后,Gennaro 和 Lindell[9]对上述定义进行了扩展,并将其应用于构造标准模型下的 PAKE 协议.

类似于文献[9],利用子集成员问题给出平滑投射 Hash 函数簇的严格定义.一个子集成员问题定义了一个易于识别的集合 X 以及一个 NP 语言 $L\subset X$,即存在证据集 W 以及 NP 关系 $R\subset X\times W$,使得 $x\in L$ 当且仅当存在 $w\in W$ 使得$(x,w)\in R$.子集成员问题的困难性要求难以区分 L 上的均匀分布和 $X\backslash L$ 上的均匀分布.

平滑投射 Hash 函数簇的严格定义如下:设$\mathcal{H}=\{H_{hk}\}_{hk\in HK}$是一簇 Hash 函数,其中,$H_{hk}$的定义域为 X,值域为群 G,Hash 密钥空间为 HK.α:$HK\times X\to HP$ 是和函数簇$\mathcal{H}$相关联的函数,称为密钥投射函数,其中,HP 被称为投射密钥空间.如果存在函数簇$\{ProjH_{hp}\}_{hp\in HP}$,使得对任意的 $x\in L$,Hash 值 $H_{hk}(x)$由投射密钥 $hp=\alpha(hk,x)$和 $x\in L$ 的证据 w(即$(x,w)\in R$)唯一确定 $ProjH_{hp}(w)=H_{hk}(x)$,则称$\mathcal{H}=\{H_{hk}\}_{hk\in HK}$是投射 Hash 函数簇.

定义 5(平滑性)[19]. 称一个投射 Hash 函数簇$\mathcal{H}=\{H_{hk}\}_{hk\in HK}$是平滑的,如果对任意的元素 $x\in X\backslash L$,Hash 值 $H_{hk}(x)$不仅统计接近于 G 上的均匀分布,还独立于投射密钥 hp 以及 x,即当 hk 是 HK 均匀选取的随机变量时,随机变量总体$\{x,hp=\alpha(hk,x),H_{hk}(x)\}$和随机变量总体$\{x,hp=\alpha(hk,x),g\in_R G\}$是统计不可区分的.

对于相应于困难子集成员问题的平滑投射 Hash 函数簇,Gennaro 等人[9]证明了它们满足下述伪随机性:

引理 6(伪随机性)[9]. 假设 x 是从语言 L 中均匀选取的随机值,hk 是从 Hash 密钥空间 HK 中均匀选取的随机值,对于一个不知道相应于 $x\in L$ 的证据 w(即$(x,w)\in R$)以及 Hash 密钥 hk 的攻击者来说,总体$\{x,hp=\alpha(hk,x),H_{hk}(x)\}$和$\{x,hp=\alpha(hk,x),g\in_R G\}$是计算不可区分的.

2 承诺体制的构造

本节基于 N 次剩余假设构造不可延展的、可提取的并且是弱模拟可靠的陷门承诺体制 $CS=(\mathcal{K},C,Ver)$.该体制的构造利用了文献[20,21]中的思想.

2.1 基于N次剩余假设的公钥加密体制

基于 N 次剩余假设的公钥加密体制自 1999 年首次由 Paillier 等人[22]提出后,人们对其进行了一系列的扩展研究.本文将用到 Gennaro 等人[9]提出的体制.该体制不仅是 CCA2 安全的,还具有易于构造相应的平滑投射 Hash 函数簇的特点.

设 $N=pq$,其中,p,q 是安全素数.即存在相应的素数 p',q'满足等式 $p=2p'+1,q=2q'+1$.

考虑乘法群 $Z^*_{N^2}=\{x\in Z_{N^2}\mid \gcd(x,N^2)=1\}$,易知 $Z^*_{N^2}$ 的阶为 $4NN'$,其中 $N'=p'q'$.设 J_N 表示群 $Z^*_{N^2}$ 中相对于 N 的 Jacobi 符号为 1 的那些元素组成的集合,则 J_N 也是循环群,且 J_N 的阶为 $2NN'$.在 $Z^*_{N^2}$ 中随机地选择元素 $\mu\in_R Z^*_{N^2}$,令 $g=-\mu^{2N}$,则除了可忽略的概率外,g 是 J_N 中阶为 $2N'$的元素.

Gennaro 等人[9]提出的公钥加密体制 $PKE=(\mathcal{K},\mathcal{E},\mathcal{D})$由如下 3 种算法组成:

- **密钥生成算法$\mathcal{K}(\mathbf{1}^n)$**. 随机地生成 $N=pq$,其中 p,q 都是长为 $n/2$ 的安全素数.在 $Z^*_{N^2}$ 中随机地选择元素 $\mu\in_R Z^*_{N^2}$, 令 $g=-\mu^{2N}$.选择随机值 $z,\tilde{z},\hat{z}\in[0..N^2/2]$, 计算 $h=g^z,\tilde{h}=g^{\tilde{z}},\hat{h}=g^{\hat{z}}$.假设 H 是一个抗碰撞的 Hash 函数,则加密体制的公钥为 $pk=(H,N,g,h,\tilde{h},\hat{h})$,私钥为 $sk=(z,\tilde{z},\hat{z})$.
- **加密算法$\mathcal{E}_{pk}(\boldsymbol{m};\boldsymbol{r})$**. 给定公钥 pk 和明文 $m\in Z_N$,选择随机值 $r\in[0..N/4]$,计算 $u=g^r \bmod N^2$,$e=(1+N)^m h^r=(1+mN)h^r \bmod N^2$, $v=\|(\tilde{h}\hat{h}^{\theta})^r \bmod N^2\|$, 其中,$\theta=H(u,e)$.函数$\|\cdot\|$定义为:如果 $x\leqslant N^2/2$,则取$\|x\|=x$;否则,取$\|x\|=N^2-v$.最后,加密算法输出相应于 m 的密文 $c=(u,e,v)=\mathcal{E}_{pk}(m;r)$.
- **解密算法$\mathcal{D}_{sk}(\boldsymbol{c})$**. 给定密文 $c=(u,e,v)$以及私钥 $sk=(z,\tilde{z},\hat{z})$,首先验证 $v\leqslant N^2/2$,然后计算$\theta=H(u,e)$,并验证 $v^2=u^{2(\tilde{z}+\theta\hat{z})} \bmod N^2$, 如果至少 1 个验证不成立,则输出$\perp$;否则,计算 $\tilde{m}=e\cdot u^{-z}$, 如果 $\tilde{m}=1 \bmod N$, 则输出 $m=(\tilde{m}-1)/N$, 否则输出$\perp$.

为了表述简单,下文在不引起混淆的情况下省略描述中的 $\bmod N^2$ 运算.

可以按照如下方式利用 $PKE=(\mathcal{K},\mathcal{E},\mathcal{D})$构造带标记的加密体制 $PKE^1=(\mathcal{K},\mathcal{E}^1,\mathcal{D}^1)$:给定明文 m 以及标记 l,加密算法 $\mathcal{E}^1_{pk}(m;l;r)$ 利用随机数 $r\in[0..N/4]$计算 $u=g^r$,$e=(1+N)^m h^r=(1+mN)h^r$, $v=\|(\tilde{h}\hat{h}^{\theta})^r\|$,其中,$\theta=H(u,e,l)$.解密算法 $\mathcal{D}^1_{sk}(c;l)$ 的修改方法类似.于是,当 PKE 是 CCA2 安全时,带标记的加密体制 PKE^1 也是 CCA2 安全的,并且其中所带的标记 l 也是不可延展的.

可以按照如下方法构造 CPA 安全的加密体制 $PKE^0=(\mathcal{K},\mathcal{E}^0,\mathcal{D}^0)$:假设利用随机数 r 和加密体制 PKE 加密明文 m 得到的密文是$\mathcal{E}_{pk}(m;r)=(u,e,v)$,取该密文的前两项,定义加密算法 $\mathcal{E}^0_{pk}(m;r)=(u,e)$,相应的解密算法为 $\mathcal{D}^0_{sk}(u,e)=e\cdot u^{-z}$,则如此定义的 PKE^0 是一个 CPA 安全的公钥加密体制,并且满足 $\mathcal{E}_{pk}(m;r)=(\mathcal{E}^0_{pk}(m;r),v)$.

2.2 不可延展的、可提取的且是弱模拟可靠的陷门承诺体制

设 $PKE^1=(\mathcal{K},\mathcal{E}^1,\mathcal{D}^1)$和 $PKE^0=(\mathcal{K},\mathcal{E}^0,\mathcal{D}^0)$分别是基于加密体制 $PKE=(\mathcal{K},\mathcal{E},\mathcal{D})$构造的 CCA2 安全的带标记的公钥加密体制和 CPA 安全的加密体制,并且它们共用相同的密钥生成算法,则不可延展的、可提取的并且是弱模拟可靠的陷门承诺体制 $CS=(\mathcal{K},C,Ver)$的详细构造步骤如下:

- **公共参考串生成**. 利用加密体制的密钥生成算法$\mathcal{K}$生成公钥 $pk=(H,N,g,h,\tilde{h},\hat{h})$ 及私钥 $sk=(z,\tilde{z},\hat{z})$.选择随机值 $b^*\in_R[0..2^{n/4}]$以及随机值 $r^*\in_R[0..N/4]$,利用 r^*对明文$-b^*$加密得到:

$$B^*=(u^*,e^*)=\mathcal{E}^0_{pk}(-b^*;r^*)=(g^{r^*},(1-b^*N)h^{r^*}),$$

  则承诺体制的公共参考串为 $pk_{CS}=(pk,B^*)$,相应的私钥为 $sk_{CS}=(sk,b^*,r^*)$.
- **承诺阶段**. 给定公共参考串 pk_{CS} 和消息 $m\in Z_N$,承诺者选择随机值 $b\in_R[0..2^{n/4}]$,$r_1,r\in_R[0..N/4]$,然后计算:

$$\begin{aligned}B_1&=(u_1,e_1)=\mathcal{E}^0_{pk}(b;r_1),\\ B_2&=(u_2,e_2)=(B_1\cdot B^*)^m\cdot\mathcal{E}^0_{pk}(0;r)\\ &=(g^{(r^*+r_1)m+r},(1+(b-b^*)mN)\cdot h^{(r^*+r_1)m+r})\\ &=\mathcal{E}^0_{pk}((b-b^*)m;(r^*+r_1)m+r),\\ v_1&=\|(\tilde{h}\hat{h}^{\theta})^{r_1}\|.\end{aligned}$$

  其中,$\theta=H(B_1,B_2)$.承诺阶段输出的承诺值为 $com=(B_1,v_1,B_2)$.
- **承诺打开**. 为了打开承诺 $com=(B_1,v_1,B_2)$,承诺者将(b,r_1,m,r)发送给接收者.接收者验证 $B_1=\mathcal{E}^0_{pk}(b;r_1)$, $B_2=(B_1\cdot B^*)^m\cdot\mathcal{E}^0_{pk}(0;r)$ 以及 $v_1=\|(\tilde{h}\hat{h}^{\theta})^{r_1}\|$ 是否成立,如果验证全部通过就接受 com 是 m 的承诺值;否则拒绝.

下面证明承诺体制 CS 是不可延展的、可提取的并且是弱模拟可靠的陷门承诺体制.

隐藏性质. 当 r 是$[0..N/4]$上独立均匀的随机值时,$(r^*+r_1)m+r \bmod (N/4)$也是$[0..N/4]$上均匀分布的随机

值.由于 $B_2=\mathcal{E}_{pk}^0((b-b^*)m;(r^*+r_1)m+r)$,再根据加密体制 PKE^1 和 PKE^0 的语义安全性可知,任意两个不同的承诺消息所对应的承诺值是计算不可区分的.因此,承诺体制 CS 满足计算意义上的隐藏性质.

不可延展性. 注意到 $com=(B_1,v_1,B_2)$ 中的 (B_1,v_1) 实际上是带标记加密体制 $PKE^1=(\mathcal{K},\mathcal{E}^1,\mathcal{D}^1)$ 的密文,即 $(B_1,v_1)=\mathcal{E}_{pk}^1(b;B_2;r_1)$,从而可以利用 PKE^1 的不可延展性证明承诺体制 CS 的不可延展性.

给定任意关于承诺消息 m 的多项式时间可计算关系 $R(m,m')=1$,定义一个新的关于消息 (b,B) 的关系 $\overline{R}$,使得 $((b,B),(b',B'))\in\overline{R}$ 当且仅当 $(\mathcal{D}_{sk}^0(B)/(b-b^*),\mathcal{D}_{sk}^0(B')/(b'-b^*))\in R$.假设承诺体制的攻击者 $\mathcal{A}$ 在收到关于 m 的承诺值 $com=(B_1,v_1,B_2)$ 后能够以不可忽略的概率输出多项式时间可计算的关系 R 和一个新的承诺值 $com'=(B_1',v_1',B_2')$,使得 com' 是消息 m' 所对应的承诺值且满足 $R(m,m')=1$,于是,由于 (B_1,v_1) 和 (B_1',v_1') 分别作为 (b,B_2) 和 (b',B_2') 的密文且满足 $\overline{R}((b,B_2),(b',B_2'))=1$,违反了加密体制 PKE^1 的不可验证性性质.

可提取性. 在限制 $b,b^*\in_R[0..2^{n/4}]$ 时,除了不可忽略的概率外,$b-b^*$ 与 N 是互素的.因此,$b-b^*$ 在 Z_N 中乘法可逆.注意到 $B_2=\mathcal{E}_{pk}^0((b-b^*)m;(r^*+r_1)m+r)$,则对于任意诚实生成的承诺 $com=(B_1,v_1,B_2)$,利用私钥 sk 计算 $m=\mathcal{D}_{sk}^0(B_2)\cdot(b-b^*)^{-1}$ 就可以提取得到承诺消息 m.

陷门性质. 给定承诺体制的私钥 $sk_{CS}=(sk,b^*,r^*)$,可按如下方式生成能够以任何方式打开的伪承诺:选择随机值 $r_1,r\in[0..N/4]$,取 $b=b^*$,计算 $B_1=\mathcal{E}_{pk}^0(b^*;r_1),B_2=\mathcal{E}_{pk}^0(0;r)$ 以及 $v_1=\|(\tilde{h}\hat{h}^\theta)^{r_1}\|$,最后输出伪承诺 $com^*=(B_1,v_1,B_2)$.给定任意的消息 m',只需计算 $r'=r-(r^*+r_1)m'$,就可以得到 com^* 的一个合法打开 (b^*,r_1,m',r').

弱模拟可靠的绑定性质. 下面证明即使攻击者可以询问 O_{TfakeC} 谕示,也不能以不可忽略的概率生成一个新的伪承诺.由于 $B_2=\mathcal{E}_{pk}^0((b-b^*)m;(r^*+r_1)m+r)$ 是 $(b-b^*)m$ 所对应的密文,因此要使承诺值 $com=(B_1,v_1,B_2)$ 能打开为任意的消息 m',只能是 $b-b^*=0$.定义关于形如 (b,B) 消息的关系 $\tilde{R}$,使得 $((b,B),(b',B'))\in\tilde{R}$ 当且仅当 $b=b'=b^*$.如果承诺体制的攻击者 $\mathcal{A}$ 在多项式次询问 O_{TfakeC} 谕示后能输出一个伪承诺 com',且这个承诺和 O_{TfakeC} 输出的承诺值 com 都不相同,则 com 和 com' 所对应的明文满足 $\tilde{R}$,违反了加密体制 PKE^1 的不可延展性性质.

3 平滑投射 Hash 函数簇的构造

本节给出相应于承诺体制 CS 的平滑投射 Hash 函数簇.首先,定义相应于带标记的加密体制 PKE^1 的语言 $L_{(PKE^1,pk),(l,*)}$ 和相应于 CPA 安全的加密体制 PKE^0 的语言 $L_{(PKE^0,pk),m}$ 为

$$L_{(PKE^1,pk),(l,*)}=\{c_1=(u_1,e_1,v_1)\mid\exists m,\exists r,\ \text{s.t. } c_1=\mathcal{E}_{pk}^1(m;l;r)\},$$

$$L_{(PKE^0,pk),m}=\{c_2=(u_2,e_2)\mid\exists r,\ \text{s.t. } c_2=\mathcal{E}_{pk}^1(m;r)\}.$$

构造相应于语言 $L_{(PKE^1,pk),(l,*)}$ 的平滑投射 Hash 函数簇 $\mathcal{H}_1$ 如下:设 UH 是任意取定的通用 Hash 函数.定义平滑投射 Hash 函数簇的密钥空间为 $HK_1=[0..N^2/2]^2$,即 Hash 密钥 $hk_1=(a_{1,1},a_{1,2})$,其中,$a_{1,i}\in_R[0..N^2/2]$.对任意给定密文 $c_1=(u_1,e_1,v_1)\in L_{(PKE^1,pk),(l,*)}$,定义相应的密钥投射函数为 $hp_1=\alpha_1(hk_1,c_1)=g^{2a_{1,1}}(\tilde{h}\hat{h}^\theta)^{2a_{1,2}}$,其中,$\theta=H(u_1,e_1,l)$.利用 Hash 密钥 hk_1 计算相应于 c_1 的平滑投射 Hash 值的函数 $Hash_{1,hk_1}(c_1)$ 定义为 $Hash_{1,hk_1}(c_1)=UH[f_{1,hk_1}(c_1)]$,其中,$f_{1,hk_1}(c_1)=u_1^{2a_{1,1}}v_1^{2a_{1,2}}$.利用投射密钥 hp_1 以及相应于 $c_1\in L_{(PKE^1,pk),(l,*)}$ 的证据 r_1 计算 Hash 值的函数 $ProjH_{1,hp_1}$ 定义为 $ProjH_{1,hp_1}(r_1)=UH[hp_1^{r_1}]$.

构造相应于语言 $L_{(PKE^0,pk),m}$ 的平滑投射 Hash 函数簇 $\mathcal{H}_2$ 如下:设 UH 是任意取定的通用 Hash 函数.定义平滑投射 Hash 函数簇的 Hash 密钥空间为 $HK_2=[0..N^2/2]^2$,即 $hk_2=(a_{2,1},a_{2,2})$,其中,$a_{2,i}\in[0..N^2/2]$.给定任意密文 $c_2=(u_2,e_2)\in L_{(PKE^0,pk),m}$,定义相应的投射密钥为 $hp_2=\alpha_2(hk_2,c_2)=g^{2a_{2,1}}h^{2a_{2,2}}$.利用 Hash 密钥 hk_2 计算相应于 c_2 的平滑投射 Hash 函数值的函数 $Hash_{2,hk_2}(c_2)$ 定义为 $Hash_{2,hk_2}(c_2)=UH[f_{2,hk_2}(c_2)]$,其中,

$$f_{2,hk_2}(c_2)=u_2^{2a_{2,1}}\left(\frac{e_2}{1+mN}\right)^{2a_{2,2}}.$$

利用投射密钥 hp_2 以及相应于 $c_2 \in L_{(PKE^0, pk),m}$ 的证据 r_2 计算 Hash 值的函数 $ProjH_{2,hp_2}$ 定义为

$$ProjH_{2,hp_2}(r_2) = UH[hp_2^{r_2}].$$

定义与承诺体制 CS 相关的语言 $L_{pk,m}$ 为

$$L_{pk,m} = \{com = (B_1, v_1, B_2) \mid (B_1, v_1) \in L_{(PKE^1, pk),(B_2,*)}, B_2 \cdot (B_1 \cdot B^*)^{-m} \in L_{(PKE^0, pk),0}\} \quad (1)$$

则 com 是相应于消息 m 的承诺值当且仅当 $com \in L_{pk,m}$.设 $\mathcal{H}_1$ 和 $\mathcal{H}_2$ 分别是前面定义的相应于语言 $L_{(PKE^1, pk),(B_2,*)}$ 和 $L_{(PKE^0, pk),0}$ 的承诺体制,则可以类似于文献[11]中的方法构造相应于语言 $L_{pk,m}$ 的平滑投射 Hash 函数 $\mathcal{H}$如下:函数簇的 Hash 密钥空间为 $HK=HK_1\times HK_2$,即 Hash 密钥为 $hk=(hk_1,hk_2)=(a_{1,1},a_{1,2},a_{2,1},a_{2,2})$.投射密钥空间为 $HP=HP_1\times HP_2$,相应的密钥投射函数为 $hp=(hp_1,hp_2)=\alpha(hk,com)=(\alpha_1(hk_1,c_1),\alpha_2(hk_2,c_2))$.利用 Hash 密钥 hk 计算相应于 com 的平滑投射 Hash 值的函数 $Hash_{hk}(com,m)$被定义为

$$Hash_{hk}(com, m) = Hash_{1,hk_1}(c_1) \oplus Hash_{2,hk_2}(c_2).$$

利用投射密钥 hp 以及相应于 $com \in L_{pk,m}$ 的证据 r_1,r_2 计算 Hash 值的函数 $ProjH_{hp}$ 被定义为

$$ProjH_{hp}(r_1, r_2) = ProjH_{1,hp_1}(r_1) \oplus ProjH_{2,hp_2}(r_2).$$

由文献[11]的结论可知,函数簇 $\mathcal{H}$是相应于语言 $L_{pk,m}$ 的平滑投射 Hash 函数簇.

4 高效的 UC 安全的 PAKE 协议

本节给出高效的 UC 框架下安全的两方 PAKE 协议,简称为 EUC-PAKE 协议(如图 1 所示).协议设计用到了基于 N 次剩余假设构造的不可延展的、可提取的且是弱模拟可靠的陷门承诺体制 CS,以及相应于承诺体制 CS 的平滑投射 Hash 函数簇等组件.

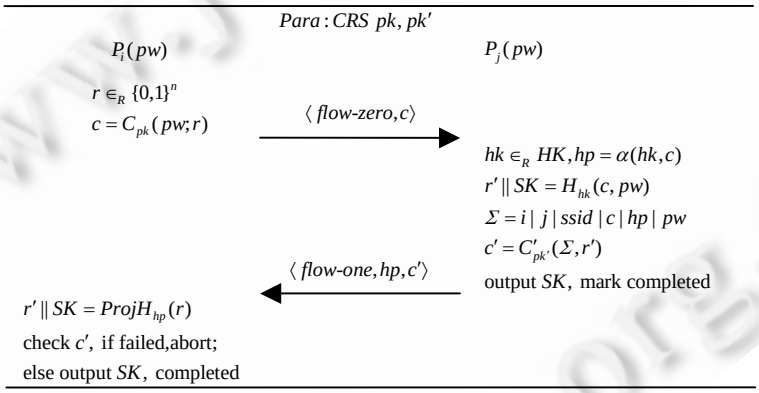

Fig.1 EUC-PAKE protocol

图 1 EUC-PAKE 协议

设 n 为安全参数,D 是口令字典.CS 是不可延展的、可提取的并且是弱模拟可靠的陷门承诺体制.不失一般性,我们仍用记号 C_{pk} 表示其中计算承诺值的函数,即用 $com=C_{pk}(pw;r)$表示利用公共参考串 pk、随机数 r 对消息 $pw\in D$ 进行承诺所得到的承诺值.设 $L_{pk,pw}$ 是按照公式(1)中定义的语言,$\mathcal{H}=\{H_{hk}\}_{hk\in HK}$ 是相应于语言 $L_{pk,pw}$ 的平滑投射 Hash 函数簇,其中,HK 为 Hash 密钥空间;H_{hk} 是从空间 $L_{pk,pw}\times D$ 到循环群 G 上的映射,密钥投射函数为 $\alpha:HK\times(L_{pk,pw}\times D)\to HP$.设 $\{ProjH_{hp}\}_{hp\in HP}$ 是利用投射密钥和相应于 NP 语言的证据计算 Hash 值的函数簇.设 CS' 是一个非交互、可提取的且不可延展的承诺体制,记 $C'_{pk'}$ 是其中计算承诺值的函数,即用 $com' = C'_{pk'}(pw;r)$ 表示利用公共参考串 pk'、随机数 r 对消息 $pw\in D$ 进行承诺所得到的承诺值.

假设 P_i 和 P_j 是想要利用 EUC-PAKE 协议协商得到高熵会话密钥的两个用户实例,它们运行协议的具体步骤如下:

(1) 在用户 P_i 被来自环境$\mathcal{Z}$的消息(*NewSession*,*ssid*,P_i,P_j,*pw*,*role*)激活后,首先检查 *role* 的值.如果 *role*=*server*,则等待形如⟨*flow-zero*,*c*⟩的消息;如果 *role*=*client*,则选择一个随机数 *r*,计算承诺值 $c=C_{pk}(pw;r)$,并发送消息⟨*flow-zero*,*c*⟩给用户 P_j.下文中,不失一般性地假定 P_i 是作为 *client* 被激活的用户,P_j 是作为 *server* 被激活的用户.

(2) 服务器 P_j 收到来自用户的消息⟨*flow-zero*,*c*⟩后,随机地选择 Hash 密钥 $hk\in_R HK$,计算相应的投射密钥 $hp=\alpha(hk,c)$和 Hash 值 $H_{hk}(c,pw)$,并将 Hash 值分成两个等长的子串 $r'\|SK$,将 r'用作随机数计算承诺值 $c'=C'_{pk'}(i\,|\,j\,|\,ssid\,|\,c\,|\,hp\,|\,pw,r')$,发送消息⟨*flow-one*,*hp*,*c*′⟩给客户 P_i,输出(*ssid*,*SK*)并标记这个会话的会话状态为 completed.

(3) 收到来自服务器的消息⟨*flow-one*,*hp*,*c*′⟩后,客户 P_i 利用投射密钥 *hp* 和相应于 $c\in L_{pk,m}$ 的证据 *r* 计算 $r'\|SK=Proj H_{hp}(r)$,利用得到的子串 r'重新生成承诺值 c',并验证它与消息⟨*flow-one*,*hp*,*c*′⟩中的最后一项是否相等,称这个过程为承诺值验证程序.如果不等,则认为验证失败,P_i 放弃相应的会话并生成随机的会话密钥;如果相等,则认为验证成功,客户 P_i 输出(*ssid*,*SK*)并标记这个会话为 completed.

注:协议中的承诺体制 *CS*′仅仅要求是非交互、可提取的且不可延展的,不要求和承诺体制 *CS* 一样满足陷门性质,也不要求存在相应的平滑投射 Hash 函数簇,因此可以比 *CS* 更有效地实现.例如,可以直接取任意 CCA2 安全的加密体制作为 *CS*′.

5 安全性证明

本节证明 EUC-PAKE 协议满足 UC 框架下 PAKE 的安全性定义,即在$\mathcal{F}_{CRS}$-混合模型中证明该协议能够安全地实现理想功能$\mathcal{F}_{pwKE}$的多会话扩展 $\hat{\mathcal{F}}_{pwKE}$.

定理 7. 设 *CS* 是不可延展的、可提取的且是弱模拟可靠的承诺体制,$\mathcal{H}=\{H_{hk}\}_{hk\in HK}$ 是相应于语言 $L_{pk,pw}$ 的平滑投射 Hash 函数簇,*CS*′是一个非交互、可提取且不可延展的承诺体制.则在考虑静态腐化攻击者的条件下,EUC-PAKE 协议在$\mathcal{F}_{CRS}$-混合模型中 UC 实现了理想功能$\mathcal{F}_{pwKE}$的多会话扩展 $\hat{\mathcal{F}}_{pwKE}$.

注:EUC-PAKE 协议的安全性证明是基于静态腐化模型的,其中,攻击者在协议开始运行之后不能再腐化用户,从而不能得到用户的内部状态.如果考虑自适应腐化模型,攻击者可以得到用户的内部状态,则相当于打开了未完成会话中的承诺,此时的安全性证明就需要比弱模拟可靠更强的安全性定义.

本节的以下部分给出上述定理的详细证明.证明过程就是针对任意的现实攻击者$\mathcal{A}$构造理想攻击者(或称为模拟者)$\mathcal{S}$,使得任意 PPT 环境$\mathcal{Z}$都不能区分是在和$\mathcal{A}$以及 EUC-PAKE 协议进行交互还是在和$\mathcal{S}$以及理想功能 $\hat{\mathcal{F}}_{pwKE}$ 进行交互.

5.1 模拟者$\mathcal{S}$的构造

给定任意的现实攻击者$\mathcal{A}$,模拟者$\mathcal{S}$的构造方式如下:$\mathcal{S}$在被激活后,首先运行承诺体制 *CS* 和 *CS*′的密钥生成算法得到公私钥对(*pk*,*sk*)和(*pk*′,*sk*′),从口令字典中随机地选择虚拟口令 pw_0.模拟者$\mathcal{S}$调用现实攻击者$\mathcal{A}$,将公共参考串 *pk*,*pk*′提供给$\mathcal{A}$,同时还将承诺体制以及平滑投射 Hash 函数簇的描述提供给$\mathcal{A}$.随后,模拟者$\mathcal{S}$按照如下步骤模拟诚实用户并与$\mathcal{A}$进行交互,同时模仿$\mathcal{A}$的行为:

- 如果模拟者$\mathcal{S}$收到了来自理想功能 $\hat{\mathcal{F}}_{pwKE}$ 的消息(*NewSession*,*ssid*,P_i,P_j,*role*),$\mathcal{S}$就以用户身份标识 P_i、会话标识 *ssid* 激活一个新的会话,记为 Π_i^{ssid}.然后,$\mathcal{S}$将会话 Π_i^{ssid} 的口令设为 $pw_i=pw_0$ 并开始运行该会话.即如果 *role*=*server*,则简单地等待一个标号为 *flow-zero* 的消息;如果 *role*=*client*,则利用口令 pw_i 按照如下方式产生消息:利用 *TfakeC* 算法生成一个伪承诺 $\tilde{c}$,并发送消息⟨*flow-zero*,$\tilde{c}$⟩给用户 P_j.
- 如果一个诚实的服务器会话 Π_j^{ssid} 收到了一条形如⟨*flow-zero*,*c*⟩的消息,$\mathcal{S}$首先检查这个消息是否是在以前的某个时刻由某个模拟的客户会话 $\Pi_i^{ssid'}$ 生成的.如果是,则将服务器会话 Π_j^{ssid} 的口令设为 $pw_j=pw_0$;如果不是,则$\mathcal{S}$利用私钥 *sk* 对承诺值 *c* 进行提取得到口令 $pw=Extract(sk,c)$,然后向理想功能 $\hat{\mathcal{F}}_{pwKE}$

发送(*TestPwd*,*ssid*,P_j,*pw*)询问.进一步来说,若理想功能返回 correct guess,则意味着 $pw=Extract(sk,c)$的确是实例 Π_j^{ssid} 所拥有的口令,$\mathcal{S}$将其口令替换成 $pw_j=pw$ 并用其进行所有后续的计算.如果理想功能返回 wrong guess,保持用 $pw_j=pw_0$作为会话 Π_j^{ssid} 的口令不变.无论哪种情况,$\mathcal{S}$都利用当前口令 pw_j模拟实例 Π_j^{ssid} 进行处理,即选择随机的 Hash 密钥 $hk\in_R HK$,计算投射密钥 $hp=\alpha(hk,c)$和 Hash 值 $H_{hk}(c,pw_j)$,并将 Hash 值分成两个等长的子串 $r'||SK$,利用 r'计算 $c'=C'_{pk'}(i|j|ssid|c|hp|pw_j,r')$,最后发送消息〈*flow-two*,*hp*,*c*′〉给客户 P_i,输出(*ssid*,*SK*),并标记这个会话的会话状态为 completed.同时,$\mathcal{S}$向理想功能 $\hat{\mathcal{F}}_{pwKE}$ 发送消息(*NewKey*,*ssid*,P_j,*SK*).

- 如果一个诚实的客户会话 Π_i^{ssid} 在发送了一条形如〈*flow-zero*,$\tilde{c}$〉的消息后收到了一条形如〈*flow-one*,*hp*,*c*′〉的消息,则首先检查上述消息是否由某个诚实的服务器会话 $\Pi_j^{ssid'}$ 在过去的某个时刻生成的.如果是,则仍旧保持 $pw_i=pw_0$并按照协议规范对该消息进行回复;如果不是,则模拟者$\mathcal{S}$利用私钥 *sk*′对 *c*′进行提取,得到 $pw=Extract'(sk',c')$,然后向理想功能 $\hat{\mathcal{F}}_{pwKE}$ 发送(*TestPwd*,*ssid*,P_i,*pw*)询问.进一步来说,如果理想功能对上述询问返回 correct guess,则将会话 Π_i^{ssid} 中的口令替换成 $pw_i=pw$;如果理想功能返回上述 wrong guess,则保持 $pw_i=pw_0$作为会话 Π_i^{ssid} 的口令不变.无论哪种情况,利用 *TfakeDecom* 算法将先前生成的伪承诺 $\tilde{c}$ 的打开为(pw_i,*r*),并将其用于平滑投射 Hash 函数的计算,得到 Hash 值 $r'||SK=ProjH_{hp}(r)$,重新生成承诺值 *c*′,并验证它与消息〈*flow-one*,*hp*,*c*′〉中的最后一项是否相等.如果不等,则认为验证失败,Π_i^{ssid} 放弃相应的会话并生成随机的会话密钥;如果相等,则认为验证成功,Π_i^{ssid} 输出(*ssid*,*SK*),并标记这个会话的会话状态为 completed.同时,$\mathcal{S}$向理想功能 $\hat{\mathcal{F}}_{pwKE}$ 发送消息(*NewKey*,*ssid*,P_i,*SK*).
- 如果攻击者$\mathcal{A}$需要发送输出给环境$\mathcal{Z}$,模拟者$\mathcal{S}$忠实地传递这些消息.如果$\mathcal{S}$收到了来自于环境$\mathcal{Z}$的输入,则也将其写入攻击者$\mathcal{A}$的输入带.

5.2 不可区分性的证明

本节证明,对于第 5.1 节构造的理想攻击者$\mathcal{S}$,任意 PPT 环境$\mathcal{Z}$都不能区分它是在与$\mathcal{A}$以及 EUC-PAKE 协议的实例进行交互还是在与$\mathcal{S}$以及理想功能 $\hat{\mathcal{F}}_{pwKE}$ 的实例进行交互.为此,我们定义了一系列的游戏,起始于真实的协议运行,结束于一个和理想模型中的协议运行计算不可区分的游戏,并证明任何两个相邻的游戏是不可区分的.

先给出如下记号:如果一个消息是在过去的某个时刻由某个诚实的客户会话或者是服务器会话生成,则称该消息是谕示生成的;否则,称其为攻击者生成的.如果一个服务器会话 Π_j^{ssid} 收到的形如〈*flow-zero*,*c*〉消息中的承诺值 *c* 满足 $pw_j=Extract(sk,c)$,其中,pw_j 是环境提供给会话 Π_j^{ssid} 的输入,则称承诺值 *c* 是有效的;类似地,如果一个服务器会话 Π_i^{ssid} 收到的形如〈*flow-one*,*hp*,*c*′〉中的承诺值 *c*′恰好是环境提供给会话 Π_i^{ssid} 的口令 pw_i 所对应的承诺值,则称承诺值 *c*′是有效的.

设 G_0 是环境$\mathcal{Z}$和现实攻击者$\mathcal{A}$以及 EUC-PAKE 协议实例进行交互的游戏,我们按照如下方式对 G_0 逐步修改,并分析相邻游戏之间的差别:

游戏 G_1:这个游戏对诚实的客户会话 Π_i^{ssid} 中消息〈*flow-zero*,*c*〉的生成方式予以修改,不再按照协议规范选择随机数并生成相应于 Π_i^{ssid} 口令的承诺值,而是利用陷门承诺的信息直接生成一个能够以多种方式打开的伪承诺 $\tilde{c}$.

引理 8. 游戏 G_1 和游戏 G_0 对于任意 PPT 环境$\mathcal{Z}$都是不可区分的.

证明:根据陷门承诺体制的定义可知,利用陷门私钥生成的伪承诺总体{(*pk*,com^*,*m*,dec^*)}和诚实承诺者生成的承诺值总体{(*pk*,*com*,*m*,*dec*)}是不可区分的.特别地,取 $m=pw_i$,则游戏 G_1 中的随机变量总体{($pk,\tilde{c},pw_i$)} 和游戏 G_0 中的随机变量总体{(*pk*,*c*,pw_i)}是不可区分的.又由于游戏 G_1 仅对承诺值 *c* 的生成方式进行了改变,环境$\mathcal{Z}$只能依据 *c* 来区分两个游戏,因此,游戏 G_1 和游戏 G_0 对任何环境$\mathcal{Z}$都是不可区分的. □

游戏 G_2:在这个游戏中,我们对诚实的客户会话中的承诺值验证程序予以修改,以保证后续游戏中修改时的一致性.首先维护一个初始为空的列表Γ,其中,存储元素的形式为$(c',*)$.当任意诚实的服务器会话 $\Pi_j^{ssid'}$ 生成 $c'=C'_{pk'}(i\,|\,j\,|\,ssid'\,|\,c\,|\,hp\,|\,pw,r')$ 时,将新的项$(c',i|j|ssid'|c|hp)$添加到列表Γ中.然后,当诚实的客户会话 Π_i^{ssid} 收到来自服务器 $\Pi_j^{ssid'}$ 谕示生成的消息$\langle flow\text{-}one,hp,c'\rangle$时,查询列表$\Gamma$得到 $i|j|ssid'|c|hp$,验证这些值和 Π_i^{ssid} 收到的 hp 以及 Π_i^{ssid} 先前发送的值 c 是否一致.同时,还验证 Π_i^{ssid} 和 $\Pi_j^{ssid'}$ 所拥有的口令是否相等.如果两个验证都通过,就认为承诺值验证程序通过;否则,就认为承诺值验证程序不通过,放弃该会话,选择均匀随机值作为会话密钥.

引理 9. 游戏 G_2 和游戏 G_1 对于任意 PPT 环境$\mathcal{Z}$都是不可区分的.

证明:实际上,可以证明游戏 G_1 中的原始承诺值验证程序成立当且仅当游戏 G_2 中修改后的承诺值验证程序成立,从而环境$\mathcal{Z}$在游戏 G_2 和游戏 G_1 中的视图是完全一样的.首先,如果游戏 G_1 中的原始承诺值验证程序成立,则客户会话 Π_i^{ssid} 收到的消息$\langle flow\text{-}one,hp,c'\rangle$中的 c'是消息串 $i|j|ssid|c|hp|pw_i$ 所对应的承诺值,其中,c 是 Π_i^{ssid} 存储的以前生成的值,pw_i 是 Π_i^{ssid} 拥有的口令,hp 是 Π_i^{ssid} 接收到的值.由于这个消息是谕示生成的,根据承诺体制的绑定性质可知,发送消息$\langle flow\text{-}one,hp,c'\rangle$的服务器会话 $\Pi_j^{ssid'}$ 拥有相同的 $i|j|ssid|c|hp|pw_i$,因此, Π_i^{ssid} 和 $\Pi_j^{ssid'}$ 拥有相同的 $i|j|ssid|c|hp$ 以及相同的口令.故,如果按照游戏 G_2 中修改后的承诺值验证程序进行验证也将通过验证;其次,如果游戏 G_2 中修改后的承诺值验证程序成立,且假设客户会话 Π_i^{ssid} 收到了服务器会话 $\Pi_j^{ssid'}$ 发送的消息$\langle flow\text{-}one,hp,c'\rangle$并接受了其中的承诺值验证,则 Π_i^{ssid} 和 $\Pi_j^{ssid'}$ 拥有相同的 $i|j|ssid|c|hp$ 以及相同的口令,这说明服务器会话 $\Pi_j^{ssid'}$ 收到的消息$\langle flow\text{-}zero,c\rangle$是 Π_i^{ssid} 发送的.因此,双方将计算得到相同的平滑投射 Hash 值,从而得到相同的 r',意味着游戏 G_1 中的原始承诺值验证程序成立. □

游戏 G_3:这个游戏和上一个游戏的不同在于,如果一个服务器会话 $\Pi_j^{ssid'}$ 收到了谕示生成的消息$\langle flow\text{-}zero,c\rangle$,则将 Π_j^{ssid} 中的平滑投射 Hash 函数的输出替换成均匀选取的随机值,也就是将 $r'||SK$ 取为均匀分布的随机值.同时,如果一个客户会话 Π_i^{ssid} 收到了来自服务器会话 $\Pi_j^{ssid'}$ 的谕示生成的消息$\langle flow\text{-}one,hp,c'\rangle$并接受了关于其中承诺值 c'的验证程序,则将 Π_i^{ssid} 中的平滑投射 Hash 函数的输出替换成和会话 $\Pi_j^{ssid'}$ 中相同的随机值.

引理 10. 游戏 G_3 和游戏 G_2 对于任意 PPT 环境$\mathcal{Z}$都是不可区分的.

证明:首先,如果客户会话 Π_i^{ssid} 收到了来自服务器会话 $\Pi_j^{ssid'}$ 的消息$\langle flow\text{-}one,hp,c'\rangle$并接受了关于其中承诺值 c'的验证程序,则说明 Π_i^{ssid} 和 $\Pi_j^{ssid'}$ 拥有相同的 $i|j|ssid|c|hp|pw_i$,双方互为匹配会话.因此,它们将计算得到相同的平滑投射 Hash 函数值.

其次,注意到游戏 G_1 已经将谕示产生的消息$\langle flow\text{-}zero,c\rangle$中的密文 c 修改为一个能够以多种方式打开的伪承诺.根据协议规范,服务器会话 $\Pi_j^{ssid'}$ 收到消息$\langle flow\text{-}zero,c\rangle$后,将利用其口令 pw_j 计算 Hash 值 $H_{hk}(c,pw_j)$.不过,由于$\langle flow\text{-}zero,c\rangle$是谕示产生的,攻击者并不知道 $c\in L_{pk,pw_j}$ 的证据 r.加上攻击者不知道 $\Pi_j^{ssid'}$ 选取的 Hash 密钥 hk,根据平滑投射 Hash 函数的伪随机性可知, $\Pi_j^{ssid'}$ 计算得到的 Hash 值和均匀分布是不可区分的. □

游戏 G_4:如果一个服务器会话 $\Pi_j^{ssid'}$ 收到了谕示生成的消息$\langle flow\text{-}zero,c\rangle$,就将 $\Pi_j^{ssid'}$ 中承诺值 c'的生成方式修改为利用虚拟口令 pw_0 生成,即改为 $c'=C'_{pk'}(i\,|\,j\,|\,ssid\,|\,c\,|\,hp\,|\,pw_0,r')$.

引理 11. 游戏 G_4 和游戏 G_3 对于任意 PPT 环境$\mathcal{Z}$都是不可区分的.

证明:首先证明 c'计算方式的修改不会改变客户会话中承诺值验证程序的结果.因为在游戏 G_2 中承诺值验证程序已经被修改为:验证查询列表Γ得到的 $i|j|ssid|c|hp$ 和客户会计算得到的 $i|j|ssid|c|hp$ 是否一致,同时验证会话 Π_i^{ssid} 和 $\Pi_j^{ssid'}$ 是否拥有相同的口令,所以验证是否成立与计算 c'所用的口令无关.因此,环境$\mathcal{Z}$只能通过密文 c'来区分游戏 G_4 和游戏 G_3,利用承诺体制 CS'的隐藏性质可知,不同口令所对应的承诺值总体是计算不可区分的,从而游戏 G_4 和游戏 G_3 具有不可区分性. □

游戏 G_5:如果一个服务器会话 $\Pi_j^{ssid'}$ 收到了攻击者生成的消息$\langle flow\text{-}zero,c\rangle$,那么模拟者$\mathcal{S}$检查密文 c 是否是

有效的.如果 c 不是有效的,将会话 $\Pi_j^{ssid'}$ 中平滑投射 Hash 函数的输出替换成均匀选取的随机值.

引理 12. 游戏 G_5 和游戏 G_4 对于任意 PPT 环境 $\mathcal{Z}$ 都是不可区分的.

证明:根据平滑投射 Hash 函数的平滑性的定义可知,当输入非有效时,其输出统计接近于均匀分布的随机变量.如果服务器会话 $\Pi_j^{ssid'}$ 收到的消息⟨*flow-zero*,*c*⟩中承诺值 c 不是有效的,即 $c \notin L_{pk,pw_j}$,其中,pw_j 为会话 $\Pi_j^{ssid'}$ 所拥有的密文,会话 $\Pi_j^{ssid'}$ 计算得到 Hash 密钥将统计接近于均匀分布.因此,将会话 $\Pi_j^{ssid'}$ 中平滑投射 Hash 函数的输出替换成均匀选取的随机值并不改变其分布,从而游戏 G_5 和游戏 G_4 是不可区分的. □

游戏 G_6:如果一个服务器会话 $\Pi_j^{ssid'}$ 收到了攻击者生成的消息⟨*flow-zero*,*c*⟩,并且承诺值 c 不是有效的,那么将会话 $\Pi_j^{ssid'}$ 中承诺值 c'的生成方式修改为利用虚拟口令 pw_0 生成,即改为 $c' = C'_{pk'}(i \mid j \mid ssid \mid c \mid hp \mid pw_0, r')$.

引理 13. 游戏 G_6 和游戏 G_5 对于任意 PPT 环境 $\mathcal{Z}$ 都是不可区分的.

证明:如果承诺值 c 不是有效的,游戏 G_5 已经将会话 $\Pi_j^{ssid'}$ 中平滑投射 Hash 函数的输出替换成均匀选取的随机值,故攻击者和任何环境都不能通过密文验证程序并生成和会话 $\Pi_j^{ssid'}$ 相同的会话密钥.因此,环境 $\mathcal{Z}$ 只能依赖于承诺值 c'本身来区分游戏 G_6 和游戏 G_5.进一步注意到,承诺体制 CS' 是计算隐藏的,因此任何 PPT 攻击者都不能区分不同口令所对应的承诺值,从而不能区分游戏 G_6 和游戏 G_5. □

到此为止,我们已经将所有诚实客户会话生成的承诺值替换成能够打开成任意口令的伪承诺,将收到谕示生成消息的服务器会话中的承诺值 c'也已经提出成虚拟口令 pw_0 所对应的密文;并且已经将收到谕示产生消息以及收到无效的攻击者生成消息的客户会话、服务器会话中的会话密钥替换成了随机值.又由于游戏 G_6 和理想模型中的客户会话和服务器会话拥有完全一样的行为,从而这两个游戏对于环境 $\mathcal{Z}$ 是不可区分的.进一步综合引理 8~引理 13 的结论可知,环境 $\mathcal{Z}$ 不能区分是在和现实的攻击者 $\mathcal{A}$ 以及现实的 EUC-PAKE 协议进行交互还是在和理想攻击者 $\mathcal{S}$ 以及理想功能 $\hat{\mathcal{F}}_{pwKE}$ 进行交互.定理 7 证毕.

6 协议效率比较

由于 Canetti 等人[7]给出的 PAKE(简记为 C-PAKE)协议在已知的标准模型下可证明 UC 安全的 PAKE 协议中效率最高,本节给出 EUC-PAKE 协议和 C-PAKE 协议的详细效率比较(见表 1).

Table 1 Efficiency comparisons of the EUC-PAKE protocol

表 1 EUC-PAKE 协议效率比较

	Round	Enc (Com)	Key-Proj	SPHF	ZKP	Sign-Ver
C-PAKE protocol[7]	6	3	2	4	1	1
EUC-PAKE protocol	2	3	1	2	0	0

在上述表格中,“Round”代表协议所需的通信轮数,实际计算轮数时,需要将零知识证明协议实例化为并发安全的、模拟可靠的具体体制[23];Enc(Com),Key-Proj,SPHF,ZKP,Sign-Ver 分别表示一次完整的协议运行中通信双方总共需要计算加密(或承诺)、密钥投射函数、平滑投射 Hash 函数、零知识证明协议以及一次性签名体制中签名-验证算法的次数.

如表 1 所述,EUC-PAKE 协议在通信轮数上要远优于 C-PAKE 协议.另外,避免一次性签名体制的使用还将降低协议对通信带宽的需求.从计算效率上讲,EUC-PAKE 协议只需要相同或更少的计算加密或承诺、密钥投射函数、平滑投射 Hash 函数的次数,而且还避免了零知识证明和一次性签名体制的计算,提高了计算效率.尽管 C-PAKE 协议中采用的 CCA2 安全加密体制可以比弱模拟可靠的承诺体制更有效地构造,当用基于 N 次剩余假设的公钥加密体制对 C-PAKE 协议进行实例化时,其中的加密、密钥投射函数、平滑投射 Hash 函数(不包括零知识证明协议所需的指数运算)需要的指数运算次数约为 28 次,EUC-PAKE 协议中所有运算总共需要的指数计算次数是 29 次.注意到 C-PAKE 协议中采用的零知识证明协议所需的指数运算次数远超过 1 次,因此,EUC-PAKE 协议整体上的计算效率仍旧优于 C-PAKE 协议.

7 结束语

首先,给出了弱模拟可靠的陷门承诺体制的定义;其次,通过构造不可延展的、可提取的且是弱模拟可靠的陷门承诺体制以及相应的平滑投射 Hash 函数簇,设计了一个标准模型下高效的 UC 安全的两方 PAKE 协议.由于采用了与 Canetti 等人的协议[7]不同的设计方法,新协议避免了零知识证明协议的使用,在保持计算复杂度相当的前提下有效地提高了通信效率,使得 PAKE 协议在 UC 框架下达到了最优的两轮.然而,该协议的安全性证明是基于静态腐化模型的,如何设计高效的、基于自适应腐化模型可证明 UC 安全的 PAKE 协议,还有待于进一步研究.

References:

[1] Bellovin SM, Merritt M. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In: Proc. of the IEEE Symp. on Research in Security and Privacy. Los Alamitos: IEEE Computer Society, 1992. 72−84. [doi: 10.1109/RISP.1992.213269]

[2] Feng DG. Research on theory and approach of provable security. Journal of Software, 2005,16(10):1743−1756 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/16/1743.htm [doi: 10.1360/jos161743]

[3] Bellare M, Pointcheval D, Rogaway P. Authenticated key exchange secure against dictionary attack. In: Preneel B, ed. Proc. of the EUROCRYPT 2000. LNCS 1807, Berlin: Springer-Verlag, 2000. 140−156.

[4] Boyko V, MacKenzie P, Patel S. Provably secure password-authenticated key exchange using Diffie-Hellman. In: Preneel B, ed. Proc. of the EUROCRYPT 2000. LNCS 1807, Berlin: Springer-Verlag, 2000. 156−171.

[5] Abdalla M, Fouque PA, Pointcheval D. Password-Based authenticated key exchange in the three-party setting. In: Vaudenay S, ed. Proc. of the PKC 2005. LNCS 3386, Berlin: Springer-Verlag, 2005. 65−84. [doi: 10.1007/978-3-540-30580-4_6]

[6] Canetti R. Universally composable security: A new paradigm for cryptographic protocols. In: Proc. of the 42nd IEEE Symp. on Foundations of Computer Science (FOCS). New York: IEEE Computer Society, 2001. 136−145. [doi: 10.1109/SFCS.2001.959888]

[7] Canetti R, Halevi S, Katz J, Lindell Y, MacKenzie P. Universally composable password-based key exchange. In: Cramer R, ed. Proc. of the EUROCRYPT 2005. LNCS 3494, Berlin: Springer-Verlag, 2005. 404−421. [doi: 10.1007/11426639_24]

[8] Katz J, Ostrovsky R, Yung M. Efficient password-authenticated key exchange using human-memorable passwords. In: Pfitzmann B, ed. Proc. of the EUROCRYPT 2001. LNCS 2045, Berlin: Springer-Verlag, 2001. 475−494.

[9] Gennaro R, Lindell Y. A framework for password-based authenticated key exchange. In: Biham E, ed. Proc. of the EUROCRYPT 2003. LNCS 2656, Berlin: Springer-Verlag, 2003. 524−543. [doi: 10.1007/3-540-39200-9_33]

[10] Jiang SQ, Gong G. Password based key exchange with mutual authentication. In: Handschuh H, Hasan A, eds. Proc. of the SAC 2004. LNCS 3357, Berlin: Springer-Verlag, 2004. 267−279.

[11] Yin Y, Li B. Provable secure encrypted key exchange protocol under standard model. Journal of Software, 2007,18(2):422−429 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/18/422.htm [doi: 10.1360/jos180422]

[12] Abdalla M, Chevalier C, Pointcheval D. Smooth projective hashing for conditionally extractable commitments. In: Halevi S, ed. Proc. of the CRYPTO 2009. LNCS 5677, Berlin: Springer-Verlag, 2009. 671−689. [doi: 10.1007/978-3-642-03356-8_39]

[13] MacKenzie P, Yang K. On simulation-sound trapdoor commitments. In: Cachin C, Camenisch J, eds. Proc. of the EUROCRYPT 2004. LNCS 3027, Berlin: Springer-Verlag, 2004. 382−400. [doi: 10.1007/978-3-540-24676-3_23]

[14] Canetti R, Rabin T. Universal composition with joint state. In: Boneh D, ed. Proc. of the CRYPTO 2003. LNCS 2729, Berlin: Springer-Verlag, 2003. 265−281. [doi: 10.1007/978-3-540-45146-4_16]

[15] Canetti R, Krawczyk H. Universally composable notions of key exchange and secure channels. In: Knudsen LR, ed. Proc. of the EUROCRYPT 2002. LNCS 2332, Berlin: Springer-Verlag, 2002. 337−351. [doi: 10.1007/3-540-46035-7_22]

[16] Dolev D, Dwork C, Naor M. Nonmalleable cryptography. SIAM Journal on Computing, 2000,30(2):391−437. [doi: 10.1137/S0097539795291562]

[17] Canetti R, Fischlin M. Universally composable commitments. In: Kilian J, ed. Proc. of the CRYPTO 2001. LNCS 2139, Berlin: Springer-Verlag, 2001. 19−40. [doi: 10.1007/3-540-44647-8_2]

[18] Beaver D. Adaptive zero-knowledge and computational equivocation. In: Proc. of the 28th Annual ACM Symp. on Theory of Computing (STOC). New York: ACM Press, 1996. 187−196. [doi: 10.1145/237814.238014]

[19] Cramer R, Shoup V. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen LR, ed. Proc. of the EUROCRYPT 2002. LNCS 2332, Berlin: Springer-Verlag, 2002. 45−64. [doi: 10.1007/3-540-46035-7_4]

[20] Nishimaki R, Fujisaki E, Tanaka K. Efficient non-interactive universally composable string-commitment schemes. In: Pieprzyk J, Zhang F, eds. Proc. of the ProvSec 2009. LNCS 5848, Berlin: Springer-Verlag, 2009. 3−18. [doi: 10.1007/978-3-642-04642-1_3]

[21] Peikert C, Waters B. Lossy trapdoor functions and their applications. In: Proc. of the 40th Annual ACM Symp. on Theory of Computing (STOC). ACM Press, 2008. 187−196. [doi: 10.1145/1374376.1374406]

[22] Paillier P. Public-Key cryptosystems based on composite degree residuosity classes. In: Stern J, ed. Proc. of the EUROCRYPT'99. LNCS 1592, Berlin: Springer-Verlag, 1999. 223−228.

[23] Garay J, MacKenzie P, Yang K. Strengthening zero-knowledge protocols using signatures. Journal of Cryptology, 2006,19(2): 169−209. [doi: 10.1007/s00145-005-0307-3]

附中文参考文献:

[2] 冯登国.可证明安全性理论与方法研究.软件学报,2005,16(10):1743−1755. http://www.jos.org.cn/1000-9825/16/1743.htm [doi: 10.1360/jos161743]

[11] 殷胤,李宝.标准模型下可证安全的加密密钥协商协议.软件学报,2007,18(2):422−429. http://www.jos.org.cn/1000-9825/18/422.htm [doi: 10.1360/jos180422]

胡学先(1982－),男,湖北红安人,博士,主要研究领域为安全协议.

刘文芬(1965－),女,博士,教授,博士生导师,CCF 会员,主要研究领域为密码学.

张振峰(1972－),男,博士,研究员,博士生导师,主要研究领域为安全协议.