中国科学 E 辑: 信息科学
2007 年 第 37 卷 第 2 期: 272~284
http://www.scichina.com

《中国科学》杂志社
SCIENCE IN CHINA PRESS

通用可组合的匿名 HASH 认证模型

张 帆 马建峰

(西安电子科技大学计算机网络与信息安全教育部重点实验室, 西安 710071)

文相在

(国立庆北大学移动网络安全技术研究中心, 韩国 大邱广域市 702-701)

摘要 理想函数是通用可组合安全的核心组成部分, 但是目前通用可组合安全框架中定义的认证理想函数通过将身份与消息和签名值绑定的方式来实现对身份的认证, 没能充分体现出采用其他形式进行匿名认证的特殊需求. 受到 Marten 的启发, 文中利用通用可组合安全定义并实现了一种适用于无线网络的匿名 Hash 认证理想函数, 并在此基础上定义了一个具有普遍意义的 Hash 证书权威模型. 定义了匿名 Hash 认证机制的安全需求和安全概念, 并且证明在标准模型(非随机预言机模型)下所提匿名 Hash 认证机制的安全属性可以通过安全对称加密机制、安全数据签名机制、伪随机函数以及单向无碰撞 Hash 函数的组合得到保证. 考虑到无线网络的特殊限制, 以及移动终端设备的有限计算能力, 本理想函数主要采用对称密码原语来实现身份认证.

关键词 通用可组合安全 匿名 认证 Hash

安全协议的研究一直是人们关注的话题, 一般来说, 安全协议研究分为两大阵营: 符号逻辑观点和计算复杂性观点 [1]. 符号逻辑观点用简单的形式化语言和符号方法进行协议设计和推理分析, 它包括: BAN逻辑、GSP方法、NRL协议机等等 [2]. 计算复杂性观点用一种计算模型和复杂性理论进行协议设计和推理分析, 它包括当前流行的安全性证明Random Oracle模型 [3]、安全多方计算模型 [4,5]、Canetti-Krawczyk模型 [6,7]、通用可组合安全模型 [8]等等.

通用可组合安全(Universally Composable Security, UC Security)的概念是由Ran Canetti在总结前人工作 [9]的基础上于 2001 年提出的计算复杂性理论模型. 通用可组合的安全框架主要用于描述和分析并发环境下的协议安全问题 [10,11]. 并发组合是现实网络环境中的实际情形, 在孤立模型中证明安全的协议在组合情形下不一定是安全的. 因此, 在孤立模型中证明一个

收稿日期: 2006-07-13; 接受日期: 2006-09-30
韩国MIC以及ITRC(批准号: IITA-2006-C1090-0603-0026)和国家自然科学基金(批准号: 30204012, 60573035, 60573036)资助项目

协议的安全性还是不够的. 通用可组合安全框架中最重要的属性是它能够确保协议在任意的和未知的多方环境中运行时仍然是安全的 [8]. 该框架采用的方法是将一个协议在独立的环境中进行分析, 然后利用其安全定义来确保它在组合情形下的安全. 在复杂环境中, 协议的安全性可以通过通用可组合定理来得到保证. 通用可组合安全是更高级别的安全定义, 它的抽象层次远远超过其他模型, 因此它的安全定义更为严格 [12,13]. 通用可组合安全最优秀的性质就是模块化的设计思想: 可以单独设计协议, 只要协议满足UC安全, 那么就可以保证和其他协议并行运行的安全. 这个框架对于不同的协议采用相同的方法来处理安全的概念.

目前对UC安全的研究主要集中在以下 3 点 [14−17]:

(1) 寻找与其他研究手段的结合点, 比如与*Spi*演算、Model Checking等非计算性观点的结合 [18], 在这一方面已经取得了一定的进展 [19];

(2) 对UC模型进行优化, 或者抛弃一般性, 提出一些限制条件下可安全实现的UC模　型 [20];

(3) 提出新的理想函数(ideal functionality), 并且对其形式化, 争取安全实现.

理想函数是 UC 安全框架中非常重要的安全概念, 它扮演着一个不可攻陷的可信第三方的角色, 能够完成协议所执行的特定功能. 目前已经定义了多个最基本的理想函数, 如认证消息传输 F_{AUTH}、安全消息传输 F_{SMT}、密钥交换 F_{KE}、公钥加解密 F_{PKE}、签名 F_{SIG}、承诺 F_{COM}、不经意传输 F_{OT} 等. UC 安全协议设计的困难所在和核心内容就在于形式化和抽象一个完美的并且可以安全实现的理想函数.

目前 UC 框架中定义的认证理想函数 F_{CERT} 对身份的认证采用的是将身份与消息和签名值绑定的方式来实现的, 但是这种绑定方式主要反映了利用公钥证书进行的身份认证, 而没能充分体现出采用其他形式的证书实现匿名认证的特殊需求.

在无线环境下, 用户的真实身份、当前位置及其运动模式是重要而又敏感的信息, 在通信中必须保证它们的机密性. 由于移动用户与通信网之间采用无线通信, 用户的身份认证必须通过无线信道来进行, 因此易受截获、窃听和攻击, 所以必须采用有效的手段实现匿名认证. 目前广泛采用以下两种方式实现匿名认证, 第一种是移动终端利用认证器的公钥或它与认证器共享的密钥加密其身份, 这种方式只能做到对第三方(除了移动终端和认证器外)的匿名, 无法做到对认证器的匿名认证; 第二种是移动终端利用认证服务器的公钥或它与认证服务器共享的密钥加密其身份, 这种情形可以实现对认证器的身份匿名, 但是在认证身份时需要借助认证服务器来实现身份鉴别, 因而效率太低, 并且增加了安全风险. 我们认为对于理想的匿名身份认证, 应该只有认证服务器知道移动终端的身份, 而认证器不应该识别出移动终端的真实身份. 此外, 出于安全及效率方面的考虑, 认证过程应该直接在移动终端和认证器之间进行, 而不应该再涉及认证服务器. 对于这种理想的匿名认证, 移动终端不能预先与认证器建立某种信任关系, 如预先建立共享密钥, 也不能使用公钥证书进行身份认证.

我们利用通用可组合安全的框架定义并实现了一种适用于无线环境的匿名Hash认证理想函数 F_{Cred}, 以及具有普遍意义的、可颁发匿名 Hash 证书的 CA 模型 F_{HCA}. 我们证明所提匿名身份认证理想函数的安全属性可以通过安全对称加密机制、安全数据签名机制以及单向无碰撞 Hash 函数的组合得到保证. 证明过程采用简单(plain)模型(即假设网络是开放、未认证、异步的), 而不是随机预言机(random-oracle)模型.

本认证机制没有采用公钥证书或预共享密钥的形式进行认证. 考虑到无线环境的特殊限制, 以及移动终端设备的有限计算能力, 本认证机制主要采用对称加密原语, 如对称加密机制、Hash 函数和伪随机函数来实现身份认证. 移动终端在认证时只需要进行计算量极低的伪随机函数的操作, 而认证器只需要进行一次签名验证和伪随机函数的操作, 甚至在大部份情形下都可以省略签名验证的过程, 因而具有较高的效率.

本文章节安排如下: 在第 1 节简要介绍了通用可组合安全的概念; 第 2 节给出了文中用到的定义及预备知识; 在第 3 节和第 4 节, 分别介绍了匿名 Hash 认证的理想函数 F_{Cred} 和真实协议 π_{Cred}; 在第 5 节, 详细描述了仿真器 $\mathcal{S}$ 的功能和作用, 并证明理想函数 F_{Cred} 与真实协议 π_{Cred} 是不可区分的; 最后一部分对全文进行了总结.

1 通用可组合安全

通用可组合安全(UC安全)是用于定义密码协议安全性的框架 [8], 在该框架中, 定义了一个可以提供某种服务的不可攻陷的理想函数 $\mathcal{F}$、虚拟参与者 $\tilde{P}$ 以及理想攻击者 $\mathcal{S}$. 只有虚拟参与者 $\tilde{P}$ 和理想攻击者 $\mathcal{S}$ 可以访问理想函数 $\mathcal{F}$, 每个虚拟参与者之间不能直接通信, 理想攻击者 $\mathcal{S}$ 可以在任何时间攻陷任意的虚拟参与者, 它只能获悉消息的发送时间, 但不能得到具体的消息内容, 它可以延迟发送消息, 但不能改变消息的内容. 与此相对应, 在该框架中还定义了能够实现上述特殊服务的真实协议 π、实际参与者 P 以及真实环境下的攻击者 $\mathcal{A}$. 每一个实际参与者间可以直接通信, 攻击者 $\mathcal{A}$ 可以控制他们之间的所有通信, 也就是说, $\mathcal{A}$ 可以读取及篡改实际参与者间传递的任何通信内容, $\mathcal{A}$ 也可以在任何时候攻陷任何的实际参与者. 在UC的安全框架中, 利用一个环境机$\mathcal{Z}$来模拟协议运行的整个外部环境(包括其他并行的协议、攻击者等等), $\mathcal{Z}$可以与所有的参与者($\tilde{P}$ 和 P)以及攻击者 $\mathcal{A}$ 和 $\mathcal{S}$ 直接通信, $\mathcal{Z}$不允许直接访问理想函数 $\mathcal{F}$. 通用可组合的安全框架如图 1 所示.

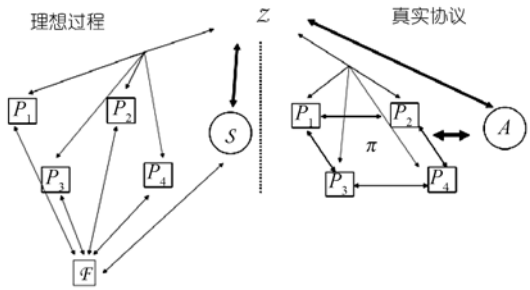

图 1 通用可组合的安全框架

定理 1(通用可组合安全universal composable security)[8] 在UC的安全框架中, 如果真实协议 π 可以在任何环境对于任何攻击者 $\mathcal{A}$ 都有与理想函数 $\mathcal{F}$ 同样的“行为”时, 就认为这是协议的一个安全实现. 具体地说: 如果对于任意的攻击者 $\mathcal{A}$ 和环境机$\mathcal{Z}$而言, 始终存在有一个理想对手 $\mathcal{S}$, 使得$\mathcal{Z}$无法区分是与虚拟参与者 $\tilde{P}$ 和 $\mathcal{S}$ 的交互, 还是与实际参与者 P 和 $\mathcal{A}$ 的交互, 就认为协议 π 安全地实现了理想函数 $\mathcal{F}$ 的功能(属性). Canetti等人证明了这个安全的定义具有一定的可组合性, 并在此基础上开展了许多的工作.

定理 2(组合理论composition theorem)[8] UC安全最重要的意义就在于可以利用已经设计好的子协议, 安全地构建一个更为复杂的协议, 从而实现指定的任务, 并保证相应的安全属性. 通常一个复杂的系统可以分解成多个子系统, 每一个子系统都可以实现某个安全任务. Canetti将这条性质定义为组合理论. 组合理论可以保证通过使用已经证明是UC安全的子协议来构建

一个更为复杂的、满足UC安全的密码协议.

定理 3(混合模型hybrid model)[8] 为了描述上述理论以及形式化表述一个真实协议访问理想函数的多个副本(copy)的情形(如签名), Canetti引入了混合模型的概念. 参与者除了在彼此之间发送信息, 还可以与无限数量的理想函数 $\mathcal{F}$ 的副本进行交互. 理想函数的副本通过会话标识SID来区分, 发送给某一副本以及从该副本发出的所有消息都对应唯一的标识SID.

目前对 UC 安全的理论已经做了大量研究, 尽管已经提出了多个实现某种安全任务的理想函数, 但是还没有提出与匿名 Hash 认证相关的理想函数, 这就是本文工作的主要出发点.

2 定义及预备知识

定义 1(无碰撞Hash函数)[21] 对于密钥为a的Hash函数 $H_a: A \to \{0, 1\}^n$, 如果在多项式时间$|a|$内, 找到两个值 $x, y \in A$,且 $x \neq y$, 使得 $H(x)=H(y)$ 的概率是可忽略的, 则 H_a 是无碰撞的Hash函数.

定义 2(单向Hash函数)[21] 对于Hash函数 $H: A \to \{0, 1\}^n$ 而言, 如果给定某个随机数 x, 在多项式时间无法找到其原像 $y \in A$ 满足 $H(y)=x$, 则H是单向的Hash函数.

定义 3(伪随机函数)[21] 对于函数 $R_a: A \to B$ 而言(其中a为参数), 如果多项式时间$|a|$内的图灵机无法区分 R_a 和均匀分布函数 $f: A \to B$, 则 R_a 是伪随机函数.

定义 4(语义安全)[21] 对于对称加密机制(E,D), 如果多项式时间的概率图灵机T, 在挑选消息m_0 和m_1 的情形下, 不能以不可忽略的优势区分$E(m_0)$和$E(m_1)$, 则对称加密机制(E,D)是语义安全的.

定义 5(签名机制的有效性和选择消息安全CMA)[22] 对于签名机制 $SS=(Kg, Sig, Vf)$,如果对于由密钥生成算法 Kg 产生的公/私钥对 (pk, sk) 和任意消息 m 而言, 都能满足 $Vf_{pk}(m, Sig_{sk}(m))=1)$, 则签名机制 SS 是有效的. 如果攻击者在访问签名预言机 $Sig_{sk}(\ )$ 的情形下, 对任何消息而言都无法产生有效的消息-签名对, 则签名机制 SS 可以安全抵抗选择消息攻击, 是CMA安全的.

定义 6(Merkle树Hash链)[23] Merkle树(Merkle tree)是一种特殊的二叉树, 其集合中的元素作为树的叶子结点, 树的每一内部结点是其左右孩子级联后的Hash值, 最后再以签名等方式认证根结点. 在建立Merkle树后, 要想构造一条包含有不属于树中叶子结点的Hash链表就意味着能够找到该Hash函数H的一个碰撞, 而这违背了所选Hash函数的无碰撞性假设, 因此Merkle树可以用来证明成员的隶属关系, 即某个元素是否属于一个集合.

对于一棵 Merkle 树, 可以构造一条从叶子结点到树的根结点的 Hash 链表, Hash 链表的每一单元中都包含一个元素和一个位置标识 o_i, 该标识用于表明元素应该从左侧(l)还是右侧(r)进行级联.

基于无碰撞性 Hash 函数 H 的 Hash 链在满足下列条件下, 我们称其为有效的: 如果 $h_0=h'_o$, $h'_{d-1}=v$, 且 $h'_{i-1}=H(h_i \| h'_i)/H(h'_i \| h_i)$ (其中 $o_i=l/r$), 对于 $i=d-1, d-2, \cdots, 1$ 都成立, 用 $isvalid(h)=1$ 来表示. 我们定义选取 Hash 链根结点的算法为 $root(h)=h_0$, 选取 Hash 链叶子结点的算法为 $leaf(h)=v$, 采用 Hash 函数 H 将集合 C 中的元素构建成 Merkle 树的算法为 $buildtree_H(C)$, 获取 Merkle 树 T 中元素 e 的路径算法为 $getchain_T(e)$.

3 匿名 Hash 认证理想函数 F_{Cred}

UC框架中定义的认证理想函数 F_{Cerd} 具有普遍意义，是安全消息传输、密钥交换和安全会话的基础模型，它采用将身份与消息和签名值绑定的方式来实现身份认证，并依靠公钥证书权威理想函数 F_{CA} 的辅助来实现 [24]. 但是这种身份与消息和签名值的绑定没能充分体现出采用其他形式的证书进行匿名认证的特殊需求. 在文献 [25]中, Marten设计了一种电子货币机制, 本匿名Hash认证理想函数受到了它的启发, 并完善了UC框架下的安全性证明.

认证机制的工作原理如下: 在初始阶段, 移动终端需要向认证服务器发送建立身份凭证的请求; 为了实现匿名性, 认证服务器采用对称加密机制加密用户的身份. 移动终端利用伪随机函数产生一系列数值并将这些数值的 Hash 结果发送给认证服务器. 认证服务器利用加密的用户身份以及这些 Hash 值构成移动终端对应的身份凭证. 为了提高效率, 本认证机制采用 MerkleHash 树的方式存储身份凭证, 即将移动终端的身份凭证作为叶子结点插入 MerkleHash 树中, 认证服务器需要对此 MerkleHash 树的根结点进行签名, 并公布该根结点. 在认证时, 为了证明拥有正确的身份凭证, 移动终端需要向认证器公布一定数量的 Hash 原像值以及从此凭证所在的叶子结点到根结点的路径. 认证器通过验证 Hash 原像的正确性, 以及路径的有效性, 确定移动终端的身份是否可靠.

3.1 匿名 Hash 认证理想函数 F_{Cred} 的安全需求

在 UC 模型中, 每个实体都是一个交互式图灵机. 本匿名 Hash 认证理想函数中的参与者分别用 ASU(认证服务器), $P_1,\cdots,P_m$ (包括认证器)来描述. 出于简化的目的, 我们用 P_0 来标识 ASU, 用 $P_1,\cdots,P_m$ 来标识移动用户和认证器.

模型中使用了两个安全参数 k_1 和 k_2, k_1 是对称加密机制的密钥长度, k_2 是标识认证器身份的数据串的长度. 模型中使用了一个影射函数 ℓ 将认证器的身份影射到 $[k_2]$ 空间, 并且保证 $\ell(p_j)$ 的基数为 $k_2/2$, 且 $p_i \neq p_j$ 时 $\ell(p_i) \neq \ell(p_j)$.

当移动终端 P_i 需要接入认证器 P_j 时, 它计算 $\ell(p_j)$ 值并将秘密信息中所有与“1”对应位置的原像提供给 P_j 检验, P_j 也可以通过计算 $\ell(p_j)$ 来判断是否发送给自己的认证凭证.

设身份凭证表示为 $c_i=(c,p_i,k,h)$, 其中 c 是加密的用户身份, P_i 是用户的身份标识, k 是该身份凭证的秘密信息, 长度为 k_2 位, h 是该凭证在 Merkle 树中的 Hash 链路径. 身份凭证 c_i 的值定义为 $\mathrm{val}_H(c_i)=c\|H(k_1)\|H(k_2)\|\cdots\|H(k_{k_2})$.

理想函数 F_{Cred} 中设有一个计数器 t, 初始值为 0, 用于索引在阶段 i 发布的身份凭证 c_i, 准备使用的身份凭证集合 $T_{prepared}$, 以及身份凭证集合 $C=\bigcup c_i$, 它们在初始时设为空 ϕ.

定义 7 设k是一个安全参数, $\varepsilon(k)$ 是基于k的可忽略函数. s 是签名密钥, v 是验证密钥. 如果下列性质成立, 则匿名Hash认证函数满足安全需求 [26]:

■ **完备性**. 对于任意的有效凭证 (c,p_i,k,h), $Prob[(s,v)\leftarrow gen(1k);0\leftarrow Verify\ \ Credential(c,z,k,p_j,h,\sigma,v)]<\delta(k)$ 都成立, 其中 σ 是根结点 $root(h)$ 的签名值.

■ **一致性**. 对于任意的有效凭证 (c,p_i,k,h) 而言, 两次独立调用验证算法 *Verify*

$Credential(c,z,k,p_j,h,\sigma,v)$ 产生不一样的输出结果的概率小于 $\varepsilon(k)$.

■**不可篡改性**. 对于一个多项式时间的篡改者 F，在 F 没有请求签名 Oracle F_{SIG} 签名根结点 $root(h)$ 的情形下，概率公式 $\text{Prob}[(s,v)\leftarrow gen(1^k);(c,p_j,k,h)\leftarrow F^{\pi_{Cred}}(v),1\leftarrow a\ Verify\ Credential(c,z,k,p_j,h,\sigma,v)]<\delta(k)$ 始终成立.

3.2 匿名 Hash 认证理想函数 F_{Cred} 的构造

匿名 Hash 认证直接将 Hash 值与实体绑定在一起，理想函数 F_{Cred} 中涉及对称加密机制 $CS=(Kg,E,D)$、签名机制 $SS=(Kg,Sig,V_f)$、伪随机函数簇 $\mathcal{R}$及无碰撞的单向 Hash 函数 $\mathcal{H}$. 我们假设签名机制 SS 满足 CMA 安全.

Present Credential

一接收到实体 p_i 发来的消息 $(Present\ Credential,p_i,c,z,p_j)$，将该消息转发给攻击者，一收到攻击者返回的消息，将其转发给 p_i.

Verify Credential

一接收到实体 p_i 发来的消息 $(Verify\ Credential,p_i,c,z,\tilde{k},p_j,h',\sigma,v)$，将该消息转发给攻击者，一收到攻击者返回的消息，将其转发给 p_i.

Check Reuse

一接收到实体 p_i 发来的消息 $(Check\ Reuse,p_s,c,z,\tilde{k}_1,\tilde{k}_2,p_{j_1},p_{j_2}h,\sigma,v)$，执行下列操作 $(Verify\ Credential,c,z,\tilde{k}_i,h,\sigma,p_{j_i})$，其中 i=1,2:

(1) 如果至少一个操作返回消息 $(Verify\ Credential,c,P_{ji},invalid)$，则返回消息 $(P_i,Check\ Reuse,c,invalid)$ 给仿真器 $\mathcal{S}$.

(2) 如果 $P_{j_1}=P_{j_2}$，则返回消息 $(P_l,Check\ Reuse,c,no)$，否则返回消息 $(P_i,Check\ Reuse,c,yes)$ 给仿真器 $\mathcal{S}$.

匿名 Hash 认证理想函数的 UC 安全定义如下:

定义 8 设 F_{Cred} 为匿名 Hash 认证理想函数，$\tilde{P}$ 是虚拟参与者集合，$\mathcal{S}$ 为理想攻击者，π_{Cred} 为实现 F_{Cred} 的真实协议，P 为真实参与者集合，$\mathcal{A}$ 为真实攻击者，$\mathcal{Z}$ 为与 P 和 $\mathcal{A}$ 交互的环境机. 如果对于任意的 $\mathcal{A}$ 和 $\mathcal{Z}$ 而言，都存在一个 $\mathcal{S}$，使得 $\mathcal{Z}$ 无法区分是与虚拟参与者 $\tilde{P}$ 和 $\mathcal{S}$ 的交互，还是与实际参与者 P 和 $\mathcal{A}$ 的交互，则 π_{Cred} 可以安全实现 F_{Cred}.

4 构造 UC 安全的匿名 Hash 认证真实协议

下面我们在混合模型(hybrid)下构造有签名理想函数 F_{SIG} 辅助的真实协议 π_{Cred}，假设它可以理想地仿问一个可信匿名 Hash 证书权威，我们将该证书权威形式化为具有普遍意义的理想函数 F_{HCA}. 我们首先描述 F_{SIG} 和 F_{HCA}，然后进一步构造真实协议 π_{Cred}.

签名理想函数 F_{SIG} [24,26]可以提供一种注册服务，一个确定的实体可以注册一个(*message*, *signature*)对. 任何正确提供验证密钥的实体都可以检查给定的消息/签名对是否已经注册.

F_{SIG} 通过接受***KeyGen***, ***Sign***, ***Verify***命令来执行生成密钥、消息签名及验证签名的操作. 根据UC安全理论, 我们将签名理想函数 F_{SIG} 修改成如下形式:

Key generation

一接收到 P 发来的消息 $(KeyGen, P)$, 将该消息转发给仿真器 S, 从 S 获得 P 的公钥 $(VerificationKey, P, \theta)$, 记录 (P, θ), 并将消息 $(VerificationKey, P, \theta)$ 发送给 P.

Signature generation

一接收到 P 发来的消息 $(Sign, P, m)$, 将该消息转发给仿真器 S, 从 S 处获得签名值 $(Signature, P, m, \sigma)$, 查找记录 $(m, \sigma, \theta, 0)$, 如果存在的话, 发送一条出错信息并退出, 否则存储记录 $(m, \sigma, \theta, 1)$ 并发送 $(Signature, P, m, \sigma)$ 给 P.

Signature verification

一接收到验证者 V 发来的消息 $(Verify, P, m, \sigma, \theta')$, 将该消息转发给仿真器 S, 从 S 处获得验证签名消息 $(Verified, P, m, \phi)$,

1. 如果 $\theta' = \theta$ 且存在记录 $(m, \sigma, \theta, 1)$, 则令 $f = 1$;
2. 否则, 如果 $\theta' = \theta$, P 未被 S 攻陷, $\forall \sigma'$ 而言都不存在记录 $(m, \sigma', \theta, 1)$, 则令 $f = 0$;
3. 否则, 如果 $\theta' \neq \theta$, 且存在记录 (m, σ, θ', f'), 则令 $f = f'$;
4. 否则, 令 $f = \phi$;

存储记录 $(m, \sigma, \theta', \phi)$, 并将消息 $(Verified, P, m, f)$ 发送给 V.

下面定义匿名 Hash 证书权威理想函数 F_{HCA}:

Key generation

■ 一接收到 ASU 发来的 $(Generate\ Key)$ 消息, 将该消息发送给攻击者 S, 一旦接收到 S 发来的消息 $(Verification\ Key, ASU, v, encryption\ key, k)$, 记录 (ASU, v, k), 并返回消息 $(Verification\ Key, ASU, v)$.

Identity encryption

■ 一接收到 p_i 发来的消息 $(Identity\ encryption, p_i)$,

1. 验证 P_i 是否在成员列表中, 如果没有, 则返回消息 $(Not\ A\ Member, p_i)$ 并退出.
2. 否则, 发送 $(Identity\ encryption, p_i)$ 给仿真器 S, 获得 P_i 的加密身份 c, 返回消息 $(Encrypted\ identiy, p_i, c)$.

Credential generation

■ 一接收到 p_i 发来的消息 $(Credential\ generation, p_i, (c, p_i, k, z))$, 将该消息发送给仿真器 S, 等待 S 返回的"*OK*"消息, 将身份凭证 $e = (c, p_i, k, \phi)$ 存储到集合 C_t 中, 返回消息 $(S, New\ Credential, p_i)$ 和 $(p_i, New\ Credential, c, z)$ 给 S.

Build tree

■一接收到 ASU 发来的 $(Build\ tree, ASU)$ 消息, 建立 Merkle 树 $T \leftarrow buildtree_H(val_H(C_t))$, 并将集合 C_t 中的身份凭证 $e = (c, P_i, k, \phi)$ 的路径修改为 $(c, p_i, k, getchain_T(val_H(e)))$. 发送消息 $(Sign, ASU, root(T))$ 给仿真器 S, 等待仿真器 S 返回的消息 $(Signature, ASU, root(T), \sigma)$, 检验有

没有记录 $(root(T),\sigma,1)$ 存在，如果有，则输出一条出错消息并停止，否则，记录下 $(root(T),\sigma,1)$，返回消息 $(Build\ Tree, ASU, T, \sigma)$ 给仿真器 $\mathcal{S}$, 并令 $t \leftarrow t+1$.

Add prepared credential

■一接收到 p_i 发来的消息 $(Add\ \ prepared\ credential, p_i, (c, p_j))$，将该消息发送给仿真器 $\mathcal{S}$, 等待 $\mathcal{S}$ 返回的"*OK*"消息，将 (c, p_j) 存储到集合 $T_{prepared}$ 中，返回消息"*OK*".

Check prepared credential

■ 一接收到 p_i 发来的消息 $(Check\ \ prepared\ credential, p_i, (c, p_j))$，将该消息发送给仿真器 $\mathcal{S}$, 等待 $\mathcal{S}$ 返回的"*OK*"消息，随后在集合 $T_{prepared}$ 中查找 (c, p_j)，如果找到返回消息"*OK*".

Check exist of credential

■ 一接收到 p_i 发来的消息 $(Check\ exist\ of\ credential, p_i, (c, p_j, k, h))$，将该消息发送给仿真器 $\mathcal{S}$, 等待 $\mathcal{S}$ 返回的"*OK* "消息，随后在集合 C 中查找 (c, p_j, k, h)，如果找到返回消息"*OK* ".

Reveal ID

■ 一接收到 ASU 发来的 $(Reveal\ ID, ASU, c)$ 消息，在集合 C 中查找身份凭证 $(c, p, \cdot, \cdot)$. 如果没找到对应的身份凭证，发送消息 $(Reveal\ ID, ASU, c)$ 给仿真器 $\mathcal{S}$, 等待仿真器 $\mathcal{S}$ 返回的消息 (c, p)，然后返回消息 $(Reveal\ ID, ASU, c, p)$.

在 (F_{SIG}, F_{HCA}) 辅助的混合模型下，协议 π_{Cred} 的描述如下:

Present Credential

1. p_i 一接收到 $(Present\ Credential, c, z, p_j)$ 消息,

2. 如果 p_i 本身还没有产生身份凭证的话，它利用伪随机函数产生一个对称密钥 $R^i \xleftarrow{R} R^{k_1}$，并将消息 $(Identity\ encryption, p_i)$ 发送给 F_{HCA}. 一接收到 F_{HCA} 返回的消息 $(Encrypted\ identiy, p_i, c)$，$p_i$ 计算秘密信息 $k_j \leftarrow (R^i(c \parallel j))_{j=1}^{k_2}$ $z \leftarrow H(k)$，发送消息 $(Credential\ \ generation, p_i, (c, p_i, k, z))$ 给 F_{HCA}

3. 否则，p_i 验证 $k_l \leftarrow (R^i(c \parallel l))_{l=1}^{k_2}$ 且 $z = H(k)$，计算 $\tilde{k} \leftarrow k_{\ell(P_j)}$ 并输出$(Present\ \ Credential, c, \tilde{k})$.

Verify Credential

■ 一接收到 p_i 发来的 $(c, z, \tilde{k}, p_j, h, \sigma, v)$ 消息

1. P_i 向签名函数 F_{Sig} 发送验证签名消息 $(Verify\ \ p_i, root(h), \sigma, v)$，执行签名验证过程;

2. P_i 向 F_{HCA} 发送 $(Check\ \ prepared\ credential, p_i, (c, p_j))$，等待 F_{HCA} 返回的"*OK* "消息;

3. P_i 验证身份凭证是否属于 Hash 树的一个叶子结点，即 $H(c, z) = leaf(h)$，并计算路径的有效性 $isvalid_H(h) = 1$;

4. 如果 F_{SIG} 返回消息"0"，或者不满足条件 2 和 3 中的任意一条，则返回消息 $(Verify\ \ Credential, p_i, c, p_j, invalid)$ 并退出;

5 否则 P_i 返回验证成功的消息($(Verify\ \ Credential, c, P_j, valid)$.

Check Reuse

■ 一接收到检查重复认证的消息 $(Check\ Reuse, c, z, \tilde{k}_1, \tilde{k}_2, h, \sigma, p_{j_1}, p_{j_2})$，$P_i$ 执行如下操作 $(Verify\ Credential, c, z, \tilde{k}_i, h, \sigma, p_{j_i})$，其中 i = 1, 2:

1. 如果至少有一次操作返回无效的结果 $(Verify\ Credential, c, P_{j_i}, invalid)$，则返回消息 $(Check\ Reuse, c, invalid)$ 并退出;

2. 如果 $P_{j_1} = P_{j_2}$，则不认为发生了重复接入(该检查交由 P_{j_i} 来完成)，并返回消息 $(Check\ Reuse, c, no)$，否则返回消息 $(Check\ Reuse, c, yes)$.

5 在 (F_{SIG}, F_{HCA}) 辅助的混合模型下 π_{Cred} 安全实现 F_{Cred}

定理 4 根据 UC 安全的定义，对于任意对手而言，在 (F_{SIG}, F_{HCA}) 辅助的混合模型下，协议 π_{Cred} 可以安全地实现匿名 Hash 认证理想函数 F_{Cred}.

证明 设 $\mathcal{A}$ 是在 (F_{SIG}, F_{HCA}) 辅助的混合模型下与真实协议 π_{Cred} 交互的攻击者，我们可以构造一个理想过程的攻击者 $\mathcal{S}$，使得对于任何环境机 $\mathcal{Z}$ 而言，它与攻击者 $\mathcal{A}$ 和协议 π_{Cred} 以及攻击者 $\mathcal{S}$ 和理想函数 F_{Cred} 的交互都是不可区分的.

(1) 攻击者 S 的构造.

攻击者 $\mathcal{S}$ 在其内部对环境机 $\mathcal{Z}$、攻击者 $\mathcal{A}$ 以及参与者 p_i 进行仿真；对于真实环境下的攻击者 $\mathcal{A}$ 攻陷的每一个参与者 p_i，理想对手 $\mathcal{S}$ 攻陷对应的虚拟参与者 $\tilde{p}_i$，当被攻陷的虚拟参与者 $\tilde{p}_i$ 接收到环境机 $\mathcal{Z}$ 发来的消息 m 后，攻击者 $\mathcal{S}$ 让 $\mathcal{Z}'$ 将消息 m 发送给 p_i. 当被攻陷的参与者 p_i 向环境机 $\mathcal{Z}'$ 输出消息 m 后，$\mathcal{S}$ 指导被攻陷的虚拟参与者 $\tilde{p}_i$ 相应的向 $\mathcal{Z}$ 发送消息 m，就好像参与者 p_i 直接与环境机 $\mathcal{Z}$ 相连一样.

攻击者 $\mathcal{S}$ 的定义及操作如图 2 所示.

(2) 攻击者 $\mathcal{S}$ 执行的操作.

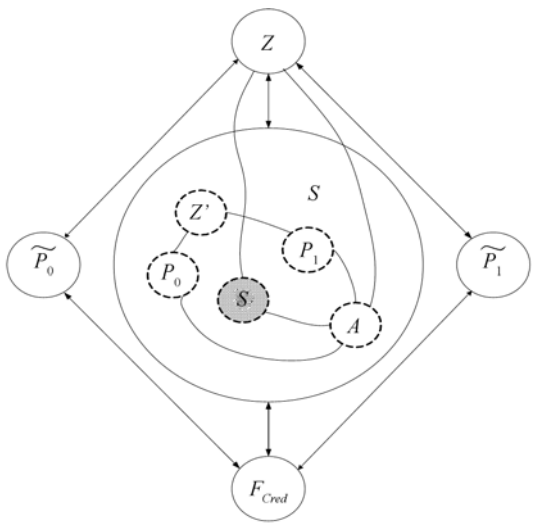

图 2 攻击者 S 的结构示意图

Simulating Present Credential

当 $\mathcal{S}$ 接收到理想函数 F_{Cred} 发来的消息(*Pres ent Credential*, p_i, c, z, p_j)后，执行如下操作:

1. 如果 p_i 还没有拥有身份凭证的话，对 $\mathcal{A}$ 仿真身份凭证的生成过程. 即，以 F_{HCA} 的名义将消息 $(Identity\ encryption, p_i)$ 发送给 $\mathcal{A}$，一旦获得 $\mathcal{A}$ 返回的消息，选择一个随机数 $u^i \xleftarrow{R} \{0,1\}^{k_1}$ 作为 P_i 的密钥，将 (P_i, u^i) 记录在成员列表中，然后计算秘密信息 $k_j \leftarrow (U^i(c \| j))_{j=1}^{k_2}$，$z \leftarrow H(k)$，其中 $k = (k_1, k_2, \cdots, k_{k_2})$，并以 F_{HCA} 的名义将消息(*Credential generation*, p_i, (c, p_i, k, z))发送给攻击者 $\mathcal{A}$.

2. 对 $\mathcal{A}$ 仿真提供身份凭证过程, 即, 选取 $m \xleftarrow{R} \{0,1\}^{k_2}$, 且保证"1"的个数为 $k_2/2$ 个, 且 m 从未产生过, 将秘密信息 k 中与"1"对应的位构成挑战消息 $\tilde{k} \leftarrow k_m$, 以 F_{HCA} 的名义发送消息 $(Add\ prepared\ credential, p_i, (c, p_j))$ 消息发送给攻击者 $\mathcal{A}$, 并将消息 $(Present\ Credential, p_i, c, \tilde{k})$ 发送给 F_{Cred}.

Simulating Verify Credential

如果接收到 F_{Cred} 发来的消息 $(Verify\ Credential, p_i, c, z, \tilde{k}, p_j, h', \sigma, v)$, 执行如下操作:

1. 以 F_{HCA} 的名义将消息 $(Check\ exist\ of\ credential, p_i, (c, p_i, k, h))$ 发送给 $\mathcal{A}$, 如果 F_{HCA} 返回的消息不是"*OK*", 则发送消息 $(Verify\ Credential, p_i, c, p_j, invalid)$ 给 F_{Cred} 并退出;

2. 否则, 检查路径, 如果 $h' \neq h$, 则输出 $(Verify\ Credential, p_i, c, p_j, invalid)$ 给 F_{Cred} 并退出.

3. 否则, 验证签名, 以 F_{SIG} 的名义发送 $(Verify\ p_i, root(h), \sigma, v)$ 给 $\mathcal{A}$, 一接收到 $\mathcal{A}$ 返回的消息 $(Verified\ p_i, root(h), \phi)$,

1) 如果有记录 $(root(h), \sigma, 1)$ 存在, 则令 $f = 1$,

2) 否则, 如果签名者未被攻陷, 且对于任何签名消息 σ' 而言, 都找不到相应的记录 $(root(h), \sigma', 1)$, 则令 $f = 0$, 并且记录 $(root(h), \sigma, 0)$.

3) 否则, 如果有记录 $(root(h), \sigma, f')$ 存在, 令 $f = f'$,

4) 否则, 令 $f = \phi$, 并且记录下 $(root(h), \sigma, \phi)$.

如果 $f = 0$, 则输出 $(P_i, Verify\ Credential, c, P_j, invalid)$ 给仿真器 $\mathcal{S}$ 并退出.

4. 否则, 验证凭证的有效性,

1) 如果 P_i 未被攻陷,

a) 以 F_{HCA} 的名义发送消息 $(Check\ prepared\ credential, p_i, (c, p_j))$ 给 $\mathcal{A}$;

b) 如果 F_{HCA} 返回的消息不是 "OK", 或者 $\tilde{k} \neq k_m$, 则将消息 $(Verify\ Credential, p_i, c, p_j, invalid)$ 发送给 F_{HCA} 并退出;

2) 或者 P_i 被攻陷且 $H(\tilde{k}) \neq z_m$, 则输出 $(Verify\ Credential, p_i, c, p_j, invalid)$ 给 F_{Cred} 并退出. 否则返回消息 $(Verify\ Credential, p_i, c, p_j, valid)$ 给 F_{Cred}.

Simulating party corruptions

如果 $\mathcal{A}$ 攻陷了一个实体 p_i, $\mathcal{S}$ 在理想过程中攻陷对应的实体 $\tilde{P}_i$, 并将 $\tilde{P}_i$ 的内部数据发送给 $\mathcal{A}$.

对于其他的操作, 如 *Check Reuse*, 因为他们的定义在理想函数和真实协议中是一致的, 不需要对 $\mathcal{A}$ 进行仿真.

(3) 证明 π_{Cred} 安全实现 F_{Cred}.

假设存在一个环境机 $\mathcal{Z}$, 对于任意理想对手而言都可以区分是与理想函数还是真实协议的交互, 则对于上面描述的攻击者 $\mathcal{S}$, 它自然也可以区分理想函数 F_{Cred} 和真实协议 π_{Cred}. 我

们构造 3 个过渡协议 $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$, 依次证明 $\pi_{Cred} \approx H_1 \approx H_2 \approx H_3 \approx F_{Cred}$, 最后得到 $F_{Cred} \approx \pi_{Cred}$ 的结论. 过度协议的定义如下:

$\mathcal{H}_1$ 与有 (F_{SIG}, F_{HCA}) 辅助的混合模型下的真实协议 π_{Cred} 的唯一区别在于, 当 π_{Cred} 指导发起者利用伪随机函数 PRF 生成对称密钥的时候, 实体选择一个无关的随机数 $k \xleftarrow{R} \{0,1\}^{k_1}$ 作为对应的密钥.

$\mathcal{H}_2$ 与 $\mathcal{H}_1$ 的唯一区别在于, 当 π_{Cred} 指导发起者利用伪随机函数 PRF 建立挑战消息的时候, 实体利用一个无关的随机函数生成对应的挑战.

$\mathcal{H}_3$ 与 $\mathcal{H}_2$ 的唯一区别在于, 在验证身份凭证的有效性时, 当 π_{Cred} 指导发起者检查路径 h 的有效性时, 实体通过在集合 C 中查找身份凭证验证其有效性.

(1) 假设 PRF 是安全的伪随机函数, 则有 $\pi_{Cred} \approx H_1$.

假设存在一个环境机 $\mathcal{Z}$ 和攻击者 $\mathcal{A}$, 使得 $\mathcal{Z}$ 可以以不可忽略的概率区分 π_{Cred} 与 $\mathcal{H}_1$, 则可以利用 $\mathcal{Z}$ 构造一个算法 $\mathcal{D}$, 使得 $\mathcal{D}$ 可以攻破伪随机函数 PRF 的安全. 即 $\mathcal{D}$ 可以访问一个 oracle f, 并以不可忽略的概率区分 f 是随机函数还是伪随机函数.

$\mathcal{D}$ 在一个仿真的交互过程中运行 $\mathcal{Z}$ 的一个拷贝, 使得它与有 (F_{SIG}, F_{HCA}) 辅助的混合模型下的协议 π_{Cred} 中的攻击者 $\mathcal{A}$ 和实体 $\mathcal{P}$ 进行交互. $\mathcal{D}$ 对 $\mathcal{Z}$ 模拟攻击者 $\mathcal{A}$ 和实体 $\mathcal{P}$. 唯一的例外在于当实体 $\mathcal{P}$ 生成密钥 k 的时候, $\mathcal{D}$ 让 f 生成 k. 如果 $\mathcal{Z}$ 在实体 $\mathcal{P}$ 产生输出前攻陷 $\mathcal{P}$, 则 $\mathcal{D}$ 退出并输出一个随机位. 否则, $\mathcal{D}$ 输出 $\mathcal{Z}$ 的输出.

如果 $f = \mathcal{U}$, 则仿真的 $\mathcal{Z}$ 的输出与它与 π_{Cred} 交互时的输出是一样的(假设 $\mathcal{D}$ 没有退出), 类似的, 如果 $f = \mathcal{R}$ 且 $\mathcal{D}$ 没有退出的话, $\mathcal{Z}$ 的输出与它与 $\mathcal{H}_1$ 交互的输出是一样的. 因此, 如果 $\mathcal{Z}$ 可以以不可忽略的概率 p 来区分 π_{Cred} 和 $\mathcal{H}_1$, 那么它可以以相同的概率区分伪随机函数和随机函数.

(2) 假设 PRF 是安全的伪随机函数, 则有 $H_1 \approx H_2$.

证明过程和 $\pi_{Cred} \approx H_1$ 的证明相似.

(3) 假设 Hash 函数 $\mathcal{H}$ 是无碰撞的, 则 $H_2 \approx H_3$.

假设存在一个环境机 $\mathcal{Z}$ 可以以不可忽略的概率区分 $\mathcal{H}_2$ 与 $\mathcal{H}_3$, 则可以利用 $\mathcal{Z}$ 构造一个算法 $\mathcal{D}$, 使得 $\mathcal{D}$ 可以找到无碰撞 Hash 函数 $\mathcal{H}$ 的一个碰撞.

我们首先指导 ASU 构造一个与 $\mathcal{Z}$ 交互的 Hash 树 T, 其根结点为 R. 假设 $\mathcal{Z}$ 能以不可忽略的概率产生一个身份凭证 $(c, \langle z_i \rangle)$, 使其能够通过路径有效性检测, 且保持根结点 R 不变, 则 $\mathcal{Z}$ 可以利用 T 构造一棵新的 $\mathcal{H}ash$ 树 T', 使得 $(c, \langle z_i \rangle)$ 作为新树 T' 的叶子结点, 通过比较两棵 Hash 树 T 和 T', 就可以找到 Hash 函数的一个碰撞. #

(4) 假设 Hash 函数 $\mathcal{H}$ 是单向无碰撞的, 则 $H_3 \approx F_{Cred}$.

$\mathcal{H}_3$ 与 F_{Cred} 的唯一区别在于, 在验证一个身份凭证的时候, F_{Cred} 采用在集合 T_{Prepared} 中查找对应值的方式, 而 $\mathcal{H}_3$ 则是检查是否有 $k_2/2$ 个 k_i 值能够保证 $H(k_i) = z_i$.

假设 $\mathcal{Z}$ 可以以不可忽略的概率 p 来区分 $\mathcal{H}_3$ 和 F_{Cred}, 则它可以提供某个 c, z 和 k_i' 能够成功

的通过验证, 并且满足:

- (c, P_j, m) 并不存在于集合 $T_{\mathrm{Pr}epared}$ 中, 或者

- $k_i' \neq k_i$, 但是 $H(k_i') = z_i$.

对于第一种情形, 由于集合 $T_{Prepared}$ 中没有 (c, P_j, m) 的记录, 因此还没有执行过相应的 *Present Credential* 操作, k_i 值还没有泄露. 如果 $\mathcal{Z}$ 可以以不可忽略的概率 p 来区分 $\mathcal{H}_3$ 和 F_{Cred}, 则可以利用 $\mathcal{Z}$ 构造一个算法 $\mathcal{D}$, 使得 $\mathcal{D}$ 可以计算出给定值 y 的 Hash 原像 $x \in H^{-1}(y)$, 从而违背了 $\mathcal{H}$ 的单向性假设.

对于第二种情形, 显然, 我们可以构造一个算法 $\mathcal{A}$, 使得 $\mathcal{A}$ 能够找到 Hash 函数的一个碰撞, 从而违背了 Hash 函数 $\mathcal{H}$ 的无碰撞性假设.

通过以上证明, 我们得到最终的结论, 即 $F_{Cred} \approx \pi_{Cred}^{Sig}$.

6　结论

通用可组合的安全框架可用于描述和分析并发环境下的协议安全问题, 相对其他安全模型而言, 通用可组合安全具有更严格、更高级别的安全定义. 目前对 UC 安全的理论已经做了大量研究, 尽管已经提出了多个实现某种安全任务的理想函数, 但是还没有提出与匿名认证相关的理想函数.

UC 框架中定义的认证理想函数 F_{CRET} 主要反映了利用公钥证书进行身份认证的方式, 但是没有反映出采用其他形式进行匿名认证的特殊需求, 它依靠证书权威理想函数 F_{CA} 的辅助来实现. 考虑到无线环境下匿名认证的特殊需求, 我们提出了匿名 Hash 认证理想函数模型 F_{Cred} 和匿名 Hash 证书权威理想函数模型, 其作用相当于证书理想函数 F_{CERT} 和 CA 理想函数 F_{CA} 的功能, 可以用来实现认证理想函数 F_{AUTH}, 其中对身份的认证采用的是将身份与特定 Hash 值绑定的方式来实现的. UC 模型最重要的性质在于模块化的设计思想, 因而我们提出的匿名 Hash 认证理想函数, 即可以作为安全模块辅助实现其他多种理想函数, 如采用 DH 交换来实现密钥交换理想函数 F_{KE}, 也可以用于设计具有匿名性要求的安全协议.

针对本文提出的匿名 Hash 认证理想函数, 我们设计了一个真实协议, 并证明所提匿名 Hash 认证机制的安全属性可以通过安全对称加密机制、安全数据签名机制、伪随机函数以及单向无碰撞 Hash 函数的组合得到保证. 在本文提出的匿名 Hash 认证机制中, 只有认证服务器知道移动终端的身份, 认证器则无法识别出移动终端的真实身份. 出于安全及效率方面的考虑, 认证过程直接在移动终端和认证器之间进行, 而不再涉及认证服务器. 考虑到无线环境的特殊限制, 以及移动终端设备的计算能力有限, 本文提出的匿名 Hash 认证机制采用对称加密原语实现, 因而具有较高的效率和安全性, 适合无线环境下使用.

我们下一步的工作在于研究匿名 Hash 认证理想函数如何作为基本安全组件进行协议设计, 另外进一步提升匿名 Hash 证书权威理想函数的抽象层次, 使其具有更普遍的意义.

致谢　作者衷心感谢 Marten 先生所做的开创性工作.

参 考 文 献

1 Martin A, Phillip R. Reconciling two views of cryptography. J Crypt, 2002, 15(2): 103—127

2 Wenbo M. Modern Cryptography: Theory and Practice. NJ: Prentice-Hall PTR. 2004

3 Bellare M, Rogaway P. Random oracles are practical: a paradigm for designing efficient protocols. In: First ACM conference on Computer and Communications Security. New York: ACM Press, 1993. 62—73

4 Beaver D. Foundations of secure interactive computing. In: Joan Feigenbaum, Advances in Cryptology-Crypto'91, 1991, 576. Berlin: Springer-Verlag Press, 1991. 377—391

5 Andrew C, Qizhi Y. Protocols for secure computations (extended abstract). In: 23rd Annual Symposium on Foundations of Computer Science. Chicago: Illinois, November 1982. 160—164

6 Ran C, Hugo K. Analysis of key exchange protocols and their use for building secure channels. In: Pfitzmann B, ed. Advances in Cryptology-EUROCRYPT 2001. LNCS 2045. Berlin: Springer-Verlag, 2001. 453—474

7 Ran C, Hugo K. Security Analysis of IKE's Signature-based Key Exchange Protocol. 2002. Advances in Cryptology-Crypto 2002. LNCS2442, Heidelberg: Springer Verlag, 2002. 143—161

8 Ran C. Universally composable security: A new paradigm for cryptographic protocols. In: 42th IEEE Annual Symposium on Foundations of Computer Science, 2001. Oakland: IEEE, 136—145

9 Birgit P, Michael W. A model for asynchronous reactive systems and its application to secure message transmission. IEEE Symposium on Security and Privacy, May 2001. Oakland: IEEE, 184—200

10 Yehuda L. Composition of secure multi-party protocols —— A comprehensive study. In: Lecture Notes in Computer Science. New York: Springer-Verlag, 2003. 2815

11 Yehuda L. General composition and universal composability in Secure multi-party computation. In: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science. Cambridge. 2003. 394—403

12 Ran C. Universally composable two-party and muti-party secure computation. In 34th STOC. New York: ACM Press, 2002. 494—503

13 季庆光. 对几类重要网络安全协议形式模型的分析. 计算机学报, 2005, 7: 128—141

14 Ran C, Marc F. Universally composable commitments. In: Lecture Notes in Computer Science, 2001, London: Springer-Verlag, 2139: 19—28

15 Ran C, Hugo K. Universally composable notions of key exchange and secure channels. Theory and application of cryptographic techniques. In: Lecture Notes in Computer Science. New York: Springer, 2002. 337—351

16 Ran C, Shai H, Jonathan K, et al. Universally Composable Password-Based Key Exchange. Eurocrypt, 2005. Denmark: Springer-Verlag, 404—421

17 Ivan D, Jesper B N. Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. In: CRYPTO. LNCS 2442, Santa Barbara, 2002. 581—596

18 Mateus P, Mitchell J, Scedrov C. A composition of cryptographic protocols in a probabilistic polynomial-time process calculus. In 14th International Conference on Concurrency Theory. LNCS 2761. New York: Springer-Verlag, 2003. 327—349

19 Ran C, Hugo K. Universally composable symbolic analysis of cryptographic protocols (The case of encryption-based mutual authentication and key-exchange). DIMACS Workshop on Protocols Security Analysis, 2004

20 Manoj P, Amit S. New notions of security: Achieving universal composability without trusted setup. In STOC'04: Proceedings of the 36th Annual ACM Symposium on Theory of Computing. New York: ACM Press, 2004. 242—251

21 Oded G. Foundations of Cryptography (Fragments of a book). Weizmann Inst of Science, 1995

22 Shafi G, Silvio M, Ron L R. A digital signature scheme secure against adaptive chosen-message attacks. J Comput, 1988, 17(2): 281—308

23 Ralph C M. Protocols for public key cryptosystems. In: IEEE Symposium on Security and Privacy. IEEE Computer Society Press, 1980. 122—133

24 Ran C. Universally composable signature, certification, and authentication. 17th IEEE computer security foundations workshop (CSFW). New York: IEEE Computer Society Press, 2004. 219—245

25 Marten T. A universally composable scheme for electronic cash. Indocrypt, 2005. 347—360

26 Michael B, Dennis H. How to break and repair a universally composable signature functionality. Information Security Conference-ISC. LNCS 3225. Berlin: Springer-Verlag, 2004. 61—74