可信计算环境下基于主机身份的一次性密钥交换协议

张 淼　徐国爱　胡正名　杨义先
(北京邮电大学网络与交换技术国家重点实验室　北京　100876)

摘　要：该文介绍了可信计算环境下可信网络连接的基本概念，分析了 TNC 协议扩展存在的问题，介绍了直接匿名证明 DAA 协议。提出了一种新的，基于主机身份的一次性密钥交换协议 I-OKEP，并分析了其安全性。经安全性分析证明，该协议可以在可信计算环境下保证密钥交换的机密性与可靠性，同时还可以保证主机完整性与主机匿名性。

关键词：密钥交换；可信计算；可信网络连接技术；DAA；AIK 证书
中图分类号：TP309　　**文献标识码**：A　　**文章编号**：1009-5896(2007)06-1348-04

A Host Identity Based One-Time Key Exchange Protocol in Trusted Computing

Zhang Miao　Xu Guo-ai　Hu Zheng-ming　Yang Yi-xian
(*State Key Lab. of Networking and Switching Tech.*, *Beijing Univ. of Posts and Telecomms*, *Beijing* 100876, *China*)

Abstract: The conception of Trusted Network Connection (TNC) is introduced, and the problem of TNC protocol extend is analyzed. Direct Anonymous Attestation(DAA) protocol is descripted. After this, the paper takes out a new host Indentity based One-time Key Exchange Protocol(I-OKEP), and analyzes its security character. The security analysis can prove that the protocol can ensure the confidentiality and reliability of key-exchange, and the integrity and anonymous of host in trusted computing environments.

Key words: Key-exchange; Trusted computing; Trusted network connection; Direct Anonymous Attestation (DAA); AIK-certificate

1　引言

随着网络技术的不断发展，可信计算正成为目前安全研究的热点，其从硬件入手，在终端上通过安全芯片等手段，从根本上解决了目前存在的病毒、木马等主机安全问题。以TCG为例，目前TCG组织成立了架构、移动、PC客户端、服务器、软件协议栈、存储、可信网络(TNC)、TPM等几个重要的工作组[1]。制定工作组自身相关领域可信计算的标准、规范、用例等内容。目前，安全芯片TPM以及对应的TSS软件协议栈的发展比较迅速，国内外已经出现了很多相关产品；但是对于可信计算的应用、部署、网络等方面，还在进一步研究与探讨当中。

可信网络连接技术TNC(Trusted Network Connection)[2]是建立在基于主机的可信计算技术之上的，其主要目的在于通过使用可信主机提供的终端基础，实现网络访问控制的协同工作。又因为完整性校验被终端作为安全状态的证明，所以TNC的权限控制策略可以估算目标网络的终端适应度。TNC网络构架会结合已存在的网络访问控制策略(例如802.1x协议)来实现访问控制功能。而通过扩充目前的网络协议(如IKE，802.1x，Radius，TLS等)，在实现其原有功能的基础上，对终端进行完整性校验。

在目前的可信计算架构中，没有专门为TNC设计的密钥交换协议，目前的TLS-Attestation，IKE-Attestation[3,4]都是对原有协议的扩充，或者说只是在原有协议中增加了一些交换属性，并增加了完整性校验流程。这种方式优点是可以兼容原有的协议，升级相对容易，但是增大了密钥交换的复杂度，并对原有协议自身带来一定的安全威胁。

本文提出了一种在可信网络中基于身份证书AIK的密钥交换协议I-OKEP(Identity based One-time Key Exchange Protocol)，主要目的是在可信网络中实现一种安全的，快速的密钥交换协议。其主要参考了目前几种常见的密钥交换算法[5–8]，支持通过直接匿名证明DAA协议对主机身份进行校验，保证主机的可信、可靠。

2　预备知识

2.1 DAA 协议

DAA(Direct Anonymous Attestations)协议由TCG组织制定[1,9]，主要目的是用于支持匿名身份认证技术。其由两个子协议组成。第 1 个叫DAA-Join协议，TPM/主机需要从DAA-Issuer获取DAA证书。这个证书可以用在第 2 个子协议DAA-Sign协议中用来与Verifer交互。DAA协议的好处是为可信平台提供最高级别的匿名性，当这个可信平台与外部进

2005-11-28 收到，2006-05-18 改回
国家自然科学基金(60372094)资助课题

行通信的时候，其身份是匿名的[1]，并且还可以证明这个平台依然是由TCG定义的可信平台。

DAA 协议包含 4 个实体：Issuer，TPM，Host，Verifier，以及两个协议，分别是 Join 协议和 Sign 协议。主机 Host 被定义为一个包含 TPM 的平台。执行 DAA 协议要求计算被分为 Host 以及 TPM 两部分。任何不需要安全的计算都被 TPM 交给 Host 完成，以提高协议的效率。这里，假设 Host 比 TPM 处理能力更强。其流程如下：

(1) Issuer 生成一个公钥 IKEY 以及对应的私钥。

(2) 首先 TPM 与 Issuer 进行 DAA Join 协议，目的是产生 TPM 使用的 IKEY-Certificate。在 Join 协议中，TPM 生成一个私有的 DAA 密钥 privDK。TPM 通过自身的 Endorsement 密钥来对 Issuer 验证自身。如果 Join 协议完成，Issuer 会提供一个 certDK 的证书给 TPM 芯片。这样主机就可以通过 IKEY-Certificate 以及 certDK 以及 privDK 对消息进行签发了。

(3) DAA Sign 协议是一个在 Verifier 与 TPM 之间的协议，在这个协议中 Verifier 获取了有效的 TPM 颁发的身份证明密钥 AIK。在 Sign 协议中，TPM 会生成一个身份证明密钥 AIK。TPM 会将 AIK-pubic-key 发送到 Verifier。TPM 还会使用 privDK 以 certDK 来生成一个基于 AIK-public-key 的签名，并把签名提供给 Verifier。Verifier 根据 IKEY Certificate 来校验这个签名是否是有效的。如果是有效的，Verifier 认为对应的 AIK-private-key 是由 TPM 所持有。

2.2 模型与假设

在I-OKEP协议中，定义Initiator为初始化密钥交换的一个实体；一个Responder作为相应Initiator的密钥交换请求。这里我们认为Initiator以及Responder都是用户。此外，还存在一个实体，其一直在尝试攻击密钥交换协议的弱点，我们称其为adversary。这里adversary不是用户，而是一个可以监控全部网络通信的设备。在密钥协商开始前，双方均拥有自己的主机身份证书AIK，分别为AIK-C_I和AIK-C_R及对应的公私钥对AIK-Pub_I, AIK-Pub_R，AIK-Priv_I, AIK-Priv_R。以及Privacy-Ca颁发的身份证书IKEY-C_I, IKEY-C_R。用户证书CSK-C_I，CSK-C_R。除此以外，还定义单向散列函数，素数p，生成器g，等等(见表 1)。

3 I-OKEP 协议

3.1 密钥交换流程

(1) Initiator 生成随机会话ID SID_I, 并通过AIK-Priv_I对 CSK-C_I 签名，而使用CSK-C_I对Proof_I，进行签名，结果为$(\text{Proof}_\text{I})_{\text{CSK-CI}}$。此外，还对利用用户私钥会话$\text{SID}_\text{I}$进行签发，主要目的是保护$\text{SID}_\text{I}$不被恶意篡改。Initiator 将$\text{SID}_\text{I}$，$\text{AIK-C}_\text{I}$，$(\text{CSK-C}_\text{I})_{\text{ALK-CI}}$，$\text{CSK-C}_\text{I}$，$\text{Proof}_\text{I}$，$(\text{Proof}_\text{I})_{\text{CSK-CI}}$，$(\text{SID}_\text{I})_{\text{CSK-CI}}$，$\text{IKEY-C}_\text{I}$发送到Responder。

Initiator → Responder

表 1　I-OKEP 协议参数表

AIK-C_I，AIK-C_R	Initiator 以及 responder 的 AIK 证书
AIK-Pub_I, AIK-Pub_R	Initiator 以及 responder 的 AIK 公钥
AIK-Priv_I, AIK-Priv_R	Initiator 以及 responder 的 AIK 私钥
IKEY-C_I , IKEY-C_R	DAA 证书
PCR_I, PCR_R	Initiator 以及 responder 的完整性状态
CSK-C_I，CSK-C_R	Initiator 以及 responder 的用户证书
CSK-Pub_I, CSK-Pub_R	Initiator 以及 responder 的用户公钥
CSK-Priv_I, CSK-Priv_R	Initiator 以及 responder 的用户私钥
prf	伪随机数发生器函数
h, h1, h2	单向散列函数
SID_I SID_R	由 initiator 以及 responder 生成的随机数，用以会话 ID
g	生成器
Seq_I Seq_R	由 initiator 以及 responder 生成的随机数，用以序列号
$\text{E}_\text{X}(\text{Y})$	使用 X 加密 Y
M, M′	消息
i	当前会话次数
j,k	当前通信的数据包数

SID_I，AIK-C_I，$(\text{CSK-C}_\text{I})_{\text{ALK-CI}}$，$\text{CSK-C}_\text{I}$，$\text{Proof}_\text{I}$，$(\text{Proof}_\text{I})_{\text{CSK-CI}}$，$(\text{SID}_\text{I})_{\text{CSK-CI}}$，$\text{IKEY-C}_\text{I}$

(2) Responder接收到消息后，首先通过DAA协议对AIK-C_I进行验证[9]，并通过AIK-C_I对用户证书CSK-C_I进行校验，而使用CSK-C_I对签名$(\text{Proof}_\text{I})_{\text{CSK-CI}}$进行校验。验证通过后，还需要使用$\text{CSK-C}_\text{I}$校验$(\text{SID}_\text{I})_{\text{CSK-CI}}$，利用载荷中的用户证书$\text{CSK-C}_\text{I}$解密$(\text{SID}_\text{I})_{\text{CSK-CI}}$获得$\text{SID}_\text{I}'$，如果$\text{SID}_\text{I}=\text{SID}_\text{I}'$，则认为会话ID未被篡改。并判断该用户是否已在会话建立中，以及会话ID SID_I已被使用，如果SID_I被篡改，重复或者该用户会话已经建立，则不再接受此请求。此后Responder生成会话ID SID_R，以及随机序列号Seq_R，计算$\text{KID}[i]=\text{h}(g^{\text{CSK-PrivI}\circ\text{CSK-PrivR}},\ i)$，计算$\text{H}_\text{KIDR}=\text{HMAC}_\text{KID}(\text{SID}_\text{R}\mid\text{SID}_\text{I}\mid\text{Proof}_\text{R}\mid\text{AIK-C}_\text{R}\mid\text{CSK-C}_\text{R}\mid\text{KID}[i]\mid\text{ID}_\text{R}\mid\text{Seq}_\text{R})$。并将$\text{SID}_\text{I}$，$\text{SID}_\text{R}$，$\text{Proof}_\text{R}$，$(\text{Proof}_\text{R})_{\text{CSK-CR}}$，$(\text{CSK-C}_\text{R})_{\text{ALK-CR}}$，$\text{CSK-C}_\text{R}$，$\text{AIK-C}_\text{R}$，$\text{H}_\text{KIDR}$，$\text{Seq}_\text{R}$，$\text{IKEY-C}_\text{R}$，$\{\text{ID}_\text{R}\}\text{CSK-Pub}_\text{I}$发送给Initiator。

Responder → Initiator

SID_I，SID_R，Proof_R，$(\text{Proof}_\text{R})_{\text{CSK-CR}}$，$\text{H}_\text{KIDR}$，$\text{Seq}_\text{R}$，$(\text{CSK-C}_\text{R})_{\text{ALK-CR}}$，$\text{CSK-C}_\text{R}$，$\text{AIK-C}_\text{R}$，$\text{IKEY-C}_\text{R}$，$\{\text{ID}_\text{R}\}\text{CSK-Pub}_\text{I}$

(3) Initiator收到消息后，校验SID_I是否正确，通过DAA协议对AIK-C_R进行验证，并通过AIK-C_R对用户证书CSK-C_R进行校验，而使用CSK-C_R对签名$(\text{Proof}_\text{R})_{\text{CSK-CR}}$进行校验。验证通过后，Initiator 计算 $\text{KID}[i]$

$=\mathrm{h}(g^{\text{CSK-PrivI}\circ\text{CSK-PrivR}},\ i)$，并计算$\mathrm{HMAC_{KID}}(\mathrm{SID_R} \mid \mathrm{Proof_R} \mid \mathrm{AIK\text{-}C_R} \mid \mathrm{CSK\text{-}C_R} \mid \mathrm{KID}[i] \mid \mathrm{Seq_R})$。是否与收到的$\mathrm{H_{KIDR}}$匹配。如果匹配，则读取主机配置寄存器中的完整性信息$\mathrm{PCR_I}$，并生成随机序列号$\mathrm{Seq_I}$。将$\mathrm{SID_I}$，$\mathrm{SID_R}$，$\mathrm{H_{KIDI}} = \mathrm{HMAC_{KID}}(\mathrm{SID_I} \mid \mathrm{SID_R} \mid \mathrm{Proof_I} \mid \mathrm{AIK\text{-}C_I} \mid \mathrm{CSK\text{-}C_I} \mid \mathrm{KID}[i] \mid \mathrm{Seq_I})$, $\mathrm{Seq_I}$, $\{\mathrm{ID_I}\}\mathrm{CSK\text{-}Pub_R}$, $\mathrm{PCR_I}$发送到Responder。

Initiator → Responder

$\mathrm{SID_I}$，$\mathrm{SID_R}$, $\mathrm{H_{KIDI}}$, $\mathrm{Seq_I}$, $\{\mathrm{ID_I}\}\mathrm{CSK\text{-}Pub_R}$, $\mathrm{PCR_I}$

(4) Responder收到消息后，检测$\mathrm{SID_R}$及$\mathrm{SID_I}$是否正确，解密$\{\mathrm{ID_I}\}\mathrm{ALK\text{-}Pub_R}$来获取身份信息$\mathrm{ID_I}$，并计算$\mathrm{H_{KIDI}} = \mathrm{HMAC_{KID}}(\mathrm{SID_I} \mid \mathrm{AIK\text{-}C_I} \mid \mathrm{CSK\text{-}C_R} \mid \mathrm{KID}[i] \mid \mathrm{ID_I} \mid \mathrm{Seq_I})$来进行校验。并根据接收到的完整性信息$\mathrm{PCR_I}$来进行对端主机完整性校验。校验通过后，密钥交换完成。为了完成主机完整性校验，Responder还需要读取主机配置寄存器中的完整性信息$\mathrm{PCR_R}$并将寄存器值发送到Initiator。

Responder → Initiator

$\mathrm{SID_I}$, $\mathrm{SID_R}$ $\mathrm{PCR_R}$, $\mathrm{Seq_R}$, $\mathrm{HMAC_{KID}}(\mathrm{SID_I} \mid \mathrm{SID_R} \mid \mathrm{PCR_R} \mid \mathrm{Seq_R})$

(5) Initiator接收到消息后，校验$\mathrm{PCR_R}$。校验通过，则密钥建立完成。

最终的密钥种子是 $\mathrm{Seed}[i] = \mathrm{prf}(\mathrm{KID}[i], \mathrm{ID_I}, \mathrm{ID_R}, \mathrm{H_{KIDI}} \mid \mathrm{H_{KIDR}})$。

3.2 通信序列

密钥交换完成后，可以开始进行通信。为了保证通信的安全，每一次通信时密钥计算如下：

$\mathrm{Key_I}[j] = \mathrm{h1}\ (\mathrm{SS_I}, \mathrm{Seq_I} + j)$

$\mathrm{Key_R}[\mathrm{k}] = \mathrm{h1}\ (\mathrm{SS_R}, \mathrm{Seq_R} + k)$

$\mathrm{SS_I} = \mathrm{h2}\ (\mathrm{Seed}[i], \mathrm{H_{KIDI}})$

$\mathrm{SS_R} = \mathrm{h2}\ (\mathrm{Seed}[i], \mathrm{H_{KIDR}})$

通信流程如图 1。

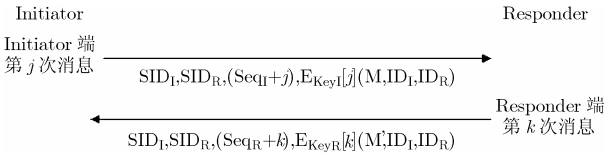

图 1 I-OKEP 通信流程图

每次通信中，序列号$\mathrm{Seq_I}$，$\mathrm{Seq_R}$每次都自增。每发送一个消息，会话ID (即$\mathrm{SID_I}$, $\mathrm{SID_R}$)不变，而序列号自增 1，并且$\mathrm{Seq_I}$, $\mathrm{Seq_R}$的自增是由发出端控制的，例如Initiator端发出了 5 个消息，则$\mathrm{Seq_I} = \mathrm{Seq_I} + 5$； 而Responder发送了 3 个消息，则$\mathrm{Seq_R} = \mathrm{Seq_R} + 3$。两者没有直接的关联，而每一次发送加密数据的密钥是根据Seq以及密钥交换过程中产生的密钥种子Seed共同产生的。

$$\mathrm{Key_I}\ [j] = \mathrm{h}(\mathrm{SS_I}, \mathrm{Seq_I} + j)$$

$$\mathrm{Key_R}\ [k] = \mathrm{h}(\mathrm{SS_R}, \mathrm{Seq_R} + k)$$

4 安全性分析

4.1 双向认证、加密、完整性

在密钥交换过程中，Responder会检查第 1 个数据包中包含了Initiator的身份证书AIK是否正确。同时协议还对用户证书CSK进行了签名$(\mathrm{CSK\text{-}C})_{\mathrm{ALK\text{-}C}}$，这样用户证书和身份证书捆绑在一起，保证了用户证书与平台的相关。同时，Responder还返回给Initiator一个自己的身份证书和用户证书，Initiator也可以对Responder进行校验，保证了双方的身份，实现了双向认证功能。

在I-OKEP协议中，双方完成身份认证后，是以相互交换的身份ID以及用户证书作为密钥生成材料的。由于身份ID是密文传送的，而$\mathrm{KID} = \mathrm{g}^{\text{CSK-PrivI}\circ\text{CSK-PrivR}}$是通过对方公钥与自身私钥共同作用得到的值，adversary不能得到密钥生成材料中的内容。并且，最终生成的密钥是根据种子Seed与序列号共同生成的，使密钥具有一次性特征。每个通信序列密钥均不相同，并且由于序列号是随密文发送的，既能控制接收顺序，也能保证密钥不会缺失。

在密钥交换的第 2 个包和第 3 个包，Respnder以及Initiator都对发送过去的数据进行了HMAC校验，$\mathrm{HMAC_{KID}}(\mathrm{SID_I} \mid \mathrm{AIK\text{-}C_I} \mid \mathrm{CSK\text{-}C_R} \mid \mathrm{KID} \mid \mathrm{ID_I} \mid \mathrm{Seq_I})$以及$\mathrm{H_{KIDR}} = \mathrm{HMAC_{KID}}(\mathrm{SID_R} \mid \mathrm{Proof_R} \mid \mathrm{AIK\text{-}C_R} \mid \mathrm{CSK\text{-}C_R} \mid \mathrm{KID} \mid \mathrm{ID_R} \mid \mathrm{Seq_R})$，保证了会话ID以及序列号Seq不被adversary窜改。

据上可知，I-OKEP 协议实现保证数据的完整性，双向认证以及密文传输。

4.2 重放攻击，DOS 攻击

在防范DOS攻击方面，在协议中Responder处理接收到的第一个消息时，除了对证书进行校验外，还对Initiator的会话ID $\mathrm{SID_I}$进行检测，使用$\mathrm{CSK\text{-}C_I}$校验$(\mathrm{SID_I})_{\text{CSK-CI}}$，保证会话ID未被篡改，并判断该用户是否已在会话建立中，以及会话ID $\mathrm{SID_I}$已被使用，如果$\mathrm{SID_I}$被篡改，重复或者该用户会话已经建立，则不再接受此请求。此外，在Responder回给Initiator的包中，计算了$\mathrm{H_{KIDR}} = \mathrm{HMAC_{KID}}(\mathrm{SID_R} \mid \mathrm{SID_I} \mid \mathrm{Proof_R} \mid \mathrm{AIK\text{-}C_R} \mid \mathrm{CSK\text{-}C_R} \mid \mathrm{KID}[i] \mid \mathrm{ID_R} \mid \mathrm{Seq_R})$。方便initiator再次确认$\mathrm{SID_I}$，防止消息一的数据包通过被伪造的方式造成泄密。

由于会话中使用了序列号 Seq，每一次发送数据时，序列号均需要增加，每当接收到新的数据包后，系统需要自动更新当前序列号。接收到的数据包中旧的序列号将不再被允许接受。接收方只能接收当前序列号 n 个(n<5)以内的数据包，大于 n 的数据包也不能被接收。所以可以防范重放攻击。

4.3 完美前向保护

假设安全数据中的一项$(\mathrm{Seed}[i], \mathrm{SS_I}, \mathrm{SS_R}, \mathrm{Key_I}\ [j]$，$\mathrm{Key_R}\ [k]$，$\mathrm{KID}[i])$被泄漏，adversary可以进行以下行为：

如果 Seed 泄漏，adversary 可以在此次会话中获取通信

中的全部数据，但是下一次会话不再能够获取。

如果SS_I泄漏, adversary可以获取此次会话中Initiator方发出的所有通信序列，但是Responder方的数据不能被获取，下一次会话数据不能被获取。

如果SS_R泄漏，adversary可以获取此次会话中Responder方发出的所有通信序列，但是Initiator方的数据不能被获取，下一次会话数据不能被获取。

如果$Key_I[j]$或$Key_R[k]$被泄漏，则adversary只能获取本次通信数据包信息。

如果 KID[i]被泄漏，则 adversary 可以伪造会话中全部数据，直到下一次会话。

如上所述，如果adversary获取了$Key_I[j]$或$Key_R[k]$，则不会对系统造成大的影响，只会泄漏一个数据包的数据，不能够猜测其他信息；而泄漏SS_I, SS_R会使adversary获取到本次会话一半的数据，但是不会影响下一次会话；如果Seed以及KID[i]泄漏，则本次会话的全部数据均会被泄漏。

当然，如果adverary窃取到了通信双方的用户证书私钥$CSK\text{-}Priv_I$, $CSK\text{-}Priv_R$以及身份信息ID_I, ID_R则其可以猜测出第i次会话中的会话密钥。

4.4 主机完整性与主机匿名性

在协议中，主机身份是基于DAA协议通过AIK证书保护的，身份信息是存放于Privacy CA中的。所以对于主机的身份认证过程是基于DAA协议的匿名认证[1,9]，即通信双方不知道对方主机的信息。同时，为了对用户身份进行认证，协议使用了用户身份证书CSK对用户身份进行校验。所以本协议对用户的身份不具有匿名性。

而 I-OKEP 协议的第 3 个消息和第 4 个消息分别对密钥交换双方的主机 PCR 寄存器进行了校验，保护了主机完整性。

5　结束语

本文指出了可信计算 TNC 协议扩展存在的问题，介绍了直接匿名证明 DAA 协议。提出了一种新的，基于主机身份的一次性密钥交换协议 I-OKEP，并分析了其安全性。希望对可信计算尤其是可信网络的发展起到一定的推动作用。

参 考 文 献

[1] Hardjono T. TCG infrestructure working group reference architecture for interperability (Part I) specification version 1.0. Trusted Computing Group. http://www.trustedcomputinggroup.org/

[2] Hardjono T. TCG trusted network connect, TNC architecture for interoperability specification version 1.0. Trusted Computing Group. http://www.trustedcomputinggroup.org/

[3] Harkins D and Carrel D. The internet key exchange(IKE). RFC2409, 1998. http://www.retf.org/

[4] Blake-Wilson S, Nystrom M, Hopwood D, Mikkelsen J and Wright T. Transport layer security (TLS) extensions. RFC3546, June 2003. http://www.retf.org/

[5] Imamoto K and Sakuraia K. Design of diffie-hellman based key exchange using one-time ID in pre-shared key model. Proceedings of the 18th International Conference on Advanced Information Networking and Application (AINA'04). Fukuoka, Japan, AINA (1) 2004: 327–333.

[6] Lin Chun-Li, Sun Hung-Min, Steiner M, and Hwang T. Three-party encrypted key exchange without server public-keys. *IEEE Communications Letters*, 2001, 5(12): 497–499.

[7] Phan R and C-W. Fixing the integrated diffie-hellman-DSA key exchange protocol. *IEEE Communications Letters*, 2005, 9(6): 570–572.

[8] Krawczyk H. SKEME: A versatile secure key exchange mechanism for internet. IEEE Proceedings of SNDSS '96, San Diego, 1996, NDSS(1)'96: 114–127.

[9] Brickell E. Camenisch J. and Chen Liqun. Direct anonymous attestation. Trusted Computing Group. http://www.trustedcomputinggroup.org/. February 11, 2004.

张　淼：　男，1980 年生，博士生，研究方向为可信计算、网络安全.

徐国爱：　男，1972 年生，副教授，研究方向为密码学与网络安全.

胡正名：　男，1931 年生，博士生导师，研究方向为编码密码学.

杨义先：　男，1961 年生，博士生导师，研究方向为编码密码学与信息安全.