# 可信计算理论与实践
# ——TCTP’2009

## 第一届中国可信计算理论与实践学术会议论文集

冯登国 编

清华大学出版社

# 可信计算理论与实践——TCTP'2009
# 第一届中国可信计算理论与实践学术会议
# 论文集

冯登国　编

清 华 大 学 出 版 社
北　京

## 内 容 简 介

本书为第一届中国可信计算理论与实践学术会议论文集，收录论文19篇，内容涉及可信计算的方方面面。主要内容包括：可信计算密码理论和信任理论、可信计算体系结构、可信计算平台和可信系统、可信计算软件、可信网络及可信计算实践与应用技术等。

本书可供从事信息安全、密码学、计算机、软件、微电子、通信等专业的科技工作者和高等院校相关专业的师生参考。



**图书在版编目（CIP）数据**

可信计算理论与实践——TCTP'2009：第一届中国可信计算理论与实践学术会议论文集/冯登国编. —北京：清华大学出版社，2009.10
ISBN 978-7-302-20875-4

Ⅰ. 可…　Ⅱ. 冯…　Ⅲ. 电子计算机－安全技术－学术会议－文集　Ⅳ. TP309-53

中国版本图书馆CIP数据核字(2009)第176732号

**责任编辑**：张　民
**责任校对**：李建庄
**责任印制**：李红英

**出版发行**：清华大学出版社　　**地　　址**：北京清华大学学研大厦A座
http://www.tup.com.cn　　**邮　　编**：100084
**社　总　机**：010-62770175　　**邮　购**：010-62786544
**投稿与读者服务**：010-62776969，c-service@tup.tsinghua.edu.cn
**质　量　反　馈**：010-62772015，zhiliang@tup.tsinghua.edu.cn
**印 装 者**：北京嘉实印刷有限公司
**经　　销**：全国新华书店
**开　　本**：185×260　　**印　张**：11.75　　**字　数**：280千字
**版　　次**：2009年10月第1版　　**印　次**：2009年10月第1次印刷
**印　　数**：1～1000
**定　　价**：30.00元

# 第一届中国可信计算理论与实践学术会议
# 程序委员会

**主席：**

冯登国　　中国科学院软件研究所

**委员**（以姓氏笔画为序）：

马建峰　　西安电子科技大学
王小云　　清华大学
石文昌　　中国人民大学
何大可　　西南交通大学
何良生　　中国人民解放军密码管理局
吴秋新　　联想研究院
张　兴　　北京工业大学
张焕国　　武汉大学
杨义先　　北京邮电大学
陈克非　　上海交通大学
陈性元　　中国人民解放军信息工程大学
俞能海　　中国科学技术大学
徐茂智　　北京大学
韩　臻　　北京交通大学

# 第一届中国可信计算理论与实践学术会议
# 组织委员会

**主席：**

强志军　　中国密码学会

**委员**（以姓氏笔画为序）：

刘　娟　　中国密码学会

刘　韧　　联想研究院

毕　宁　　电子工业出版社

李　立　　中国密码学会

张　民　　清华大学出版社

邢运超　　中国密码学会

赵立生　　国民技术股份有限公司

秦　宇　　中国科学院软件研究所

# 前　言

随着计算机应用尤其是网络应用的普及，计算机病毒、恶意代码和黑客攻击事件层出不穷，通用计算机终端的安全问题越来越突出。因此人们逐渐意识到必须从终端计算机的源头出发，综合采用安全芯片、硬件结构和操作系统等多种安全措施构建可信赖的计算环境，这就是可信计算的基本思想。与传统的安全问题解决方法不同，可信计算从计算机体系结构着手，针对信息系统的安全需求和各种攻击手段，提出一种全新的体系结构级别的系统安全解决方案。作为平台信任根的可信计算安全芯片提供了机密性、完整性、封装存储等一系列重要安全属性，这已经引起了产业界和学术界的极大兴趣，他们以极大的热情投身于可信计算核心技术和产品的研制，目前可信计算关键技术已经形成相当的积累。

国际上，1999 年由 HP、IBM、Intel、Microsoft 等 IT 巨头成立了 TCPA（Trusted Computing Platform Alliance），开始在全球范围内倡导可信计算理念，推广可信计算技术和标准。2003 年，TCPA 改组为 TCG（Trusted Computing Group），发布了 TPM 1.2 技术规范，同时从 PC 平台扩展到服务器、PDA、移动电话等各类平台，将可信计算技术渗透到可信计算平台的各个层面。我国在可信计算领域起步不晚，发展也比较迅速。在有关政府部门的认可和支持下，由众多厂商和科研机构共同成立了中国可信计算工作组（简称 TCMU），大力发展自主创新的可信计算技术和标准。目前我国已经成功研制了 TCM（Trust Cryptographic Module）安全芯片、TSM（TCM Service Module）软件、安全 PC 等，全面地掌控了可信计算核心关键技术，并于 2007 年 12 月正式颁布了《可信计算密码支撑平台功能与接口规范》，标志着我国具有自主知识产权的可信计算技术、产品和标准进入一个新的发展阶段。

正是在这种背景下，由中国密码学会主办，中国可信计算工作组协办了“第一届中国可信计算理论与实践学术会议”，旨在汇聚国内可信计算领域的专家学者、研究开发者和工程技术人员，分享近年来可信计算理论、技术和应用方面的研究成果，交流可信计算应用方面的技术挑战和研究问题，讨论可信计算未来发展方向和发展趋势。本次会议收到众多的可信计算论文来稿，内容涉及可信计算信任理论、体系结构、可信系统、可信网络及可信计算应用技术，主要包括了可信计算理论模型和方法学，信任理论；包含 TCM 芯片，可信外设，可信主机等可信计算硬件平台；包括可信操作系统，完整性度量技术，远程证明方法，完整性评估技术等可信计算基础软件；基于可信计算技术的可信网络连接，网格安全技术；包含密钥管理技术，数字版权管理技术，电子政务/电子商务安全，可信存储等可信计算应用技术。经过多轮的严格评审和审议，本次会议以“择优录取”的原则收录论文 19 篇，并从中筛选出 4 篇优秀论文。

由于篇幅所限，大量优秀论文未能在本次会议中刊出。希望本次会议能为读者提供一

个了解可信计算理论和实践应用研究进展的窗口。最后,衷心感谢广大作者和审稿专家的大力支持!

冯登国

2009 年 8 月

# 目　　录

# A Remote Anonymous Attestation Scheme from ECC

Zhang Rui[1+], Liu Ji-Qiang[1], Han Zhen[1], Zheng Li-Juan[1,2]

[1](School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)

[2](Department of Computer and Information Engineering, Shijiazhuang Railway Institute, Hebei Shijiazhuang 050043, China)

[+]Corresponding author: Phn: +86-10-51688490, E-mail: penny_dada@hotmail.com

**Abstract**: Remote anonymous attestation is concernful to attest the remote platform is trusted but not revealing any private information of the platform. In this paper, we design a remote anonymous attestation scheme based on TPM which with ECC engine. Our scheme does not need any zero knowledge proof and the involvement of the third trusted party. So this scheme has better security properties and higher execution efficiency.

**Key words**: remote anonymous attestation; ring signature; hidden property certificate; ECC; TPM

## 1. Introduction

Nowadays, private protection is increasingly important in distributed environment. Terminals security is one of the biggest problems in distributed network. It is very likely to reveal platform private information in remote attestation process.

In 2004, E. Brickell at el. introduced a new remote attestation scheme named Direct Anonymous Attestation (DAA) based on Trusted Platform Module (TPM)[1]. DAA employs Zero Knowledge Proof[2] and Group Signature[3] to realize the anonymousness and omit the trusted third party (Private CA). DAA had become a part of TCG specifications (version 1.2) in 2006[4]. After that, several different DAA schemes are posed. In 2007, He Ge and Stephen R. Tate proposed a DAA scheme for devices with low computing capabilities[5]. A. Leung and E. Brickell take account of privacy preserving in their works[6,7]. However, all these schemes are not efficient enough because of the several times of Zero knowledge Proof.

A. R. Sadeghi proposed an efficient protocol called Property-Based Attestation (PBA) in 2004[8]. A property indicates various platform configurations to protect the platform from revealing configuration information. Therefore, verifier knows nothing about specific platform configuration but the knowledge that the platform has a certain property. Based on this protocol, L. Chen and R. Landfermann introduced a new property-based attestation[9]. A property and its corresponding configuration information are contained in a Property-Configuration Certificate issued and managed by a trusted third party. While, this method also uses Zero Knowledge Proof to conceal the real configuration information, so it is also hard to realize. Moreover, the trusted third party must store the information of all platform configurations and sign them. Actually, it increases the burden on

trusted third party.

According to above reasons, J. Q. Liu proposed a Remote Anonymous Attestation (RAA) scheme based on TPM[10,11]. This scheme protects private information without additional Zero Knowledge Proof so that it can be realized effectively.

All above schemes are suitable for RSA-based TPM. However, elliptic curve cryptography is more efficient than integer factorization systems and the length of private key is shorter than RSA algorithm. Therefore, ECC-based TPM is more efficient for DAA scheme. Recently, some DAA schemes are proposed with ECC-based TPM. E. Breickell, L. Chen and J. Li put forward a DAA scheme from pairings in 2008[12]. X. Chen and D. Feng also proposed a new DAA scheme from bilinear maps[13]. However, these two schemes also use CL signature and Zero Knowledge Proof. From above analysis, the efficiency of implement is not good enough.

Based on J. Q. Liu's work[10,11], we propose a remote anonymous attestation scheme which we call it RAA-ECC scheme in the following parts. This scheme not only satisfies the requirement of ECC algorithms, but also protects private information without Zero Knowledge Proof, so it is more efficient than other DAA schemes. Moreover, we use a modified Provably Secure Elliptic Curve Encryption Scheme-V3 (PSEC-3) as the elliptic curve encryption algorithm and add a flexible algorithm-link so that a verifier can link two signatures in some specific applications. Consequently, this scheme has better security characteristics, such as conditional anonymity, conditional linkability and other security attributes.

The rest of this paper is organized as follows. In section 2, we describe the related fundamentals that are needed in RAA-ECC scheme. The RAA-ECC scheme is introduced in section 3. The security analysis is discussed in section 4. Finally, the conclusion is in section 5.

## 2. Background Knowledge

In this section we introduce some preliminaries in RAA-ECC scheme. In this scheme we use the special TPM with ECC engine instead of RSA engine, so the algorithms are based on elliptic curve algorithms. We first describe the modified Provably Secure Elliptic Curve Encryption Scheme-V3 (PSEC-3), and then we introduce ring signature and hidden property certificate used in our scheme.

### 2.1 Modified PSEC-3

PSEC-3 is a public-key encryption system that uses the elliptic curve El Gamal trapdoor function and two random functions[14]. But we need to modify the PSEC-3 for our RAA-ECC scheme. With any symmetric encryption, the modified PSEC-3 is semantically secure against chosen-ciphertext attacks (IND-CCA2), in the random oracle model[15], under the Elliptic Curve Gap Diffie-Hellman (EC-Gap-DH) assumption.

The parameters of modified PSEC-3 is $(p, a, b, G, H_1, H_2, H', G', SymE)$, where $H_1, H_2, H'$ and $G'$ are four hash functions, $a$ and $b$ are two elliptic curve coefficients of $F_p$, that define an elliptic curve $E$. $G$ is a curve point of order $p$. $SymE = (E_K, D_K)$ is a symmetric encryption

scheme which uses secret key $K$. We use two hash functions $H_1$ and $H_2$ to instead of the random choice values $R$ and $r \in \mathbf{Z}_p$ which in PSEC-3 in following algorithms, since the verifier cannot know the random values when he verifies the signature in RAA-ECC scheme. While the modified PSEC-3 algorithm is also secure and the values $R$ and $r$ are also fresh, because of the randomness of $m$ in our RAA-ECC scheme.

**Encryption Algorithm of Modified PSEC-3**

With the plaintext $m$ and the public key $Q$, the encryption algorithm outputs ciphertext $c = (C_1, c_1', c_2, c_3)$ as follows:

1) Compute $R \leftarrow H_1(m)$ and $r \leftarrow H_2(m) \in \mathbf{Z}_p$.

2) Compute the points on $E$, $C_1 \leftarrow r \cdot G$ and $T \leftarrow r \cdot Q$.

3) Compute $c_1' \leftarrow x_T \oplus R$, $K \leftarrow G'(R)$ and $c_3 \leftarrow H'(C_1, c_1', R, m)$, where $x_T$ is the x-coordinate of $T$.

4) Compute $c_2 \leftarrow E_K(m)$.

**Decryption Algorithm of Modified PSEC-3**

With the input ciphertext $c = (C_1, c_1', c_2, c_3)$, the decryption algorithm outputs the plaintext or $\perp$ as follows:

1) Compute the point on $E$, $T' \leftarrow d \cdot C_1$ and $R' \leftarrow c_1' \oplus x_T'$.

2) Compute $K' \leftarrow G'(R')$ and $m' \leftarrow D_{K'}(c_2)$.

3) Check whether the equation holds or not: $c_3 = H'(C_1, c_1', R', m')$. If it holds, output $m'$ as the decrypted plaintext. Otherwise, output $\perp$.

## 2.2 Ring Signature

The ring signature is proposed by R. L. Rivest, A. Shamir and Y. Tauman in 2001[16]. A set of possible signers compose a ring. But there is only one real signer who produces the signature. A ring signature scheme is set-up free. The signer does not need the consent or assistance of the other ring members. Different members can use different public key signature schemes with different key and signature sizes. In our scheme, we use modified PSEC-3 as the signature scheme.

A ring signature scheme is defined by two steps:

1) *Ring-sign* $(m, P_1, P_2, \cdots, P_r, s', S_{s'})$ Given public keys $P_1, P_2, \cdots, P_r$ of the $r$ ring members, the secret key $S_{s'}$ of the real signer who is the $s'$-th member and message $m$, the real signer produces a ring signature $\sigma$.

2) *Ring-verify* $(m, \sigma)$ When verifier receives a message $m$ and a signature $\sigma$, he outputs $T$ or $F$ with the public keys of all the possible signers.

## 2.3 Hidden Property Certificate

We follow the notion of property-based certificate proposed by A. R. Sadeghi[8] in our scheme. The property $P'$ corresponds to various platform configurations $C_1, C_2, \cdots, C_t$, such as a platform has a property that it installs an anti-virus software, and this kind of software have several different vendors and versions.

The hidden property certificate is issued by TPM and the host of TPM. It indicates the properties of current platform. The signature signed by TPM can assure the authenticity of platform configurations. Platform configuration $C_r(1 \leqslant r \leqslant t)$ is concealed in the hidden property certificate by computing the hidden property value $y_s$. Then TPM and its host produce a signature $\sigma$ for property hidden value $y_s$, property $P'$, platform configurations $C_1, C_2, \cdots, C_t$ and public parameters. So the hidden property certificate of property $P'$ must contain the property hidden value $y_s$, property $P'$, signature $\sigma$ and public parameters.

# 3. Our RAA-ECC Scheme

In a RAA-ECC scheme, there are two types of entities: a real signer $S$ and a verifier $V$. Each signer and verifier is a trusted platform. $T_S$ is the TPM of $S$ and $H_S$ is the host of $S$. Correspondingly, $T_V$ is the TPM of $V$ and $H_V$ is the host of $V$.

## 3.1 The Process of RAA-ECC Scheme

A RAA-ECC scheme consists of four polynomial-time algorithms: $RAA\text{-}ECC = (Initialize, Sign, Verify, Link)$. We assume the signer $S$ is the resource requestor and the verifier $V$ is the resource provider. Due to the limitation of TPM, only some essential computations are performed inside the TPM and the others are done on host.

**Initialize** $H_S$ firstly defines the global public parameters $params = (p, a, b, G, n, h)$ of elliptic curve on finite field $F_p$ which must satisfy some restrictions[18] and sends them to $T_S$. Here, Elliptic curve is $E(F_p)$: $y^2 \equiv x^3 + ax + b \pmod p$, where $a, b \in F_p$. $G = (x_G, y_G)$ is the base point on $E(F_p)$. The prime number $n$ is the order of base point $G$ and integer $h = \#E(F_p)/n$. Then $H_S$ selects other $t-1$ platforms with ECC-based TPM to compose a ring. The public keys of these platforms can be obtained from their public key certificates. To facilitate, we let $Q_1, Q_2, \cdots, Q_t$ be the $t$ public keys. For each private key $d_j$ in rogue set $R$, $H_S$ computes $Q_j = d_j \cdot G$. If $Q_j (1 \leqslant j \leqslant t)$ matches with any $d_j$ in a set of rogue signers' secret keys $R$, $H_S$ rejects the invalid platform and re-chooses another platform with valid TPM to re-compose the ring. Finally, $T_S$ chooses its private key $d_S \in [1, n-1]$ and computes point $Q_S = (x_{Q_S}, y_{Q_S})$ on the elliptic curve which $Q_S = d_S \cdot G$ as its public key. $T_S$ sends $Q_S$ to $H_S$ and keeps $d_S$ secret.

**Sign** From the above description, the ring is composed of $t$ platforms with TPM, and their public keys are $Q_1, Q_2, \cdots, Q_t$ respectively. Suppose the sequence number of the real signer $S$ is $s'$ and its corresponding key pair is $(d_{s'}, Q_{s'})$. Note that $(d_{s'}, Q_{s'}) \neq (d_S, Q_S)$, where $(d_{s'}, Q_{s'})$ is the Endorsement Key (EK) of TPM, and $(d_S, Q_S)$ is a temporary key pair generated by $T_S$. Let $bsn_V$ be a basename associated with the verifier $V$. The process of sign takes the following steps:

1) If $bsn_V \neq \perp$, $H_S$ computes $U = H_E(bsn_V)$, where $H_E$: $\{0,1\}^* \rightarrow E(F_p)$; Otherwise, $H_S$ chooses $U \in E(F_p)$ uniformly at random. Then $H_S$ sends $U$ to $T_S$.

2) $H_S$ randomly selects a private information $x_S (1 \leqslant x_S \leqslant n-1)$ and calculates $x_S \cdot G = (x', y')$ and $r = x' \bmod n (r \neq 0)$. $H_S$ sends $x_S$, $r$ to $T_S$.

3) $T_S$ verifies that $U \in E(F_p)$ and calculates $W = d_{s'} \cdot U$. Note that $d_{s'} \neq d_S$. Then $T_S$ takes

out the configuration digest values from the Platform Configuration Registers (PCRs) in TPM and computes the hidden property value $y_S = x_S^{-1}(H_3(P', C_1', C_2', \cdots, C_t') + d_S r) \bmod n$, where $y_S \neq 0$ and $H_3$ is a one way hash function. Note that a kind of property $P'$ may have several different platform configurations $C_1', C_2', \cdots, C_t'$. So the information needs to be signed by $S$ is $M = (params, Q_S, y_S, r, P', U, W)$. $T_S$ sends $M$ to $H_S$.

4) $H_S$ computes value $k = H_0(M, Q_1, Q_2, \cdots, Q_t)$, where $H_0: \{0,1\}^* \rightarrow \{0,1\}^{l_1}$ and $l_1$ is the length of $k$, and picks a random integer as initial value $v$ and a random sequence $x_1, \cdots, x_{s'-1}, x_{s'+1}, \cdots, x_t$. Let $f$ be the encryption function. Then $H_S$ respectively computes $y_i = f(x_i), 1 \leqslant i \leqslant t, i \neq s'$ using encryption algorithm of modified PSEC-3 defined in Section 2.1.1.

5) $H_S$ chooses the following ring equation:

$$\begin{aligned} & C_{k,v}(y_1, y_2, \cdots, y_t) \\ = & E_k(y_t \oplus E_k(y_{t-1} \oplus E_k(\cdots \oplus E_k(y_1 \oplus v)\cdots))) \\ = & v \end{aligned} \tag{1}$$

where $E_k(\,)$ denotes a symmetric encryption algorithm and $k = H_0(M, Q_1, Q_2, \cdots, Q_t)$ is the key, and $\oplus$ indicates bit XOR operation.

6) $H_S$ figures out $y_{s'}$ from ring equation (1):

$$\begin{aligned} y_{s'} = & E_k(y_{s'-1} \oplus E_k(y_{s'-2} \oplus E_k(\cdots \oplus \cdots E_k(y_1 \oplus v)\cdots))) \\ & \oplus D_k(y_{s'+1} \oplus D_k(y_{s'+2} \oplus D_k(\cdots \oplus D_k(y_t \oplus D_k(v))\cdots))) \end{aligned}$$

where $D_k(\,)$ is the decryption algorithm corresponding to $E_k(\,)$. After that, $H_S$ sends $y_{s'}$ into $T_S$.

7) $T_S$ works out $x_{s'} = g(y_{s'})$ using decryption algorithm of modified PSEC-3 defined in Section 2.1.2 with private key $d_{s'}$ and sends $x_{s'}$ back to $H_S$.

8) $H_S$ produces a $(2t+1)$-tuples signature $\sigma = (Q_1, Q_2, \cdots, Q_t; v; x_1, x_2, \cdots, x_t)$.

**Verify** After receives the information $M = (params, Q_S, y_S, r, P', U, W)$ and the signature $\sigma = (Q_1, Q_2, \cdots, Q_t; v; x_1, x_2, \cdots, x_t)$, the verifier $V$ does following steps:

1) If $bsn_V \neq \perp$, $H_V$ checks that $U = H_E(bsn_V)$; Otherwise, $H_V$ checks whether or not $U \in E(F_p)$.

2) For each $d_j$ in $R$, $H_V$ checks $W \neq d_j \cdot U$. If $W$ matches with any $d_j$ in $R$, $H_V$ rejects and aborts.

3) $T_V$ computes $b_1 = H_4(d_V \cdot G)$ inside TPM and sends $b_1$ to $H_V$, where $H_4: E(F_p) \rightarrow \{0,1\}^*$ and $d_V$ is the private key of verifier's key pair $(d_V, Q_V)$.

4) $H_V$ validates the ring signature as follows:

a. $H_V$ computes $k = H_0(M, Q_1, Q_2, \cdots, Q_t)$ to obtain the symmetric key.

b. For each $1 \leqslant i \leqslant t$, $H_V$ calculates $y_i = f(x_i)$ as step 4) in Sign algorithm.

c. $H_V$ checks $C_{k,v}(y_1, y_2, \cdots, y_t) = v$ with $k$ and $y_i (1 \leqslant i \leqslant t)$, if the ring equation holds, $H_V$ accepts the ring signature; Otherwise, $H_V$ rejects the signature.

5) Then $H_V$ picks a random private value $x_V (1 \leqslant x_V \leqslant n-1)$ and respectively computes $u_1 = H_3(P', C_1', C_2', \cdots, C_t')\, y_s^{-1} \bmod n$ and $u_2 = r y_s^{-1} \bmod n$. Next, $H_V$ calculates $k_1 = H_5(b_1 x_V(u_1 \cdot G + u_2 \cdot Q_S))$ by using $u_1$ and $u_2$, where $H_5: E(F_p) \rightarrow \{0,1\}^*$.

6) $H_V$ sends $x_V \cdot G$ and the resource encrypted with secret key $k_1$ to $H_S$.

7) After receives the resource and $x_V \cdot G$, $H_S$ calculates $b_2 = H_4(Q_V)$ and $k_2 = H_5(b_2 x_S(x_V \cdot G))$, If $k_1 = k_2$, $S$ can get the resource by decrypting the message with key $k_2$, namely, $S$ has property $P'$; Otherwise, $S$ can't obtain the resource from $V$.

**Link** When verifier $V$ wants to know whether or not two given signatures $\sigma_0$ and $\sigma_1$ are linked, namely, signed under the same TPM private key $d_{s'}$. $V$ verifies the validation of them by using Verify algorithm. If either of them is invalid, $V$ outputs $\perp$; Otherwise, $V$ outputs 1 (linked) if $\sigma_0$ and $\sigma_1$ include the same $(U, W)$ pair or 0 (unlinked) otherwise.

### 3.2 Rogue TPM Tracing

Once a TPM have been compromised, its private key is exposed. How to trace rogue TPMs is a problem in our scheme. However, we use ring signature instead of group signature in this RAA-ECC scheme. There has no group manager in ring. So the verifier must verify the ring members to ensure the signature is not from a corrupted TPM when he receives a signature. We follow the method in paper [12] to trace the rogue TPMs. Differ from the scheme using group signature, every terminal must maintain a revocation list $R$ which contains private keys of corrupted TPMs and refresh it frequently. All private keys in the revocation list $R$ are published. For each private key $d_i$ in the revocations list, a verifier checks whether $W = d_i \cdot U$ or not. If the equation holds, the signature may come from a compromised TPM.

If verifier wants to make sure that the signature is from a corrupted TPM, it can use **Link** algorithm to link the signature with other signatures respectively which have been compromised after the verifier received them. The verifier checks whether or not the signature and a compromised signature include same pair $(U, W)$. If the **verify** algorithm outputs 1 (the two signature are linked), the verifier can ascertain that the signature is from a compromised TPM.

## 4. Security Analysis

Following the steps of the scheme in Section 3.1, the RAA-ECC scheme is correct. Besides, we employ the modified PSEC-3 as the ECC encryption algorithm. This algorithm is semantically secure with any symmetric encryption against chosen-ciphertext attacks (IND-CCA2), in the random oracle model, under the Elliptic Curve Gap Diffie-Hellman (EC-Gap-DH) assumption. Now we discuss the security attributes of our scheme as follows.

### 4.1 Unforgeability

An adversary can generate a valid property certificate, that is the adversary can create a hidden property value $y_S$, but can't obtain any private key of ring members. However, the adversary finds it hard to forge a valid signature by adaptively querying the oracles to get the query-answer pairs.

Moreover, the real signer $S$ computes hidden property value $y_S$ with private value $x_S$ picked by itself and creates the signature certificate inside TPM. So the configuration values are non-forgeable. Private values $x_S$ and $x_V$ make communication fresh between $S$ and $V$. Adversary can't impersonate $V$ to provide malicious resource, because of the involvement of $d_V$ in computation when verification.

### 4.2 Conditional Linkability

The concept of linkability is that two signatures signed by the same signer can be linked[17]. In the RAA-ECC scheme, we use Link algorithm to allow a verifier to recognize whether two signatures are linked or not. The Link algorithm is controlled by the real signer. If the verifier wants to know the linkability between two signatures, he needs to negotiate with the signer. If the signer willing to offer such a link between his two signatures, he makes use of a basename $bsn \neq \perp$ which associated with the verifier in his signatures; Otherwise, he uses $bsn = \perp$, then the verifier cannot link any two signatures. So we call this variable linkability attribute conditional linkability in this scheme.

### 4.3 Conditional Anonymity

For the anonymity, we use ring signature to conceal the identity of platform. Any adversary has probability at most $1/t$ to determine the identity of the actual signer in a ring of size $t$, if ignore the **Link** algorithm.

Unlike the unconditionally signer-ambiguous in original ring signature scheme[16], our scheme is conditional anonymity by appending **Link** algorithm. The notion of unconditional anonymity means that even an infinitely powerful adversary with access to an unbounded number of chosen-message signatures produced by the same ring member cannot guess his identity with any advantage and cannot link additional signatures to the same signer. But in our scheme, we consider that conditional anonymity, which in paper [12] is called user-controlled-anonymity, is more suitable for RAA scheme. Informally, conditional anonymity means that an adversary without the private key of the signer finds it hard to guess the identity of the signer from its signature and cannot tell whether or not two signatures associated with two different base names are signed by a same signer.

## 5. Conclusion

In this paper, we propose a RAA scheme using ECC-based TPM. And we use a provable secure algorithm-modified PSEC-3 as ECC encryption algorithm. It makes our scheme more secure. Compared to other schemes, our scheme has the shorter signature length because of using ECC algorithms. Moreover, the RAA-ECC scheme does not need zero knowledge proof and the involvement of the third trusted party. Consequently, the feasibility of this scheme is better than other direct anonymous attestation schemes.

**Acknowledgement** The work is supported by the 973 Research Foundation Grant 2007CB307101, National High Technology Research, Development Program of China 2007AA01Z410 and 2007AA01Z177, the program for Changjiang Scholars and Innovative Research Team in University, and the Scientific Research Foundation of BJTU K08J0030.

## References

[1] E. Brickell, J. Camenisch, and L. Chen, Direct Anonymous Attestation. In: *Proceedings of the 11th ACM*

*Conference on Computer and Communications Security*, Washington, DC, USA, Oct. 2004.

[2] S. Goldwasser, S. Micali, and C. Racko, The Knowledge Complexity of Interactive Proofs. *SIAM J. Comput.*, 1989, vol. 18(1), 186-208.

[3] D. Chaum and E. van Heyst. Group signatures. In: *Advances in Cryptology-EUROCRYPT'91*, LNCS 950, pp. 257-265, Springer-Verlag, 1992.

[4] Trusted Computing Group, TPM Main Specification, main Specification Version 1.2 rev. 94. 29 March 2006.

[5] He Ge, Stephen R. Tate: A Direct Anonymous Attestation Scheme for Embedded Devices. Public Key Cryptography, 2007: 16-30.

[6] A. Leung, L. Q. Chen and C. J. Mitchell, On a Possible Privacy Flaw in Direct Anonymous Attestation (DAA). In: *Proceedings of the 1st international conference on Trusted Computing and Trust in Information Technologies: Trusted Computing-Challenges and Applications*, Villach, Austria, pp. 179-190, 2008.

[7] E. Brickell and J. T. Li, Enhanced privacy id: a direct anonymous attestation scheme with enhanced revocation capabilities. In: *Proceedings of the 2007 ACM workshop on Privacy in electronic society*, Alexandria, Virginia, USA, pp. 21-30, 2007.

[8] A. R. Sadeghi and C. Stüble, Property-based attestation for computing platforms: Caring about properties, not mechanisms. In: *The 2004 New Security Paradigms Workshop*, Virginia Beach, VA, USA, Sept. 2004.

[9] L. Chen, R. Landfermann, H. Löhr, M. Rohe, A. Sadeghi, and C. Stüble, A Protocol for Property-Based Attestation. In: *STC'06*, Alexandria, Virginia, USA, November 3, 2006.

[10] J. Q. Liu, J. Zhao and Z. Han, A remote anonymous attestation protocol in trusted computing. In: *Proceeding of IEEE International Symposium on Parallel and Distributed Processing*, pp. 1-6, 2008.

[11] J. Q. Liu, J. Zhao, Y. Zhao. Study of remote anonymous attestation in trusted computing. Chinese Journal of Computers, 2009, vol. 32: 1-7.

[12] E. Brickell, L. Chen and J. Li. Simplified Security Notions of Direct Anonymous Attestation and a Concrete Scheme from Pairings. In: *Conference on Trusted Computing (TRUST 2008)*, Villach, Austria, March 2008.

[13] X. Chen, D. Feng, A new direct anonymous attestation scheme from bilinear maps. In: *The 9th International Conference for Young Computer Scientists, 2008. ICYCS*, pp. 2308-2313, 2008.

[14] T. Okamoto and D. Pointcheval, PSCE-3: Provable secure elliptic curve encryption scheme-V3. 2000. http://www.di.ens.fr/~pointche/proposals/IEEE/../../Documents/Reports/2000_PSEC3.ps.gz.

[15] M. Bellare and P. Rogaway, Random oracles are practical: A paradigm for designing efficient protocols. In: *Proceedings of the 1st ACM conference on Computer and communications security*, Fairfax, Virginia, United States, pp. 62-73, 1993.

[16] R. L. Rivest, A. Shamir and Y. Tauman, How to leak a secret: theory and applications of ring signatures. Theoretical Computer Science, 2006, vol. 3895: 164-186.

[17] J. K. Liu, V. K. Wei and D. S. Wong, Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups. Information Security and Privacy, Springer Berlin, 2004, vol. 3108: 325-335.

[18] Certicom Research. SEC 1: Elliptic Curve Cryptography, Version 1.7. November 13, 2006.

## 附中文参考文献

刘吉强,赵佳,赵勇. 可信计算中远程自动匿名证明的研究. 计算机学报. 2009, 32: 1-7.

# 一种基于 ECC 的远程匿名证明方案

**摘　要**：远程匿名证明是用于证明远程平台可信但又不泄露任何平台信息的重要技术。本文设计了一种基于 ECC 算法的远程匿名证明方案。此方案不需要任何的零知识证明和可信第三方的参与。因此,此方案具有更高的执行效率和较好的安全性。

**关键词**：远程匿名证明；环签名；隐藏属性证书；ECC；TPM

**中图分类号**：TP309　**文献标识码**：A