直接匿名证言协议的性能估算新方法

谭 良[1),2)] 孟伟明[1)] 周明天[3)]

[1)](四川师范大学计算机学院,四川省可视化计算与虚拟现实重点实验室 成都 610066)

[2)](中国科学院计算技术研究所 北京 100190)

[3)](电子科技大学计算机学院 成都 610054)

摘 要 性能问题是阻碍DAA推广和应用的首要问题.为了进一步优化该协议的性能,找出性能瓶颈,定量地分析和测量DAA中各个实体的性能负荷分布是一个十分重要且必须的工作.文中详细分析了DAA的协议流程,提出了以机器周期为基本性能单位的性能负荷分布测量方法——归一化统计法(Normalized Statistics,NS).该方法需要首先分析DAA协议中的各种复杂运算,针对不同的运算选用当前性能较好的算法,然后统计各个算法中大整数单精度乘法、单精度加法、读内存、写内存等基本运算的数目,最后通过汇总并转换得出DAA协议中各实体以机器周期为单位的性能负荷分布和总性能负荷.比较分析表明,该方法不仅能相对准确、精细、有效地定量计算出DAA协议中各实体的性能负荷和总的性能负荷,而且测出的性能负荷具有平台无关性.最后为了说明该方法的有效性,将NS方法应用于有关可信计算匿名证明的一个典型方案的性能负荷估算.

关键词 可信计算;直接匿名证言;Camenisch-Lysyanskaya签名;知识证明;性能负荷

中图法分类号 TP311 **DOI号**: 10.3724/SP.J.1016.2012.01553

A New Method of Performance Estimate of Direct Anonymous Attestation Scheme in TCG

TAN Liang[1),2)] MENG Wei-Ming[1)] ZHOU Ming-Tian[3)]

[1)](*College of Computer, Sichuan Normal University, Key Laboratory of Visualization in Scientific Computing and Virtual Reality of Sichuan, Chengdu* 610066)

[2)](*Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190)

[3)](*School of Computer Science & Engineering, University of Electronic Science & Technology of China, Chengdu* 610054)

Abstract Performance is a most important problem to Direct Anonymous Attestation Scheme in TCG. It is very necessary and important to analyze and measure performance to every entity quantitatively for optimizing DAA. In this paper, DAA protocol is first analysed detailedly, and then a new performance measurement method, called Normalized Statistics method, which takes the machine period as the basic performance unit, is put forward. When using this method, all complex calculates in DAA protocol must be found out and statistic, and better algorithms to every complex calculate are chosen, moreover, to each algorithms, we need to compute the sum for each basic operation, such as multiplication of single big integer, addition of single big integer, reading and writing memory, and so on. Finally, the every entity performance and the whole performance burden in DAA, whose unit is the machine period, are summed. The theoretical analysis results show that the performance estimate is exact, meticulous and effective by this method which is independent of actual platform. For proving availability of the method, we apply it to estimate performance of other one DAA scheme.

Keywords trusted computing; direct anonymous attestation; Camenisch-Lysyanskaya sign; knowledge proof; performance burden

收稿日期:2011-01-24;最终修改稿收到日期:2011-05-03.本课题得到国家自然科学基金(60970113)、国家自然科学基金青年基金(60903073)资助.**谭 良**,男,1972年生,博士,教授,主要研究领域为可信计算、网络安全.E-mail:tanliang2008cn@126.com.**孟伟明**,男,1985年生,硕士研究生,研究方向为可信计算.**周明天**,男,1937年生,教授,主要研究领域为网络计算、信息安全.

1 引 言

可信计算组织 TCG 在 TPM 规范 1.2 版中采用了直接匿名证言(Direct Anonymous Attestation, DAA)协议[1],该协议主要基础包括 Camenisch-Lysyanskaya 签名方案[2]、基于离散对数的知识证明和 Fiat-Shamir 启发式方法[3],既解决了隐私 CA 的瓶颈问题,又实现对 TPM 芯片的认证和匿名,是当前可信计算平台身份证明最好的理论解决方案之一. 但是该协议非常复杂,实现过程中不仅涉及到多个实体,而且涉及大量的耗时运算. 性能问题突出制约了该协议的广泛应用.

为了进一步优化该协议的性能,找出性能瓶颈,定量地分析和测量 DAA 中各个实体的性能负荷是一个十分重要且必须的工作. 对于这一问题,最好的方法是开发出实际的 DAA 系统,在真实的环境中实际测量. 但实际上,目前产业界生产的可信计算产品如 TPM 芯片、可信计算机(台式机和笔记本)等采用 DAA 协议进行平台身份认证的产品还比较少见,再加上完整的 DAA 协议流程包括 TPM、平台 Host、发布方 Issuer、验证方 Verifier 和可信第三方 TTP 等共 5 个实体,在真实的运行环境测试性能负荷分布不仅需要 TPM 支持,而且还需要开发和建设其它 4 个实体. 显然,这是一个复杂的系统工程,需要较大的成本和时间开销. 因此,通常采用以下两种方法对 DAA 的性能进行定量分析和测试.

(1)仿真环境实测法. 文献[1]是在 IBM ThinkPad T41(1.7GHz Intel Pentium M CPU,1GB RAM,Linux)上运行 IBM 开发的 DAA 原型系统,测量的结果是,TPM 向发布者申请 DAA 证书需要 2.4 s,其中 TPM 约占用 25%,主机约占用 25%,发布者约占用 50%的时间;TPM 每次向验证者认证自己需要 4.4 s,其中 TPM 约占用 8%,主机约占用 47%,验证者约占用 45%的时间. 文献[4]是在 Intel dual-core 3.2GHz、1GB RAM、Windows 平台上开发 DAA 原型系统,设计了 tpm-module, host-module, server-module 3 个模块. 测量的结果是,在 Join 阶段,TPM 需要 15.2 s, Host 需要 12 s;在 Sign 阶段,TPM 需要 13.8 s,Host 需要 20.6 s;在 Verify 阶段,TPM 需要 0 s, Host 需要 30.4 s. 值得注意的是,此类方法通常在单台机器上对 DAA 的各协议实体进行性能测试,不计通信开销. 这和在实际的运行环境进行测试是有差别的,毕竟原型系统不是实际运行环境. 而且协议中涉及到的多个实体如果均在同一个平台上运行,会相互影响彼此的运行效率,因此该方法测试的性能负荷分布并不准确.

(2)运算符号化统计法. 该方法是将 DAA 协议中的各主要运算符号化,然后统计符号次数并求和. 例如只需将 DAA 中涉及的主要运算的表示符号约定:G_n:一指数模 n 运算,例如 $g^a \bmod n$(G_n 是 G_n^1 的简写);G_n^2:两指数连乘模 n 运算,例如 $g^a h^b \bmod n$;G_n^3:三指数连乘模 n 运算,例如 $g^a h^b y^c \bmod n$;G_n^i:依上述 G_n,G_n^2,G_n^3 的定义类推;G_Γ:一指数模 Γ 运算,例如 $g^a \bmod \Gamma$;G_Γ^2:两指数连乘模 Γ 运算,例如 $g^a h^b \bmod \Gamma$;G_Γ^3:三指数连乘模 Γ 运算,例如 $g^a h^b y^c \bmod \Gamma$;G_Γ^i:依上述 G_Γ,G_Γ^2,G_Γ^3 的定义类推;P_c:生成一个大素数的运算;P_v:验证一个数为素数的运算;H:Hash 运算;则 DAA 协议的性能分布如表 1.

表 1 基于运算符号化统计法的 DAA 性能负荷分布

阶段	参与方	主要计算开销
申请加入(Join)	TPM	$3 \cdot G_\Gamma + 2 \cdot G_n^3 + (i+3) \cdot H$
	Issuer	$j \cdot G_\Gamma + 2 \cdot G_n + 1 \cdot G_n^4 + 1 \cdot G_\Gamma^2 + 1 \cdot P_c + 2 \cdot H$
	Host	$1 \cdot G_\Gamma + 1 \cdot G_n^2 + 1 \cdot P_v + 3 \cdot H$
进行签名(Sign)	TPM	$3 \cdot G_\Gamma + 1 \cdot G_n^3 + 1 \cdot H$
	Host	$1 \cdot G_\Gamma + 1 \cdot G_n + 1 \cdot G_n^2 + 2 \cdot G_n^3 + 1 \cdot G_n^4 + 2 \cdot H$
签名验证(Verify)	Verifier	$4 \cdot G_\Gamma^2 + 2 \cdot G_n^4 + 1 \cdot G_n^6 + j \cdot G_\Gamma + 4 \cdot H$

实际上,有关 DAA 扩展和改进的大量文献(如文献[5-11])均采用该方法进行性能负荷分析. 因为该方法简单,且在同类运算性能的比较上非常有效,如 $G_n < G_n^2 < G_n^3$,$G_\Gamma < G_\Gamma^2 < G_\Gamma^3$ 等. 但该方法在不同类运算之间无法进行比较. 因此,当各协议实体的不同类型运算较多时,该方法不能定量估算性能负荷及其比例关系,不便于各阶段、各实体性能负荷的比较分析.

鉴于以上原因,提出了以机器周期为基本性能单位的性能负荷分布测量方法——归一化统计法(Normalized Statistics, NS). 该方法需要首先分析 DAA 协议中的各种复杂运算,针对不同的运算选用当前性能较好的算法,然后统计各个算法中大整数单精度乘法、单精度加法、读内存、写内存等基本运算的数目,最后通过汇总并转换得出 DAA 协议中各实体以机器周期为单位的性能负荷分布和总性能负荷. 理论分析表明,该方法不仅能相对准确、精细、有效地定量计算出 DAA 协议中各实体的性能负荷和总的性能负荷,而且测出的性能负荷具有平台无关性. 最后为了说明该方法的有效性,将 NS 方法应

用于有关可信计算匿名证明的一个典型方案的性能负荷估算.

2　DAA 协议流程分析

本节将依据文献[1]对 DAA 协议进行流程分析. 文献[1]重点描述 DAA 实现认证和匿名的复杂运算和步骤,但协议流程描述并不详细和完整. 如发布 DAA 证书时发布方对谁提供零知识证明协议证明 DAA 证书构造正确? 谁对 DAA 证书进行了签名? 等等. 本节将对此进行补充,使得 DAA 更明确和完整.

2.1　DAA 协议的常数和假设

DAA 协议涉及到的安全参数和长度要求包括,l_n为 RSA 模数长度,长度为 2048 位;l_f为 TPM 秘密 ID 的长度,长度为 104 位;l_e为指数 e 的长度,长度为 368 位;$l_{e'}$为选择 e 的区间长度,长度为 128 位;l_v为随机数 v 的长度,为 2536 位;l_ϕ为零知识协议安全常数,长度为 80 位;l_H为 Hash 函数输出长度,即 SHA-1,长为 160 位;l_Γ模数 Γ 的长度,长度为 1632 位;l_ρ为 ρ 的长度,与 Γ 一起应用于假名,长度为 208 位.

假设 1.　(Strong RSA Assumption)不存在可行的算法,对任意随机的 RSA 模数 n 和一个随机元素 $u\in Z_n^*$,计算出 $e>1$ 和 v,使得 $v^e\equiv u \bmod n$.

假设 2.　(Decisional Diffie-Hellman Assumption)假设 Γ 为 l_Γ比特的素数,ρ 为 l_ρ比特的素数且 $\rho|\Gamma-1$. 设 $\gamma\in Z_\Gamma^*$ 为阶等于 ρ 的元素,对足够大的 l_Γ 和 l_ρ,不存在可行的算法可以区分四元组$\{(\delta,\delta^a,\delta^b,\delta^{ab})\}$和$\{(\delta,\delta^a,\delta^b,\delta^c)\}$,其中 δ 为$\langle\gamma\rangle$中的一个随机元素,a、b 和 c 为区间$[0,\rho-1]$内的随机元素.

2.2　DAA 协议初始化

DAA 协议初始化是在发布方和可信第三方之间进行的. 发布方在生成 DAA 公钥的同时向可信第三方提供一个非交互式的零知识证明,证明其公钥的合法性. 这对于其后参与者在发布方作弊时也能保持匿名起到了关键的作用.

1. 发布方选择一个安全素数乘积构成的 RSA 模数 $n=pq$,其中 $p=2p'+1$,$q=2q'+1$,p,q,p',q',都是素数,n 长度为 l_n;

2. 选择 QR_n的一个随机生成元 g';

3. 选择随机整数 x_0、x_1、x_Z、x_S、x_h、$x_g\in[1,p'q']$,并计算

$$g=g'^{x_g} \bmod n,\quad h=g'^{x_h} \bmod n,\quad S=h^{x_s} \bmod n$$

$$Z=h^{x_Z} \bmod n,\quad R_0=S^{x_0} \bmod n,\quad R_1=S^{x_1} \bmod n;$$

4. 提供一个非交互式的零知识证明发布方公钥构造正确. 即发布方对 g 和 h 相对于 g'的离散对数作零知识证明,对 S 和 Z 相对于 h 的离散对数作零知识证明,对 R_0 和 R_1 对 S 的离散对数作零知识证明;

(1) 发布方选择随机数 r_{x_g}, r_{x_h}, r_{x_S}, r_{x_Z}, r_{x_0}, $r_{x_1}\in[1,p'q']$,计算

$$t_{x_g}=g'^{r_{x_g}} \bmod n,\ t_{x_h}=g'^{r_{x_h}} \bmod n,\ t_{x_S}=h^{r_{x_s}} \bmod n,$$

$$t_{x_Z}=h^{r_{x_Z}} \bmod n,\ t_{x_0}=S^{r_{x_0}} \bmod n,\ t_{x_1}=S^{r_{x_1}} \bmod n;$$

(2) 可信第三方选择一个随机比特串 $n_{TI}\in\{0,1\}^{l_H}$ 作为防重放攻击的 nonce,发送给发布方;

(3) 发布方也选择一个随机比特串 $n_{IT}\in\{0,1\}^{l_H}$ 作为防重放攻击的 nonce,并计算

$$c_{x_g}=H(t_{x_g}\|n_{TI}\|n_{IT}),\ c_{x_h}=H(t_{x_h}\|n_{TI}\|n_{IT}),$$

$$c_{x_S}=H(t_{x_s}\|n_{TI}\|n_{IT}),\ c_{x_Z}=H(t_{x_Z}\|n_{TI}\|n_{IT}),$$

$$c_{x_0}=H(t_{x_0}\|n_{TI}\|n_{IT}),\ c_{x_1}=H(t_{x_1}\|n_{TI}\|n_{IT});$$

(4) 发布方计算

$$s_{x_g}=r_{x_g}+c_{x_g}x_g,\ s_{x_h}=r_{x_h}+c_{x_h}x_h,\ s_{x_S}=r_{x_S}+c_{x_S}x_S,$$

$$s_{x_Z}=r_{x_Z}+c_{x_Z}x_Z,\ s_{x_0}=r_{x_0}+c_{x_0}x_0,\ s_{x_1}=r_{x_1}+c_{x_1}x_1,$$

将$(c_{x_g},s_{x_g},c_{x_h},s_{x_h},c_{x_S},s_{x_S},c_{x_Z},s_{x_Z},c_{x_0},s_{x_0},c_{x_1},s_{x_1},n_{IT})$传给可信第三方;

(5) 可信第三方验证 $s_{x_g}<(p'q'\times2^{h+1})$,$s_{x_h}<(p'q'\times2^{h+1})$,$s_{x_S}<(p'q'\times2^{h+1})$,$s_{x_Z}<(p'q'\times2^{h+1})$,$s_{x_0}<(p'q'\times2^{h+1})$,$s_{x_1}<(p'q'\times2^{h+1})$,并计算

$$H(g'^{s_{x_g}}t_{x_g}^{-c_{x_g}} \bmod n)=c_{x_g},\ H(g'^{s_{x_h}}t_{x_h}^{-c_{x_h}} \bmod n)=c_{x_h},$$

$$H(h^{s_{x_s}}t_{x_s}^{-c_{x_s}} \bmod n)=c_{x_S},\quad H(h^{s_{x_z}}t_{x_z}^{-c_{x_z}} \bmod n)=c_{x_z},$$

$$H(S^{s_{x_0}}t_{x_0}^{-c_{x_0}} \bmod n)=c_{x_0},\quad H(S^{s_{x_1}}t_{x_1}^{-c_{x_1}} \bmod n)=c_{x_1};$$

5. 选择素数 Γ 为 l_Γ比特,ρ 为 l_ρ比特且 $\Gamma=r\rho+1$. 选择一个随机数 $\gamma'\in_R Z_\Gamma^*$ 使得 $\gamma'^{(\Gamma-1)/\rho}\neq1 \bmod\Gamma$,记 $\gamma=\gamma'^{(\Gamma-1)/\rho} \bmod\Gamma$;

6. DAA 发布者将$(n,g',g,h,S,Z,R_0,R,\gamma,\Gamma,\rho)$发送给可信第三方,可信第三方用其私钥对此签名,随后发布方公布发布者公钥.

2.3　DAA-Join 协议

设发布者公钥 $PK_1=(n,g',g,h,S,Z,R_0,R_1,\gamma,\Gamma,\rho)$,发布方对 PK_1的签名使用的公钥是 PK_1',假设发布方基名是 bsn_1. *DAAseed* 为 TPM 产生 (f_0,f_1)的恒定常量.

1. 平台计算 $\zeta_1=(H(1\|bsn_1))^{(\Gamma-1)/\rho} \bmod\Gamma$,将结果返给 TPM;

2. TPM 检查 $\zeta_1^\rho=1 \bmod\Gamma$,计算其秘密 ID,

$$f=H(H(DAAseed\|H(PK_1'))\|cnt\|1) \bmod\rho,$$

记 $f_0=LSB_{l_f}(f)$,$f_1=CAR_{l_f}(f)$,选择一个随机数 $v'\in_R\{0,1\}^{l_\phi+l_n}$,计算 $U=R_0^{f_0}R_1^{f_1}S^{v'} \bmod n$,$N_1=\zeta_1^{f_0+f_1^2\,l_f} \bmod\Gamma$,将 U 和 N_1发送给平台,平台转发给发布方;

3. 发布方检查撤销列表中的所有 f_0 和 f_1,验证 $N_1\neq$

$\zeta_1^{f_0+f_1^{2^{l_f}}} \bmod \Gamma$. 发布方同时还检查该平台以前使用的 N_1，如果平台位于撤销列表中，发布方终止 Join 协议；

4. TPM 通过零知识协议向发布方证明其拥有 f_0、f_1 和 v' 以及 U 和 N_1 的正确构成；

(1) TPM 选择随机数 $r_{f_0}, r_{f_1} \in_R \{0,1\}^{l_f+l_\phi+l_H}$ 和 $r_{v'} \in_R \{0,1\}^{l_f+2l_\phi+l_H}$，计算 $\widetilde{U}=R_0^{r_{f_0}}R_1^{r_{f_1}}S^{r_{v'}} \bmod n$，$\widetilde{N}_1=\zeta_1^{r_{f_0}+r_{f_1}^{2^{l_f}}} \bmod \Gamma$，将结果发送给平台；

(2) 发布方选择一个随机比特串 $n_i \in \{0,1\}^{l_H}$ 作为防重放攻击的 nonce，发送给平台；

(3) 平台计算 $c_h=H(n \| R_0 \| R_1 \| S \| U \| N_1 \| \widetilde{U} \| \widetilde{N}_1 \| n_i)$，返回结果给 TPM；

(4) TPM 选择 TPM 端的 nonce，$n_t \in \{0,1\}^{l_\phi}$，计算 $c=H(c_h \| n_t)$；

(5) TPM 计算 $s_{f_0}=r_{f_0}+cf_0$，$s_{f_1}=r_{f_1}+cf_1$，$s_{v'}=r_{v'}+cv'$；

(6) TPM 发送 $c, n_t, s_{f_0}, s_{f_1}, s_{v'}$ 给平台，平台转发给发布方；

(7) 发布方计算

$$\widetilde{U}=U^{-c}R_0^{s_{f_0}}R_1^{s_{f_1}}S^{s_{v'}} \bmod n, \widetilde{N}_1=N_1^{-c}\zeta_1^{s_{f_0}+s_{f_1}^{2^{l_f}}} \bmod \Gamma,$$

然后验证

$$c=H(H(n \| R_0 \| R_1 \| S \| U \| N_1 \| \widetilde{U} \| \widetilde{N}_1 \| n_i) \| n_t),$$

$s_{f_0}, s_{f_1} \in \{0,1\}^{l_f+l_\phi+l_H+1}$，$s_{v'} \in \{0,1\}^{l_n+2l_\phi+l_H+1}$；

5. 发布方选择随机数 $\hat{v} \in_R \{0,1\}^{l_v-1}$ 和素数 $e \in_R [2^{l_e-1}, 2^{l_e-1}+2^{l_{e'}-1}]$，计算 $v''=\hat{v}+2^{l_v-1}$，$A=\left(\frac{Z}{US^{v''}}\right)^{1/e} \bmod n$；

6. 发布方向平台证明其 A 计算正确(防止发布方得到 A 后，选择一个 $b \notin \langle h \rangle$，且 $b^e=1 \bmod n$，返回 Ab 给平台，可用于后继跟踪平台)；

(1) 平台选择一个随机的 nonce，$n_h \in \{0,1\}^{l_H}$，将结果发送给发布方；

(2) 发布方随机生成 $r_e \in_R [0, p'q']$，计算 $\tilde{A}=(Z/US^{v''})^{r_e} \bmod n$；

(3) $c'=H(n \| Z \| S \| U \| v'' \| A \| \tilde{A} \| n_h)$ 和 $s_e=r_e-c'/e \bmod p'q'$，并将结果 c', s_e, A, e, v'' 发送给平台；

(4) 平台验证 e 是素数且位于区间 $[2^{l_e-1}, 2^{l_e-1}+2^{l_{e'}-1}]$，计算 $\hat{A}=A^{c'}(Z/US^{v''})^{s_e} \bmod n$，验证 $c'=H(n \| Z \| S \| U \| v'' \| A \| \hat{A} \| n_h)$；

7. 平台发送 v'' 给 TPM；

8. TPM 计算 $v=v'+v''$，保留 v、f_0 和 f_1.

2.4 DAA-Sign 协议

1. 根据验证方是否提供 bsn_v，平台计算 $\zeta \in_R \langle \gamma \rangle$ 或 $\zeta=(H_\Gamma(1 \| bsn_v))^{(\Gamma-1)/\rho} \bmod \Gamma$；

2. 平台随机选择整数 $w, r \in \{0,1\}^{l_n+l_\phi}$，计算 $T_1=Ah^w \bmod n$ 和 $T_2=g^w h^e (g')^r \bmod n$. TPM 计算 $N_v=\zeta^{f_0+f_1^{2^{l_f}}} \bmod \Gamma$ 并将结果发送给平台；

3. TPM 和平台合作生成一个零知识证明，证明 T_1、T_2 来自于 DDA 证书，且 N_v 计算中应用的 f 为该 DAA 证书的秘密；

(1) TPM 随机选择 $r_v \in_R \{0,1\}^{l_v+l_\phi+l_H}$，计算

$\widetilde{T}_{1t}=R_0^{r_{f_0}}R_1^{r_{f_1}}S^{r_v} \bmod n$，$\tilde{r}_f=r_{f_0}+r_{f_1}^{2^{l_f}} \bmod \rho$，$\widetilde{N}_v=\zeta^{\tilde{r}_f} \bmod \Gamma$，TPM 将 $\widetilde{T}_{1t}$、$\widetilde{N}_v$ 发送给平台；

(2) 平台随机生成

$r_e \in_R \{0,1\}^{l_{e'}+l_\phi+l_H}$，$r_{ee} \in_R \{0,1\}^{2l_e+l_\phi+l_H+1}$，

$r_w, r_r \in_R \{0,1\}^{l_n+2l_\phi+l_H}$，$r_{ew}, r_{er} \in_R \{0,1\}^{l_e+l_n+2l_\phi+l_H+1}$，

计算

$$\widetilde{T}_1=\widetilde{T}_{1t}T_1^{r_e}h^{-r_{ew}} \bmod n, \widetilde{T}_2=g^{r_w}h^{r_e}g'^{r_r} \bmod n,$$

$$\widetilde{T}_2'=T_2^{-r_e}g^{r_{ew}}h^{r_{ee}}g'^{r_{er}} \bmod n;$$

(3) 平台计算

$$c_h=H(n \| g \| g' \| h \| R_0 \| R_1 \| S \| Z \| \gamma \| \Gamma \| \rho \| \zeta \| T_1 \| T_2 \| N_v \| \widetilde{T}_1 \| \widetilde{T}_2 \| \widetilde{T}_2' \| \widetilde{N}_v \| n_v),$$

并将结果发送给 TPM，TPM 选择一个随机的 nonce，$n_t \in \{0,1\}^{l_\phi}$，计算 $c=H(H(c_h \| n_t) \| b \| m)$，并将 n_t，c 返回给平台；

(4) TPM 计算

$$s_v=r_v+cv, s_{f_0}=r_{f_0}+cf_0, s_{f_1}=r_{f_1}+cf_1,$$

并将 3 个结果发送给平台；

(5) 平台计算

$s_e=r_e+c(e-2^{l_e-1})$，$s_{ee}=r_{ee}+ce^2$，$s_w=r_w+cw$，

$s_{ew}=r_{cw}+cwe$，$s_r=r_r+cr$，$s_{er}=r_{er}+cer$；

(6) 平台最终输出对 m 的签名：

$$\sigma=(\zeta, (T_1, T_2), N_v, c, n, (s_v, s_{f_0}, s_{f_1}, s_e, s_{ee}, s_w, s_{ew}, s_r, s_{er})).$$

2.5 DAA-Verification 协议

1. 计算

$$\hat{T}_1=Z^{-c}T_1^{s_c+c\times 2^{l_e-1}}R_0^{s_{f_0}}R_1^{s_{f_1}}S^{s_v}h^{-s_{ew}} \bmod n,$$

$$\hat{T}_2=T_2 g^{s_w}h^{s_c+c\times 2^{l_e-1}}g'^{s_r} \bmod n,$$

$$\hat{T}_2'=T_2^{-(s_e+c2^{l_e-1})}g^{s_{ew}}h^{s_{ee}}g'^{s_{er}} \bmod n,$$

$$\hat{N}_v=N_v^{-c}\zeta^{s_{f_0}+s_{f_1}^{2^{l_f}}} \bmod \Gamma;$$

2. 验证

$$c=H(H(H((n \| g \| g' \| h \| R_0 \| R_1 \| S \| Z \| \gamma \| \Gamma \| \rho \| \zeta \| T_1 \| T_2 \| N_v \| \widetilde{T}_1 \| \widetilde{T}_2 \| \widetilde{T}_2' \| \widetilde{N}_v \| n_v) n_t) \| b) \| m)$$

以及 $N_v, \zeta \in \langle \gamma \rangle$，$s_{f_0}, s_{f_1} \in \{0,1\}^{l_f+l_\phi+l_H+1}$ 和 $s_e \in \{0,1\}^{l_{e'}+l_\phi+l_H+1}$；

3. 如果验证方提供了基名，验证

$$\zeta \equiv (H(1 \| bsn_v))^{(\Gamma-1)/\rho} \bmod \Gamma;$$

4. 检查撤销列表上所有的 f_0、f_1，验证

$$N_v \neq \zeta^{f_0+f_1^{2^{l_f}}} \bmod \Gamma.$$

3　DAA 协议的性能估算新方法——NS 方法

NS 方法的步骤如下：(1) 首先进行复杂运算的统计；(2) 确定主要运算及其算法选择；(3) 基本运算统计；(4) 以机器周期数为基本单位估算性能负荷分布.

3.1　DAA 协议的复杂运算统计

NS 方法的第 1 步是进行复杂运算的统计. DAA 包括初始化阶段、Join 协议阶段、Sign 阶段和验证阶段，我们在进行复杂运算统计时，实体仅考虑发布方 Issuer、Host 平台、TPM 以及验证方 Verifier. DAA 协议是以大数运算为主，主要的复杂运算包括大素数选择、大随机数产生、模指数运算、SHA-1 运算、大整数乘法和大整数加法等. 下面将各种复杂运算按照实体进行统计，可以很容易地看出各个协议实体在整个协议过程中的复杂运算消耗. 由于篇幅关系，统计过程略. 统计结果见表 2. 其中 l_c 是 TPM 撤销列表的长度.

表 2　DAA 各实体的主要运算统计总表

运算 实体	大素数 选择	大随机数 产生	模指数 运算	SHA-1 运算	大整数 乘法运算	大整数 加法运算
Host	0 次	9 次	20 次	4 次	9 次	6 次
TPM	0 次	7 次	14 次	4 次	6 次	8 次
Issuer	5 次	18 次	24 次	6 次	7 次	8 次
Verifier	0 次	0 次	$17+l_c$ 次	4 次	0 次	0 次

3.2　DAA 复杂运算的算法选择

NS 方法的第 2 步是确定主要运算及其算法选择. 从表 2 可以看出运算次数较多是大整数模指数运算、大随机数运算和大整数乘法运算. 其它运算如大素数选择、SHA-1 运算、大整数加法等运算次数较少，因此，在性能估算时，主要考虑大整数模指数运算、大随机数产生运算和大整数乘法运算.

另外，对主要运算的算法选择时，应遵循一个基本的原则，即选择的算法应该是该运算使用得最多、最广泛的算法，并已经应用于 GMP、NTL、Crypto++、LibTomCrypt (LibTomMath)、OpenSSL 以及 miracl 等库中.

根据以上原则，大随机数产生运算选择 Lehmer 提出的线性同余法算法. 如算法 1 所示.

算法 1.

$$\begin{cases} x_{i+1}=(ax_i+c)\bmod m \\ r_i=\dfrac{x_i}{m} \end{cases},$$

其中 a 是乘子，c 是增量，x_0 为种子，m 为模数，当 $a\approx x_i$，该算法主要运算是一次模乘运算. 著名的粒子输运 Monte Carlo 程序 MCNP、MORSE 和 KENO 的随机数发生器均基于该方法.

模指数运算选择 Montgomery 模指数算法[12]，该算法是目前最好的模指数算法之一，由滑动窗口指数算法结合 Montgomery 模乘法，如算法 2 所示.

算法 2.　Montgomery 模指数算法.

输入：整数 $m=(m_{l-1}\cdots m_1 m_0)_b$，$gcd(m,b)=1$，$R=b^l$，$m'=-m^{-1}\bmod b$，$e=(e_l e_{l-1}\cdots e_1 e_0)_2$，$e_l=1$，整数 x，$1\le x<m$

输出：$x^e \bmod m$

1. $\bar{x}\leftarrow Mont(x,R^2\bmod m)$，$A\leftarrow R\bmod m$；
2. 对于 i 从 t 递减到 0，执行：
   2.1. $A\leftarrow Mont(A,A)$；
   2.2. 若 $e_i=1$，则 $A\leftarrow Mont(A,x)$；
3. $A\leftarrow Mont(A,1)$；
4. 返回 A.

该算法主要运算如表 3. 其中 $t+1$ 表示指数 e 的二进制长度，l 表示以 b 为基的数的长度.

表 3　Montgomery 模指数算法的主要运算次数

步骤	Montgomery 模乘法次数	单精度乘法次数
1	1	$2l(l+1)$
2	$\frac{3}{2}t$	$3tl(l+1)$
3	1	$l(l+1)$

算法 1 和 2 的关键运算均是模乘运算，因此模乘运算算法的选择非常关键. 在众多的模乘算法中，选择 CIOS 算法[13]，该算法将多精度乘法和模约减算法完美地融合为一体，在读写内存等方面节省了许多资源，如算法 3 所示.

算法 3.　CIOS 模乘算法.

输入：整数 $m=(m_{n-1}\cdots m_1 m_0)_b$，$x=(x_{n-1}\cdots x_1 x_0)_b$，$y=(y_{n-1}\cdots y_1 y_0)_b$，$gcd(m,b)=1$，$R=b^n$，$m'=-m^{-1}\bmod b$

输出：$xyR^{-1}\bmod m$

1. $A\leftarrow 0$；
2. 对于 i 从 0 到 $n-1$，执行：
   2.1. $A\leftarrow A+x_i y$；
   2.2. $u_i\leftarrow a_0 m'\bmod b$；
   2.3. $A\leftarrow (A+u_i m)/b$；
3. 如果 $A\ge m$，则 $A\leftarrow A-m$；
4. 返回 A.

算法 3 包含单精度乘法、单精度加法、读内存以及写内存的次数如表 4. 其中 k 表示以 b 为基的大整数的位数.

表 4 COIS 模乘算法的基本运算及次数统计表

基本运算	单精度乘法	单精度加法	读内存	写内存
CIOS 算法	$2k^2+k$	$4k^2-k-1$	$4k^2+7k$	$2k^2+4k$

多精度乘法运算选择效率较高的 Comba 算法[14]. 如算法 4 所示.

算法 4. Comba 多精度乘算法.

输入:两个整数 x 和 y,长度分别为 n 和 t,基为 b

输出:乘积的 b 进制表示 $x\times y=(w_{n+t-1}\cdots w_1 w_0)_b$

1. $(v_2 v_1 v_0)_b=0$;

2. 对于 i 从 0 到 $n+t-2$,执行:

2.1. $(v_2 v_1 v_0)_b\leftarrow(v_2 v_1 v_0)_b+\sum_{j=0}^{i}x_j y_{i-j}$;

2.2. $w_i\leftarrow v_0$, $v_0\leftarrow v_1$, $v_1\leftarrow v_2$, $v_0\leftarrow 0$;

3. $w_{n+t-1}\leftarrow v_0$;

4. 返回$(w_{n+t-1}\cdots w_1 w_0)_b$.

算法 4 包含单精度乘法、单精度加法、读内存以及写内存的次数如表 5. 其中 k 表示以 b 为基的大整数的位数.

表 5 Comba 多精度乘算法的基本运算及次数统计表

操作	单精度乘法	单精度加法	读内存	写内存
Comba 算法	k^2	$2k^2-2$	$2k^2$	$2k$

由于算法 1 的实现主要包括算法 3,根据表 4,得算法 1 包含的基本运算统计表 6.

表 6 大随机数的基本运算及次数统计表

操作	单精度乘法	单精度加法	读内存	写内存
Moktgomery 模乘算法	$2k^2+k$	$4k^2-k-1$	$4k^2+7k$	$2k^2+4k$

由于算法 2 的实现主要包括算法 3 和 4,根据表 4、表 5 得算法 2 包含的基本运算统计表 7.

表 7 Moktgomery 模乘算法的基本运算及次数统计表

操作	单精度乘法	单精度加法	读内存	写内存
Moktgomery 模乘算法	$(2+1.5t)\times(2k^2+k)$	$(2+1.5t)\times(4k^2-k-1)$	$(2+1.5t)\times(4k^2+7k)$	$(2+1.5t)\times(2k^2+4k)$
单精度乘法次数	$3l(l+1)(t+1)$	0	0	0

3.3 DAA 协议各阶段的基本运算统计

NS 方法的第 3 步是基本运算统计. NS 方法的基本运算是单精度乘法、单精度加法、读内存和写内存. 之所以选择这些运算作为 NS 方法的基本运算,是因为这些基本运算的机器周期很容易确定. 由于当前的多数计算机为 32 位机型,因此,选择 $b=32$. 如果机型为 64 位,则 $b=64$,则统计的相关结果减半.

3.3.1 DAA Join 阶段的基本运算统计

在 Join 阶段,协议实体包含 Host 平台、TPM 和 Issuer. Join 阶段的基本运算统计表 8.

表 8 DAA 协议 Join 阶段的基本运算次数统计表

	单精度乘法	单精度加法	读内存	写内存
Host	114385681	226020918	233330857	117121378
TPM	383098821	376894638	389356930	195455631
Issuer	560984641	552135689	569901665	286058980

3.3.2 DAA Sign 阶段的基本运行统计

在 Sign 阶段,协议实体包含 Host 和 TPM. Sign 阶段的基本运算统计表 9.

表 9 DAA 协议 Sign 阶段的基本运算次数统计表

	单精度乘法	单精度加法	读内存	写内存
Host	735976684	727407463	750465815	376673079
TPM	203641660	201146242	207878436	104359354

3.3.3 DAA Verification 阶段的基本运算统计

在 Verification 阶段,协议实体包含 Verifier,取 $l_c=0$. Verification 阶段的基本运算统计表 10.

表 10 DAA 协议 Verification 阶段的基本运算次数统计表

	单精度乘法	单精度加法	读内存	写内存
Verifier	813090596	803349273	829421714	416337084

3.4 DAA 协议的性能负荷分布及总性能负荷的估算与分析

NS 方法的最后一步是以机器周期数为基本单位估算性能负荷分布.

定义 1. 令执行一次单精度乘法运算的机器周期数为 θ_0,执行一次单精度加法的机器周期数为 θ_1,读内存的机器周期数为 θ_2,写内存的机器周期数为 θ_3. 在 DAA 协议的某一阶段某实体运行单精度乘法 n_0 次,运行单精度加法 n_1 次,读内存 n_2 次,写内存 n_3 次,则该实体在此阶段运行时所需机器周期总数为

$$T=n_0\times\theta_0+n_1\times\theta_1+n_2\times\theta_2+n_3\times\theta_3 \quad (1)$$

假定 DAA 整个协议均运行在 X86 指令系统上,通常 X86 指令系统执行一次单精度乘法运算的机器周期数了 $\theta_0=2$,加法的机器周期数 $\theta_1=2$,读内存的机器周期数 $\theta_2=1$,写内存的机器周期数 $\theta_3=1$.

由式(1)根据表 8 在 Join 阶段各实体的机器周期总数如表 11. 图 1 为 Join 阶段的性能分布图. 从图 1 可以看出,在该阶段 Verifier 的性能开销占 0%,TPM 占 33.48%,Host 占 16.58%,Issuer 占 49.94%.

表 11　Join 阶段的各协议实体的机器周期总数

协议实体	机器周期数
TPM	2104799479
Host	1031265433
Issuer	3082201305
Verifier	0

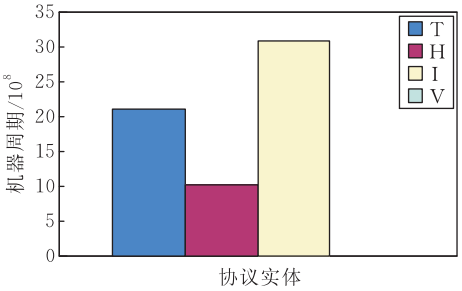

图 1　DAA Join 阶段性能负荷分布

由式(1)根据表 9 在 Sign 阶段各实体的机器周期总数如表 12. 图 2 为 Sign 阶段的性能分布图. 从图 2 可以看出，在该阶段 Issuer、Verifier 的性能开销占 0%，TPM 占 21.68%，Host 占 78.32%.

表 12　Sign 阶段的各协议实体的机器周期总数

协议实体	机器周期数
TPM	1121813594
Host	4053907188
Issuer	0
Verifier	0

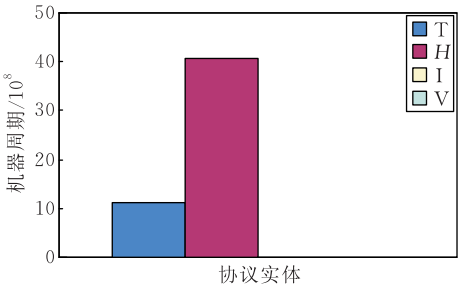

图 2　DAA Sign 阶段性能负荷分布

由式(1)根据表 10 在 Verify 阶段各实体的机器周期总数如表 13. 图 3 为 Verify 阶段的性能分布图. 从图 3 可以看出，在该阶段 TPM、Issuer 的性能开销占 0%，Verifier 占 100%.

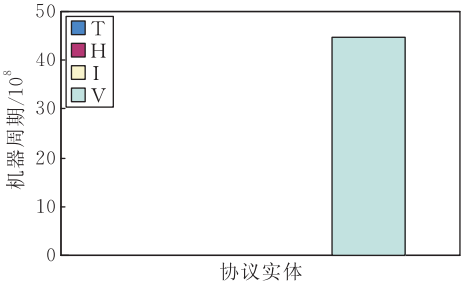

图 3　DAA Sign 阶段性能负荷分布

表 13　Verify 阶段的各协议实体的机器周期总数

协议实体	机器周期数
TPM	0
Host	0
Issuer	0
Verifier	4478638536

由式(1)根据各实体的机器周期总数如表 14. 图 4 为总性能分布图. 从图 4 可以看出 DAA 整个协议完成各实体的总性能开销分布. TPM 占 16.08%，Host 占 17.46%，Host 占 27.51%，Issuer 占 31.81%，Verifier 占 24.22 %.

表 14　各实体的机器周期总数

协议实体	机器周期数
TPM	3226613073
Host	5085172621
Issuer	5878593311
Verifier	4478638536

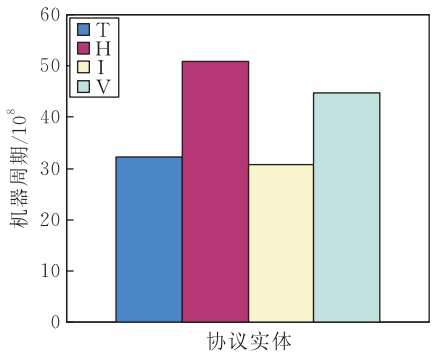

图 4　DAA 总性能负荷分布

显然，采用此方法得出的性能分布与单机系统仿真环境实测法得出的结果完全不一致[2-3].

另外，还可以对各实体在协议的各阶段之间的性能开销做一个比较：

(1) 对于 TPM，显然有 $TPM_{\text{Join}} > TPM_{\text{Sign}} > TPM_{\text{Verify}}$，其中，Join 阶段占 65.23%，Sign 阶段占 34.77%，其它两阶段占 0%.

(2) 对于 Host，显然有 $Host_{\text{sign}} > Host_{\text{Join}} > Host_{\text{Verify}}$，其中，Sign 阶段占 79.72%，Join 阶段占 20.28%，其它两阶段占 0%.

(3) 而对于 Verifier，仍有 $Verifier_{\text{Verify}} > Verifier_{\text{sign}} = Verifier_{\text{Join}}$ 其中，Verify 阶段占了 100%，其它三阶段占 0%.

4　NS 方法的应用

为了验证 NS 方法的有效性，本节将 NS 方法应用于可信计算匿名证明的文献[11]典型方案的性

能负荷估算，并将该方案与原方案进行比较.

首先统计文献[11]中的复杂运算. 文献[11]主要运算包括 $E(F_q)$ 的点加、标量乘[15]和非双线形映射，而 $E(F_q)$ 的点加、标量乘和非双线形映射的主要开销是求逆(I 表示)、模乘法(M 表示)和模平方(S 表示). 仿射坐标表示时：两不同点相加(点的加法)，需要两次模乘法，1 次模平方，1 次求逆，即 $2M+1S+1I$；相同点相加(点的倍点)，需要 2 次乘法，2 次模平方，1 次求逆，即 $2M+2S+1I$. 对采用 Tate 对计算非双线形映射[16]，需要 6 次 M_k，8 次 M_b，2 次 S_k，$(19+2k)$ 次 M，8 次 S，4 次 I，即 $6M_k+8M_b+2S_k+(19+2k)M+8S+4I$，其中 $M_k \approx k^{1.6}M$，$M_b=kM$，$S_k=2^kS$，k 是 F_{p^k} 嵌入系数. 表 15 是文献[11]中协议各阶段的主要运算统计表.

表 15 文献[11]方案的复杂运算统计

运算阶段		运算		
		点加	标量乘	映射
Join	TPM	0	3	0
	Host	1	0	6
	Issuer	2	6	0
Sign	TPM	0	1	0
	Host	0	7	1
Verify	Verifier	1	5	5

然后确定主要运算及其算法. 模乘法运算选用第 3 节中的 CIOS 算法；模平方算法选用 Montgomery 模平方算法的改进算法 OMMS1，适合通用 32 位处理器计算大整数；对于模逆算法，根据费马定理，当 q 是素数，a 是正整数且不能被 q 整除时有 $a^{-1}=a^{q-2} \bmod q$. 因此，对 $E(F_q)$ 内元素求逆都可和模指数运算统一，因此模逆算法选用 Moktgomery 模乘算法.

第 3 步是的基本运算统计. 取嵌入系统 $k=1$，$E(F_q)$ 中元素长度为 160 位，表 16 是协议各阶段运算统计表.

表 16 文献[11]方案的基本运算统计

运算阶段		运算			
		单精度乘法	单精度加法	读内存	写内存
Join	TPM	85260	81492	116871	59325
	Host	648150	569688	818022	423120
	Issuer	14718600	12667320	18191235	9422790
Sign	TPM	40770	55374	79127	38428
	Host	502075	429533	616892	319865
Verify	Verifier	786260	772375	1107453	562379

最后以机器周期数为基本单位估算性能负荷分布. 按照第 3 节的方法，表 17 是协议各阶段各协议实体的机器周期总数. 表 18 是各实体的总性能负荷分布.

表 17 文献[11]方案的性能负荷分布

阶段		负荷
		机器周期数
Join	TPM	509700
	Host	3676818
	Issuer	82385865
Sign	TPM	309843
	Host	2799973
Verify	Verifier	4787102

表 18 文献[11]方案各实体的总性能负荷分布

协议实体	机器周期总数
TPM	819543
Host	6476791
Issuer	82385865
Verifier	4787102

另外，可以将最初的原始方案和文献[11]方案作定量的比较：

(1) 对于两方案总开销，文献[11]方案机器周期总数的理论结果为 94469301，最初的原始方案机器周期总数的理论结果为 18669017541. 显然，文献[11]方案的性能远远高于最初的原始方案的整体性能，高 3 个数量级.

(2) 对于 TPM，文献[11]方案总开销为 819543，最初的原始方案的总开销为 3226613073. 显然，文献[11]方案对 TPM 的性能要求远远低于最初的原始方案，低 4 个数量级.

(3) 对于 Host，文献[11]方案总开销为 6476791，最初的原始方案的总开销为 5085172621. 显然，文献[11]方案对 Host 的性能要求远远低于最初的原始方案，低 3 个数量级.

(4) 对于 Issuer，文献[11]方案总开销为 82385865，最初的原始方案的总开销为 5878593311. 显然，文献[11]方案对 Host 的性能要求远远低于最初的原始方案，低两个数量级.

(5) 对于 Verifier，文献[11]方案总开销为 4787102，最初的原始方案的总开销为 4478638536. 显然，文献[11]方案对 Verifier 的性能要求远远低于最初的原始方案，低 3 个数量级.

5 NS 方法与其它方法的比较

本节将 NS 方法与“仿真环境模拟测试法”和“符号化运算统计法”进行比较分析.

(1) NS 方法与“仿真环境模拟测试法”相比，具有如下特点：

一方面，NS 方法不受实际测试环境、开发技

术、运行平台、网络环境等方面的影响.因此,应用本方法进行同一方案中不同协议实体之间、不同阶段以及不同方案之间性能的比较和分析要比“仿真环境模拟测试法”更方便和准确.

另一方面,“仿真环境模拟测试法”只需要在理解协议的基础上借助于一些大数运算库和加密库就可以开发实体原型进行仿真测试,理论分析难度要比 NS 方法小.但需要用户熟悉大数运算库和加密库的编程.

(2) NS 方法与“符号化运算统计法”相比,具有如下特点:

NS 方法更有效.NS 方法将所有复杂运算转化和汇总为以机器周期为单位的性能负荷分布和总性能负荷,便于不同阶段、不同实体之间的比较分析.克服了符号化统计法在不同运算间不能直接进行比较的缺点.

NS 方法精细和准确.NS 方法由于深入分析了各主要运算具体算法的内部特点,将各种负责运算归结为单精度乘、单精度加、读内存和写内存等基本操作.这样得出的结果要比符号运算统计法更精细和准确.

NS 方法更适用.NS 方法不仅适合同一方案不同实体之间的性能对比,更合适于不同 DAA 方案之间的性能对比.如第 4 节.

NS 方法理论分析难度高.与“符号化运算统计法”相比,NS 方法不仅需要分析算法中的主要运算,而且还要分析算法的实现过程以及基本运算的次数,所以,NS 方法理论分析难度高于“符号化运算统计法”的理论分析难度.

从以上的分析可以看出,尽管本文是针对原方案提出的性能估算新方法,但该方法也适用于以后相关的其它新方案.特别适合与计算类型较多、方案之间需要精细和准确比较的场合.

6　总　结

性能问题是阻碍 DAA 协议的广泛应用的瓶颈.本文提出的以机器周期为基本性能单位的性能负荷分布测量方法具有重要的意义.该方法需要首先分析 DAA 协议中的各种复杂运算,针对不同的运算选用当前性能较好的算法,然后计算各个算法中大整数单精度乘法、加法、读内存、写内存等基本运算的数目,最后通过汇总并转换得出 DAA 协议中各实体以机器周期为单位的性能负荷分布和总的性能负荷.

我们下一步的工作将在两个方面开展:(1)将该方法应用于其它方案的性能分析以及各方案之间的性能分析与比较;(2)进一步优化 DAA 协议.

参　考　文　献

[1] Brickell E, Camenisch J, Chen L. Direct anonymous attestation//Proceedings of the 11th ACM Conference on Computer and Communications Security. Washington, DC, USA, 2004: 132-145

[2] Camenisch J, Lysyanskaya A. A signature scheme with efficient protocols//Proceedings of the 3rd International Conference on Security in Communication Networks. Amalfi, Italy, 2003: 268-289

[3] Fiat A, Shamir A. How to prove yourself: Practical solutions to identification and signature problems//Proceedings of Advances in Cryptology-CRYPTO'86. London, UK, 1987: 186-194

[4] Chen Xiaofeng, Feng Dengguo. Direct anonymous attestation for next generation TPM. Journal of Computers, 2008, 3(12): 43-50

[5] Backes Michael, Maffei Matteo, Unruh Dominique. Zero-knowledge in the applied pi-calculus and automated verification of the direct anonymous attestation protocol//Proceedings of the IEEE Symposium on Security and Privacy. Oakland, California, USA, 2008: 202-215

[6] Brickell Ernie, Chen Liqun, Li Jiangtao. A new direct anonymous attestation scheme from bilinear maps//Proceedings of the 1st International Conference on Trusted Computing. Villach, Austria, 2008: 166-178

[7] Brickell Ernie, Chen Liqun, Li Jiangtao. Simplified security notions of direct anonymous attestation and a concrete scheme from pairings. International Journal of Information Security, 2009, 8(5): 315-330

[8] Brickell Ernie, Li Jiangtao. Enhanced privacy ID: A direct anonymous attestation scheme with enhanced revocation capabilities//Proceedings of the 6th ACM Workshop on Privacy in the Electronic Society. Alexandria, VA, USA, 2007: 21-30

[9] Brickell Ernie, Li Jiangtao. A pairing-based DAA scheme further reducing TPM resources//Trust and Trustworthy Computing. Lecture Notes in Computer Science 6101. Heidelberg: Springer, 2010: 181-195

[10] Chen Liqun. A DAA scheme requiring less TPM resources//Proceedings of the 5th China International Conference on Information Security and Cryptology. Beijing, China, 2009: 211-219

[11] Chen Liqun, Morrissey Paul, Smart Nigel P. Pairings in trusted computing//Proceedings of the 2nd Internation Conference on Pairing-Based Cryptography. Egham, UK, 2008: 1-17

[12] Menezes A, van Oorsehot P, Vanstone S. Handbook of Applied Cryptography. CRC Press, 1996

[13] Koe C K, Aear T, Kaliski B. Analyzing and Comparing Montgomery multiplication Algorithms. IEEE Micro, 1996, 16(3): 26-33

[14] Comba Paul G. Exponentiation cryptosystems on the IBM PC. IBM System Journal, 1990, 29(4): 526-538

[15] Skakai Y, Sakurai K. Efficient scalar multiplication on elliptic curves with direct computations of several doublings. IEICE Transactions on Fundamentals of Electronics, 2001, E84-A (1): 120-129

[16] Abdulwahed M Ismail, Mohamad Rushdan MD Said, Kamel Ariffin Mohd Atan et al. Bilinear pairing computation using the extended double-base chains algorithm. International Journal of Mathematical Analysis, 2010, 4(19): 929-941

TAN Liang, born in 1972, Ph. D., professor. His research interests include trusted computing, network security.

MENG Wei-Ming, born in 1985, M. S. candidate. His research interest is trusted computing.

ZHOU Ming-Tian, born in 1937, professor. His research interests include network computing, information security.

Background

Performance estimate of Direct Anonymous Attestation Scheme is important problem for trusted Computing, which is the focus of researches in information security field.

Performance problem is a most important problem to Direct Anonymous Attestation Scheme in TCG. The original DAA scheme's computation, is much more complex than the Privacy CA, is based on the modular exponentiations, modular squarings and multiplications, of which the great problem is complex and great time consumption. Due to the limited computing power of the TPM, the complexity of the DAA protocol is not only to the disadvantage of its practical application, but also to seriously hinder development of trusted computing widely and deeply. So the study of a simpler and more efficient DAA protocol is very significant. It is very necessary and important to analyze and measure performance to every entity quantitatively for optimizing DAA.

Currently, performance estimate of Direct Anonymous Attestation Scheme generally takes on "simulation method" or "operator statistics method", but these two methods can't exactly estimate the performance of DAA.

In this paper, a new measurement method, called Normalized Statistics method, which takes the machine period as the basic performance unit, is put forward. When using this method, all complex calculates in DAA protocol must be found out and statistic, and better algorithms to every complex calculate are chosen, moreover, to each algorithms, we need to compute the sum for each basic operation, such as multiplication of single big integer, addition of single big integer, reading and writing memory, and so on. Finally, the every entity performance and the whole performance burden in DAA, whose unit is the machine period, are summed. The theoretical analysis rusults show that the performance estimate is exact, meticulous and effective by this method which is independent of actual platform. For proving availability of the method, we apply it to estimate performance of other one DAA scheme.

This work is support by the National Natural Science Foundation of China under Grant No. 60970113 and Sichuan Funds for Distinguished Young Scientists under Grant No. 0110028. They aim to find out an optimizing DAA scheme to decrease complex and time-consuming of the original DAA scheme, and to improve the practical application of the DAA scheme.