基于 XYZ/E 的 CA 认证系统描述与求精*)

刘俭云[1,2] 张广泉[2,4] 戎 玫[3]

(无锡商业职业技术学院信息工程系 无锡 214063)[1] (苏州大学计算机科学与技术学院 苏州 215006)[2]

(暨南大学深圳旅游学院计算机中心 深圳 518053)[3] (重庆师范大学数学与计算机科学学院 重庆 400047)[4]

摘 要 时序逻辑语言 XYZ/E 在统一的逻辑框架下既能表示静态语义又能表示动态语义,可以实现从抽象描述到可执行程序的平滑过渡。本文建立了 CA 认证系统组件求精模型,对 CA 和 RA 组件用 XYZ/E 进行了描述和求精。

关键词 XYZ/E, CA, RA, 描述, 求精

The Description and Refinement of CA Attestation System Based on XYZ/E

LIU Jian-Yun[1,2] ZHANG Guang-Quan[2,4] RONG Mei[3]

(Wuxi Vocational Institute of Commercial Technology, Information Engineering Department, Wuxi 214063)[1]

(School of Computer Science and Technology, Soochow University, Suzhou 215006)[2]

(Shenzhen Tourism College, Jinan University, Shenzhen 518053)[3]

(School of Mathematics and Computer Science, Chongqing Normal University, Chongqing 400047)[4]

Abstract The temporal logic language (TLL) XYZ/E can represent both dynamic semantics and static semantic under a unified logical framework and carry out the smooth transition from the abstract specification to executable program. This paper builds up the Certification Authority attestation system component refinement model and describes and refine the CA and the RA component using XYZ/E.

Keywords XYZ/E, Certification authority, Registry authority, Describe, Refinement

1 引言

XYZ 系统是由时序逻辑语言 XYZ/E 及一组基于该语言的 CASE 工具集组成,其目标是提高软件可靠性和软件生产率。时序逻辑语言 XYZ/E 是面向软件工程设计的,以适应软件工程的如下领域:图形程序设计、分布式和基于共享存储器的并发程序设计、并发进程与数据模块相结合意义下的面向对象程序设计,在编译的编译、源代码到源代码的转换中起中间语言的作用[1]。它在统一的逻辑框架下既可以表示静态语义又可以表示动态语义,克服了 CSP、Z 等形式化工具只适合描述其一的不足,因此非常适合描述软件体系结构。文[2]中介绍了采用逐步求精方法对 CA 认证系统体系结构整体抽象描述,本文将在它的基础上对该认证系统中的 CA 和 RA 两个组件做进一步详细的描述和求精。

2 时序逻辑语言 XYZ/E

XYZ/E 是基于 Manna-Pnueli 线性时间的时序逻辑语言,它将时序逻辑算子融进程序设计语言之中,使其既是一个逻辑系统又是一个程序设计语言[3]。

2.1 基本语言成分

XYZ/E 的基本语言成分是条件元,有两种形式:

$$LB=y\wedge R=>\$O(v_1,\cdots,v_k)=(e_1,\cdots,e_k)\wedge\$OLB=z \quad (1)$$

$$LB=y\wedge R=>@(Q\wedge\$OLB=z) \quad (2)$$

式中"=>"表示蕴含;R 和 Q 表示一阶逻辑公式,分别称为条件元的条件部分和动作部分;y 和 z 分别表示条件元的定义标号和转出标号;符号@为一阶逻辑算子,可以是下一时刻算子 $O 或最终时刻算子◇。形式(1)条件元定义了程序相邻状态之间的转换关系,用于描述动态语义;形式(2)条件元表示程序抽象规范,用于描述静态语义。

2.2 通信命令

XYZ/E 提供两种通信命令:

输入命令:

$LB_i=y\wedge R=>ChNm?\ x\wedge\$O\ LB_i=w$

输出命令:

$LB_i=y\wedge R=>ChNm!\ z\wedge\$O\ LB_i=w$

其中,"ChNm"是通道名,x 是变量,z 是表达式。"ChNm? x"表示由通道 ChNm 接收信息送入变量 x 中,数据与 x 的类型必须一致;"ChNm! z"表示由通道 ChNm 送出表达式 z 的值,数据与 z 的类型必须一致。

3 CA 认证系统组件求精模型

随着网络的飞速发展,经常需要在大型异种开放网络中不明身份的实体之间进行通信,为了在这种环境中提供机密性、完整性、不可抵赖性和安全性等服务,通常将公开密钥与其拥有者之间的信息绑定到一条电子记录上,称为数字证书[4]。认证系统通常由 CA(Certification Authority)、RA(Registry Authority)和证书库组成。证书机构(CA)是用于证明数字证书本身具有真实性的重要实体,是被一个或者多个用户信任的主体,可以创建、分配和管理公钥证书。CA 组件可以细化成 AL 和 ML 子组件,AL 子组件包括密钥更新和密钥恢复组件;ML 子组件细化为 CA 的初始化、证书签发、证书撤销等组件。注册机构(RA)负责确认主体的身份和验证主体确实有权利拥有在证书申请中所要求的值[5]。RA 组

*)中国科学院计算机科学国家重点实验室开放课题(SYSKF0303)、重庆市教委科学技术研究项目(040803)、国家自然科学基金(60073020)、江苏省高校自然科学研究项目(05KJB520119)。

件由 SQ 和 XZ 子组件组成，其中比较重要的是证书申请和证书下载组件。数字证书库主要基于 LDAP 构建。轻量目录访问协议 LDAP(Lightweight Directory Access Protocol)作为一个开放的、独立于任何厂商的标准，为目前在分布式应用系统和服务中所要求的信息集中存储和管理提供了一个可扩展的结构[6]。组件的求精就是用组件的动态语义代替组件的静态语义，最后过渡到可执行程序。CA 认证系统组件的求精模型如图 1 所示。

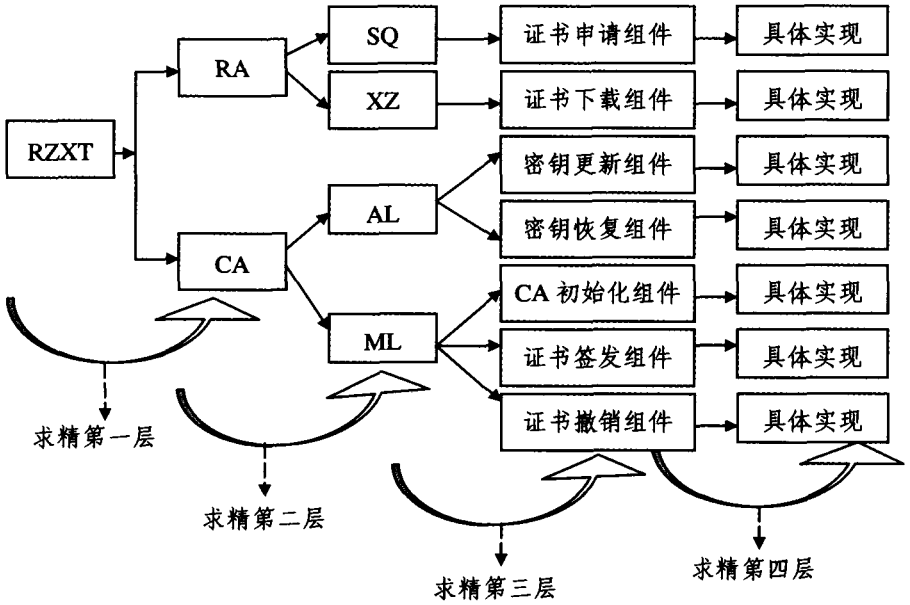

图 1　组件求精模型

两个组件实例 ALComIn$_1$：ALComType$_1$、MLComIn$_2$：MLComType$_2$，表示 ALComType$_1$ 计算规范的单元为 ALComSpec$_1$；MLComType$_2$ 计算规范的单元为 MLComSpec$_2$。

两个连接件实例 AL-CAConIn$_1$：AL-CAConType$_1$、ML-CAConIn$_2$：ML-CAConType$_2$，AL-CAConIn$_x$ 有 r_x 个角色，ML-CAConIn$_y$ 有 r_y 个角色，AL-CA 之间的交互协议为单元 AL-CAConGLUE$_j$，ML-CA 之间的交互协议为单元 ML-RAConGLUE$_i$。

在 CACOM 中的行为规范是一个 XYZ/E 单元：

(‖ $i=1\cdots2$(CAComInSpec$_i$:ComSpec$_i$)) ‖ (‖ $j=1\cdots2$(CAConInGLUE$_j$:ConGLUE$_j$))

连接件 CACON 的规范是一个 XYZ/E 单元：

GLUE ‖ (ALBehavior ‖ MLBehavior)其中在 GLUE 中出现的通道集合为{AL,ML}。

CACOM 的组合声明指示它包含有 ALCOM、MLCOM 两个组件和 AL-CA、ML-CA 两个连接件的实例，描述如下：

```
%CACOMPOSITION==[
                ALComIn1:ComponentName;
                MLComIn2:ComponentName;
                AL-CAConIn1:ConnectorName;
                ML-CAConIn2:ConnectorName;
                ]
```

CACOM 组件的连接定义表明内部体系结构配置，描述如下：

```
%ATTACHMENTS==[
                ALComIn1. PortName ## AL-CACo-
                nIn1. RoleName.
                MLComIn2. PortName ## ML-CA-
                ConIn1. RoleName.
                ]
```

ALClient 和 CAServer 之间的客户-服务器风格描述为：

组件：

```
MLClient==[LB=y'=>Request∧ $OLB=y'1;
           LB= y'1=>Computation1∧ $OLB=z']
CAServer==[LB=w'=>Provide∧ $OLB=v']
```

连接件：MP

```
Sender== [LB=y=>(c ! w∧ $OLB=z )]
Receiver==[LB=y=>(c ? u∧ $OLB=z )]
```

体系结构风格

CS_i == ‖ [ALClient;CAServer]

ALClient 和 CAServer 之间通过报文在网络之间按照 TCP/IP 协议进行通信：

```
%PROS[
CAServer(%INP Request : NM ;%IOP Result1 : NM)==%ALG
[
LB=r'=>Protocol(Request, Request1)∧ $OLB=r'1;
LB= r'1=>MP. sender(Request1)∧ $OLB= r'2;
LB= r'2=>MP. receiver(Result1)∧ $OLB= r'3;
LB= r'3=> $OLB=STOP
];
ALClient(%INP Request1:NM ;%IOP Result1:NM)==%ALG
[
LB=p'=> MP. receiver (Request1)∧ $OLB=p'1;
LB= p'1=>Computation2∧ $OLB= p'2;
LB= p'2=> Protocol (Result,Result1)∧ $OLB= p'3;
LB= p'3=> MP. sender (Result1)∧ $OLB= p'4;
LB=p'4=> $OLB=STOP
]
]
```

· MLCOM 包含有 CA 初始化(CACSH)、证书签发(ZSQF)、CRL 发布(CRLFB)、交叉认证(JCRZ)、证书撤销(ZSCX)等子组件。

MLCOM 定义了 kNum 个端口 CLCPort$_k$ ($k=1\cdots k$Num)。

5 个组件实例 CACSHComIn$_1$：CACSHComType$_1$、ZSQFComIn$_2$：ZSQFComType$_2$、CRLFBComIn$_3$：CRLFBComType$_3$、JCRZZComIn$_4$：JCRZComType$_4$、ZSCXComIn$_5$：ZSCXComType$_5$。

表示 CACSHComType$_1$ 计算规范的单元为 CACSHComSpec$_1$；ZSQFComType$_2$ 计算规范的单元为 ZSQFCom-

$Spec_2$；CRLFBComType$_3$ 计算规范的单元为 CRLFBComSpec$_3$；JCRZComType$_4$ 计算规范的单元为 JCRZComSpec$_4$；ZSCXComType$_5$ 计算规范的单元为 ZSCXComSpec$_5$。

5 个连接件实例 CACSH-MLConIn$_1$：CACSH-MLConType$_1$、ZSQF-MLConIn$_2$：ZSQF-MLConType$_2$，CRLFB-MLConIn$_3$：CRLFB-MLConType$_3$、JCRZ-MLConIn$_4$：JCRZ-MLConType$_4$、ZSCX-MLConIn$_5$：ZSCX-MLConType$_5$。

CACSH-MLConIn$_a$ 有 u_a 个角色，ZSQF-MLConIn$_b$ 有 u_b 个角色，CRLFB-MLConIn$_c$ 有 u_c 个角色，JCRZ-MLConIn$_d$ 有 u_d 个角色，ZSCX-MLConIn$_e$ 有 u_e 个角色。

CACSH-ML 之间的交互协议为单元 CACSH-MLConGLUE$_a$，ZSQF-ML 之间的交互协议为单元 ZSQF-MLConGLUE$_b$，JCRLFB-ML 之间的交互协议为单元 CRLFB-MLConGLUE$_c$，JCRZ-ML 之间的交互协议为单元 JCRZ-MLConGLUE$_d$，ZSCX-ML 之间的交互协议为单元 ZSCX-MLConGLUE$_e$。

在 MLCOM 中的行为规范是一个 XYZ/E 单元：

(‖ $i=1\cdots5$ (ComInSpec$_i$: ComSpec$_i$)) ‖ (‖ $j=1\cdots5$ (ConInGLUE$_j$: ConGLUE$_j$))

连接件的规范是一个 XYZ/E 单元：

GLUE ‖ (CACSH-MLBehavior ‖ ZSQF-MLBehavior ‖ CRLFB-MLBehavior ‖ JCRZ-ML ‖ ZSCX-MLBehavior)，其中在 GLUE 中出现的通道集合为{ CACSH-ML，ZSQF-ML，CRLFB-ML，JCRZ-ML，ZSCX-ML}。

MLCOM 的组合声明指示它包含有 CACSH、ZSQF、CRLFB、JCRZ、ZSCX 5 个组件和 CACSH-ML、ZSQF-ML、CRLFB-ML、JCRZ-ML、ZSCX-ML5 个连接件的实例，描述如下：

```
%MLCOMPOSITION==[
                    CACSHComIn1 : ComponentName;
                    ZSQFComIn2 : ComponentName;
                    CRLFBComIn3 : ComponentName;
                    JCRZComIn4 : ComponentName;
                    ZSCXComIn5 : ComponentName;
                    CACSH-MLConIn1 : ConnectorName;
                    ZSQF-MLConIn2 : ConnectorName;
                    CRLFB-MLConIn3 : ConnectorName;
                    JCRZ-MLConIn4 : ConnectorName;
                    ZSCX-MLConIn5 : ConnectorName;]
```

MLCOM 组件的连接定义表明内部体系结构配置，描述如下：

```
%ATTACHMENTS==[
                    CACSHComIn1. PortName # # CACSH-
                    MLConIn1. RoleName.
                    ZSQFComIn2. PortName # # ZSQF-ML-
                    ConIn2. RoleName.
                    CRLFBComIn3. PortName # # CRLFB-
                    MLConIn3. RoleName.
                    ZJCRZComIn4. PortName # # JCRZ-ML-
                    ConIn4. RoleName.
                    ZSCXComIn5. PortName # # ZSCX-ML-
                    ConIn5. RoleName. ]
```

· 密钥更新组件

用户通过子 RA 服务器提出密钥更换申请，CA 按申请证书的流程为用户生成新的密钥对和新的证书；接着将用户的旧私钥从密钥表中删除，添加到密钥备份表中；将用户的旧证书从证书列表中删除，添加到撤销证书列表中；随后将旧证书放在 CRL 中予以废除；然后将新私钥加密后写入到密钥库的密钥列表中；将新证书保存在证书库的证书表中；最后将新证书的一个副本发布到 LDAP 目录服务器。

```
%PROS[
  %PROSKeyrenew(%CHN ASBN p1( * ,pout:NM):STRING;
                %CHN ASBN p2(pin:NM, * ):STRING)
  ==[
    %ALG[
      % LOC [key_renew_message, new_prikey, new_ca:
             STRING;
             krm,cacopy,cano,pubkey,prikey:STRING;
             dm,am,dk,ak:STRING]
             LB=START=〉$OLB=b1;
             LB=b1=〉$O krm=0∧$OLB=b2;
             LB=b2∧p2 ? =〉$OLB=b3;
             LB=b2∧¬p2 ? =〉$OLB=b1;
             LB=b3=〉$O p2 ? krm∧$OLB=b4;
             LB=b4=〉$O key_renew_message=krm∧
             $OLB=b5;
             LB=b5=〉$O dm=1∧$OLB=b6;
             LB=b6=〉$O dm=dm+krm∧$OLB=b7;
             LB=b7=〉$O p1 ! dm∧$OLB=b8;
             LB=b8=〉$O am=1∧$OLB=b9;
             LB=b9=〉$O am=am+krm∧$OLB=b10;
             LB=b10=〉$O p1 ! am∧$OLB=b11;
             LB=b11=〉$O dk=1∧$OLB=b12;
             LB=b12=〉$O dk=dk+ krm∧$OLB=b13;
             LB=b13=〉$O p1 ! dk∧$OLB=b14;
             LB=b14=〉$O ak=1∧$OLB=b15;
             LB=b15=〉$O ak=ak+ krm∧$OLB=b16;
             LB=b16=〉$O p1 ! ak∧$OLB=b17;
             LB=b17=〉create_key(%iop pubkey|pu; %iop
             prikey|pr);
             LB=b18=〉$O new_prikey=prikey∧$OLB
             =b19;
             LB=b19=〉encrpt_key(%iop pribey|epri);
             LB=b20=〉$O p1 ! prikey∧$OLB=b21;
             LB=b21=〉create_certi(%iop new_ca|cc);
             LB=b22=〉$O new_ca=new_ca+new_
             prikey∧$OLB=b23;
             LB=b23=〉$O cano=get_certino(new_ca)∧
             $OLB=b24;
             LB=b24=〉$O cacopy=copy_certi(new_ca)
             ∧$OLB=b25;
             LB=b25=〉$O p1 ! new_ca∧$OLB=b26;
             LB=b26=〉$O p1 ! cacopy∧$OLB=b27;
             LB=b27=〉$OLB=b1;
    ]
  ]
]
```

5 RA 组件描述与求精

· RA 组件是一个复合组件，由 SQCOM 和 XZCOM 合成。

RA 组件中的主要功能用户注册、证书申请、证书查询、证书下载和 CRL 下载等将分别由这两个子组件完成。

RA 组件的功能规范描述如下：

两个组件实例 SQComIn$_1$：SQComType$_1$、XZComIn$_2$：XZComType$_2$，表示 SQComType$_1$ 计算规范的单元为 SQComSpec$_1$；XZComType$_2$ 计算规范的单元为 XZComSpec$_2$。

两个连接件实例 SQ-RAConIn$_1$：SQ-RAConType$_1$、XZ-RAConIn$_2$：XZ-RAConType$_2$，SQ-RAConIn$_j$ 有 r_j 个角色，XZ-RAConIn$_i$ 有 r_i 个角色，XZ-RA 之间的交互协议为单元 XZ-RAConGLUE$_j$，SQ-RA 之间的交互协议为单元 SQ-RAConGLUE$_i$。

在 RACOM 中的行为规范是一个 XYZ/E 单元：

(‖ i=1…2(RAComInSpec$_i$: ComSpec$_i$)) ‖ (‖ $j=1\cdots2$ (RAConInGLUE$_j$: ConGLUE$_j$))

连接件 RACON 的规范是一个 XYZ/E 单元：

GLUE ||(SQBehavior || XZBehavior)其中在 GLUE 中出现的通道集合为{SQ，XZ }。

RACOM 的组合声明指示它包含有 SQCOM、XZCOM 两个组件和 SQ-RA、XZ-RA 两个连接件的实例，描述如下：

```
%RACOMPOSITION==[
                    SQComIn1 : ComponentName;
                    XZComIn2 : ComponentName;
                    SQ-RAConIn1 : ConnectorName;
                    XZ-RAConIn2 : ConnectorName;
                    ]
```

RACOM 组件的连接定义表明内部体系结构配置，描述

如下：

```
%ATTACHMENTS==[
                SQComIn1. PortName # # SQ-RAConIn1. RoleName.
                XZComIn2. PortName # # XZ-RAConIn1. RoleName.
                ]
```

SQClient 和 RAServer 之间的客户-服务器风格描述为：

组件：

$SQClient==[LB=y''\Rightarrow Request \wedge \$OLB=y''_1;$
$LB=y''_1\Rightarrow Computation_2 \wedge \$OLB=z'']$
$RAServer==[LB=w''\Rightarrow Provide \wedge \$OLB=v'']$

连接件：MP

$Sender==[LB=y\Rightarrow(c!w \wedge \$OLB=z)]$
$Receiver==[LB=y\Rightarrow(c?u \wedge \$OLB=z)]$

体系结构风格

$CS_2==\|[SQClient;RAServer]$

·中间过程与 CA 相近，因篇幅有限，详细代码在此就不列出了。

·证书申请组件

用户填写申请人信息及其 PIN 值，提交到子 RA 服务器。子 RA 服务器将申请人信息保存在证书申请库的证书申请表中，并将其状态值标记为“未审核”。RA 管理员审核申请人真实身份，若提交的申请人信息与其真实物理身份相符，则审核通过；否则审核不能通过。若用户的证书申请审核通过，则 RA 将证书申请表中该申请人状态值更新为“已审核”，将该申请转交给安全服务器。经过安全检查后，转交给系统的 RA 服务器，等待 CA 发放证书；反之，如果未通过审核，则子 RA 服务器将该申请人的信息从证书申请表中删除，并且将其对应的用户信息从用户表中删除。最后，子 RA 服务器将审核结果以电子邮件的方式发送给用户。

```
%PROS[
  %PROSCAapplication (% CHN ASYN m1 ( * , chout: NM): STRING;
                      % CHN ASYN m2 ( chin: NM, * ): STRING)
    ==[
      %ALG[
        %LOC[client_message,exam_value:STRING;
             tellmessage,delmessage,cm,ev,sh:STRING;
             ]
        LB=START=>$OLB=y1;
        LB=y1=>$O cm=0∧$OLB=y2;
        LB=y2∧m2 ? =>$OLB=y3;
        LB=y2∧¬m2 ? =>$OLB=y1;
        LB=y3=>$O m2 ? cm∧$OLB=y4;
        LB=y4=>$O client_message=cm∧$OLB=y5;
        LB=y5=>$O exam_value=0∧$OLB=y6;
        LB=y6=>$O cm=cm+exam_value∧$OLB=y7;
        LB=y7=>$O client_message=cm∧$OLB=y8;
        LB=y8=>$O m1 ! cm∧$OLB=y9;
        LB=y9=>$O ev=exam_vale∧$OLB=y10;
        LB=y10∧(sh=1)=>$OLB=y11;
        LB=y10∧(sh=0)=>$OLB=y15;
        LB=y11=>$O exam_value=1∧$OLB=y12;
        LB=y12=>$O cm=cm+exam_value∧$OLB=y13;
        LB=y13=>$O client_message=cm∧$OLB=y14;
        LB=y14=>$O m1 ! cm∧$OLB=y1;
        LB=y15=>$O tellmessage=1∧$OLB=y16;
        LB=y16=>$O tellmessage=tellmessage+sh∧$OLB=y17;
        LB=y17=>$O m1 ! tellmessage∧$OLB=y18;
        LB=y18=>$O delmessage=1∧$OLB=y19;
        LB=y19=>$() delmessage=delmessage+cm∧$OLB=y20;
        LB=y20=>$O m1 ! delmessage∧$OLB=y21;
        LB=y21=>$OLB=y1;
            ]
      ]
  ]
```

结束语 本文以时序逻辑语言 XYZ/E 对 CA 认证系统中两个关键组件，即认证机构 CA 和注册机构 RA 进行了描述和求精。整个过程在统一的时序逻辑框架下进行，保证了系统的语义一致性，从而提升了 CA 认证软件开发的可靠性。同时，引入组件技术，便于系统分阶段开发和未来的功能扩展。用时序逻辑语言 XYZ/E 对 CA 认证系统的软件体系结构进行描述和求精有利于对整个系统的分析、演化、管理。因此，下一步将在此基础上对已经完成的描述内容进行分析和验证等工作。

参 考 文 献

1 唐稚松. 时序逻辑程序设计与软件工程[M]. 北京：科学出版社，2002

2 刘俭云，戎玫，张广泉. XYZ/E 在 CA 认证系统中的一个初步应用[J]. 计算机工程与应用(已录用)

3 Moszko B. Executing Temporal Logic Programs. Cambridge University Press, 1986

4 汪立东，余祥湛，方滨兴. PKI 中几个安全问题的研究[J]. 计算机工程，2000，26(1)：14～15

5 陈鲁川，李善平. Linux 平台下公钥基础设施(PKI)的研究[J]. 计算机应用研究，2003，20(2)：93～94

6 张辉，杨岳湘，汪诗林. 数字校园中基于 LDAP 的统一用户身份管理技术研究[J]. 计算机工程与科学，2005，17(1)：14～16

(上接第 112 页)

10 A. Joux. A one round protocol for tripartite Diffie-Hellman. In: Bosma W, eds. Proceedings of Algorithmic Number Theory Symposium - ANTS IV. Springer-Verlag, 1987, 1838: 385～394

11 Sakai R, Ohgishi K, Kasahara M. Cryptosystems based on pairing. The 2000 Symposium on Cryptography and Information Security, Okinawa, Japan, January 2000

12 Boneh D Franklin M. Identity-based encryption from the Weil pairing. In: Kilian J, eds. Advances in Cryptology-CRYPTO 2001. Springer-Verlag, 2001, 2139: 213～229

13 Cha J C, Cheon J H. An identity-based signature from gap Diffie-Hellman groups. In: Desmedt Y, editor. Public Key Cryptography - PKC 2003. Lecture Notes in Computer Science. Springer - Verlag, 2002, 2567: 18～30

14 Zheng Y. Digital signcryption or how to achieve cost(signature & encryption)<< cost(signature) + cost(encryption). In: Kalishi B S Jr. eds. Advances in Cryptology - CRYPTO 1997. Lecture Notes in Computer Science. Springer - Verlag, 1997, 1294: 165～179

15 Malone-Lee J. Identity-based signcryption. Cryptology ePrint Archive: [Report 2002/098]. 2002. http://eprint.iacr.org/

16 Libert B, Quisquater J-J. Efficient signcryption with key privacy from gap diffie-hellman groups. In: F. Bao, R. H. Deng, and J. Zhou, eds. International Workshop on Practice and Theory in Public Key Crytography - PKG 2004. Lecture Notes in Computer Science. Springer - Verlag, 2004, 2947: 187～200

17 Lynn B. Authenticated identity-based encryption. Cryptology ePrint Archive: [Report 2002/072]. 2002. http://eprint.iacr.org/

18 Nalla D, Reddy K C. Signcryption scheme for identity-based cryptosystems. Cryptology ePrint Archive: [Report 2003/066]. 2003. http://eprint.iacr.org/

19 Malone-Lee J. Identity-based signcryption. Cryptology ePrint Archive: [Report 2002/098]. 2002. http://eprint.iacr.org/

20 Bellare M, Rogaway P. Random oracles are practical: A paradigm for designing efficient protocols. In ACM Conference on Computer and Communications Security, ACM, 1993. 62～73

21 Chen Liqun, Malone-Lee J. Improved Identity-Based Signcryption. Public Key Cryptography(PKC), Lecture Notes on Computer Science Springer-Verlag, 2005, 3386: 362～379