Journal of Computer Applications
计算机应用,2013,33(11):3247-3251
ISSN 1001-9081
CODEN JYIIDU
2013-11-01
http://www.joca.cn

文章编号:1001-9081(2013)11-3247-05
doi:10.11772/j.issn.1001-9081.2013.11.3247

关于秘密共享方案在应用 Pi 演算中的实现

徐　军*
(山东理工大学 计算机科学与技术学院,山东 淄博 255049)
(*通信作者电子邮箱 Xujun@sdut.edu.cn)

摘　要:针对秘密共享方案的自动化验证问题,提出一种基于等值理论的秘密共享方案自动化验证方法。首先通过等值理论在应用 Pi 演算中对可验证的多秘密共享方案的密码学语义进行了形式化定义。在此基础上,进一步提出了一种用于将所提出等值理论转化为自动化协议验证器 ProVerif 中重写机制的编码方法,在 ProVerif 中实现了关于可验证的多秘密共享方案的自动化验证。通过证明给出了关于可验证的多秘密共享方案形式化分析结果的健壮性结论:如果自动化协议验证器 ProVerif 中可验证的多秘密共享方案的形式化分析结果满足特定安全属性,则其能够归约证明应用 Pi 演算模型中针对可验证的多秘密共享方案所建立的现实敌手可以"模拟"ProVerif 验证器中的理想敌手,其意味着现实敌手与理想敌手是不可区分的。

关键词:Pi 演算;秘密共享;形式化分析;协议验证
中图分类号:TN95　　**文献标志码**:A

Automatic verification technique for secret-sharing schemes in applied Pi-calculus

XU Jun*
(*School of Computer Science and Technology, Shandong University of Technology, Zibo Shandong* 255049, *China*)

Abstract: In this paper, an abstraction of secret-sharing schemes that is accessible to a fully mechanized analysis was given. This abstraction was formalized within the applied Pi-calculus by using an equational theory that characterized the cryptographic semantics of secret share. Based on that, an encoding method from the equational theory into a convergent rewriting system was presented, which was suitable for the automated protocol verifier ProVerif. Finally, the first general soundness result for verifiable multi-secret sharing schemes was concluded: for the multi-secret sharing schemes satisfying the specified security criterion in ProVerif, the realistic adversaries modeled on multi-secret sharing schemes in Pi-calculus can simulate the ideal adversaries in verifier ProVerif, which means that realistic adversaries and ideal adversaries are indistinguishable.

Key words: Pi-calculus; secret sharing; formal analysis; protocol verification

0　引言

安全协议是保证信息系统安全的重要手段之一。而分析安全协议中存在的缺陷却是一个非常困难的问题。目前,安全协议的安全性证明主要包括基于逻辑的形式化证明方法、基于随机预言机模型的证明方法、基于标准模型的证明方法以及基于零知识证明理论的证明方法。与其他方法相比,安全协议的形式化分析技术具有以下优点:

1)能够更准确地界定安全协议的边界,即安全协议与其运行环境的界面;

2)能够更准确地定义安全协议参与者的行为;

3)能够更准确地定义安全协议的安全属性;

4)能够证明安全协议满足其安全属性,以及证明安全协议在什么条件下才能满足其安全属性。

当前安全协议形式分析技术面对的主要挑战之一是如何对复杂密码操作进行形式化定义。传统的安全协议分析技术只对包括加密和数字签名在内的基本密码操作进行了形式化定义,此类基本密码操作仅仅具备认证性与秘密性的安全属性。而现代安全协议中出现了许多新的密码操作,此类密码操作具备了比基本密码操作更为复杂的安全属性。秘密共享就属于这样一类密码操作。

1994 年,Dawson 等[1]提出了一种多秘密共享(Multi-Secret Sharing, MSS)方案,在这个方案中,多个秘密可以在同一秘密共享方案执行过程中被共享。2004 年,Yang 等[2]提出了一种新的多秘密共享方案(Yang-Chang-Hwang, YCH),此方案通过采用单向映射函数实现了多秘密的并行构建。2005 年,Shao 等[3]基于 YCH 提出了一种可验证的多秘密共享(Verifiable Multi-Secret Sharing, VMSS)方案。2006 年,Zhao 等[4]基于 YCH 提出了一种实用的可验证多秘密共享方案。秘密共享方案的上述这些独特的安全属性及其有效实现使得其在现代安全协议中被广泛应用。

对于门限值为(l,t)的秘密共享方案,其参与者包括秘密委托者 1,2,…,l 与一个可信任的秘密分发者。秘密共享方案包括关于秘密份额的分割算法、验证算法与合并算法。在秘密分发阶段,秘密分发者通过输入秘密分发者参数产生秘密份额密钥 $SK_1,SK_2,\cdots,SK_l$ 与秘密份额验证公钥 $VK_1,VK_2,\cdots,VK_l$。之后,秘密分发者或者秘密委托者 i 可以通过输入秘密委托者参数以及由秘密分发者分配的秘密份额密钥 SK_i,生成秘密份额 D_i。秘密份额的验证算法则通过输入秘密份额 D_i 和秘密份额验证公钥 VK_i 来验证秘密份额 D_i 是否有效。秘密份

收稿日期:2013-06-04;**修回日期**:2013-07-10。
作者简介:徐军(1967-),男,山东淄博人,副教授,硕士,主要研究方向:信息安全、并行计算。

额的合并算法则通过输入t个有效的秘密份额生成秘密D。其中秘密D是通过对秘密分发者参数与秘密委托者参数进行基本密码操作而得到的信息项序列。秘密共享方案的基本安全属性表现为：

1）通过t个或者更多个秘密份额D_i可以计算得到秘密D(正确性)；

2）通过$t-1$个或者更少个秘密份额D_i计算得到秘密D是不可行的(健壮性)。

由于可验证的多秘密共享方案本身的复杂性，对其所有安全属性进行形式化定义非常困难，Yew等[5]在Coq工具中提出过关于一般秘密共享方案的形式化定义，但其仍然无法实现对可验证的多秘密共享方案进行形式化建模。

因此，本文在应用Pi演算中实现了关于可验证的多秘密共享方案相关密码操作的形式化定义，同时，利用等值理论对可验证的多秘密共享方案的密码学语义进行了描述。在此基础上，本文提出了一种用于将上述等值理论转化为自动化验证器ProVerif中重写机制的编码方法，并且，最终在ProVerif中实现了可验证的多秘密共享方案的形式化验证。

1 应用Pi演算与等值理论

应用Pi演算[6-9]是当前主要的安全协议形式化分析技术，其成功地实现了关于安全协议抗拒绝服务性、秘密性、认证性、不可否认性等安全属性的验证。基于应用Pi演算的ProVerif安全协议自动化验证器是当前主要的安全协议自动化验证工具。

2008年，Backes等[10-11]利用Pi演算给出了第一个关于零知识安全协议相关密码操作的形式化定义，同时，其在ProVerif中实现了零知识安全协议的自动化分析与验证。2010年，Backes等[12]利用Pi演算给出了第一个关于安全多方计算协议相关密码操作的形式化定义。

在Pi演算中，信息项通过函数集合Σ进行定义，信息项集合T_Σ是由标识、变量和Σ中函数生成的代数系统。信息项之间存在等值关系，所有关于信息项的等值关系则组成了关于信息项的等值理论E。$E\vdash M=N$与$E\vdash M\neq N$分别表示，根据等值理论E能够推导出信息项M与N之间存在相等或不相等关系。

$\mu(x)$关于进程的语法定义如下。空进程0不包含任何事件；限制语句$\nu n.P$表示产生新鲜标识n，然后执行进程P；分支语句if $M=N$ then P else Q表示：如果$E\vdash M=N$，则执行进程P，否则，执行进程Q；输入语句$\mu(x).P$表示：首先利用信道接收信息项N，然后执行进程$P\{N/x\}$；输出语句$\overline{\mu}(N).P$表示：首先利用信道μ输出信息项N，然后执行进程P；并行语句$P\mid Q$表示并行执行进程P和Q，复制语句$!P$表示并行执行任意多个P进程。

$\nu x.(P\mid\{M/x\})$扩展进程则在一般进程定义的基础上加入了实时替换的概念。其中，实时替换$\{M/x\}$可以作用到其接触到的任意进程。另外，为了控制实时替换的作用范围，可以加入关于变量x的限制，例如，$\nu x.(P\mid\{M/x\})$表示把实时替换$\{M/x\}$的作用范围限制在进程P内。如果不对变量x做出限制，则实时替换可以被进程输出，因此，代表敌手的进程运行环境能够立刻获取信息项M。

标识和变量的作用范围由限制语句和输入语句来决定。$fv(A)$和$fn(A)$分别用来表示扩展进程A中的自由变量和自由标识；$bv(A)$和$bn(A)$分别用来表示扩展进程A中的被限制变量和被限制标识。同时，$free(A):=fv(A)\cup fn(A)$，$bound(A):=bv(A)\cup bn(A)$。

对于序列$\tilde{M}=M_1,\cdots,M_k$与$\tilde{x}=x_1,\cdots,x_k$，$\{\tilde{M}/\tilde{x}\}$表示$\{M_1/x_1\}\mid\cdots\mid\{M_k/x_k\}$。根据扩展进程的定义，对于每一变量，扩展进程最多只能包含一个关于此变量的实时替换，而对于被限制的变量，扩展进程则应该恰好包含一个关于此变量的实时替换。

上下文是一个包含有未知项的进程，其用于表示特定的进程运行环境。如果上下文中不存在包含有未知项的复制语句、分支语句、输入语句、输出语句，则其被称为评估上下文。对于进程A，上下文C，$C[A]$表示通过将A代替C中的未知项而形成的新进程。如果$C[A]$是封闭的，则A在$C[_]$中是封闭的。

框架是一个由空进程0和实时替换构建起来的进程。对于框架φ，$dom(\varphi)$代表φ的域，其包含所输出的所有变量，例如，如果φ中包含实时替换$\{M/x\}$，并且，φ中不存在关于变量x的限制，则x被包含在$dom(\varphi)$之中。另外，通过把进程A中的每一个普通进程用0代替，A可以被映射到框架$dom(A)$，其可以用来表示进程A暴露给外部运行环境的静态知识，但其并不能用来表示进程A的动态行为。

应用Pi演算的语义是通过结构等价关系($\equiv$)与内部推演关系($\rightarrow$)来定义的。其中，结构等价关系表示进程之间存在的句法重组关系，其最小包含下列规则：

$A_1\mid(A_2\mid A_3)\equiv(A_1\mid A_2)\mid A_3$	PAR-A
$A_1\mid A_2\equiv A_2\mid A_1$	PAR-C
$!P\equiv P\mid!P$	REPL
$\nu n.0\equiv 0$	RES-0
$\nu u.\nu u'.A\equiv\nu u'.\nu u.A$	RES-C
$A_1\mid\nu u.A_2\equiv\nu u.(A_1\mid A_2)\ u\notin free(A_1)$	RES-PAR
$\nu x.\{M/x\}\equiv 0$	ALLAS
$\{M/x\}\mid A\equiv\{M/x\}\mid A\{M/x\}$	SUBST
$\{M/x\}\equiv\{N/x\}$ if $E\vdash M=N$	REWRITE

并且在α-转换与评估上下文的应用中，其是封闭的。

内部推演关系则表示扩展进程的语义。其最小包含下列规则：

$\overline{a}(x).P\mid a(x).Q\rightarrow P\mid Q$

IF $E\vdash M=N$ THEN if $M=N$ then P else $Q\rightarrow P$

IF $E\vdash M\neq N$ THEN if $M=N$ then P else $Q\rightarrow Q$

并且在α-转换与评估上下文的应用中，其是封闭的。

接下来给出关于进程间观察等同性($\approx$)的定义。观察等同性用来表示扩展进程之间存在的动态等价关系。首先给出以下定义：$A\Downarrow\mu$表示进程A通过信道μ发送消息，例如，$A\rightarrow*C[\overline{\mu}(M).P]$，其中，在评估上下文$C[_]$中，$\mu$是被限制变量或标识。

定义1 如果扩展进程A，B之间具备观察等同性($\approx$)，其表示为$A\approx B$，则A，B具有相同的域，并且，其满足下列条件：

1）如果$A\Downarrow\mu$，则$B\Downarrow\mu$；

2）如果$A\rightarrow*A'$，则存在B'，其满足$B\rightarrow*B'$，并且$A'\approx B'$；

3）对于任意封闭的评估上下文$C[_]$，其满足$C[A]\approx$

$C[B]$。

2　关于可验证的多秘密共享方案的等值理论

2.1　基本等值理论

本文所基于的基本等值理论 E_{base} 如下所示:函数 *ntuples* 表示信息项合并操作;函数 enc_{sym}、dec_{sym}、enc_{asym}、dec_{asym} 分别表示信息项对称加、解密操作,以及公钥加、解密操作;函数 *sign* 与 *check* 分别表示数字签名及验证操作;*pk* 为公钥构造函数;*h* 为哈希函数。上述基本等值理论已经在应用 Pi 演算中被充分检验,并且,其已经被成功应用于相关协议的形式化分析。另外,基本等值理论包含 3 个二元函数 *eq*、∧ 和 ∨,其分别用于对相等性测试以及与、或逻辑运算进行建模。因此,应用 Pi 演算实现了关于布尔逻辑表达式的相关建模。

$x_2 ith_n(ntuples(x_1 x_n)) = x_i$ 基本等值理论 E_{base} 包括如下关于 x,y,z 的等值规则:

$dec_{\text{sym}}(enc_{\text{sym}}(x,y),y) = x$

$dec_{\text{asym}}(enc_{\text{asym}}(x,pk(y)),y) = x$

$check(sign(x,y),x,pk(y)) = \text{true}$

$eq(x,x) = \text{true}$

$\wedge(\text{true},\text{true}) = \text{true}$

$\vee(\text{true},x) = \text{true}$

$\vee(x,\text{true}) = \text{true}$

2.2　可验证的多秘密共享方案实例

接下来本文将结合可验证的多秘密共享方案实例 (l,t)-门限签名方案[13] 对其所提出的关于秘密共享方案的一般性自动化验证方法进行阐述。(l,t)-门限签名方案实现秘密委托者 $ID_1,ID_2,\cdots,ID_l$ 中的 t 个能够合作生成秘密分发者签发的关于信息 M 的有效数字签名;但是,少于 t 个参与者则不能合作生成秘密分发者签发的关于信息 M 的有效数字签名。

(l,t)-门限签名方案的应用描述如下。

1) 秘密份额的生成。

① 秘密分发者随机选取整数 p,q,其中:$p = 2p' + 1$,$q = 2q' + 1$。然后,秘密分发者计算 $n = pq$,$m = p'q'$。

② 秘密分发者随机选取整数 $e > l$,发布其公钥为 (n,e)。同时,秘密分发者计算得到其用于签名的私钥 d,其满足 $de \equiv 1 \bmod m$。

③ 秘密分发者选取整数 $a_0 = d$。同时,其从 $\{1,2,\cdots,m-1\}$ 中分别随机选取整数 $a_1,a_2,\cdots,a_{k-1}$,并且,建立多项式函数 $f(x) = \sum_{i=0}^{k-1} a_i X^i$。

④ 秘密分发者计算得到秘密份额密钥 $sk_i = f(i) \bmod m$,并且,其分别将 sk_i 分发给秘密委托者 ID_i。

⑤ 秘密分发者从 $\{1,2,\cdots,n-1\}$ 中随机选取整数 vk,分别计算得到 $vk_i = vk^{sk_i}$,并且,秘密分发者发布其秘密份额验证公钥为 (vk,vk_i)。

⑥ 秘密委托者 ID_i 通过秘密份额密钥 sk_i 计算得到秘密份额 (x,x_i),其中,$x = H(M)$,$x_i = (H(M))^{2l!sk_i}$,H 为哈希函数。

2) 秘密份额的验证。

① 秘密委托者 ID_i 从 $\{0,1,\cdots,2^{L(n)+2L_1}-1\}$ 中随即选取整数 r,计算 $vk' = vk^r$,$x' = (H(M))^{4l!r}$,$z = sk_i + r$,$c = H'(vk,(H(M))^{4l!},vk_i,x_i^2,vk',x')$,其中:$H'$ 为哈希函数,$L(n)$ 为整数 n 的二进制长度,L_1 为 H' 输出数据的二进制长度。然后,秘密秘密委托者发送信息 (c,z) 至秘密份额验证者。

② 秘密份额验证者接收到信息 (c,z) 后,其判断是否满足 $c = H'(vk,(H(M))^{4l!},vk_i,x_i^2,vk^z vk_i^{-c},(H(M))^{4l!z}x_i^{-2c})$。如果满足,则验证成功。

3) 秘密份额的合并。

① 秘密份额合并者获得集合 S 中所有秘密委托者所持有的秘密份额,其中,$S = \{ID_{i_1},ID_{i_2},\cdots,ID_{i_t}\}$,并且,$\{i_1,i_2,\cdots,i_t\} \subseteq \{1,2,\cdots,l\}$。

② 秘密份额合并者分别计算 $\lambda_{ij} = l! \dfrac{\prod_{i_j' \in \{i_1,i_2,\cdots,i_t\}\setminus\{i_j\}} (i_j')}{\prod_{i_j' \in \{i_1,i_2,\cdots,i_t\}\setminus\{i_j\}} (i_j' - i_j)}$,$w = \prod_{j=1}^{t} (x_{i_j})^{2\lambda i_j}$,同时,秘密份额合并者计算得到整数 a,b,其满足 $4(l!)^2 a + eb = 1$。最后,秘密份额合并者可以计算得到 $w^a(H(M))^b$,其满足 $w^a(H(M))^b = (H(M))^d$。至此,秘密份额合并者获得秘密,即秘密分发者所签发的关于信息 M 的有效数字签名 $(H(M))^d$。

2.3　关于可验证的多秘密共享方案的等值理论

本文提出的关于可验证的多秘密共享方案的等值理论 E_{SS} 如下所示。

在等值理论 E_{SS} 中,$SSP_{l,t}(\tau)$ 表示门限值为 (l,t) 的秘密共享过程。其中,标识 τ 用于识别特定的秘密共享过程。$SSP_{l,t}(\tau)$ 可以被简写为 $\tau_{l,t}$。

$SSK_{i,j,k}(\overline{M},m,\tau_{l,t},\overline{F})$ 表示特定秘密共享过程中的秘密份额密钥,其中,$\overline{M}$ 代表秘密分发者参数,其表示序列 $M_1,\cdots,M_i$;整数 m 称为密钥标识,其用于识别同一秘密共享过程中的不同秘密份额密钥,并且其满足 $m \leqslant l$。

$SS_{i,j,k}(\overline{N},SSK_{i,j,k}(\overline{M},m,\tau_{l,t},\overline{F}),\overline{F})$ 表示通过秘密份额密钥生成的秘密份额,其中:$\overline{N}$ 代表秘密委托者参数,其表示序列 $N_1,\cdots,N_j$;$\overline{F}$ 表示 (i,j)-函数序列 $F_1,\cdots,F_k$。由上述定义可以得出,$SSK_{i,j,k}$ 函数的参数个数为 $i+k+2$,$SS_{i,j,k}$ 函数的参数个数为 $j+k+1$。接下来给出关于 (i,j)-函数的详细定义。需要补充说明的是,(i,j)-函数中的 α_m 和 β_n 分别对应于 $\overline{M}$ 和 $\overline{N}$ 序列中的信息项 M_m 和 N_n。举例来说,$SS_{2,1,1}(h(M);SSK_{2,1,1}(SK,pk(SK);i,\tau_{l,t},F),F);F = sign(\beta_1,\alpha_1)$ 表示 2.2 节中通过 (l,t)-门限签名方案所产生的关于秘密 $(H(M))^d$ 的秘密份额 (x,x_i)。其中:M 表示需要进行数字签名的信息,SK 表示秘密分发者的私钥 d,$pk(SK)$ 表示 d 所对应的公钥 (n,e)。

对应于秘密份额密钥 $SSK_{i,j,k}(\overline{M},m,\tau_{l,t},\overline{F})$,引入函数 $SVK_{i,j,k}(\overline{M},m,\tau_{l,t},\overline{F})$,其表示特定秘密共享过程中的秘密份额验证公钥,并且,其满足 $m \leqslant l$。

函数 *SVer* 表示利用秘密份额验证公钥对秘密份额进行验证,其包含 3 个参数。关于秘密份额验证的等值规则定义如下:

$$SVer_{i,j,k}(SVK_{i,j,k}(\overline{M},m,\tau_{l,t},\overline{F}),SS_{i,j,k}(\overline{N},SSK_{i,j,k}(\overline{M},m,\tau_{l,t},\overline{F}),\overline{F}),\overline{F}) = \text{true} \quad (1)$$

同时,函数 $SCombin_{i,j,k,r}$ 表示秘密份额的合并算法,其包含 $r+k$ 个参数。对于同一秘密的 r 个秘密份额 $SS_{i,j,k}(\overline{N},SSK_{i,j,k}(\overline{M},i_1,\tau_{l,t},\overline{F}),\overline{F}),\cdots,SS_{i,j,k}(\overline{N},SSK_{i,j,k}(\overline{M},i_r,\tau_{l,t},\overline{F}),\overline{F})$,如果其满足下列条件:

1) $i_m \neq i_n$,其中,$1 \leqslant m,n \leqslant r, m \neq n$。

2) $r \geqslant t$,则

$$SCombin_{i,j,k,r}(SS_{i,j,k}(\bar{N},SSK_{i,j,k}(\bar{M},i_1,\tau_{l,t},\bar{F}),\bar{F}),\cdots,SS_{i,j,k}(\bar{N},SSK_{i,j,k}(\bar{M},i_r,\tau_{l,t},\bar{F}),\bar{F}),\bar{F}) = \bar{F}\{\bar{M}/\bar{\alpha}\}\{\bar{N}/\bar{\beta}\} \tag{2}$$

通过以上规则,使得所提出的抽象模型充分反映了秘密份额所具备的安全特性,即通过 t 个或者更多个秘密份额可以合并计算得到秘密;通过 $t-1$ 个或者更少个秘密份额计算得到秘密是不可行的。同时,由于秘密共享过程以及秘密份额具有唯一的标识,这使得抽象模型反映了秘密共享方案的健壮性和正确性。

3 可验证的多秘密共享方案的实现

3.1 关于可验证的多秘密共享方案的有限等值理论

本节所提出的有限等值理论的核心思想是其只考虑进程定义中涉及的(i,j)-函数,而忽略由环境或者攻击者可能产生的其他不同(i,j)-函数。这使得前面介绍的等值理论成为有限。

首先定义 TR 集合,其具备$(i,j,k,\bar{F})$的形式,其中 $\bar{F}$ 为 k 个(i,j)-函数组成的序列。对于信息项 M 和进程 P,$terms(M)$ 包含 M 所含有的所有子项,$terms(P)$ 则包含 P 所含有的所有信息项。下面给出关于信息项和进程的(TR,h)-有效性的相关定义。

定义2 信息项 Z 是(TR,h)-有效的,当且仅当以下条件满足:

1) 对于任意 $SSK_{i,j,k}(\bar{M},M,N,\bar{F})$,$SVK_{i,j,k}(\bar{M},M,N,\bar{F})$,$SS_{i,j,k}(\bar{M},M,\bar{F})$,$SVer_{i,j,k}(M,N,\bar{F})$ 和 $SCombin_{i,j,k,r}(\bar{M},\bar{F}) \in terms(Z)$,其满足下列条件:

①$(i,j,k,\bar{F}) \in TR$;

②对于任意$(i,j,k,\bar{F}') \in TR$,如果 $E_{SS} \vdash \bar{F}' = \bar{F}$,则 $\bar{F}' = \bar{F}$。

2) 对于任意 $l \in N$,如果 Z 含有 α_l 与 β_l,则 α_l 与 β_l 只出现在(i,j)-函数中。

3) 对于任意 $SSP_{l,t}(\tau) \in terms(Z)$,其满足 $l \in [1,h]$。

进程 P 是(TR,h)-有效的,当且仅当对于任意 $M \in terms(P)$,M 是(TR,h)-有效的。

以上定义保证 TR 集合包含了所有涉及秘密份额的生成,验证,与合并的(i,j)-函数(条件1)。同时,其保证了在协议定义中,秘密数据最多只能被拆分成为 h 个秘密份额(条件3)。

接下来通过定义信息项的有效范式来限制环境或者攻击者可能产生的信息项,因此可以减少环境或者攻击者所产生的信息项中存在的冗余。

定义3 信息项 M 具有关于框架 φ 的(TR,h)-有效范式当且仅当下列条件满足:

1) 对于任意 $SSP_{l,t}(\tau) \in terms(M)$,其满足 $l \in [1,h]$,$t \in [1,l]$。

2) 对于任意 $SSK_{i,j,k}(\bar{z},x,y,\bar{F})$,$SVK_{i,j,k}(\bar{z},x,y,\bar{F})$,$SS_{i,j,k}(\bar{z},x,\bar{F})$,$SVer_{i,j,k}(x,y,\bar{F}) \in terms(M)$,其满足$(i,j,k,\bar{F}) \in TR$。

3) 对于任意 $SCombin_{i,j,k,r}(z_1,\cdots,z_r,\bar{F}) \in terms(Z)$,其满足下列条件:

①$(i,j,k,\bar{F}) \in TR$;

②$r \in [1,h]$。

4) 对于任意 $SSK_{i,j,k}(\bar{z},x,y,\bar{F})$ 和 $SVK_{i,j,k}(\bar{z},x,y,\bar{F}) \in terms(M)$,其满足 $E_{SS} \vdash y\varphi = SSP_{l,t}(N)$,$E_{SS} \vdash x\varphi = m$,并且,$m \in [1,l]$。

5) 对于任意 $SS_{i,j,k}(\bar{z},x,\bar{F}) \in terms(M)$,其满足 $E_{SS} \vdash x\varphi = SSK_{i,j,k}(\bar{M},m,SSP_{l,t}(\tau),\bar{F})$。

6) 对于任意 $SCombin_{i,j,k,r}(z_1,\cdots,z_r,\bar{F}) \in terms(M)$,如果 $E_{SS} \vdash z_m\varphi = SS_{i,j,k}(\bar{N},SSK_{i,j,k}(\bar{M},n_m,SSP_{l,t}(\tau),\bar{F}),\bar{F})$,$m = 1;2;r$,则 $n_e \neq n_f$,其中:$1 \leqslant e,f \leqslant r, e \neq f$。

特别地,在上述定义中,只考虑环境或者攻击者可能产生的同一秘密的不同秘密份额之间的合并计算。

接下来给出框架与扩展进程的有效性定义。如果框架是(TR,h)-有效的,则代表其输出信息的自由变量被关联到(TR,h)-有效的信息项;代表其输入信息的被限制变量被关联到具有(TR,h)-有效范式的信息项。同时,如果扩展进程是(TR,h)-有效的,则其可以被拆分为(TR,h)-有效的进程和框架。

定义4 对于框架 $\varphi = v\bar{n}.v\bar{y}.\{\bar{Z}/\bar{x}\}$,$\bar{y} \subseteq \bar{x}$,$\varphi$ 是(TR,h)-有效的,当且仅当以下条件满足:

1) 对于任意 $x_k \in fv(\varphi)$,其满足 x_k 是(TR,h)-有效的。

2) 对于任意 $x_k \in bv(\varphi)$,其满足 Z_k 具有关于框架 φ 的(TR,h)-有效范式,并且,$free(Z_k) \bigcap bound(\varphi) = \varnothing$。

对于扩展进程 $A = v\bar{n}.v\bar{y}.(\{\bar{Z}/\bar{x}\} \mid P)$,$\bar{y} \subseteq \bar{x}$,$A$ 是(TR,h)-有效的当且仅当以下条件满足:

1)$v\bar{n}.v\bar{y}.\{\bar{Z}/\bar{x}\}$ 是(TR,h)-有效的;

2)P 是(TR,h)-有效的。

3.2 动态编译

本节提出一种关于将等值理论 E_{SS} 转化为 ProVerif 中重写机制的编码方法。本节提出的有限等值理论 $E_{SS}^{TR,h}$ 包括函数 $SSK_{i,j,k}^{\bar{F}}$,$SVK_{i,j,k}^{\bar{F}}$,$SS_{i,j,k}^{\bar{F}}$,$SVer_{i,j,k}^{\bar{F}}$ 和 $SCombin_{i,j,k,r}^{\bar{F}}$。通过动态编译,$E_{SS}$ 中的函数 $SSK_{i,j,k}(\bar{M},M,N,\bar{F})$,$SVK_{i,j,k}(\bar{M},M,N,\bar{F})$,$SS_{i,j,k}(\bar{M},M,\bar{F})$,$SVer_{i,j,k}(M,N,\bar{F})$ 和 $SCombin_{i,j,k,r}(\bar{M},\bar{F})$ 分别被 $SSK_{i,j,k}^{\bar{F}}(\bar{M},M,N)$,$SVK_{i,j,k}^{\bar{F}}(\bar{M},M,N)$,$SS_{i,j,k}^{\bar{F}}(\bar{M},M)$,$SVer_{i,j,k}^{\bar{F}}(M,N)$ 和 $SCombin_{i,j,k,r}^{\bar{F}}(\bar{M})$ 所代替。

根据关于扩展进程有效性的定义,函数 $SSK_{i,j,k}^{\bar{F}}$,$SVK_{i,j,k}^{\bar{F}}$,$SS_{i,j,k}^{\bar{F}}$,$SVer_{i,j,k}^{\bar{F}}$ 和 $SCombin_{i,j,k,r}^{\bar{F}}$ 唯一地确定了 $\bar{F}$,因此,$\bar{F}$ 可以在协议定义中被忽略。另外,由于只考虑环境或者攻击者可能产生的同一秘密的不同秘密份额之间的合并计算,在对门限值为(l,t)的秘密共享过程中秘密份额之间的合并计算进行有限建模时,仅仅需要判断其中秘密份额的数目是否大于 t。因此,$E_{SS}^{TR,h}$ 引入了函数 $SCVer_{i,j,k,r}^{\bar{F}}$ 和 $PCombin_{i,j,k,r}^{\bar{F}}$。

函数 $SCVer_{i,j,k,r}^{\bar{F}}$ 用于判断 r 个秘密份额是否可以合并生成秘密。其可以通过以下等值规则进行建模:

$$SCVer_{i,j,k,r}^{\bar{F}}(SS_{i,j,k}^{\bar{F}}(\bar{N},SSK_{i,j,k}^{\bar{F}}(\bar{M},i_1,\tau_{l,t})),\cdots,SSK_{i,j,k}^{\bar{F}}(\bar{M},i_r,\tau_{l,t}))) = eq(t,r) \vee SCVer_{i,j,k,r-1}^{\bar{F}}(SS_{i,j,k}^{\bar{F}}(\bar{N},SSK_{i,j,k}^{\bar{F}}(\bar{M},i_1,\tau_{l,t})),\cdots,SSK_{i,j,k}^{\bar{F}}(\bar{M},i_{r-1},\tau_{l,t}))); \quad r > 1 \tag{3}$$

$$SCVer_{i,j,k,1}^{\bar{F}}(SS_{i,j,k}^{\bar{F}}(\bar{N},SSK_{i,j,k}^{\bar{F}}(\bar{M},i_m,\tau_{l,t}))) = eq(t,1) \tag{4}$$

在此基础上,门限值为(l,t)的秘密共享过程中 r 个秘密份额的合并操作可以通过以下等值规则进行建模:

$$SCombin^{\bar{F}}_{i,j,k,r}(\bar{M}) = PCombin^{\bar{F}}_{i,j,k,r}(\bar{M}, SCVer^{\bar{F}}_{i,j,k,r}(\bar{M})) \quad (5)$$

$$PCombin^{\bar{F}}_{i,j,k,r}(SS^{\bar{F}}_{i,j,k}(\bar{N}, SSK^{\bar{F}}_{i,j,k}(\bar{M}, i_1, \tau_{l,t})), \cdots, SS^{\bar{F}}_{i,j,k}(\bar{N}, SSK^{\bar{F}}_{i,j,k}(\bar{M}, i_r, \tau_{l,t})), \text{true}) = \bar{F}\{\bar{M}/\bar{\alpha}\}\{\bar{N}/\bar{\beta}\} \quad (6)$$

其中函数 $SCVer^{\bar{F}}_{i,j,k,r}$ 和 $PCombin^{\bar{F}}_{i,j,k,r}$ 并不能被攻击者所利用。

因此,针对 (TR,h)-有效的扩展进程,无限等值理论 E_{SS} 可以通过编译技术转化成为有限等值理论 $E^{TR,h}_{SS}$。接下来介绍关于信息项的动态编译技术。

定义 5　(TR,h)-动态编译是偏序函数 $\sigma: T_{\Sigma_{SS}} \rightarrow T_{\Sigma^{TR,h}_{SS}}$,其可以按照如下方式被递归调用:

$$SSK_{i,j,k}(\bar{M},M,N,\bar{F})\sigma = SSK^{\bar{F}}_{i,j,k}(\bar{M}\sigma, M\sigma, N\sigma)$$

$$SVK_{i,j,k}(\bar{M},M,N,\bar{F})\sigma = SVK^{\bar{F}}_{i,j,k}(\bar{M}\sigma, M\sigma, N\sigma)$$

$$SS_{i,j,k}(\bar{M},M,\bar{F})\sigma = SS^{\bar{F}}_{i,j,k}(\bar{M}\sigma, M\sigma)$$

$$SVer_{i,j,k}(M,N,\bar{F})\sigma = SVer^{\bar{F}}_{i,j,k}(M\sigma, N\sigma)$$

$$SCombin_{i,j,k,r}(\bar{M},\bar{F})\sigma = SCombin^{\bar{F}}_{i,j,k,r}(\bar{M}\sigma)$$

$$f(M_1,\cdots,M_i)\sigma = f(M_1\sigma,\cdots,M_i\sigma)$$

$$x\sigma = x$$

$$n\sigma = n;\ (i,j,k,\bar{F}) \in TR, r \in [1,h]。$$

对于任意 (TR,h)-有效的进程 $A = v\bar{n}.\ v\bar{y}.\ (\{\bar{M}/\bar{x}\} \mid P)$,$A\sigma$ 可用于表示 $v\bar{n}.\ v\bar{y}.\ (\{\bar{M}\sigma/\bar{x}\} \mid P\sigma)$。接下来的定理表明,经过动态编译,扩展进程之间的观察等同性得以保持,因而,本节所提出的动态编译方法是健壮的。

定理 1　对于 (TR,h)-有效的扩展进程 A,B,以及 (TR,h)-动态编译 σ,如果 $A\sigma \approx_{E^{TR,h}_{SS}} B\sigma$,则 $A \approx_{E_{SS}} B$。

证明　略。

4　结语

本文基于等值理论在应用 Pi 演算中对秘密共享方案的密码学语义进行了形式化定义。在此基础上,本文进一步给出了一种用于将所提出有限等值理论转化为自动化验证器 ProVerif 中重写机制的编码方法,并最终在 ProVerif 中实现了关于可验证的多秘密共享方案的形式化验证。

本文通过证明给出了关于可验证的多秘密共享方案形式化分析结果的健壮性结论:如果自动化协议验证器 ProVerif 中可验证的多秘密共享方案的形式化分析结果满足特定安全属性,则其能够归约证明应用 Pi 演算模型中针对可验证的多秘密共享方案所建立的现实敌手可以"模拟"ProVerif 验证器中的理想敌手,其意味着现实敌手与理想敌手是不可区分的。

参考文献:

[1] HE A J, DAWSON E. Multistage secret sharing based on one-way function [J]. Electronics Letters, 1994, 30(9): 1591 – 1592.

[2] YANG C-C, CHANG T-Y, HWANG M-S. A (t, n) multi-secret sharing scheme [J]. Applied Mathematics and Computation, 2004, 151(2): 483 – 490.

[3] SHAO J, CAO Z F. A new efficient (t, n) verifiable multi-secret sharing (VMSS) based on YCH scheme [J]. Applied Mathematics and Computation, 2005, 168(1): 135 – 140.

[4] ZHAO J, ZHANG J, ZHAO R. A practical verifiable multi-secret sharing scheme [J]. Computer Standards and Interfaces, 2007, 29(1): 138 – 141.

[5] YEW K M, RAHMAN M Z, LEE S P. Formal verification of secret sharing protocol using Coq [C]// Proceedings of the 5th Asian Computing Science Conference on Advances in Computing Science. Berlin: Springer-Verlag, 1999: 381 – 382.

[6] ABADI M, FOURNET C. Mobile values, new names, and secure communication[C]// Proceedings of the 28th Symposium on Principles of Programming Languages. New York: ACM Press, 2001: 104 – 115.

[7] ABADI M. Secrecy by typing in security protocols [J]. Journal of the ACM, 1999, 46(5): 749 – 786.

[8] ABADI M, BLANCHET B, FOURNET C. Just fast keying in the Pi calculus [J]. ACM Transactions on Information and System Security, 2007, 10(3): 9.

[9] ABADI M, GORDON A D. A calculus for cryptographic protocols: the Spi calculus [J]. Information and Computation, 1999, 148(1): 1 – 70.

[10] BACKES M, MAFFEI M, UNRUH D. Zero-knowledge in the applied Pi-calculus and automated verification of the direct anonymous attestation protocol [C]// Proceedings of the 29th IEEE Symposium on Security and Privacy. Washington, DC: IEEE Computer Society, 2008: 202 – 215.

[11] BACKES M, HRITCU C, MAFFEI M. Type-checking zero-knowledge [C]// Proceeding of the 16th ACM Conference on Computer and Communications Security. New York: ACM Press, 2008: 357 – 370.

[12] BACKES M, MAFFEI M, MOHAMMADI E. Computationally sound abstraction and verification of secure multi-party computations [C]// Proceedings of IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science. Chennai, India: [s. n.], 2010: 352 – 363.

[13] SHOUP V. Practical threshold signatures [C]// Proceedings of EURO-CRYPT 2000. Berlin: Springer-Verlag, 2000: 207 – 220.

[14] BACHMAIR L, GANZINGER H. Rewrite-based equational theorem proving with selection and simplification [J]. Journal of Logic and Computation, 1994, 4(3): 217 – 247.

(上接第 3189 页)

[8] WANG Y, YANG C, WU X, *et al*. Kinect based dynamic hand gesture recognition algorithm research [C]// Proceedings of the 4th International Conference on Intelligent Human-Machine Systems and Cybernetics. Piscataway: IEEE Press, 2012: 274 – 279.

[9] 卜富清. 基于人工神经网络的图像识别和分类[D]. 成都: 成都理工大学, 2010.

[10] IKEMURA S, FUJIYOSHI H. Real-time human detection using relational depth similarity features [C]// Proceedings of the 10th Asian Conference on Computer Vision. Berlin: Springer, 2011: 25 – 38.

[11] SHOTTON J, SHARP T, KIPMAN A, *et al*. Real-time human pose recognition in parts from single depth images [J]. Communications of the ACM, 2013, 56(1): 116 – 124.

[12] 余涛. Kinect 应用开发实战用最自然的方式与机器对话[M]. 北京: 机械工业出版社, 2013.

[13] WEBB J, ASHLEY J. Beginning Kinect programming with the Microsoft Kinect SDK [M]. New York: APress, 2012: 93 – 94.

[14] 黄思博. 基于计算机视觉的异常驾驶行为检测方法研究[D]. 广州: 华南理工大学, 2011.