需求,间接隐私数据项的安全等级根据信息流的安全策略决定.由于用户需求中规定了隐私数据项组合使用的安全等级,当多个信息流流向同一实体时,安全策略实施时需要考虑隐私数据项聚合问题,主要表现在两方面.

(1) BPEL 中的变量是临时存放数据的容器,由若干隐私数据项组成,变量安全等级的计算需要考虑用户隐私需求中对数据项组合的使用约束.例如,假设用户部分隐私需求为 r_1=({*email*},(*M*,*top-retention*,{*current*,*contact*})),r_2=({*name*},(*M*,1*day*,{*current*})),r_3=({*email*,*name*},(*H*,1*day*,{*current*})).计算变量 *v*={*email*,*name*}的安全等级时,因为用户在需求中定义了 *email*,*name* 组合使用时的安全等级,变量的安全等级不应为 $\overline{v}=\overline{name}\oplus\overline{email}$ =(*M*,*top-retention*,{*current*,*contact*})⊕(*M*,1*day*,{*current*})=(*M*,1*day*,{*current*}).根据隐私规则 r_3, $\overline{v}=\overline{\{email,name\}}$ =(*H*,1*day*,{*current*}).

(2) 在组合执行过程中,成员服务可能多次接收用户隐私数据,虽然单次接收满足安全策略,但多次接收后可能不满足用户隐私需求.仍以情形(1)中的隐私需求为例,假设成员服务 *s* 的安全等级为 $\overline{s}$ =(*M*,1*day*,{*current*})),设 v_1={*email*},v_2={*name*},虽然 $\overline{v}_1\to\overline{s}\wedge\overline{v}_2\to\overline{s}$ 成立,但这两个信息流分别发生后,*s* 接收了数据项集合{*email*,*name*},违反了隐私规则 r_3.

4.3 安全等级绑定

4.3.1 成员服务安全等级绑定

成员服务 *s* 的安全等级可以静态绑定为 *s.sc*.假设 *s* 未接收过任何隐私数据,当其更新或新产生间接隐私数据项时,为满足安全策略的“不向下写”原则,*s* 输出数据的安全等级也应为 *s.sc*,这显然是不合理的.为此,针对成员服务 *s* 提出如下降密策略:由于 *s* 接收到的隐私数据是动态变化的,为防止用户隐私信息泄露,服务 *s* 的安全等级取决于其历史接收到的隐私数据,即 $\overline{s}=\overline{s.hds}$,其中,*s.hds* 表示服务 *s* 曾经接收到的隐私数据项集合.如果 *s* 未曾接收过任何隐私数据项,则 $\overline{s}=L_{sc}$.只要服务的操作满足安全策略, $\overline{s.hds}$ 总等于或低于 *s.sc*.

值得说明的是:检查组合向服务 *s* 释放隐私数据的操作时,*s* 的安全等级仍然绑定为 *s.sc*.

4.3.2 变量安全等级绑定

变量的安全等级绑定有两种方法:静态绑定和动态绑定.BPEL 中的变量是临时存放数据的容器,其安全等级取决于保存的内容,适合采用动态绑定方法.

变量 *v* 中包含的直接和间接隐私数据项的集合分别记为 v_{dir} 和 v_{ind},*v* 依赖的直接隐私数据项集合记为 *DDep*(*v*).由于安全策略的“不向下写,不向上读”原则以及成员服务的降密策略,所以有 $\overline{v}=\overline{DDep(v)}$.*DDep*(*v*)的计算公式如下:

$$DDep(v)=v_{dir}\cup DDep(d_{ind\_1})\cup\ldots\cup DDep(d_{ind\_m}) \tag{1}$$

其中,$d_{ind\_i}\in v_{ind}$,$1\leqslant i\leqslant m$,表示变量 *v* 中的间接隐私数据项.由于用户隐私需求中定义了隐私数据项组合的安全等级,从而引入隐私数据项聚合问题,因此,*v* 的安全等级的计算公式如下:

$$\overline{v}=\overline{ds_1}\oplus\ldots\oplus\overline{ds_i}\oplus\ldots\oplus\overline{ds_n} \tag{2}$$

其中,$ds_i\in 2^{DDep(v)}\wedge ds_i\in PR\_DSet$,$1\leqslant i\leqslant n$,$2^{DDep(v)}$表示 *DDep*(*v*)的幂集,*PR_DSet*={$r_k.ds|r_k\in PR$,$1\leqslant k\leqslant|PR|$},*PR* 表示隐私需求,且 $\overline{ds_i}=r_k.sc, ds_i=r_k.ds$.

4.3.3 间接隐私数据项安全等级绑定

传统的数据流分析技术关注系统实体之间的数据依赖关系,例如:服务 s_1 定义了变量 *v*,服务 s_2 使用变量 *v*,服务 s_2 数据依赖于服务 s_1.这类研究主要应用于数据流相关属性验证、系统演化分析等工作[23−25].信息流控制中的数据流分析则关注数据之间的依赖关系,例如:若存在赋值操作 *a*=*b*,则变量 *a* 依赖于变量 *b*.虽然文献[26]为实施信息流控制机制提出数据依赖分析规则,且这些规则可以分析因变量赋值引入的显式依赖以及流程结构引入的隐式依赖,但在本文的隐私保护场景中,由于考虑了面向成员服务的降密策略及隐私数据项聚合问题,需要提出特定的依赖分析方法.

组合执行过程中更新或新产生的间接隐私数据项记为 d_{new}.由于安全策略的“不向下写,不向上读”原则以及成员服务的降密策略,所以有 $\overline{d_{new}}=\overline{DDep(d_{new})}$,即其安全等级取决于依赖的直接隐私数据项集合

$DDep(d_{new}).d_{new}$ 依赖的直接和间接隐私数据项集合记为 $Dep(d_{new})$,只要得到 $Dep(d_{new})$,根据依赖的传递性,很容易得到 $DDep(d_{new})$.

针对组合的隐私工作流网模型,提出隐私数据项依赖关系分析规则如下.

- **DAR**$_1$. $T_A(t)=ASGN\Rightarrow Dep(d_{new})=r(t),\forall d_{new}\in w(t)$;
- **DAR**$_2$. $T_A(t)=RECV\Rightarrow Dep(d_{new})=T_S(t).hds,\forall d_{new}\in w(t)$;
- **DAR**$_3$. $T_A(t)=SND\Rightarrow T_S(t).hds=T_S(t).hds\cup r(t)$;
- **DAR**$_4$. $\forall d_{new\_i},d_{new\_j},\exists d_{new\_k},d_{new\_k}\in Dep(d_{new\_i})\wedge d_{new\_j}\in Dep(d_{new\_k})\Rightarrow Dep(d_{new\_i})=Dep(d_{new\_i})\cup\{d_{new\_j}\}$.

设 $assign(d_{new},v)$表示 assign 活动,显然 d_{new} 依赖于变量 v,该活动建模为 ASGN 类型变迁.对应 **DAR**$_1$ 有 $Dep(d_{new})=r(t),\forall d_{new}\in w(t)$.其中,$r(t)$表示变迁 t 的读集,$w(t)$表示变迁 t 的写集.设 $receive(s,v)$表示 receive 活动,v 中任意 d_{new} 依赖于 s 曾经接收到的隐私数据项集合,该活动建模为 RECV 类型变迁.对应 **DAR**$_2$ 有 $Dep(d_{new})=T_S(t).hds,\forall d_{new}\in w(t)$.其中,$T_S(t)$表示变迁 t 对应的成员服务,$T_S(t).hds$ 表示该服务曾经接收到的隐私数据项集合.在本文的隐私保护场景中,用户是可信实体,且我们假设用户不会更新或新产生间接隐私数据项,因此,若组合通过 receive 活动从用户接收数据,即 $T_S(t)$=USER,则分析该变迁时不需要应用规则 **DAR**$_2$.设 $reply(s,v)$表示 reply 活动,$invoke(s,v)$表示单向 invoke 活动,本质上均为向成员服务发送变量 v 的内容,均建模为 SND 类型变迁.两者需更新服务 s 接收到的隐私数据项集合,因此对应规则 **DAR**$_3$,即 $T_S(t).hds=T_S(t).hds\cup r(t)$.值得说明的是:由于用户是可信实体,若 $T_S(t)$=USER,该变迁不需要应用规则 **DAR**$_3$.设 $invoke(s,v_1,v_2)$表示请求响应 invoke 活动,建模为顺序执行的 SND 类型和 RECV 类型变迁,需依次应用规则 **DAR**$_3$ 和 **DAR**$_2$.**DAR**$_4$ 用于处理依赖的传递关系,即:如果 d_{new_i} 依赖于 d_{new_k},d_{new_k} 依赖于 d_{new_j},则 d_{new_i} 也依赖于 d_{new_j}.由于方法不考虑流程结构引入的隐蔽通道,所以不需要分析 STRC 类型变迁.

通过规则 **DAR**$_4$,可以从 $Dep(d_{new})$推导出 $DDep(d_{new})$.显然,通过分析隐私数据项的依赖关系,最终可以得到隐私数据项依赖图,其定义如下:

定义 4.6(隐私数据项依赖图). 隐私数据项依赖图 $PDIDG=(V,E)$是有向无环图,其中:V 是隐私数据项的有限集合;E 是边的集合,表示隐私数据项节点之间的依赖关系.节点对应的隐私数据项分为直接和间接隐私数据项两类,其中:出度为 0 的节点为直接隐私数据项节点;出度不为 0 的节点为间接隐私数据项节点.

4.4 信息流控制规则

为了验证路径的隐私信息流安全性,给出隐私信息流控制规则,其中,$secure(t)$=true 表示变迁 t 的触发是安全的.

- **IFCR**$_1$. $T_A(t)=ASGN\Rightarrow secure(t)$=true iff $\overline{r(t)}\rightarrow\overline{d_{new}}$, $\forall d_{new}\in w(t)$;
- **IFCR**$_2$. $T_A(t)=RECV\Rightarrow secure(t)$=true iff $\overline{T_S(t).hds}\rightarrow\overline{d_{new}}$, $\forall d_{new}\in w(t)$;
- **IFCR**$_3$. $T_A(t)=SND\Rightarrow secure(t)$ = true iff $\overline{r(t)\cup T_S(t).hds}\rightarrow\overline{T_S(t)}$.

其中,**IFCR**$_1$ 表明 assign 活动是安全的,当且仅当 $\overline{r(t)}\rightarrow\overline{d_{new}}$.由于 d_{new} 的安全等级采用动态绑定,由 **DAR**$_1$ 可知 $\overline{d_{new}}=\overline{r(t)}$,因此 $\overline{r(t)}\rightarrow\overline{d_{new}}$ 总是满足的.**IFCR**$_2$ 表明 receive 活动是安全的,当且仅当 $\overline{T_S(t).hds}\rightarrow\overline{d_{new}}$, $\forall d_{new}\in w(t)$.同理,由于 d_{new} 的安全等级采用动态绑定,由 **DAR**$_2$ 可知 $\overline{d_{new}}=\overline{T_S(t).hds}$,因此规则 **IFCR**$_2$ 总是满足的.由于用户隐私需求中定义了隐私数据项组合的安全等级,从而引入隐私数据项聚合问题.**IFCR**$_3$ 表明 reply 活动或单向 invoke 活动是安全的,当且仅当 $\overline{r(t)\cup T_S(t).hds}\rightarrow\overline{T_S(t)}$.值得说明的是:若 $T_S(t)$=USER,该变迁不需要应用规则 **IFCR**$_2$ 和 **IFCR**$_3$.对于请求响应 invoke 活动,建模为顺序执行的 SND 类型和 RECV 类型变迁,分别应用规则 **IFCR**$_3$ 和 **IFCR**$_2$ 即可.

4.5 静态分析算法

通过静态分析算法检测待验证的每条路径是否会产生非法隐私信息流,该算法伪码见算法 1.

算法 1. 静态分析算法.

输入:*PR*/*用户隐私需求*/;
PW/*待验证路径集合*/
PWF-net/*组合隐私工作流网模型*/
输出:*bResult*/*布尔型,标识是否存在泄露*/
ILLEGAL_FLOWS/*非法信息流集合*/.

```
1.  bResult=true;
2.  ILLEGAL_FLOWS=∅;
3.  for each p in PW do
4.    init_pdidg(PR,PDIDG)
5.    for each t in p do
6.      if (t.subject==USER) continue; end if
7.      if (t.type==STRC) continue; end if
8.      if (t.type==ASGN) update(PDIDG,t); end if
9.      if (t.type==RECV) update(PDIDG,t); end if
10.     if (t.type==SND)
11.       if (chk_IFCR3(PR,t.subject)==false)
12.         bResult=false;
13.         ILLEGAL_FLOWS.add(p,t);
14.         break;
15.       else
16.         update_hds(t.rs,t.subject);
17.       end if
18.     end if
19.   end for
20. end for
```

算法需要分析 *PW* 中的每条路径 *p*,并依次处理 *p* 中每个变迁 *t*.设 *PW* 中路径数为 *n*,*p* 中变迁数为 *m*.在进行路径分析之前,首先通过第 4 行初始化隐私数据项依赖图 *PDIDG*,根据隐私需求初始化后的 *PDIDG* 中只包含直接隐私数据项节点.第 6 行的含义为:由于用户为可信实体,若变迁对应主体为 USER,不管变迁类型是 RECV 还是 SND,都不需要进行检测.第 7 行的含义为:由于本文分析方法不考虑控制流程引入的隐式泄露,所以不需要检测类型为 STRC 的变迁.第 8 行、第 9 行的含义为:由于间接隐私数据项的安全等级采用动态绑定,根据规则 $\mathbf{IFCR}_1$,$\mathbf{IFCR}_2$ 可知:ASGN,RECV 类型的变迁是安全的,只需分别通过规则 $\mathbf{DAR}_1$,$\mathbf{DAR}_2$ 更新隐私数据项依赖图 *PDIDG* 即可.若间接隐私数据项不在当前 *PDIDG* 中,则创建相应节点及依赖关系;否则,仅需更新相应的依赖关系.设组合中的隐私数据项数量为 *k*,变迁操作的数据项集合的大小、用户需求 *PR* 中规则的数量等均可以认为与 *k* 具有线性关系.由于每个 ASGN 类型变迁只更新一个间接隐私数据项,因此第 8 行的时间复杂度为 $O(k^2)$.RECV 类型变迁更新写集中间接隐私数据项,所以第 9 行的时间复杂度为 $O(k^3)$.第 10 行~第 18 行处理 SND 类型变迁,其中,第 11 行的 *chk_IFCR*3 函数用于检测变迁的触发是否满足信息流控制规则 $\mathbf{IFCR}_3$.

为处理隐私数据项聚合问题,该函数首先根据公式(1)得到 $DDep(t.rs\cup t.subject.hds)$,其中,*t.rs* 表示变迁的读集,*t.subject.hds* 表示变迁对应服务 *t.subject* 接收到的隐私数据项集合,该步骤时间复杂度为 $O(k^3)$;然后,根据公式(2)计算 $\overline{DDep(t.rs\cup t.subject.hds)}$,该步骤的时间复杂度为 $O(k^3)$;最后,判断是否满足规则 $\mathbf{IFCR}_3$ 中的条件 $\overline{DDep(t.rs\cup t.subject.hds)}\rightarrow\overline{t.subject}$,该步骤计算量主要体现在判断使用目的集合的包含关系上,设格 *P* 中使用目的的有限集合 *PS* 中元素数量为 *h*,则该步骤的时间复杂度为 $O(h^2)$.若不满足规则 $\mathbf{IFCR}_3$,则在第 13 行记录非法信息流信息,并检测下一条路径;若满足,则在第 16 行更新对应成员服务接收到的隐私数据项集合,时间复杂

度为 $O(k^2)$.

综上所述,算法 1 的总体算法复杂度为 $O(nmk^3)$.

5 实例分析与实验

5.1 实例分析

通过旅行代理(travel agent,简称 TA)对本文提出的方法进行实例分析.TA 根据用户旅行计划提供机票预订、酒店预订、在线支付一站式服务,组合了机票预订(flight)、酒店预订(hotel)及在线支付(pay)这 3 个服务,假设完成机票预订和酒店预订后一并进行支付.针对其隐私数据,假设用户 Bob 定义的隐私需求如下.

- r_1=({*name*},(*M*,1*day*,{*current*,*contact*}));
- r_2=({*phone*},(*M*,1*day*,{*current*,*contact*}));
- r_3=({*id_number*},(*H*,1*day*,{*current*,*contact*}));
- r_4=({*credit_card_info*},(*H*,0*day*,{*current*}));
- r_5=({*name*,*id_number*,*credit_card_info*},(*TH*,0*day*,{*current*})).

其中,用户针对隐私数据项组合{*name*,*id_number*,*credit_card_info*}定义了更为严格的使用约束条件.此外,假设服务 flight,hotel,pay 对应的安全等级分别为(*H*,1*day*,{*current*,*contact*}),(*M*,1*day*,{*current*,*contact*}),(*H*,0*day*,{*current*}).为简化表述,略掉用户查询航班和酒店信息的过程及组合中的 assign 活动,TA 的 PWF_net 模型的控制流视图如图 7 所示.

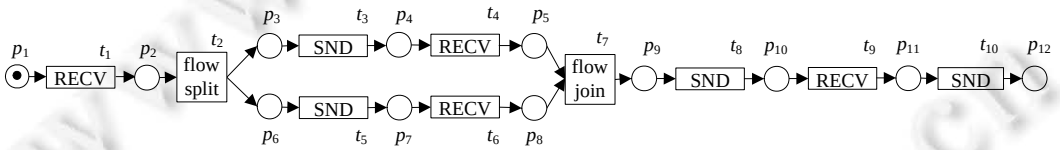

Fig.7 Control view of TA

图 7 TA 控制流视图

{t_3,t_4}与{t_5,t_6}这两组变迁分别对应酒店预订和机票预订操作,每组中的变迁会与另一组中的变迁并发执行,且任意两个并发执行的变迁不对应相同的成员服务,因此根据性质 1,只需验证 TA 的独立路径集,该路径集中只包含一条独立路径.假设选取 p=t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8,t_9,t_{10} 作为独立路径,通过算法 1 依次验证 p 中的每个变迁,验证过程信息及结果见表 2.

验证过程如下:

(1) t_1 接收用户预订信息,对应用户实体,t_2 属于 STRC 类型变迁,两者不需要验证;

(2) t_3 将隐私数据{*name*,*phone*}发送给酒店预订服务 hotel.因 $\overline{\{name, phone\}}$=(*M*,1*day*,{*current*,*contact*}), $\overline{hotel}$=(*M*,1*day*,{*current*,*contact*}),所以根据规则 **IFCR**$_3$, $\overline{\{name, phone\}} \rightarrow \overline{hotel}$ 成立,t_3 是信息流是安全的.同时,需要根据规则 **DAR**$_3$ 更新服务 hotel 使用的隐私数据项集合 *hotel.hds*={*name*,*phone*};

(3) t_4 接收酒店预订结果,根据规则 **DAR**$_2$,*Dep*(*hotel_order_id*)=*hotel.hds*={*name*,*phone*}.由于间接隐私数据项的安全等级采用动态绑定方法,t_4 是信息流是安全的.同时,需要根据规则 **DAR**$_2$ 更新路径对应的隐私数据项依赖图;

(4) t_5 将隐私数据{*name*,*id_number*}发送给机票预订服务 flight.因 $\overline{\{name,id\_number\}}$=(*H*,1*day*,{*current*,*contact*}), $\overline{flight}$=(*H*,1*day*,{*current*,*contact*}),所以根据规则 **IFCR**$_3$, $\overline{\{name,id\_number\}} \rightarrow \overline{flight}$ 成立,t_5 是信息流是安全的.同时,需要根据规则 **DAR**$_3$ 更新服务 flight 使用的隐私数据项集合:

flight.hds={*name*,*id_number*};

(5) t_6 接收机票预订结果,根据规则 **DAR**$_2$,*Dep*(*flight_order_id*)=*flight.hds*={*name*,*id_number*}.由于间接隐

私数据项的安全等级采用动态绑定方法,t_6是信息流是安全的.同时,需要根据规则 $\mathbf{DAR}_2$ 更新路径对应的隐私数据项依赖图;

(6) t_7属于 STRC 类型变迁,不需要验证;

(7) t_8将隐私数据集合 D={*hotel_order_id*,*flight_order_id*,*credit_card_info*}发送给服务 pay 以完成支付操作.根据之前验证操作建立的依赖关系易知,$DDep(D)$={*id_number*,*name*,*phone*,*credit_card_info*}.由于用户隐私规则 r_5针对隐私数据项组合{*name*,*id_number*,*credit_card_info*}定义了更为严格的使用约束条件,根据公式(2), $\overline{D}$ =(*TH*,0*day*,{*current*}).因 $\overline{pay}$ =(*H*,0*day*,{*current*}),所以 $\overline{D} \rightarrow \overline{pay}$ 不成立,变迁 t_8对应活动会引入用户隐私信息的非法泄露.

值得说明的是:一旦路径中的某变迁引入非法信息流,针对该路径的验证过程即可终止.

Table 2 Verification process and result

表 2 验证过程及结果

变迁	类型	含义	活动输入	活动输出	实体	实体访问的隐私数据集*hds*	是否安全
t_1	RECV	TravelBook	none	*name, id_number, credit_card_info, phone*	user	N/A	Yes
t_2	flow split	flow	none	none	TA	N/A	Yes
t_3	SND	HotelBook	*name, phone*	none	hotel	*name, phone*	Yes
t_4	RECV	HotelBookResp	none	*hotel_order_id*	hotel	*name, phone*	Yes
t_5	SND	FlightBook	*name, id_number*	none	flight	*name, id_number*	Yes
t_6	RECV	FlightBookResp	none	*flight_order_id*	flight	*name, id_number*	Yes
t_7	flow join	flow	none	none	TA	N/A	Yes
t_8	SND	PayRequest	*hotel_order_id, flight_order_id, credit_card_info*	none	pay	null	No
t_9	RECV	PayResponse	none	*pay_result*	pay	null	–
t_{10}	SND	BookResultResp	*hotel_order_id, flight_order_id, pay_result*	none	user	N/A	–

5.2 实 验

首先对算法 1 进行仿真实验,用以评估隐私数据项数量及路径中变迁数量对算法性能的影响.实验的系统环境为 Intel Pentium CPU 3.2GHz,4G 内存,32 位 Windows7 操作系统.编程环境为 Eclipse 4.4.2+JDK1.7.

算法 1 的时间复杂度为 $O(nmk^3)$,其中,n 是路径数量,m 是路径中变迁数量,k 是隐私数据项数量.我们针对单条路径对算法 1 的性能进行实验分析.实验相关参数设置为:隐私数据项数量为 k,其中,直接和间接隐私数据项的数量均为 k/2;用户隐私需求中隐私规则数量为 k,其中,涉及数据项组合的规则数量为 k/2;隐私数据及成员服务的安全等级随机生成,安全等级的集合采用本文第 2 节中的 *SC*;路径中计算量最大的 RECV 和 SND 类型变迁数量均为 m/2,读集和写集中包含的隐私数据项随机生成,且数量均为 k/2.m 分别取 10,50,90,评估 k 不同取值时算法的性能.实验结果如图 8 和图 9 所示.

算法 1 的最坏时间复杂度为 $O(nmk^3)$,主要计算量体现在处理 SND 和 RECV 类型变迁.图 8 中的实验结果表明:在实验设置条件下,当变迁数 m 值确定时,执行时间随着隐私数据项数 k 的增大而显著增大.例如:当 m=10,k=10 时,执行时间为 0.79ms;当 m=10,k=90 时,执行时间为 12.77ms.而当 k 的值确定时,m 值的增大对执行时间的影响则相对较小.

从图 9 可知:在实验设置条件下,当 m=10,k=10 时,算法 1 内存消耗为 0.69MB;当 m=90,k=90 时,内存消耗为 2.68MB.实验结果表明,算法 1 的内存消耗不会随着 m 和 k 的增大而急剧增长.这是因为依赖图中数据项数及边的数量、用户需求中的规则数、消息中的隐私数据项数均与 k 呈线性关系.

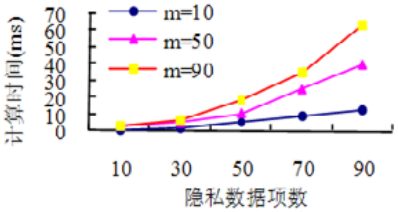

Fig.8 Variation of time cost with the number of privacy data items and transitions

图 8 计算时间随数据项数和变迁数的变化

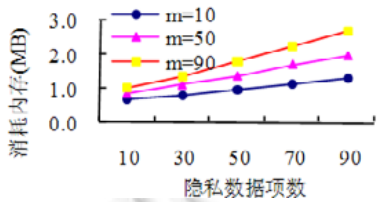

Fig.9 Variation of memory cost with the number of privacy data items and transitions

图 9 消耗内存随数据项数和变迁数的变化

由于组合的PWF-net建模及路径信息获取在部署组合前就已完成,验证开销取决于算法1.上述实验结果表明,方法的计算时间和内存消耗并不会随着变迁数量和隐私数据项数的增大而快速提升.在实际应用中,由于组合的路径、变迁、隐私数据项数均较小,因此本文方法并不会给用户与组合的交互引入过多的负载.

本文方法采用的技术路线与文献[18]类似,为便于比较,以第5.1节中的旅行代理为例,从需要分析的可达标识图中状态数量和需验证的路径数量两个方面进行实验分析,实验结果如图10和图11所示.

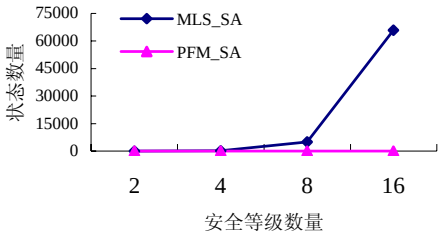

Fig.10 Variation of state number with the number of security classes

图 10 状态数随安全等级数量的变化

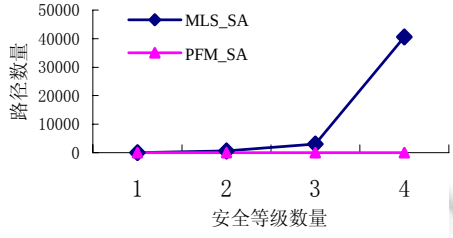

Fig.11 Variation of path number with the number of security classes

图 11 路径数随安全等级数量的变化

MLS_SA 表示文献[18]中的方法,PFM_SA 表示本文方法.随着安全等级数量的增加,方法 MLS_SA 需要分析的状态数和路径数会急剧增长.这是由于方法 MLS_SA 处理 RECV 类型变迁时,为数据库所标注所有可能的安全等级得到扩展可达标识图,穷举出系统所有可能的状态.而本文方法分析的可达标识图中状态数与路径数并不依赖于安全等级数量.本文方法中,可达图对应的状态数为 14,路径数为 6.由于 TA 中并发执行的变迁对应不同的成员服务,故只需对 1 条独立路径进行分析验证.而且,涉及安全等级的计算量主要体现在判断使用目的集合的包含关系上,时间复杂度为 $O(|PS|^2)$,其中,PS 是使用目的的有限集合.安全模型中安全等级的数量为$|SC|=|RS|\times|RT|\times|PC|$.由于 PC 是使用目的集合 PS 的幂集,且在实际应用中 PS 的元素较多,例如 P3P 中规定了 12 种使用目的,故$|SC|$往往较大.因此在实际应用中,本文方法的性能明显优于文献[18]中的方法.

6 总结与未来工作

隐私数据一旦提交给服务组合,用户难以控制组合如何使用和暴露用户隐私信息.如何保证组合执行过程中不发生用户隐私信息非法泄露,成为当前服务计算领域的研究热点之一.本文针对隐私保护特征,从服务信誉度、隐私数据使用目的及保留期限这 3 个维度提出一种面向服务组合的隐私信息流安全模型,从而形式化规约了隐私信息流的安全策略.为静态实施安全策略,提出了支持隐私信息流分析的隐私工作流网模型 PWF-net,在组合的 PWF-net 模型基础上,通过静态分析算法检测组合执行是否会发生用户隐私信息的非法泄露.最后,通过实例分析说明了方法的有效性,并对方法性能进行了实验分析.

在基于本文方法构建隐私保护框架时,服务提供者在部署组合前创建对应的 PWF-net 模型及路径信息.用户可以通过隐私代理向组合发送隐私需求,组合通过静态分析算法检测是否存在隐私信息流非法泄露.若不存

在非法泄露,则继续进行使用服务组合.若存在非法泄露,则可以采用如下解决方法:(1) 用户降低隐私数据的释放约束限制;(2) 组合选择其他功能相同的成员服务以满足安全策略.

本文方法仍然存在一些限制,这也是下一步研究的重点.首先,将独立路径集作为待验证路径集的前提条件过于严格,为更好地解决并发变迁引入的路径爆炸问题,后续拟在深入研究并发变迁对隐私数据依赖关系影响的基础上,提出能适用更多情况的路径约简方法.此外,分析方法并未考虑流程结构引入的隐蔽通道,后续研究拟对方法进行扩展,以能够对流程结构引入的隐蔽通道进行检测.

References:

[1] Pearson S. Taking account of privacy when designing cloud computing services. In: Proc. of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing. New York: IEEE Press, 2009. 44−52. [doi: 10.1109/CLOUD.2009.5071532]

[2] Warren SD, Brandeis LD. The right to privacy. Harvard Law Review, 1890,4(5):193−220. [doi: 10.2307/1321160]

[3] Westin A. Privacy and Freedom. New York: Atheneum, 1967.

[4] Goldberg I, Wagner D, Brewer E. Privacy-Enhancing technologies for the Internet. In: Proc. of the 42nd IEEE Int'l Computer Conf. New York: IEEE Press, 1997. 103−109. [doi: 10.1109/CMPCON.1997.584680]

[5] Ke CB, Huang ZQ, Tang M. Supporting negotiation mechanism privacy authority method in cloud computing. Knowledge-Based Systems, 2013,51:48−59. [doi: 10.1016/j.csi.2010.09.001]

[6] Allison DS, EL Yamany HF, Capretz M. Meta model for privacy policies within SOA. In: Proc. of the 2009 Int'l Conf. on Software Engineering (ICSE) Workshop on Software Engineering for Secure Systems. New York: IEEE Press, 2009. 40−46. [doi: 10.1109/IWSESS.2009.5068457]

[7] Organization for Economic Co-operation and Development. OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data. Organization for Economic Co-operation and Development, 2013.

[8] Liu LY, Li Q, Zhu Y, Zhou H, Xiao FX, Huang ZQ. Specification and verification of privacy requirements in Web service compositions. Journal of PLA University of Science and Technology (Natural Science Edition), 2012,13(1):27−33 (in Chinese with English abstract). [doi: 10.3969/j.issn.1009-3443.2012.01.006]

[9] Li YH, Paik HY, Benatallah B. Formal consistency verification between BPEL process and privacy policy. In: Proc. of the 2006 Int'l Conf. on Privacy, Security and Trust (PST): Bridge the Gap Between PST Technologies and Business Services. New York: ACM Press, 2006. 1−10. [doi: 10.1145/1501434.1501466]

[10] Yan D, Tian Y, Huang J, Yang F. Privacy-Aware RBAC model for Web services composition. The Journal of China Universities of Posts and Telecommunications, 2013,20(1):30−34. [doi: 10.1016/S1005-8885(13)60253-8]

[11] Peng HF, Huang ZQ, Fan DJ, Zhang YL. Specification and verification of user privacy requirements for service composition. Ruan Jian Xue Bao/Journal of Software, 2016,27(8):1948−1963 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4945.htm [doi: 10.13328/j.cnki.jos.004945]

[12] Bacon J, Eyers D, Pasquier TFJM, Singh J, Papagiannis I, Pietzuch P. Information flow control for secure cloud computing. IEEE Trans. on Network and Service Management, 2014,11(1):76−89. [doi: 10.1109/TNSM.2013.122313.130423]

[13] Nakajima S. Model-Checking of safety and security aspects in Web service flows. In: Proc. of the 4th Int'l Conf. on Web Engineering. Berlin: Springer-Verlag, 2004. 488−501. [doi: 10.1007/978-3-540-27834-4_60]

[14] Denning DE. A lattice model of secure information flow. Communications of the ACM, 1976,19(5):236−243. [doi: 10.1145/360051.360056]

[15] Hutter D, Volkamer M. Information flow control to secure dynamic Web service composition. In: Proc. of the 3rd Int'l Conf. on Security in Pervasive Computing. Berlin: Springer-Verlag, 2006. 196−210. [doi: 10.1007/11734666_15]

[16] Accorsi R, Lehmann A, Lohmann N. Information leak detection in business process models: Theory, application, and tool support. Information Systems, 2015,47:244−257. [doi: 10.1016/j.is.2013.12.006]

[17] Bell DE, Lapadula LJ. Secure computer systems: Mathematical foundations. MITRE Technical Report, 2547, Bedford: MITRE Corporation, 1996.

[18] Knorr K. Multilevel security and information flow in Petri net workflows. In: Proc. of the 9th Int'l Conf. on Telecommunication Systems—Modeling and Analysis. Las Vegas: WASET, 2001. 1−16.

[19] Cranor L, Dobbs B, Egelman S, Hogben G, Humphrey J, Langheinrich M, Marchiori M, Presler-Marshall M, Reagle J, Schunter M, Stampley DA, Wenning R. The Platform for Privacy Preferences 1.1 (P3P1.1) Specification. W3C, 2006.

[20] Van der Aalst WMP. Verification of workflow nets. In: Proc. of the 18th Int'l Conf. on Application and Theory of Petri Nets. Berlin: Springer-Verlag, 1997. 407−426. [doi: 10.1007/3-540-63139-9_48]

[21] Zhou GF, Du ZM. Petri nets model of implicit data and control in program code. Ruan Jian Xue Bao/ Journal of Software, 2011, 22(12):2905−2918 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3956.htm [doi: 10.3724/SP.J.1001.2011.03956]

[22] Lohmann N. A feature-complete petri net semantics for WS-BPEL 2.0. In: Proc. of the 4th Int'l Workshop on Web Services and Formal Methods. Berlin: Springer-Verlag, 2007. 77−91. [doi: 10.1007/978-3-540-79230-7_6]

[23] Kazhamiakin R, Pistore M. Static verification of control and data in Web service compositions. In: Proc. of the 4th IEEE Int'l Conf. on Web Services. New York: IEEE Press, 2006. 83−90. [doi: 10.1109/ICWS.2006.124]

[24] Song M, Wei ZX, Yin GS. Evolution analysis of data flow oriented internetware service. Ruan Jian Xue Bao/Journal of Software, 2013,24(12):2797−2813 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4396.htm [doi: 10.3724/SP.J.1001.2013.04396]

[25] Song W, Ma XX, Lu J. Instance migration in dynamic evolution of Web service compositions. Chinese Journal of Computers, 2009, 32(9):1816−1831 (in Chinese with English abstract). [doi: 10. 3724/SP.J.1016.2009.01816]

[26] She W, Yen IL, Thuraisingham B, Huang SY. Rule-Based run-time information flow control in service cloud. In: Proc. of the 9th IEEE Int'l Conf. on Web Services. New York: IEEE Press, 2011. 524−531. [doi: 10.1109/ICWS.2011.35]

附中文参考文献:

[8] 刘林源,李清,祝义,周航,肖芳雄,黄志球.Web 服务组合中的隐私需求规约与验证.解放军理工大学学报(自然科学版),2012,13(1):27−33. [doi: 10.3969/j.issn.1009-3443.2012.01.006]

[11] 彭焕峰,黄志球,范大娟,章永龙.面向服务组合的用户隐私需求规约与验证方法.软件学报,2016,27(8):1948−1963. http://www.jos.org.cn/1000-9825/4945.htm [doi: 10.13328/j.cnki.jos.004945]

[21] 周国富,杜卓敏.程序代码中隐含数据与控制的 Petri 网建模技术.软件学报,2011,22(12):2905−2918. http://www.jos.org.cn/1000-9825/3956.htm [doi: 10.3724/SP.J.1001.2011.03956]

[24] 宋敏,韦正现,印桂生.面向数据流的网构软件服务动态演化分析.软件学报,2013,24(12):2797−2813. http://www.jos.org.cn/1000-9825/4396.htm [doi: 10.3724/SP.J.1001.2013.04396]

[25] 宋巍,马晓星,吕建.Web 服务组合动态演化的实例可迁移性.计算机学报,2009,32(9):1816−1831. [doi: 10.3724/SP.J.1016.2009.01816]

彭焕峰(1978－),男,山东临沂人,博士生,副教授,CCF 专业会员,主要研究领域为云计算与服务计算,隐私保护,软件形式化验证.

黄志球(1965－),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为云计算与服务计算,模型检测,嵌入式软件安全性,软件形式化验证.

刘林源(1981－),男,博士,讲师,主要研究领域为云计算与服务计算,系统可靠性与安全.

李勇(1983－),男,博士生,讲师,主要研究领域为实证软件工程,机器学习.

柯昌博(1984－),男,博士,讲师,CCF 专业会员,主要研究领域为基于本体的软件工程,SaaS 服务中的隐私增强技术.