Hindawi
Mobile Information Systems
Volume 2017, Article ID 7469342, 13 pages
https://doi.org/10.1155/2017/7469342

Research Article

Distributed Secure Service Composition with Declassification in Mobile Clouds

Ning Xi, Di Lu, Cong Sun, Jianfeng Ma, and Yulong Shen

School of Computer Science and Technology and School of Cyber Engineering, Xidian University, Xian, China

Correspondence should be addressed to Di Lu; nijino2002@163.com

Received 7 October 2016; Accepted 21 December 2016; Published 14 February 2017

Academic Editor: Changqiao Xu

The regional and dynamic characteristics of mobile clouds pose a great challenge on information flow security during service composition. Although secure verification approaches based on standard noninterference provide a solid assurance on information flow security of composite service, too strict constraints on service components may cause the failure of composition procedure. In order to ensure the availability of composite service, we specify the declassification policies based on cryptographic operations to allow data to be legally declassified. And we propose the improved distributed secure service composition framework and approach, which can realize different cloud platforms in multiple domains, cooperate with each other to complete the declassification, and secure composition procedure. Through the experiment and evaluation, it is indicated that our approach provides a more reliable and efficient way for secure service composition in mobile clouds.

1. Introduction

Mobile devices (e.g., smartphone and tablet PC) are increasingly becoming more and more popular in human life as their portability, pervasive connectivity, and various applications (e.g., iPhone and Android Apps). Particularly, in recent years, more kinds of basic functions (e.g., computation, storage, and network) are offered by cloud computing as the software services for elastic management and rapid service delivery with low cost, such as SDS (Software Defined Storage) [1], SDN (Software Defined Network) [2], and cloud- based mobile Apps. With the explosion of mobile applications and the support of cloud computing, mobile computing based on clouds provides a new and promising paradigm for delivering IT services more effectively and conveniently [3]. Moreover, services provided by different clouds and mobile terminals can be composed together to form a more powerful applications [4, 5], for example, trip mode selection application composed by Positioning service, Walking Speed service, Bus Tracking service, and Arrival Estimate service [6].

However, because of the regional and heterogeneous characteristic of mobile networks, there are multiple clouds deployed in different network domains. Due to the multidomain feature of the mobile clouds, data located in different mobile terminals and domains may have different security levels, which poses a great challenge on the security of service composition across multiple mobile clouds. For instance, the personal medical records in e-health data center are with high security level, while the position of the ambulance is with lower security level. When these services are composed together for the patient's emergency, data with different security levels are transmitted among these services, respectively. If these services are composed in an insecure way, an operation in a service may transmit confidential data to a public object and cause the information leakage. Access control has been widely used for protecting sensitive information of individual service from being released to unauthorized attackers [7]. However, for a composite service in mobile networks, data may be processed by serval services from multiple clouds dynamically. Access control cannot detect the information leakage caused by the subsequent operations in other services. Therefore, information flow security is one of the major concerns about the service composition in mobile clouds.

In order to enforce the data security during the service composition, various security mechanisms have been proposed to validate the information flow in composite service based on type system, Petri nets, model checking, program

static analysis, and real-time monitoring. By using type system [8], Hutter and Volkamer [9] define a set of information flow security rules that check the service composition in a secure way during the compilation of the workflow code. Petri nets provide a formal way to model composite service and Accorsi and Wonnemann [10] can identify leaks by analyzing it. Model checking is an automatic verification way that can be used to detect information leaks [11]. Nakajima [12] embedded the lattice model into the Business Process Execution Language (BPEL), and verified the absence of invalid information flows based on model checking. Program analysis is used to construct the dependence among different inputs or outputs; then information flow control (IFC) policies can be designed according to the security requirements. There are two ways to analyze the software according to the different objects, that is, static analysis for source code and dynamic analysis for executable program. For static analysis, She et al. [13, 14] define the transformation factor to measure how likely the output would depend on the input data in different candidate services. In order to improve the accuracy of static analysis, PDG (Program Dependence Graph) is used to specify the dependence between the objects in composite service [15, 16]. Compared with static analysis, dynamic analysis is built on the real-time monitoring of executing program, which can provide more accurate way to check the illegal information flow during the running time [17]. But real-time monitoring increases the cost of service execution, which may decrease the QoS and interfere with users' experience, especially when dozens of services are composed together.

Based on the above approaches, many schemes for secure service composition among clouds are proposed to address the issues of the information leakage on cloud services. Bacon et al. [18] review a range of IFC models and implementations to identify opportunities for using IFC within a cloud computing context, including type system, static analysis, and runtime dynamic analysis. Chou [19] presents the CloudIFC (Cloud Information Flow Control) model to strictly control output information flows in cloud services. Based on the specific information flow control rules and the variables dependency obtained by static analysis, they propose a novel checking way by MapReduce to decrease the verification cost. Solanki et al. [20] develop a new access and information flow control paradigm for service based systems, namely, WS-AIFC, to secure the information flow among services. Based on the dependence list for each data object, WS-AIFC supports flexible cross-domain access and information flow validation. Considering multiple domain nature of clouds, we [21] propose a distributed information flow security verification framework and approach to provide a better load balance and reduce the verification cost effectively across multiple clouds.

Although the above approaches provide a solid assurance on information flow security of composite service, implementing these IFC policies in real applications is still a challenge. These policies aim at standard noninterference that characterizes the complete absence of any information flow or any causal flow from high-level entities to low level ones. However, this requirement is too strict that few services can satisfy it in real application. If all the candidate services fail in the verification, there is no available execution path, which causes the failure of the whole composite service. Meanwhile, in mobile clouds, services are bound together in a dynamic way during service composition, which means the security sensitivities of the input and output data may change when mobile terminal move into a new domain. Considering dozens of candidate services with similar service function, it will be a complex work on selecting appropriate components to compose users required application by type system, global model checking, or centralized static analysis. For type system, when user's initial inputs change, the service codes need to be rebuilt, which brings extra cost for the secure service composition. For global model checking and centralized static analysis, it is impractical to employ a centralized entity in multiple clouds to verify the information flow security. Moreover, the cost of verification can increase rapidly when the application involves more components and the number of the candidate services increases. First, the same service component has to be reverified in different composite services. Second, the state explosion problem arises if each service component is complicated.

Therefore, a distributed and efficient information flow control mechanism supporting declassifying or downgrading information is needed for the secure and reliable service composition in mobile clouds. Compared to the paper [22], we provide the following new extensions. Firstly, mobile cloud is a more complex scenario, which involves the cooperation of different cloud platforms in multiple domains during the composition, and we add more related works for a clear description. Secondly, we give more specific definitions on declassification operations and design an improved formal information flow security model supporting declassification. Thirdly, considering the limited energy and computing resource of mobile terminal, we improve the distributed secure service composition framework and algorithms for the involvement of cloud platforms, which can take over some load on service verification. Besides, more experiments and evaluations are executed for a deep analysis on our approach.

The rest of the paper is structured as follows. Section 2 gives a formal definition of the service chain model in mobile clouds. Section 3 presents the improved computation rules with declassifying information flow in service chain. In Section 4, we propose the secure service composition with declassification mechanism for service chain in mobile clouds. Section 5 evaluates the proposed approach. Section 6 concludes the paper.

2. Preliminaries

As shown in Figure 1, mobile cloud MC is a large-scale distributed environment which consists of multiple heterogeneous domains; that is, MC $= \{d_0, d_1, \ldots\}$. Domain d has various types of data resources R. And services provided by mobile terminals MT or cloud platforms can be CP composed into a more powerful application according to the different customer's requirement. For a clear description, each service provided by either terminals or clouds can be uniformly regarded as a service node in the domain; that is,

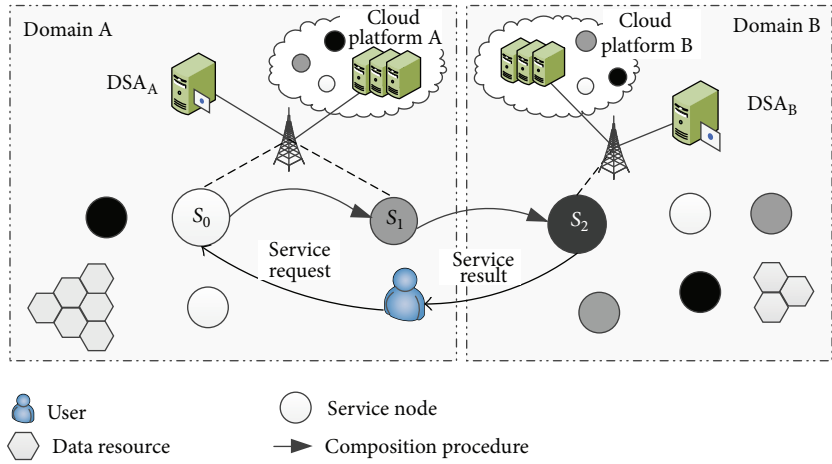

Figure 1: Service composition in mobile clouds.

SN $= \{sn_0, sn_1, \ldots\}$. There is also a security authority DSA in each domain for the management on security policies expressed by domain certificate DCe. Due to the limited energy and computation resources of mobile terminal, there is a cloud platform CP for processing more complex tasks. So domain d can be represented as $d = \langle SN, R, SA, DCe, CP\rangle$.

Referring to the definition in [21], each service s_i provided by service node SN can be represented as a tuple $s_i = \langle dom_i, In_i, Out_i, F_i, SCe_i\rangle$, where dom_i is the domain s_i belongs to; In_i is the input set of service; Out_i is the output set of service; F_i is the service function. SCe_i is the service certificate which specifies the security properties. For each service s_i, there is $In_i = \{In_i^M, In_i^D, In_i^L\}$, where In_i^M is the set of all inputs that s_i receives from its predecessor s_{i-1}; In_i^D is all the inputs from the domain resources $dom \cdot R$; In_i^L is all inputs from service node itself. In the same way, there is $Out_i = \{Out_i^M, Out_i^D, Out_i^L\}$, where Out_i^M is the set of all outputs that s_i sends to its successor s_{i+1}. Out_i^D is all outputs updated to the domain resources $dom \cdot R$. Out_i^L is all outputs written to its local storage.

Service chain SC is a simplified composite service with sequence structure, which can be represented as $SC = \langle CH, In_c, Out_c\rangle$. CH is the execution chain of services $\langle s_0, s_1, \ldots, s_{n+1}\rangle$. In CH, each service s_{i+1} only has one predecessor s_{i-1} and one successor s_{i+1}. For a clear description, s_0 and s_{n+1} are used to denote the initial user. In_{ch} and Out_{ch} are the inputs and outputs of SC including all the service components; that is, $In_{ch} = \cup\{In_i^M \cup In_i^D \cup In_i^L\}$, $0 \le i \le n+1$.

Due to the complex operations in service chain and dynamic network environment, the inner-service dependency $Dep_{inner}(o)$ and interservice dependency $Dep_{inter}(o)$ are defined to represent the flows between different inputs and outputs based on Program Dependence Graph (PDG) [16].

3. Secure Information Flow Model with Declassification in Service Chain

3.1. Multilevel Security Model. In order to represent different sensitivities of data resources in mobile clouds, multilevel security model is defined as $\langle SL, \le\rangle$, where SL is a finite set of security levels that is totally ordered by $\le$ [23].

For each input or output object o in s_i, we define Re: $In_i \cup Out_i \to SL_{ex}$ maps o to the required security level of data stored in it, while Pr: $In_i \cup Out_i \to SL_{ex}$ maps o to its clearance level which represents o can access the corresponding-level data. The required security levels will be computed according to the dependence of the input and output data, which is described as computation rules in the following sections. The clearance levels are provided by the objects who want to access the data, which can be specified in service certificates.

3.2. Secure Information Flow with Standard Noninterference. For data with different security requirements, the computation rules (CRs) on required security level are defined in [16] as follows:

CR1. $\forall u \in In_i^D \cup In_i^L, Re(u) = Pr(u)$.

CR2. $\forall u \in In_i^M, Re(u) = Re(u)$ where $v \in Out_{i-1}^M \wedge v \in Dep_{inter}(u)$.

CR3. $\forall u \in Out_i, Re(u) = \sqcup_{max} Re(v) v \in In_i \cup Out_j$, $j \le i$, and $v \in Dep(u) \cup Dep_{inter}(u)$.

Based on the standard noninterference, we propose a strong security definition on information flow for composite service in [16].

Definition 1. The information flow in service chain SC is considered secure if $\forall u \in In_{ch} \cup Out_{ch}$, u satisfies $Pr(u) \ge Re(v)$, where $v \in In_{ch} \cup Out_{ch}$ and $v \in Dep_{inner}(u) \cup Dep_{inner}(u)$.

In this definition, it is considered secure when there is no flow from a high-level object to another low level one across all service components. However, the strong security constraints enforce the fact that the flow of information must comply with the security level ordering and do not tolerate any exceptions. To deal with real application, with the execution of the composite service, the required security levels of inputs or outputs become higher and higher according to the above CRs, which is so strict that fewer candidate service components can satisfy. In this case, it would lead to

a high failure rate on service composition. Therefore, more general flow policy allowing data declassification needs to be proposed to improve the availability of composite service.

3.3. Secure Service Composition Model with Cryptographic Operations. Due to the strong security condition, declassification operations are needed for the secure service composition. Cryptographic operations are promising ways of maintaining data confidentiality and integrity, for example, encryption and digital signature. Through the cryptographic operations, processed secret data can be transmitted into a public object, which realizes the declassification of data. Therefore, extra cryptographic operations $\mathrm{En}(o, \mathrm{key})$ and $\mathrm{De}(o, \mathrm{key})$ can be add to the service function F_i for each service.

$$
\begin{aligned}
f &::= a; f \\
a &::= \mathrm{skin} \mid \mathrm{input}(\mathrm{var}, e) \mid \mathrm{var}_p := e \mid a; a \\
&\quad \mid \text{if } (e) \text{ then } a \text{ else } a \\
&\quad \mid \text{while } (e)\ a \mid \mathrm{output}(\mathrm{var}, e) \\
&\quad \mid \mathrm{En}(\mathrm{var}, \mathrm{key}) \mid \mathrm{De}(\mathrm{var}, \mathrm{key}) \\
e &::= \mathrm{var}_p \mid e\,\mathrm{Re} \\
R &::= + \mid - \mid = \mid <
\end{aligned}
$$

For $\forall \mathrm{var} \in \mathrm{In}_i \cup \mathrm{Out}_i$, var_p and var_c represent the plaintext and ciphertext of var, and the encryption and decryption operation on var are defined as $\mathrm{En}(\mathrm{var}, \mathrm{key})$ and $\mathrm{De}(\mathrm{var}, \mathrm{key})$. Because of the low efficiency on homomorphic encryption [24], the traditional cryptographic operations are considered in this paper. As shown in F_i's definition, the classified data var_c cannot be directly processed by regular operations which may cause the plaintext of var to not be recovered. But the basic input, output, encryption, and decryption are still supported by F_i for classified data var_c.

When the data in var is encrypted, it provides more secure way to transmit var, and the attacker needs to work harder to crack the ciphertext which depends on the security of encryption algorithm E and the key *key*. Thus we use $\mathrm{Re}(\langle E, \mathrm{key}\rangle)$ to represent the security level of classified data var_c. Encryption with more complex algorithm and key means $\mathrm{Re}(\langle E, \mathrm{key}\rangle)$ is lower. And the security level of var with reencryption depends on the strongest algorithm and key. When var_c is decrypted, the data of var is no longer protected by encryption, and the security level of var returns to its original value. According to the analysis above, we can extend the basic computation rules as follows:

CR4. $\forall u \in \mathrm{In}_i \cup \mathrm{Out}_i$, if u is encrypted by $\langle E, \mathrm{key}\rangle$, there is $\mathrm{Re}(u) = \mathrm{Re}(u_p) \wedge \mathrm{Re}(\langle E, \mathrm{key}\rangle)$.

CR5. For the ciphertext u_c, if u is decrypted, there is $\mathrm{Re}(u) = \mathrm{Re}(\mathrm{De}(u_c, \mathrm{key})) = \mathrm{Re}(u_p)$.

In traditional definition on standard noninterference, high security level data are not allowed to transfer to an object with lower level. The encryption operation may violate the requirements on standard noninterference. But the attacker still cannot obtain the sensitive data if he cannot crack the ciphertext, which is still considered secure although the sensitive data is transferred to an object with lower clearance. In order to specify the special downgrading flow in composite service, an extended definition on inner dependence is proposed as follows.

Definition 2. For $\forall u \in \mathrm{Out}_i, \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc,Dec}}(u)$ represents the set of inputs that u depends on, where Enc is the pair of encryption algorithm and key that u adopts; Dec is the pair of decryption algorithm and key that dependent inputs adopt. Then $\forall v \in \mathrm{In}_i$ and $\wedge v \in \mathrm{Dep}_{\mathrm{inner}}(u)$, there are four cases to consider:

(1) v is plaintext and u outputs as the plaintext; there is $v \in \mathrm{Dep}^{\mathrm{inner}}_{\phi,\phi}(u)$.

(2) v is plaintext but u outputs as the ciphertext encrypted by $\langle E_u, \mathrm{key}_u\rangle$; there is $v \in \mathrm{Dep}^{\mathrm{inner}}_{\langle E_u, \mathrm{key}_u\rangle,\phi}$.

(3) v is ciphertext but u outputs as the plaintext; it means v is decrypted with $\langle E_v, \mathrm{key}_v\rangle$ during the execution of service. Then there is $v \in \mathrm{Dep}^{\mathrm{inner}}_{\phi,\langle E_v, \mathrm{key}_v\rangle}$.

(4) v is ciphertext and u also outputs as the ciphertext; there are three different cases:

  (1) If v is decrypted with $\langle E_v, \mathrm{key}_v\rangle$ during the execution of service, it means v is operated as plaintext and u is encrypted by another encryption algorithm and key. Then there is $v \in \mathrm{Dep}^{\mathrm{inner}}_{\langle E_u, \mathrm{key}_u\rangle,\langle E_v, \mathrm{key}_v\rangle}$.

  (2) If v is not decrypted but u is reencrypted by $\langle E_u, \mathrm{key}_u\rangle$, we can obtain $v \in \mathrm{Dep}^{\mathrm{inner}}_{\langle E_u, \mathrm{key}_u\rangle,\phi}$.

  (3) If v is not decrypted and u is not reencrypted, there is $v \in \mathrm{Dep}^{\mathrm{inner}}_{\phi,\phi}$.

Based on the extend inner dependence, interdependence can be defined recursively as follows.

Definition 3. $\forall u \in \mathrm{In}^M_i \cup \mathrm{Out}_i, \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(u)$ represents the set of inputs or outputs in different services that u depends on. EnS is the set of pairs of the encryption algorithm and key that is used during the execution path, while DeS represents the set of all decryption operations. For each $v \in \mathrm{In}_j \cup \mathrm{Out}^M_j, v \in \mathrm{Dep}_{\mathrm{inter}}(u), 0 \le j < i \le N$, there are three cases to consider:

(1) $i = j + 1$: $\forall u \in \mathrm{In}^M_i$ and $v \in \mathrm{Out}^M_j$, if $v \in \mathrm{Dep}_{\mathrm{inter}}(u)$, there is $v \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(u)$.

(2) $i = j + 1$: $\forall u \in \mathrm{In}^M_i$ and $v \in \mathrm{In}^M_j$, if $\exists w \in \mathrm{Out}^M_j$, $v \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc}_v,\mathrm{Dec}_v}(w)$, and $w \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS}_w,\mathrm{DeS}_w}(u)$, there is $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(u)$, where $\mathrm{EnS} = \mathrm{Enc}_v \cup \mathrm{EnS}_w$ and $\mathrm{DeS} = \mathrm{Dec}_v \cup \mathrm{DeS}_w$.

(3) $i = j + 1$: $\forall u \in \mathrm{Out}_i, v \in \mathrm{Out}^M_j$, if $\exists w \in \mathrm{In}^M_i, v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS}_v,\mathrm{DeS}_v}(w)$ and $w \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{Enc}_w,\mathrm{Dec}_w}(u)$, there is $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(u)$, where $\mathrm{EnS} = \mathrm{EnS}_v \cup \mathrm{Enc}_w$ and $\mathrm{DeS} = \mathrm{DeS}_v \cup \mathrm{Dec}_w$.

(4) $i = j + 1$: $\forall u \in \mathrm{Out}_i$ and $v \in \mathrm{In}_j$, if $\exists w_1 \in \mathrm{Out}^M_j, w_2 \in \mathrm{In}^M_i$, $v \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc}_v,\mathrm{Dec}_v}(w_1)$, $w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS}_{w1},\mathrm{DeS}_{w1}}(w_2)$

and $w_2 \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc}_{w2},\mathrm{Dec}_{w2}}(u)$, there is $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS},\mathrm{DeS}}(u)$, where $\mathrm{EnS} = \mathrm{Enc}_v \cup \mathrm{EnS}_{w1} \cup \mathrm{Enc}_{w2}$ and $\mathrm{DeS} = \mathrm{Dec}_v \cup \mathrm{Des}_{w1} \cup \mathrm{Dec}_{w2}$.

(5) $i > j + 1$: $\forall u \in \mathrm{In}^M_i$ and $v \in \mathrm{In}_j \cup \mathrm{Out}^M_j$, if $\exists w \in \mathrm{In}_k \cup \mathrm{Out}_k$, $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{Ens}_v,\mathrm{Des}_v}(w)$, and $w \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{Ens}_w,\mathrm{Des}_w}(w)$, $j < k < i$, there is $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS},\mathrm{DeS}}(u)$, where $\mathrm{EnS} = \mathrm{EnS}_v \cup \mathrm{EnS}_w$ and $\mathrm{DeS} = \mathrm{DeS}_v \cup \mathrm{DeS}_w$.

Based on the extend inner and interdependence, the improved security definition on information flow for composite service can be presented as follows.

Definition 4. The information flow in service chain SC is considered secure if $\forall u, v \in \mathrm{In}_c \cup \mathrm{Out}_c$, $v \in \mathrm{Dep}_{\mathrm{inner}}(u) \cup \mathrm{Dep}_{\mathrm{inter}}(u)$ satisfies the following conditions:

(1) $\forall u, v \in \mathrm{In}_i \cup \mathrm{Out}_i$, and $v \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc},\mathrm{Dec}}(u)$,

(i) if $\mathrm{Enc} - \mathrm{Dec} = \phi$, there is $\Pr(u) \geq \mathrm{Re}(v)$;
(ii) if $\mathrm{Enc} - \mathrm{Dec} \neq \phi$, there is $\Pr(u) \geq \mathrm{Re}(\langle E_u, \mathrm{key}_u \rangle)$, where $\langle E_u, \mathrm{key}_u \rangle \in \mathrm{Enc} - \mathrm{Dec}$.

(2) $\forall u \in \mathrm{In}_i \cup \mathrm{Out}_i$, $v \in \mathrm{In}_j \cup \mathrm{Out}_j$, $0 \leq j < i \leq N$, and $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{Ens},\mathrm{Des}}(u)$,

(i) if $\mathrm{EnS} - \mathrm{DeS} = \phi$, there is $\Pr(u) \geq \mathrm{Re}(v)$;
(ii) if $\mathrm{EnS} - \mathrm{DeS} \neq \phi$, there is $\Pr(u) \geq \prod^{1 \leq x \leq N_e}_{\min} \mathrm{Re}(\langle E_x, \mathrm{key}_x \rangle)$, where $\langle E_x, \mathrm{key}_x \rangle \in \mathrm{Ens} - \mathrm{Des}$.

According to Definition 4, two different types of flow are considered separately, that is, unclassified and classified flow. For the unclassified flow, it must satisfy the traditional information noninterference constraints, that is, the clearance on each input or output in s_i must be no less than the required security level, which depends on all related inputs and outputs in s_i and its predecessor. For the classified flow, data security depends on the encryption operation, so it can be considered secure that the clearance of the input or output is equal or greater than the required security level of the strongest encryption operation.

Based on improved information flow security definition, we can deduce the security constraints on each service as the following theorem.

Theorem 1. *The information flow in service chain SC with N steps is considered secure if each s_i in SC satisfies the following conditions:*

(1) $\forall u \in Out_i, v \in In_i$, *and* $v \in Dep_{\mathrm{inner}}(u)$,

(a) *if u is not encrypted, there is* $\Pr(u) \geq \mathrm{Re}(v)$;
(b) *if u is encrypted by* $\langle E_u, key_u \rangle$, *there is* $\Pr(u) \geq \mathrm{Re}(\langle E_u, key_u \rangle)$.

(2) $\forall u \in In^M_i, v \in Out^M_{i-1}$, *and* $v \in Dep_{\mathrm{inter}}(u)$,

(a) *if u is not encrypted, there is* $\Pr(u) \geq \mathrm{Re}(v)$;
(b) *if u is encrypted by* $\langle E, key \rangle$, *there is* $\Pr(u) \geq \mathrm{Re}(\langle E, key \rangle)$.

Proof. First, let $N = 1$; then there are two service components involved in the service chain, that is, s_0 and s_1. □

Case 1. Inner information flow in each service component is considered first; that is, $\forall u \in \mathrm{Out}_0$, $v \in \mathrm{In}_0$, and $v \in \mathrm{Dep}_{\mathrm{inner}}(u)$.

(1) Condition (1)(a) provides that for each $v \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc},\mathrm{Dec}}(u)$ where $\mathrm{Enc} - \mathrm{Dec} = \phi$, there is $\Pr(u) \geq \mathrm{Re}(v)$.

(2) Condition (1)(b) provides that for each $v \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc},\mathrm{Dec}}(u)$ where $\mathrm{Enc} - \mathrm{Dec} \neq \phi$, there is $\Pr(u) \geq \mathrm{Re}(\langle E_u, \mathrm{key}_u \rangle)$.

In the same way, we can get the information flow is also secure in s_1.

Case 2. Information flow between s_0 and s_1 is considered; that is, $\forall u \in \mathrm{In}_1 \cup \mathrm{Out}_1$, $v \in \mathrm{In}_0 \cup \mathrm{Out}_0$, and $v \in \mathrm{Dep}_{\mathrm{inter}}(u)$.

(1) $\forall u \in \mathrm{In}_1$, $v \in \mathrm{Out}_0$, and $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS},\mathrm{DeS}}(u)$, according to Definition 3(1), there is $v \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{EnS},\mathrm{DeS}}(u)$ where $\mathrm{EnS} = \phi$ and $\mathrm{DeS} = \phi$, and condition (2) provides $\Pr(u) \geq \mathrm{Re}(v)$.

(2) $\forall u \in \mathrm{In}_1$, $v \in \mathrm{In}_0$, and $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS},\mathrm{DeS}}(u)$, according to Definition 3(2), there is $\exists w \in \mathrm{Out}^M_j$, $v \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc}_v,\mathrm{Dec}_v}(w)$, and $w \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(u)$.

(i) If $\forall v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS},\mathrm{DeS}}(u)$ satisfies $\mathrm{EnS} - \mathrm{DeS} = \phi$, w is not encrypted. Condition (1)(a) provides $\Pr(w) \geq \mathrm{Re}(v)$, and condition (2)(a) provides $\Pr(u) \geq \mathrm{Re}(w)$. Therefore, $\Pr(u) \geq \mathrm{Re}(v)$.
(ii) If $\forall v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS},\mathrm{DeS}}(u)$ satisfies $\mathrm{EnS} - \mathrm{DeS} \neq \phi$, w is encrypted by $\langle E_w, \mathrm{key}_w \rangle$. There is $\mathrm{EnS} - \mathrm{DeS} = \{\langle E_w, \mathrm{key}_w \rangle\}$. Condition (1)(b) provides $\Pr(w) \geq \mathrm{Re}(\langle E_w, \mathrm{key}_w \rangle)$. Condition (2)(a) and CR 4 provide $\Pr(u) \geq \mathrm{Re}(w) = \mathrm{Re}(\langle E_w, \mathrm{key}_w \rangle)$.

(3) $\forall u \in \mathrm{Out}_1$ and $v \in \mathrm{Out}_0$, according to Definition 3(3), there is $\exists w \in \mathrm{In}^M_1$ and $v \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(u), w \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc}_w,\mathrm{Dec}_w}(u)$.

(i) If $\forall v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS},\mathrm{DeS}}(u)$ satisfies $\mathrm{EnS} - \mathrm{DeS} = \phi$, u is not encrypted. Condition (1)(a) provides $\Pr(u) \geq \mathrm{Re}(w)$, and condition (2)(a) a provides $\Pr(w) \geq \mathrm{Re}(v)$. Therefore, $\Pr(u) \geq \mathrm{Re}(v)$.
(ii) If $\forall v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS},\mathrm{DeS}}(u)$ satisfies $\mathrm{EnS} - \mathrm{DeS} \neq \phi$, u is encrypted by $\langle E_u, \mathrm{key}_u \rangle$. There is $\mathrm{EnS} - \mathrm{DeS} = \{\langle E_u, \mathrm{key}_u \rangle\}$. Condition (1)(b) provides $\Pr(u) \geq \mathrm{Re}(\langle E_u, \mathrm{key}_u \rangle)$.

(4) $\forall u \in \mathrm{Out}_1$, and $v \in \mathrm{In}_0$, according to Definition 3(4), there is $\exists w_1 \in \mathrm{Out}^M_0$, $w_2 \in \mathrm{In}^M_1$, $v \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc}_v,\mathrm{Dec}_v}(w_1)$, $w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\varphi,\varphi}(w_2)$, and $w_2 \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc}_{w2},\mathrm{Dec}_{w2}}(u)$, and there is $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS},\mathrm{DeS}}(u)$ where $\mathrm{EnS} = \mathrm{Enc}_v \cup \mathrm{Enc}_{w2}$ and $\mathrm{DeS} = \mathrm{Dec}_v \cup \mathrm{Dec}_{w2}$.

(i) If $\forall v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS},\mathrm{DeS}}(u)$ satisfies $\mathrm{EnS} - \mathrm{DeS} = \phi$, there are two different cases:

(a) For $\mathrm{Enc}_v \cup \mathrm{Enc}_{w2} = \phi$ and $\mathrm{Dec}_v \cup \mathrm{Dec}_{w2} = \phi$, CR 3 provides $\mathrm{Re}(w_1) \geq \mathrm{Re}(v)$. Condition (1)(a) provides $\Pr(u) \geq \mathrm{Re}(w_2)$ and condition (2)(a) provides $\Pr(w_2) \geq \mathrm{Re}(w_2) = \mathrm{Re}(w_1)$. Therefore, $\Pr(w) \geq \mathrm{Re}(v)$.

(b) For $\mathrm{Enc}_v = \{\langle E_v, \mathrm{key}_v\rangle\}$, $\mathrm{Dec}_v = \phi$, $\mathrm{Enc}_{w2} = \phi$, and $\mathrm{Dec}_{w2} = \{\langle E_v, \mathrm{key}_v\rangle\}$, CR 5 provides $\mathrm{Re}(w_{2_p}) \geq \mathrm{Re}(v)$ and condition (1)(a) provides $\Pr(u) \geq \mathrm{Re}(w_{2_p})$, so $\Pr(u) \geq \mathrm{Re}(v)$.

(ii) If $\forall v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(u)$ satisfies $\mathrm{EnS} - \mathrm{DeS} \neq \phi$, there are four different cases:

(a) For $\mathrm{Enc}_v = \{\langle E_v, \mathrm{key}_v\rangle\}$, $\mathrm{Dec}_v = \phi$, $\mathrm{Enc}_{w2} = \phi$, and $\mathrm{Dec}_{w2} = \phi$ where $\mathrm{EnS} - \mathrm{DeS} = \{\langle E_v, \mathrm{key}_v\rangle\}$, condition (2)(b) provides $\Pr(u) \geq \mathrm{Re}(\langle E_v, \mathrm{key}_v\rangle)$.

(b) For $\mathrm{Enc}_v = \phi$, $\mathrm{Dec}_v = \phi$, $\mathrm{Enc}_{w2} = \{\langle E_{w2}, \mathrm{key}_{w2}\rangle\}$, and $\mathrm{Dec}_{w2} = \phi$ where $\mathrm{EnS} - \mathrm{DeS} = \{\langle E_{w2}, \mathrm{key}_{w2}\rangle\}$, condition (2)(b) provides $\Pr(u) \geq \mathrm{Re}(\langle E_v, \mathrm{key}_v\rangle)$.

(c) For $\mathrm{Enc}_v = \{\langle E_v, \mathrm{key}_v\rangle\}$, $\mathrm{Dec}_v = \phi$, $\mathrm{Enc}_{w2} = \{\langle E_{w2}, \mathrm{key}_{w2}\rangle\}$, and $\mathrm{Dec}_{w2} = \{\langle E_v, \mathrm{key}_v\rangle\}$ where $\mathrm{EnS} - \mathrm{DeS} = \{\langle E_{w2}, \mathrm{key}_{w2}\rangle\}$, condition (2)(b) provides $\Pr(u) \geq \mathrm{Re}(\langle E_v, \mathrm{key}_v\rangle)$.

(d) For $\mathrm{Enc}_v = \{\langle E_v, \mathrm{key}_v\rangle\}$, $\mathrm{Dec}_v = \phi$, $\mathrm{Enc}_{w2} = \{\langle E_{w2}, \mathrm{key}_{w2}\rangle\}$, and $\mathrm{Dec}_{w2} = \phi$ where $\mathrm{EnS} - \mathrm{DeS} = \{\langle E_v, \mathrm{key}_v\rangle, \langle E_{w2}, \mathrm{key}_{w2}\rangle\}$, condition (2)(b) provides $\Pr(u) \geq \mathrm{Re}(\langle E_{w2}, \mathrm{key}_{w2}\rangle) \geq \min\{\mathrm{Re}(\langle E_v, \mathrm{key}_v\rangle), \mathrm{Re}(\langle E_{w2}, \mathrm{key}_{w2}\rangle)\}$.

Based on the above analysis and Definition 4, information flow between s_0 and s_1 is secure.

Therefore, Theorem 1 is true when $N = 1$.

Then we assume Theorem 1 is true when $N = n$, and the proof on $N = n + 1$ is presented as follows.

Case 1. Inner information flow in service component s_{n+1} is considered; that is, $\forall u \in \mathrm{Out}_{n+1}, v \in \mathrm{In}_{n+1}$, and $v \in \mathrm{Dep}_{\mathrm{inner}}(u)$.

(1) Condition (1)(a) provides that for each $v \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc,Dec}}(u)$ where $\mathrm{Enc} - \mathrm{Dec} = \phi$, there is $\Pr(u) \geq \mathrm{Re}(v)$.

(2) Condition (1)(b) provides that for each $v \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc,Dec}}(u)$ where $\mathrm{Enc} - \mathrm{Dec} \neq \phi$, there is $\Pr(u) \geq \mathrm{Re}(\langle E_u, \mathrm{key}_u\rangle)$.

And above assumption provides that information flow in $s_0, s_1, \ldots, s_n$ is secure.

Case 2. The assumption provides that information flow among first n service step is secure. Then the interinformation flows between s_{n+1} and former services are considered; that is, $\forall u \in \mathrm{In}_{n+1} \cup \mathrm{Out}_{n+1}$, $v \in \mathrm{In}_j \cup \mathrm{Out}_j$, and $v \in \mathrm{Dep}_{\mathrm{inter}}(u)$, $0 \leq j \leq n + 1$.

According to Definition 3(5) and Lemma 1 in [16], there is $\exists w_1 \in \mathrm{Out}^M_n$, $w_2 \in \mathrm{In}^M_{n+1}$, $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(w_1)$, $w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\varphi}(w_2)$, and $w_2 \in \mathrm{Dep}^{\mathrm{inner}}(\mathrm{Enc}, \mathrm{Dec})(u)$.

(1) For $u \in \mathrm{In}_{n+1}$ there is $u = w_2$.

(i) If $\forall v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(u)$ satisfies $\mathrm{EnS} - \mathrm{DeS} = \phi$, there is for $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(w_1)$, and $w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(u)$ where $\mathrm{EnS}_v - \mathrm{DeS}_v = \phi$. Condition (2)(a) provides $\Pr(u) \geq \mathrm{Re}(w_1)$ and the assumption provides $\mathrm{Re}(w_1) \geq \mathrm{Re}(v)$. So there is $\Pr(u) \geq \mathrm{Re}(v)$.

(ii) If $\forall v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(u)$ satisfies $\mathrm{EnS} - \mathrm{DeS} \neq \phi$, there is for $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(w_1)$ and $w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(u)$ where $\mathrm{EnS} - \mathrm{DeS} = \{\langle E_{vi}, \mathrm{key}_{vi}\rangle, 1 \leq i \leq n\}$. Condition (2)(b) provides $\Pr(u) \geq \mathrm{Re}(\langle E_n, \mathrm{key}_n\rangle)$. So there is $\Pr(u) \geq \bigcap^{1\leq i\leq n}_{\min} \mathrm{Re}(\langle E_{vi}, \mathrm{key}_{vi}\rangle)$.

(2) For $u \in \mathrm{Out}_{n+1}$, the following cases are considered:

(i) If $\forall v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(u)$ satisfies $\mathrm{EnS} - \mathrm{DeS} = \phi$, there are two cases:

(a) For $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(w_1)$, and $w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(w_2), w_2 \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc,Dec}}(u)$ where $\mathrm{EnS}_v - \mathrm{DeS}_v = \phi$ and $\mathrm{Enc}_{w2} - \mathrm{Dec}_{w2} = \phi$, condition (1)(a) provides $\Pr(u) \geq \mathrm{Re}(w_2)$. CR 2 provides $\mathrm{Re}(w_1) = \mathrm{Re}(w_2)$. The assumption provides $\mathrm{Re}(w_1) \geq \mathrm{Re}(v)$. So there is $\Pr(u) \geq \mathrm{Re}(v)$.

(b) For $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(w_1), w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(w_2)$, and $w_2 \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc,Dec}}(u)$ where $\mathrm{EnS}_v - \mathrm{DeS}_v = \{\langle E_v i, \mathrm{key}_v i\rangle, 1 \leq i \leq n\}$ and $\mathrm{Dec}_{w2} = \{\langle E_{vi}, \mathrm{key}_{vi}\rangle, 1 \leq i \leq n\}$, CR 5 provides $\mathrm{Re}(w_{2_p}) \geq \mathrm{Re}(v)$ and condition (1)(a) provides $\Pr(u) \geq \mathrm{Re}(w_{2_p})$, so there is $\Pr(u) \geq \mathrm{Re}(v)$.

(ii) If $\forall v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(u)$ satisfies $\mathrm{EnS} - \mathrm{DeS} \neq \phi$, there are five cases:

(a) For $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(w_1), w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(w_2)$, and $w_2 \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc,Dec}}(u)$ where $\mathrm{EnS}_v - \mathrm{DeS}_v = \{\langle \mathrm{E}_{vi}, \mathrm{key}_{vi}\rangle, 1 \leq i \leq n\}$ and $\mathrm{Enc}_{w2} - \mathrm{Dec}_{w2} = \phi$, condition (1)(b) provides $\Pr(u) \geq \mathrm{Re}(\langle E_n, \mathrm{key}_n\rangle)$. So there is $\Pr(u) \geq \bigcap^{1\leq i\leq n}_{\min} \mathrm{Re}(\langle E_{vi}, \mathrm{key}_{vi}\rangle)$.

(b) For $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(w_1), w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(w_2)$, and $w_2 \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc,Dec}}(u)$ where $\mathrm{EnS}_v - \mathrm{DeS}_v = \{\langle E_{vi}, \mathrm{key}_{vi}\rangle, 1 \leq i \leq n\}$ and $\mathrm{Enc}_{w2} - \mathrm{Dec}_{w2} = \{\langle E_{w2}, \mathrm{key}_{w2}\rangle\}$, there is $\mathrm{EnS} - \mathrm{DeS} = \{\langle E_{vi}, \mathrm{key}_{vi}\rangle, 1 \leq i \leq n\} \cup \langle E_{w2}, \mathrm{key}_{w2}\rangle$. Condition (1)(b) provides $\Pr(u) \geq \mathrm{Re}(\langle E_{w2}, \mathrm{key}_{w2}\rangle)$. So there is $\Pr(u) \geq \bigcap^{1\leq i\leq n}_{\min} \mathrm{Re}(\langle E_{vi}, \mathrm{key}_{vi}\rangle)$.

(c) For $\in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(w_1), w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(w_2)$, and $w_2 \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc,Dec}}(u)$ where $\mathrm{EnS}_v - \mathrm{DeS}_v = \{\langle E_{vi}, \mathrm{key}_{vi}\rangle, 1 \leq i \leq n\}$ and $\mathrm{Enc}_{w2} - \mathrm{Dec}_{w2} = \phi$ but $\mathrm{Dec}_{w2} = \{\langle E_{vn}, \mathrm{key}_{vn}\rangle\}$, there is $\mathrm{EnS} - \mathrm{DeS} = \{\langle E_{vi}, \mathrm{key}_{vi}\rangle, 1 \leq i \leq n\}$. Condition (1)(b)

provides $\Pr(u) \geq \mathrm{Re}(\langle E_{n-1}, \mathrm{key}_{n-1}\rangle)$. So there is $\Pr(u) \geq \bigcap_{\min}^{1\leq i\leq n} \mathrm{Re}(\langle E_{vi}, \mathrm{key}_{vi}\rangle)$.

(d) For $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(w_1)$, $w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(w_2)$, and $w_2 \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc,Dec}}(u)$ where $\mathrm{EnS}_v - \mathrm{DeS}_v = \{\langle E_{vi}, \mathrm{key}_{vi}\rangle,\ 1 \leq i \leq n\}$ and $\mathrm{Enc}_{w2} - \mathrm{Dec}_{w2} = \{\langle E_{w2}, \mathrm{key}_{w2}\rangle\}$ but $\mathrm{Dec}_{w2} = \{\langle E_{vn}, \mathrm{key}_{vn}\rangle\}$, there is $\mathrm{EnS} - \mathrm{DeS} = \{\langle E_{vi}, \mathrm{key}_{vi}\rangle,\ 1 \leq i \leq n\} \cup \{\langle E_{w2}, \mathrm{key}_{w2}\rangle\}$. Condition (1)(b) provides $\Pr(u) \geq \mathrm{Re}(\langle E_{w2}, \mathrm{key}_{w2}\rangle)$. So there is $\Pr(u) \geq \bigcap_{\min}^{1\leq i\leq n} \mathrm{Re}(\langle E_{vi}, \mathrm{key}_{vi}\rangle)$.

(e) For $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(w_1)$, $w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(w_2)$, and $w_2 \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc,Dec}}(u)$ where $\mathrm{EnS}_v - \mathrm{DeS}_v = \phi$ and $\mathrm{Enc}_{w2} - \mathrm{Dec}_{w2} = \{\langle E_{w2}, \mathrm{key}_{w2}\rangle\}$, there is $\mathrm{EnS} - \mathrm{DeS} = \{\langle E_{w2}, \mathrm{key}_{w2}\rangle\}$. Condition (1)(b) provides $\Pr(u) \geq \mathrm{Re}(\langle E_{w2}, \mathrm{key}_{w2}\rangle)$.

Based on the above analysis and Definition 4, information flows between s_{n+1} and former services s_i where $i < n + 1$ are secure.

Therefore, Theorem 1 is also true when $N = n + 1$.

In conclusion, Theorem 1 is true.

Based on the above Theorem 1, we can propose an improved service composition mechanism supporting declassification operations. The specific declassification policies (DPs) are presented as follows.

DP 1. For $\forall v \in \mathrm{In}_i, u \in \mathrm{Out}_i, v \in \mathrm{Dep}_{\mathrm{inner}}(u)$, if $\Pr(u) < \mathrm{Re}(v)$, then u needs to be encrypted by $\langle E_u, \mathrm{key}_u\rangle$ which satisfies $\Pr(u) \geq \mathrm{Re}(\langle E_u, \mathrm{key}_u\rangle)$.

DP 2. For $\forall v \in \mathrm{Out}_i^M$, $u \in \mathrm{In}_{i+1}^M$, $v \in \mathrm{Dep}_{\mathrm{inter}}(u)$, if $\Pr(u) < \mathrm{Re}(v)$, then u needs to be encrypted by $\langle E_u, \mathrm{key}_u\rangle$ which satisfies $\Pr(u) \geq \mathrm{Re}(\langle E_u, \mathrm{key}_u\rangle)$.

According to the declassification policies, when the provided security level of u cannot satisfy the strict conditions, cryptographic operations are adopted to assist in declassifying the required security level which can also hold the information flow security.

4. Secure Service Composition with Declassification in Mobile Network

4.1. Secure Service Composition with Declassification Framework in Mobile Network. In the mobile cloud system with multiple domains, there are serval candidate services with same functions but different providers, which can be denoted by $s_{i,j} \in S_i$, $0 \leq i \leq N$, $0 \leq j \leq |S_i|$. Traditional secure service composition approaches are based on standard information flow verification technique where insecure candidate service is filtered. However, it may be so strict that few candidate services can satisfy in real application, which leads to the failure of service composition. Based on the declassification policies, we can propose an improved secure service composition framework supporting declassification operations, which is shown in Figure 2.

This framework is constructed as a distributed secure service composition framework involving three main kinds of entities, that is, Cloud Platform (CP), Candidate Services (CS), and Domain Security Authorities (DSA). Considering the limited energy and computation resources of mobile terminals, the verification procedure is executed by CPs. DSAs are responsible for the management on the security certificate SCe for each service node. SCe includes the provided security levels of input and outputs, the dependencies between the input and output and its public key. If the service node is fixed one, that is, services are provided by cloud platform, the certificate is generated when the service is first deployed in cloud platform. If the service node is mobile one, that is, services are provided by mobile terminal, the certificate is generated when the terminal first moves into this domain.

During the verification, all candidate services send their dynamic input data and certificates to the cloud platform to finish the verification procedure. There are two different scenarios, that is, inner-domain and interdomain verification. For inner-domain verification, candidate services CP and DSA in the same domain are involved in the verification. For interdomain verification, the participant entities include not only candidate services but also two CPs and SAs in the corresponding domains.

Comparing to the traditional verification procedure in [16], declassification based on cryptographic operations is executed automatically to recover the insecure information flows against the declassification policies. If the information flow security verification returns failure, each insecure component needs to negotiate a session key with its adjacent nodes for the encryption and decryption during the service execution. For clear description, we mainly focus on the declassification procedure in this paper.

4.2. Cryptographic Operations for Declassification in Service Composition

4.2.1. Cryptographic Operation Agent. Based on the Theorem 1, basic cryptographic operations must be supported by each service node to realize the declassification of information flow during the service composition. There are many relevant security specifications which have been proposed to protect data confidentiality and integrity during service execution, such as XML Encryption and Signature, WS-Security, SAML (Security Assertion Markup Language), XACML (XML Access Control Markup Language), and XKM- S (XML Key Management Specification) [25]. By developing the basic security functions supported by these specifications, a cryptographic operation agent (COA) can be designed and deployed in each service node, mobile terminal, or cloud platform, to execute the declassification operations, which is shown in Figure 3.

The cryptographic operation agent is composed of three function modules, that is, key negotiator, encryptor and decryptor. Key negotiator is responsible for the key management including key generation, negotiation with other services, key storage, and update. Encryptor and decryptor are responsible for data encryption and decryption during the service. There are two phases for agent to complete the declassification procedure, that is, key negotiation and data encryption and decryption.

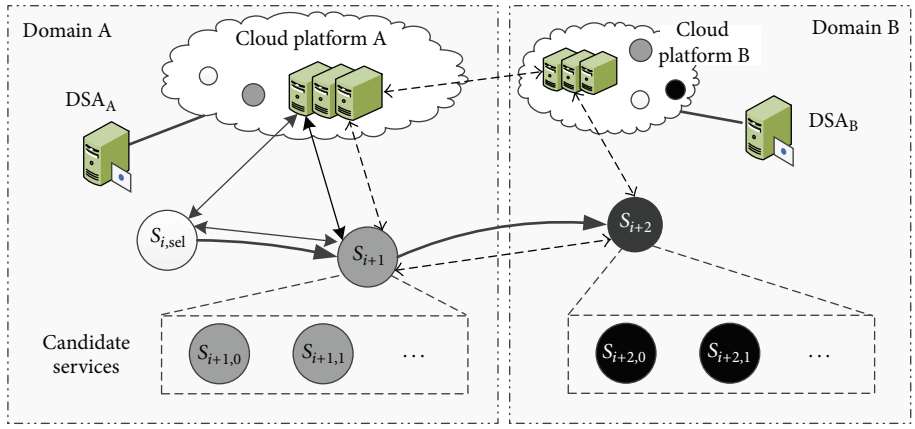

Figure 2: Distributed secure service composition with declassification framework.

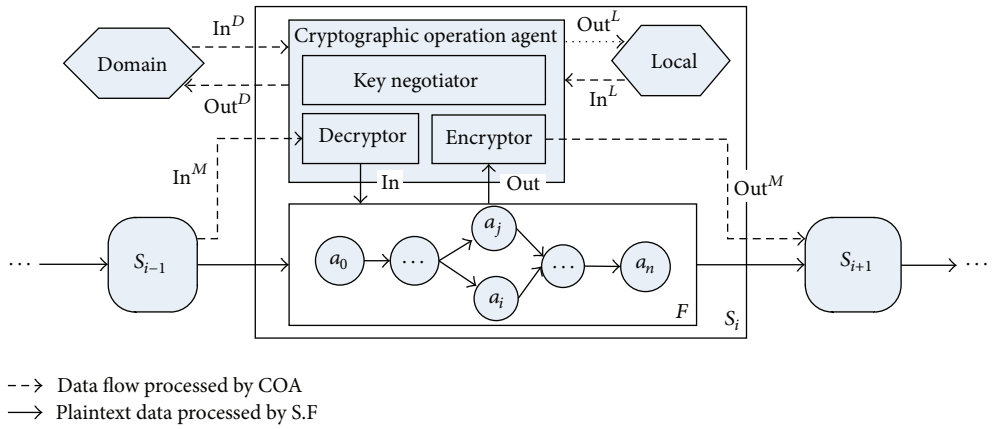

Figure 3: Cryptographic operation agent in each service node.

4.2.2. Key Negotiation Phase. Key negotiation phase is the preparation phase for the data declassification, which is also the most critical step. In this phase, for each insecure input or output $u \in \text{In}_i \cup \text{Out}_i$, related two services negotiate for generating the appropriate encryption algorithm and key $\langle E, \text{key} \rangle$ to ensure the information flow security according to DPs. There are two kinds of negotiation process due to multiple domains, that is, inner-domain negotiation and interdomain negotiation, which is shown in Figures 4 and 5. The procedure of key negotiation follows the specification of XKMS (XML Key Management Specification).

When the key negotiation begins between two adjacent service nodes, both certificates containing their own public keys are delivered to the opponents. Then the random number protected by public key is transferred to each other at the fourth and seventh step. And finally the session key *key* is computed based on these random numbers with a standard key generation algorithm. Meanwhile the encryption algorithm E can also be negotiated during this procedure. In order to ensure the information flow security in the following composition, the length of the key, the complexity of the random number, the key generation algorithm, and the encryption algorithm must satisfy the requirements on security level. The pseudocode of key negotiation is presented as Algorithm 1.

4.2.3. Data Declassification Phase. The data declassification phase is activated after the procedure of secure service composition. During the service execution, the COA encrypts the insecure inputs and outputs to realize the declassification on high-level data by using the session key. Meanwhile, it also realizes decryption on the cipher data for normal processing of service function.

4.3. Distributed Secure Service Composition with Declassification Algorithm across Multiple Mobile Clouds. During the secure service composition, cloud platform CP verifies the service chain by service step based on Theorem 1. For each candidate service s_i, CP first verifies whether the input objects satisfy the security condition, then compute the required

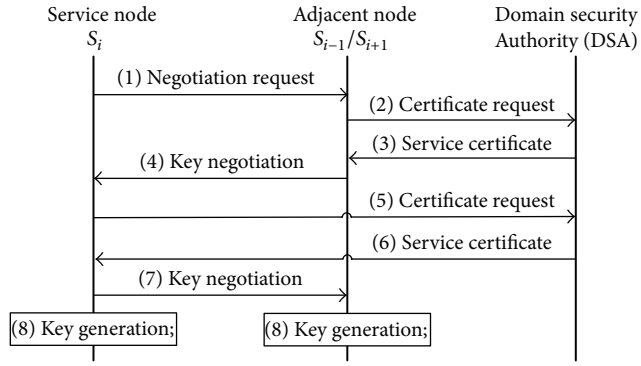

Figure 4: Key negotiation in the domain.

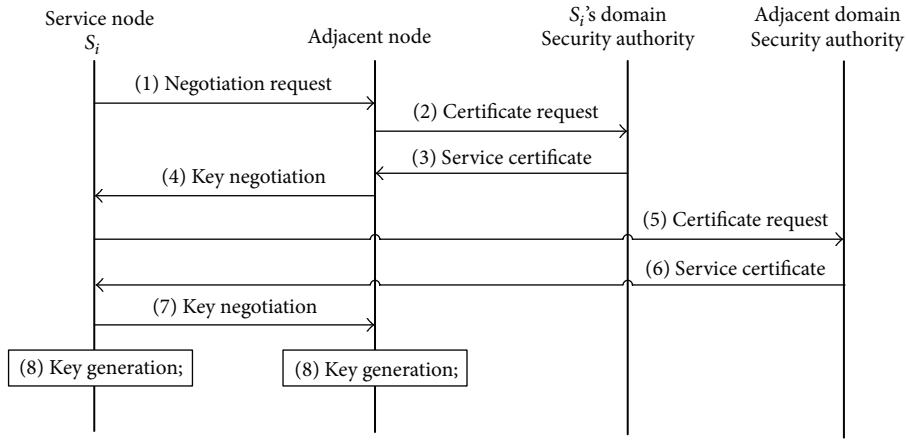

Figure 5: Key negotiation across the domain.

security level for each output objects, and finally verify whether the output objects satisfy the security condition. Meanwhile, if there is an input or output object of s_i which fails to satisfy the strict security constraints, the key negotiation is executed automatically between the related services. In this case, the procedure also returns true unless key negotiation is failed. The pseudocode of verification and declassification for adjacent services is presented as Algorithm 2.

Based on the verification and declassification procedure, we propose a distributed secure service composition with declassification algorithm for mobile clouds. The composition procedure is executed in a distributed way, that is, different cloud platforms in multiple domains need cooperation with each other to finish the whole procedure. There are three types of messages defined for the control on the execution of the procedure, that is, start_message, failure_message, and success_message. First, each cloud platform CP waits for the start message to start the composition procedure. Then CP receives the intermediate result of composition from the start message, including the required security level of predecessor's output and all executable path. After that, CP generates all possible execution paths based on intermediate result and the candidate services located in its domain and verifies them. For each path p that passed the verification, CP pushes it into passed path set PP and records its required security levels of outputs, which can be grouped as an intermediate result for the next step composition. If there is no legal path, CP would send the failure message to user to announce the failure of composition. If the final service step located in this domain, CP would send the success message with all passed path to user. If there are other steps in different domain, CP would send the start message with the intermediate result to the next cloud platform to continue the verification procedure. The distributed secure service composition with declassification algorithm is shown as Algorithm 3.

5. Experiments and Evaluations

The information flow security can be ensured by Theorem 1. And the basic comparison of different verification approaches is presented in Table 1.

According to Table 1, compared to [9, 13–16, 19–21], only [12] and our approach in this paper support the declassification during service composition which can ensure the availability of the composite service. Besides, approaches in [9, 12–15] all work in a centralized way while [16, 19–21] and

Input: s_i, s_{i-1} or s_{i+1}
Output: True or **False** $\langle \text{key}_{i,0}^{M}, E_{i,0}^{M} \rangle, \ldots, \langle \text{key}_{i,n}^{M}, E_{i,n}^{M} \rangle$ **and** $\langle \text{key}_{i,0}^{\text{Out}}, E_{i,0}^{\text{Out}} \rangle, \ldots, \langle \text{key}_{i,n}^{\text{Out}}, E_{i,n}^{\text{Out}} \rangle$.
(1) //$\text{In}_{i,\text{insec}}^{M}$ and $\text{Out}_{i,\text{insec}}$ represents insecure input and outputs in s_i
(2) **if** input is s_{i-1} **then**
(3) **for** each input $\text{in}_{i,x}^{M} \in \text{In}_{i,\text{insec}}^{M}$ **do**
(4) Negotiate_Requests(s_{i-1}, $\text{dom}_{i-1} \cdot \text{SA}$)
(5) **if** Key_Computation($\text{key}_{i,x}^{M}$, $E_{i,x}^{M}$, $\Pr(\text{in}_{i,x}^{M})$) == **False then**
(6) //**False** means two service components can't generate appropriate $\langle \text{key}_{i,x}^{M}, E_{i,x}^{M} \rangle$ which satisfies $\Pr(\text{in}_{i,x}^{M}) \geq \text{Re}(\text{key}_{i,x}^{M})$, else it return **True**.
(7) **return False**
(8) **end if**
(9) **end for**
(10) **else**
(11) **for** each output $\text{out}_{i,y} \in \text{Out}_{i,\text{insec}}$ **do**
(12) Negotiate_Requests(s_{i+1}, $\text{dom}_{i+1} \cdot \text{SA}$)
(13) **if** Key_Computation($\text{key}_{i,y}^{\text{Out}}$, $E_{i,y}^{\text{Out}}$, $\Pr(\text{out}_{i,y})$) == **False then**
(14) **return False**
(15) **end if**
(16) **end for**
(17) **end if**
(18) **return True.** $\langle \text{key}_{i,0}^{M}, E_{i,0}^{M} \rangle, \ldots, \langle \text{key}_{i,n}^{M}, E_{i,n}^{M} \rangle$ and $\langle \text{key}_{i,0}^{\text{Out}}, E_{i,0}^{\text{Out}} \rangle, \ldots, \langle \text{key}_{i,n}^{\text{Out}}, E_{i,n}^{\text{Out}} \rangle$

Algorithm 1: Key Negotiation(·).

Input: s_i, s_{i-1}, s_{i+1}
Output: True or **False**
(1) **for** each $\text{in}_{i,x}^{M} \in \text{In}_{i}^{M}$ **do**
(2) **if** Verification($\text{in}_{i,x}^{M}$) == **False then**
(3) $\text{In}_{i,\text{insec}}^{M} \leftarrow \{\text{in}_{i,x}^{M}\} \cup \text{In}_{i,\text{insec}}^{M}$
(4) **end if**
(5) **end for**
(6) **if** Key_ Negotiation(s_i, s_{i-1}) == **False then**
(7) **return False**
(8) **end if**
(9) Compute Out Required (s_i, s_{i-1})
(10) **for** each $\text{out}_{i,y} \in \text{Out}_{i}$ **do**
(11) **if** Verification($\text{out}_{i,y}$) == **False then**
(12) $\text{Out}_{i,\text{insec}} \leftarrow \{\text{out}_{i,y}\} \cup \text{Out}_{i,\text{insec}}$
(13) **end if**
(14) **end for**
(15) **if** Key_ Negotiation(s_i, s_{i+1}) == **False then**
(16) **return False**
(17) **end if**
(18) **return True**

Algorithm 2: Adjacent Verify & Declass(·).

Table 1: Basic comparison.

	Approach	Framework	Information declassifying
Hutter and Volkamer [9]	Type system	Centralized	×
Nakajima [12]	Model checking	Centralized	√
She et al. [13–15]	Program analysis	Centralized	×
Chou [19]	Program analysis	Distributed	×
Solanki et al. [20]	Program analysis	Distributed	×
Xi et al. [21]	Model checking	Distributed	×
Xi et al. [16]	Program analysis	Distributed	×
This paper	Program analysis	Distributed	√

Table 2: Simulation Configuration.

Network settings	
Simulator	NS-3
Field (m^2)	1000×1000
Cloud domain	3
Cloud platform	3
Mobile nodes	100
Radio type	802.11g
Mobility model	RandomWalk
Backbone network	1 Gbps
Security settings	
Security level	U, C, S, T

our approach is distributed, which is more appropriate for the verification across multiple domains.

In addition, we evaluate the performance of typical approaches, that is, our approach, global model checking [12], centralized program analysis [15], and distributed verification [20, 21] in multiple scenarios by using NS-3 [26]. Basic encryption functions are provided by OPENSSL library [27]. The basic settings of multiple clouds in mobile network are shown as Table 2.

Our simulated mobile network covers about $1000 \times 1000\,\text{m}^2$, which involves three cloud domains, three cloud platforms, and about 100 mobile nodes. For each domain in mobile network, there is one cloud platform and random number of mobile nodes. The communication adopts advanced 802.11g technology, and the mobility model for each node uses the standard *RandomWalk* model. For the backbone network, it connects different cloud domains with wired Gigabit Ethernet. Based on the settings of network, we develop multiple services and deploy them to the cloud platform and specific mobile nodes for dynamic service composition. Meanwhile, we define four different security levels for the information used in service composition, that is, unclassified (U), confidential (C), secret (S), and top secret (T), according to the standard multilevel security model which has been widely applied in government and military systems [23].

Based on the designed mobile network, we simulate the service composition process in multiple mobile clouds. During the simulation, we investigate the success rate and

Input: $s_i, s_{i+1}, \ldots, s_{i+n} \in \mathrm{dom}_{\mathrm{CP}}, \mathrm{CP}_{s_{i-1}}, \mathrm{CP}_{s_{i+n+1}}$
Output: **True** or **Flase**
(1) wait for start_message
(2) $\mathrm{Res}_{s_{i-1}} = \mathrm{ReceInterRes}(\mathrm{CP}_{s_{i-1}})$
(3) $\mathrm{AP} = \mathrm{GenAllPath}\ (\mathrm{Res}_{s_{i-1}} \cdot \mathrm{PP}, s_{i+1}, \ldots, s_{i+n})$
(4) **for** each path $p \in \mathrm{AP}$ **do**
(5) **for** each step m from $i-1$ to $i+n$ **do**
(6) **if** Adjacent Verify & Declass $(s_i, s_{i-1}, s_{i+1}) ==$ **True then**
(7) RCount++
(8) **end if**
(9) **end for**
(10) **if** RCount $== n$ **then**
(11) Push p into $\mathrm{Res}_{s_{i+n}} \cdot \mathrm{PP}$
(12) **for** each $\mathrm{out}^M_{i+n} \in s_{i+n}$ **do**
(13) $\mathrm{Res}_{s_{i+n}} \cdot \mathrm{Re}(\mathrm{out}^M_{i+n}) = \mathrm{Re}(\mathrm{out}^M_{i+n})$
(14) **end for**
(15) **end if**
(16) **end for**
(17) **if** $|\mathrm{PP}| == 0$ **then**
(18) send failure_message to user S_0
(19) **else**
(20) **if** $i + n == N$ **then**
(21) send success message to user s_{N+1}
(22) **else**
(23) send start message to next cloud platform $\mathrm{CP}_{s_{i+n+1}}$
(24) **end if**
(25) **end if**

Algorithm 3: Distributed Compos & Declass(·).

Table 3: Experiment scenario.

Success rate	
Candidate Number	0–20
Service step	1–15
Approaches	Our approach, Solanki et al. [20], She et al. [15], Xi et al. [21]
Composition Time	
Candidate Number	1–15
Service step	1–8
Approaches	Our approach, Xi et al. [21], She et al. [15], Nakajima [12]

time cost on the composition with the different number of service steps and candidate services. The variations of the simulation are shown in the Table 3.

Figures 6 and 7 show the success rate of service composition with different number of candidate services and service steps. With the increase of candidate service number, the composition procedures are more likely to be succeed. Because of the looser security constraints, the success rate of our approach increases much faster than the other approaches. Besides, for [15, 20, 21], the rate decreases vastly when there are too many service steps involved in composite service. With the execution of the verification, the requirements on the inputs and outputs become more strict and fewer candidate services can satisfy it. For our approach, most candidate services can still satisfy the information flow security constraints because of the declassification on data.

Figures 8 and 9 show the cost on composition time with dynamic candidate service number and service step. When the number of candidate service or service step is small, the difference on time cost is not too much. However, for global model checking way [12] and centralized program analysis [15], they are all centralized verification approaches in which all possible composite services must be verified one by one. Therefore, with the increase of candidate services, the complexity of modeling the composite service increases vastly, and it is a time-consuming work to check the complicated model. For our approach and [21], they avoid the repetitive verification on some candidate services, so it provides an more efficient way for secure service composition as the increase of candidate services. Besides, because the key negotiation procedure is executed by related candidate services, the cloud platform can continue verifying other candidate services. Both procedures can br executed in a parallel way. Therefore, the extra effort is not evidence compared with that in [21].

6. Conclusion

In this paper, we propose a declassification mechanism for secure service composition based on cryptographic operations and information flow security requirements. Considering the multidomain characters of mobile clouds, a distributed secure service composition with declassification framework and approach is proposed to overcome the high-rate

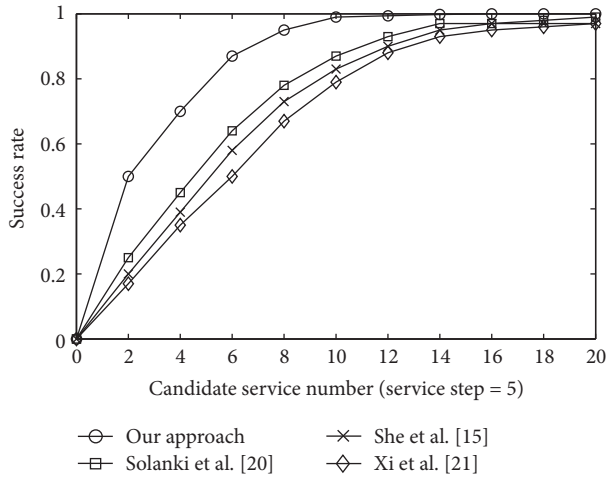

Figure 6: Success rate of composition with candidate service number.

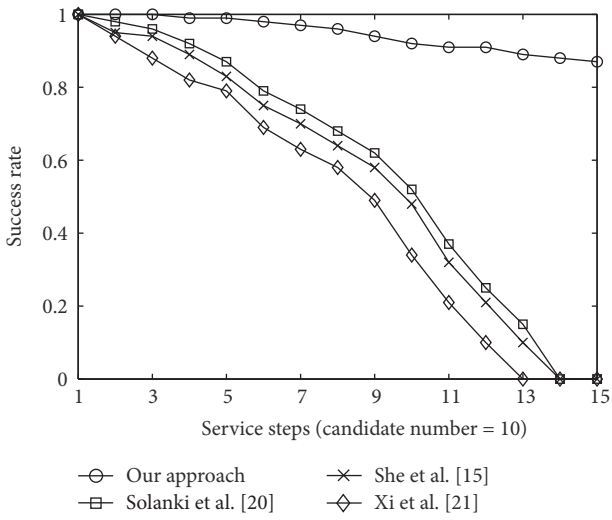

Figure 7: Success rate of composition with service step.

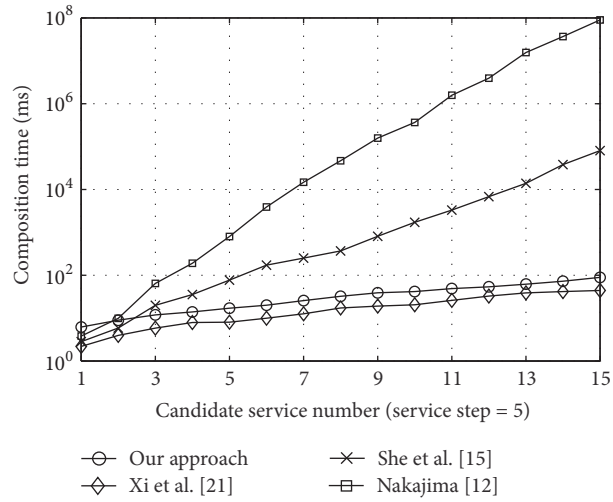

Figure 8: Composition time with candidate service number.

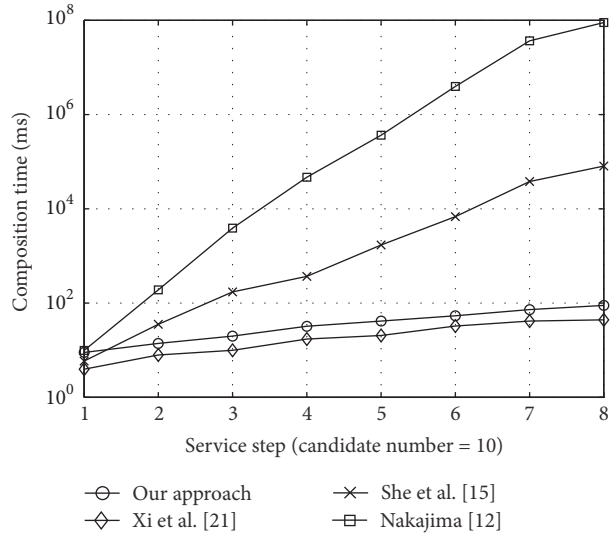

Figure 9: Composition time with service step.

failure of composition, which is caused by too strict security constraints in the traditional composition methods. Through the evaluation on NS-3, the results show our approach can improve the success rate of service composition effectively while the additional cost can be affordable. More dynamic declassification policies for service composition with complex structure will be considered in the future.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (61502368, 61303033, and U1405255), the National High Technology Research and Development Program (863 Program) of China (no. 2015AA017203 and no. 2015AA016007), the Fundamental Research Funds for the Central Universities (XJS14072 and JB150308), Natural Science Basis Research Plan in Shaanxi Province of China (Grant no. 2016JM6034), Xi'an Technology Research Project (CXY1402), and the Aviation Science Foundation of China (no. 20141931001).

References

[1] P.-J. Maenhaut, H. Moens, B. Volckaert, V. Ongenae, and F. De Turck, "Design of a hierarchical software-defined storage system for data-intensive multi-tenant cloud applications," in *Proceedings of the 11th International Conference on Network and Service Management (CNSM '15)*, pp. 22–28, IEEE, Barcelona, Spain, November 2015.

[2] S. Sezer, S. Scott-Hayward, P. Chouhan et al., "Are we ready for SDN? Implementation challenges for software-defined networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 36–43, 2013.

[3] N. Chalaemwongwan and W. Kurutach, "Mobile cloud computing: a survey and propose solution framework," in *Proceedings of the 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON '16)*, pp. 1–4, Chiang Mai, Thailand, June 2016.

[4] N. Chen, N. Cardozo, and S. Clarke, "Goal-driven service composition in mobile and pervasive computing," *IEEE Transactions on Services Computing*, 2016.

[5] F. Paganelli, M. Ulema, and B. Martini, "Context-aware service composition and delivery in NGSONs over SDN," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 97–105, 2014.

[6] C. Groba and S. Clarke, "Opportunistic composition of sequentially-connected services in mobile computing environments," in *Proceedings of the IEEE 9th International Conference on Web Services (ICWS '11)*, pp. 17–24, IEEE, Washington, DC, USA, July 2011.

[7] C. Danwei, H. Xiuli, and R. Xunyi, "Access control of cloud service based on UCON," in *Proceedings of the IEEE International Conference on Cloud Computing*, pp. 559–564, Springer, Beijing, China, December 2009.

[8] D. Volpano, C. Irvine, and G. Smith, "Sound type system for secure flow analysis," *Journal of Computer Security*, vol. 4, no. 2-3, pp. 167–187, 1996.

[9] D. Hutter and M. Volkamer, "Information flow control to secure dynamic web service composition," in *Security in Pervasive Computing*, pp. 196–210, Springer, Berlin, Germany, 2006.

[10] R. Accorsi and C. Wonnemann, "Static information flow analysis of workflow models," in *Proceedings of the 2nd International Symposium on Services Science, ISSS 2010 and 3rd International Conference on Business Process and Services Computing (BPSC '10— INFORMATIK '10)*, pp. 194–205, October 2010.

[11] R. Dimitrova, B. Finkbeiner, M. Kovács, M. N. Rabe, and H. Seidl, "Model checking information flow in reactive systems," in *Verification, Model Checking, and Abstract Interpretation*, pp. 169–185, Springer, Berlin, Germany, 2012.

[12] S. Nakajima, "Model-checking of safety and security aspects in web service flows," in *Web Engineering*, pp. 488–501, Springer, Berlin, Germany, 2004.

[13] W. She, I.-L. Yen, B. Thuraisingham, and E. Bertino, "The SCIFC model for information flow control in web service composition," in *Proceedings of the IEEE International Conference on Web Services (ICWS '09)*, pp. 1–8, IEEE, Los Angeles, Calif, USA, July 2009.

[14] W. She, I.-L. Yen, B. Thuraisingham, and E. Bertino, "Policy-driven service composition with information flow control," in *Proceedings of the IEEE 8th International Conference on Web Services (ICWS '10)*, pp. 50–57, Miami, Fla, USA, July 2010.

[15] W. She, I.-L. Yen, B. Thuraisingham, and E. Bertino, "Security-aware service composition with fine-grained information flow control," *IEEE Transactions on Services Computing*, vol. 6, no. 3, pp. 330–343, 2013.

[16] N. Xi, J. Ma, C. Sun, and T. Zhang, "Decentralized information flow verification framework for the service chain composition in mobile computing environments," in *Proceedings of the IEEE 20th International Conference on Web Services (ICWS '13)*, pp. 563–570, Santa Clara, Calif, USA, July 2013.

[17] E. J. Schwartz, T. Avgerinos, and D. Brumley, "All you ever wanted to know about dynamic taint analysis and forward symbolic execution (but might have been afraid to ask)," in *Proceedings of the 31st IEEE Symposium on Security and Privacy (SP '10)*, pp. 317–331, IEEE, Oakland, Calif, USA, May 2010.

[18] J. Bacon, D. Eyers, T. F. J.-M. Pasquier, J. Singh, I. Papagiannis, and P. Pietzuch, "Information flow control for secure cloud computing," *IEEE Transactions on Network and Service Management*, vol. 11, no. 1, pp. 76–89, 2014.

[19] S.-C. Chou, "Controlling information flows in SaaS cloud services," in *Proceedings of the 7th International Conference on Computing and Convergence Technology (ICCIT, ICEI and ICACT) (ICCCT '12)*, pp. 651–656, Seoul, Korea, December 2012.

[20] N. Solanki, T. Hoffman, I. L. Yen, F. Bastani, and S. S. Yau, "An access and information flow control paradigm for secure information sharing in service-based systems," in *Proceedings of the IEEE 39th Annual Computer Software and Applications Conference (COMPSAC '15)*, vol. 1, pp. 60–67, July 2015.

[21] N. Xi, C. Sun, J. Ma, and Y. Shen, "Secure service composition with information flow control in service clouds," *Future Generation Computer Systems*, vol. 49, pp. 142–148, 2015.

[22] N. Xi, C. Sun, J. Ma, Y. Shen, and D. Lu, "Distributed secure service composition with declassification in mobile network," in *Proceedings of the International Conference on Networking and Network Applications (NaNA '16)*, pp. 254–259, IEEE, Hakodate, Japan, July 2016.

[23] D. E. Denning, "A lattice model of secure information flow," *Communications of the Association for Computing Machinery*, vol. 19, no. 5, pp. 236–243, 1976.

[24] C. Gentry and S. Halevi, "Implementing Gentry's fully—homomorphic encryption scheme," in *Advances in cryptology—EUROCRYPT 2011*, vol. 6632 of *Lecture Notes in Computer Science*, pp. 129–148, Springer, Heidelberg, Germany, 2011.

[25] S. Luo, J. Hu, and Z. Chen, "Implementing attribute-based encryption in web services," in *Proceedings of the IEEE 8th International Conference on Web Services (ICWS '10)*, pp. 658–659, IEEE, Miami, Fla, USA, July 2010.

[26] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, "Network simulations with the ns-3 simulator," in *Proceedings of the SIGCOMM Demonstration*, vol. 14, 2008.

[27] P. Chandra, M. Messier, and J. Viega, *Network Security with Openssl*, O'Reily, 2002.