Distributed Secure Service Composition with Declassification in Mobile Networks

Ning Xi, Cong Sun, Jianfeng Ma, Yulong Shen, Di Lu
School of Computer Science and Technology,
School of Cyber Engineering,
XIDIAN University, Xian, China
nxi@xidian.edu.cn

***Abstract*—The regional and heterogeneous characteristic of mobile network 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 the improved distributed secure service composition framework and approach built on these declassification policies are proposed, which can realize that mobile service nodes cooperate with each other to complete the declassification and secure composition procedure. Through the experiment and evaluation, it indicates that our approach provides a more reliable and efficient way for secure service composition in mobile network.**

***Keywords*-Information Declassification; Cryptographic Operations; Service Chain; Mobile Network;**

I. Introduction

Pervasive network connectivity, intelligent mobile terminal and service-oriented computing environment are creating a new service delivery paradigm for users supporting flexible composition of different basic value-added services [1]. Because of the regional and heterogeneous characteristic of mobile networks, merging data with different sensitivities among different services may cause privacy leakage, which poses a great challenge on secure service composition.

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 and program analysis. By using type system [2], Hutter et al [3] defines 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 provides a formal way to model composite service and Accorsi et al [4] can identify leaks by analyzing it. Model checking is an automatic verification way that can be used to detect information leaks [5]. Nakajima et al [6] embed the lattice model into the Business Process Execution Language (BPEL), and verify the absence of invalid information flows based on model checking. Program analysis is used to construct the dependences among different inputs or outputs based on which information flow control (IFC) policies can be designed according to the security requirements. Transformation factor is defined to measure how likely the output depend on the input data in different candidate services [7] [8]. PDG (Program Dependence Graph) is used to specify the dynamic dependences between the objects in composite service [9] [10]. Considering energy-limited nature of mobile terminal, we propose a distributed information flow security verification framework and approach to provide a better load balance on energy cost in mobile network [10].

Although these 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 aims 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 candidate service can satisfy it in real application. If all the candidate services fail on the verification, there is no available execution path, which causes the failure of the whole composite service. Besides, In mobile Internet, services are bound together in a dynamic way during service composition. Considering dozens of candidate services with similar service function, it is a complex work on selecting appropriate components to compose users required application. If the enforcement mechanism is executed in a centralized way, it causes a great burden on single mobile terminal which only has limited energy.

Therefore, a distributed information flow control mechanism supporting declassifying or downgrading information is needed for the secure and reliable service composition in mobile network. The rest of the paper is structured as follows. Section II gives a formal definition of the service chain model in mobile network. Section III presents the improved computation rules on declassifying rules for information flow in service chain. In section IV, we propose the secure service composition with declassification mechanism for service chain in mobile network. Section V evaluates the proposed approach. Section VI concludes the paper.

II. PRELIMINARIES

Mobile network [10] MN is a large-scale distributed environment which consists of multiple heterogeneous wireless

978-1-4673-9803-9/15 $31.00 © 2015 IEEE
DOI 10.1109/NaNA.2016.53

network domains e.g., $MN = \{d_0, d_1, ...\}$. A domain d could have various types of data resources R. And mobile terminals could use these resources and provide various services to users which are regarded as the service nodes $SN = \{sn_0, sn_1, ...\}$ in the domain. There is also a security authority SA in each domain for the management on security policies expressed by domain certificate DSe. So the domain d can be represented as $d = \langle SN, R, SA, DCe \rangle$.

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 network domain resources $dom_i.R$; 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 $dom_i.R$. 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 $\langle s_0, s_1, ...s_{n+1} \rangle$. In CH , each service s_i only has one predecessor s_{i-1} and one successor s_{i+1}. In_c and Out_c are the inputs and outputs of SC including all the service componentsi.e., $In_c = \bigcup\{In_i^M \cup In_i^D \cup In_i^L\}$ and $Out_c = \bigcup\{Out_i^M \cup Out_i^D \cup Out_i^L\}$,$0 \leq i \leq n+1$. Due to the complex operations in service chain, the inner-service dependency $Dep_{inner}(o)$ and inter-service dependency $Dep_{inter}(o)$ are defined to represent the flows between different inputs and outputs based on Program Dependence Graph (PDG) [10].

III. Secure Information Flow Model with Declassification in Service Chain

A. Multi-Level Security Model

In order to represent different sensitivities of data resources in mobile network, Multi-Level Security Model is defined as $\langle SL, \leq \rangle$, where SL is a finite set of security levels that is totally ordered by $\leq$. [11]

For each input or output object o in s_i, we define $Re : In_i \cup Out_i \rightarrow SL_{ex}$ maps o to the required security level of data stored in it, while $Pr : In_i \cup Out_i \rightarrow SL_{ex}$ 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 and domain certificates respectively.

B. Secure Information Flow with Standard Noninterference

For data with different security requirements, the computation rules(**CR**s) on required security level is defined in [10] as follows:

CR 1. *For* $\forall u \in In_i^D \cup In_i^L$, $Re(u) = Pr(u)$.

CR 2. *For* $\forall u \in In_i^M$, $Re(u) = Re(v)$ *where* $v \in Out_{i-1}^M \wedge v \in Dep_{inter}(u)$.

CR 3. *For* $\forall u \in Out_i$, $Re(u) = \sqcup_{max}\, Re(v) \quad v \in In_i \cup Out_j, j \leq iv \in Dep(u) \cup Dep_{inter}(u)$.

Based on the standard noninterference, we propose a strong security definition on information flow for composite service in [10].

Definition 1. The information flow in service chain SC is considered secure if it satisfies that for $\forall u, v \in In_c \cup Out_c, v \in Dep_{inner}(u) \cup Dep_{inter}(u)$, there is $Pr(u) \geq Re(v)$, where $Re(v)$ is computed according to the above **CRs**.

In this definition, it is considered secure when there is no flow from a high-level object to another low-level one. However, the strong security constraints enforces that the flow of information must comply with the security level ordering and does 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 more higher according to the above **CRs**, which is over-restrictive that fewer service components can satisfy and results in 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.

C. Secure Service Composition Model with Cryptographic Operations

Due to the strict security condition, declassification operations are needed for the secure service composition. Cryptographic operations are promising ways of maintaining data confidentiality and integrity, e.g., encryption and digital signature. Through the cryptographic operations, processed secret data can be transferred to a public object, which realizes the declassification of data. Therefore, extra cryptographic operations $En(o, key)$ and $De(o, key)$ can be added into the service function F_i for each service.

For $\forall u \in In_i \cup Out_i$, u_p and u_c represents the plaintext and ciphertext of u, and the encryption and decryption on u are defined as $En(u, key)$ and $De(u, key)$. For the ciphertext u_c , its security level depends on the encryption algorithm E and the key key . When the data in u is encrypted, it provide a more securer way to transfer u , and the attacker needs to work harder to crack the ciphertext. Thus we use $Re(\langle E, key \rangle)$ to represent the security level that encryption operation can ensure the ciphertext cannot be cracked. Encryption with more complex algorithm and

key means $Re(\langle E, key\rangle)$ is lower. According to the analysis above, we can extend the basic computation rules as follow:

CR 4. *For* $\forall u \in In_i \cup Out_i$, *if* u *is encrypted, there is* $Re(u) = Re(u_c) = Re(En(u_p, key)) = Re(u_p) \cap Re(\langle E, key\rangle)$.

CR 5. *For the ciphertext* u_c , *if* u *is decrypted, there is* $Re(u) = Re(De(u_c, key)) = Re(u_p)$.

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 can't obtain the sensitive data if he cannot crack the ciphertext, which is considered secure although the sensitive data is transferred to an object with lower data clearance. In order to specify the special flow in composite service, declassification dependence $Dep^{\downarrow}$ is defined as following:

Definition 2. For $\forall u, v \in In_c \cup Out_c$, v is in $Dep^{\downarrow}(u)$ which satisfies one of the following three conditions.
(1)For $\forall v \in In_i, u \in Out_i, v \in Dep_{inner}(u)$, if u is encrypted, there is $v \in Dep^{\downarrow}_{inner}(u)$.
(2)For $\forall v \in Out^M_{i-1}, u \in In^M_i, v \in Dep_{inter}(u)$, if u is encrypted, there is $v \in Dep^{\downarrow}_{adjacent}(u)$.
(3)For $\forall v \in In_i \cup Out_i, u \in In_j \cup Out_j, j > i, v \in Dep_{inter}(u)$, if $\nexists w_1 \in In_k \cup Out_k, w_2 \in In_l \cup Out_l, i < k \leq l < j, v \in Dep_{inter}(w_1), w_2 \in Dep_{inter}(u)$ satisfy above two conditions, there is $v \notin Dep^{\downarrow}_{inter}(u)$, else $v \in Dep^{\downarrow}_{inter}(u)$.

Considering special declassification dependence, the improved security definition on information flow for composite service can be defined as follows:

Definition 3. The information flow in service chain SC is considered secure if $\forall u, v \in In_c \cup Out_c, v \in Dep_{inner}(u) \cup Dep_{inter}(u)$ satisfies the following two conditions:
(1)For $v \notin Dep^{\downarrow}(u)$, there is $Pr(u) \geq Re(v)$.
(2)For $v \in Dep^{\downarrow}(u)$, there is $Pr(u) \geq Re(u) = \sqcap_{min} Re(\langle E, key\rangle)$.

Based on improved information flow security definition, we can deduce the following Theorem:

Theorem 1. *The information flow in service chain SC is considered secure if for each* s_i *in SC satisfies the following two conditions:*
(1)For $\forall v \in In_i, u \in Out_i, v \in Dep_{inner}(u)$*: If* u *is not encrypted, there is* $Pr(u) \geq Re(v)$. *If* u *is encrypted by* $\langle E, key\rangle$, *there is* $Pr(u) \geq Re(\langle E, key\rangle)$.
(2)For $\forall v \in Out^M_i, u \in In^M_{i+1}, v \in Dep_{inter}(u)$*: if* u *is not encrypted, there is* $Pr(u) \geq Re(v)$. *If* u *is encrypted by* $\langle E, key\rangle$, *there is* $Pr(u) \geq Re(\langle E, key\rangle)$.

Based on these above computation rules and Theorem , we can propose an improved service composition mechanism that can declassify the required security level on inputs and outputs, which can assist the unsecure service components to prevent information leaks. The declassification policies (DPs) are presented as follows:

DP 1. *For* $\forall v \in In_i, u \in Out_i, v \in Dep_{inner}(u)$, *if* $Pr(u) < Re(v)$, *then* u *needs to be encrypted which satisfies* $Pr(u) \geq Re(\langle E, key\rangle)$.

DP 2. *For* $\forall v \in Out^M_i, u \in In^M_{i+1}, v \in Dep_{inter}(u)$, *if* $Pr(u) < Re(v)$, *then* u *needs to be encrypted which satisfies* $Pr(u) \geq Re(\langle E, key\rangle)$.

According to the declassification policies, when the provided security level of u cannot satisfies the strict conditions, cryptographic operations are adopted to assist declassifying the required security level which is no longer higher than its provided clearance level.

IV. SECURE SERVICE COMPOSITION APPROACH WITH DECLASSIFICATION IN MOBILE NETWORK

A. Distributed Secure Service Composition with Declassification Framework in Mobile Network

Based on the declassification policies presented above, we design the distributed secure service composition with declassification framework in mobile network as shown in Fig.1. In this framework, mobile service nodes can cooperate with each other to complete the composition and the declassification procedure. If the information flow security verification returns failure, the declassification procedure is executed, where each insecure component negotiates a session key with its adjacent nodes for the encryption and decryption during the service execution. There are two different scenarios, i.e., inner-domain and inter-domain declassification. For inner-domain declassification, two related service components and the domain security authority(SA) are involved in the key negotiation, while for inter-domain declassification, the participants not only includes two adjacent service components but also two security authorities in the corresponding domains.

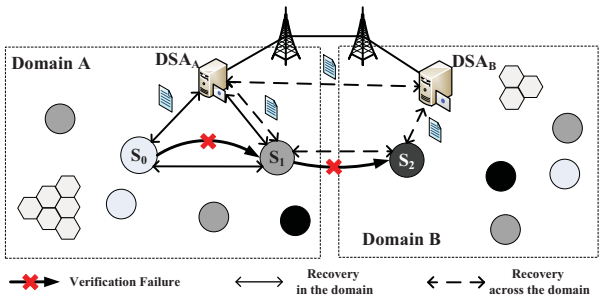

Figure 1. Distributed Secure Service Composition with Declassification Framework

B. Cryptographic Operations for Secure Information Flow

1) Cryptographic Operation Agent: In order to protect sensitive information from illegal users during service composition, many relevant security specifications have been proposed, such as XML Encryption and Signature, WS-Security, SAML(Security Assertion Markup Language), X-ACML(XML Access Control Markup Language), XKMS(XML Key Management Specification), etc [12]. By using the basic security functions supported by these specifications, a cryptographic operation agent (COA) can be designed and deployed in the mobile service node to execute the declassification procedure including key negotiation, data encryption and decryption, which is shown in Fig.2.

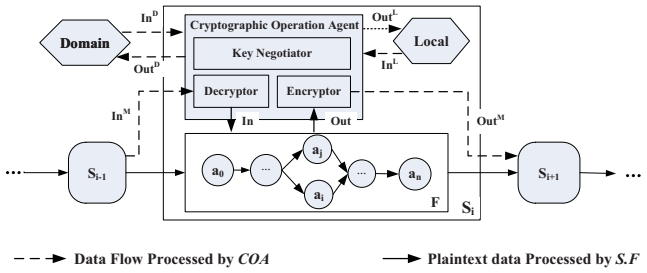

Figure 2. Cryptographic Operation Agent in Each Service Node

The cryptographic operation agent is composed of three function modules, i.e., 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 according to the declassification policies. There are two phases for agent to complete the declassification procedure, i.e., key negotiation and data encryption.

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 $u \in In_i \cup Out_i$, adjacent service components need to negotiate to generate the appropriate encryption algorithm and key $\langle E, key \rangle$ to ensure the information flow security according to **DP**s. The procedure of key negotiation is shown in Fig.3 and Fig. 4.

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

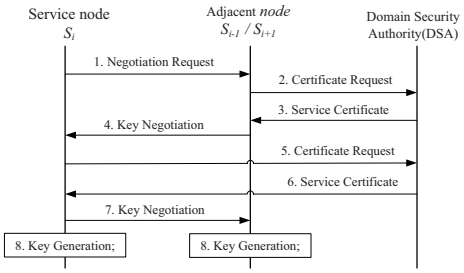

Figure 3. Key Negotiation in the Domain

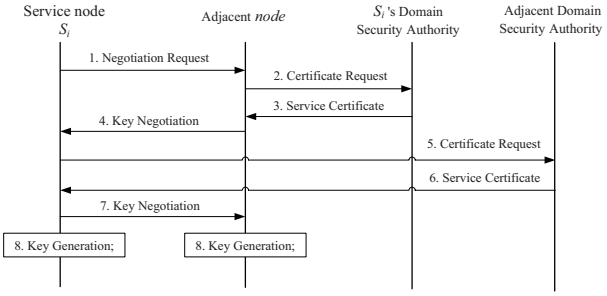

Figure 4. Key Negotiation across the Domain

Algorithm 1 $Key\_Negotiation()$

Input: s_i, s_{i-1}, s_{i+1}.
Output: **True** or **False** $\langle key^M_{i,0}, E^M_{i,0} \rangle$,...,$\langle key^M_{i,n}, E^M_{i,n} \rangle$ and $\langle key^{Out}_{i,0}, E^{Out}_{i,0} \rangle$, ..., $\langle key^{Out}_{i,n}, E^{Out}_{i,n} \rangle$.

1: //$In^M_{i,insec}$ *and* $Out_{i,insec}$ *represents insecure input and outputs in* s_i
2: **for** each input $in^M_{i,x} \in In^M_{i,insec}$ **do**
3: Negotiate_Request(s_{i-1},$dom_{i-1}.SA$)
4: **if** Key_Computation($key^M_{i,x}$,$E^M_{i,x}$,$Pr(in^M_{i,x})$ == **False then**
5: //***False*** *means two service components can't generate appropriate* $\langle key^M_{i,x}, E^M_{i,x} \rangle$ *which satisfies* $Pr(in^M_{i,x}) \geq Re(key^M_{i,x},)$, *else it return* ***True***.
6: **return False**
7: **end if**
8: **end for**
9: **for** each output $out_{i,y} \in Out_{i,insec}$ **do**
10: Negotiate_Request(s_{i+1},$dom_{i+1}.SA$)
11: **if** Key_Computation($key^{Out}_{i,y}$,$E^{Out}_{i,y}$,$Pr(out_{i,y})$)==**False then**
12: **return False**
13: **end if**
14: **end for**
15: **return True**, $\langle key^M_{i,0}, E^M_{i,0} \rangle$,...,$\langle key^M_{i,n}, E^M_{i,n} \rangle$ and $\langle key^{Out}_{i,0}, E^{Out}_{i,0} \rangle$, ..., $\langle key^{Out}_{i,n}, E^{Out}_{i,n} \rangle$

the encryption algorithm must satisfies the requirements on security level. The pseudocode of key negotiation is presented as Algorithm 1.

3) Data Declassification Phase: The data declassification phase is activated after the procedure of secure service composition. During the service execution, the COA encrypt 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.

C. Distributed Secure Service Composition with Declassification Algorithm in Mobile Network

During the secure service composition, if there is no appropriate candidate service components which satisfy the strict security constraints, the key negotiation executed in a distributed way. If encryption key and algorithm are generated successfully, the composition procedure continues until it returns a secure and success execution path for composite service. If key generation is failed, other possible candidate service components continue the key negotiation procedure. The distributed secure service composition with declassification algorithm is shown as Algorithm 2.

Algorithm 2 *Distributed_Composition & Declassify()*

Input: s_i, s_{i-1}, s_{i+1}.
Output: **True** or **False**

```
1: wait start_message
2: if i == N then
3:     send success_message to start node s_0
4:     return True
5: else
6:     for each in_{i,x}^M ∈ In_i^M, out_{i,y} ∈ Out_i do
7:         if Verification(in_{i,x}^M) == False then
8:             In_{i,insec}^M ← {in_{i,x}^M} ∪ In_{i,insec}^M
9:         end if
10:        if Verification(out_{i,y}) == False then
11:            Out_{i,insec} ← {out_{i,y}} ∪ Out_{i,insec}
12:        end if
13:    end for
14:    if Key_Negotiation(s_i, s_{i-1}, s_{i+1}) == False then
15:        send failure_message to s_{i-1}
16:        return False
17:    end if
18:    send start_message to s_{i+1}
19:    if receive failure_message then
20:        failure_message_count++
21:        if failure_message_count == |S_{i+1}| then
22:            if i == 0 then
23:                send composition_failure_message to user
24:            else
25:                send failure_message to s_{i-1}
26:            end if
27:            return False
28:        else
29:            send start_message to s'_{i+1}
30:        end if
31:    end if
32: end if
```

V. 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 I.

Table I
BASIC COMPARISON

	Approach	Framework	Information Declassifying
Dieter et al [3]	Type System	Centralized	×
Nakajima et al [6]	Model Checking	Centralized	√
She et al [7] [8] [9]	Program Analysis	Centralized	×
Xi et al [10]	Program Analysis	Distributed	×
This paper	Program Analysis	Distributed	√

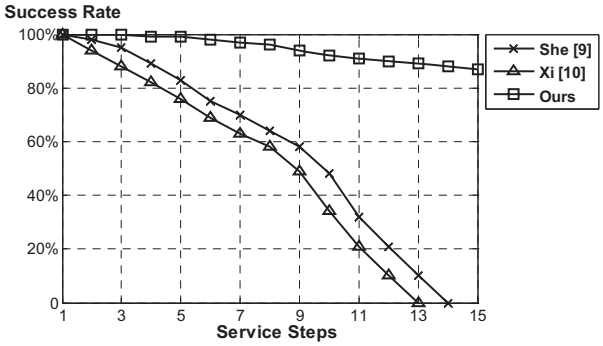

Figure 5. Comparsion on The Success Rate of Service Composition

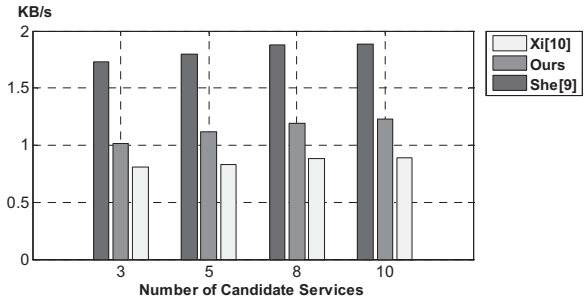

Figure 6. Average Communication Effort on Composition Node/Nodes

According to Table I, [6] and our approach in this paper support the declassification during service composition compared to [3], [7], [8], [9] and [10], which can ensure the availability of the composite service. Besides, approaches in [3], [6], [7], [8], and [9] all work in a centralized way while our approach is distributed, which is more appropriate for the mobile terminal with limited energy and computation ability. We evaluate the performance of typical approaches in multiple scenarios by using NS-3 [13]. Basic encryption functions are provided by OPENSSL library [14].

Fig.5 shows the success rate of the different service composition approaches with different service steps. For [9] and [10], the rate decreases vastly when there are too much 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, much candidate services can still satisfy the information flow security constraints because of the declassification on data.

Fig.6 shows the communication effort on verification node/nodes of different approaches. For [9], it works in a centralized way, i.e. a single service composer, while [10] and our approach performs in a distributed way that each service node is involved in. Therefore, the average effort on composition node is evidently lower than the other approach. Because of the key negotiation in declassification procedure,

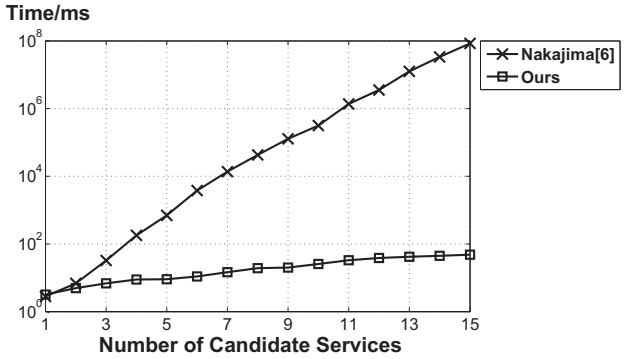

Figure 7. Verification Time of Centralized and Distributed Approaches

the effort is a little higher than that in [10]. But it is not evident because the negotiation procedure also works in a distributed way.

Fig.7 shows the verification time cost on two approaches supporting declassification. When the number of candidate service is small, the difference on time cost is not too much. However, for centralized model checking way [6], it is a centralized verification approach that all possible composable services must be verified by the single node, and the time complexity is $O(n^m)$. 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, each service participant only needs to verify the adjacent candidate service and the time complexity of the whole verification is $O(n * m)$, so it provides an more efficient way for secure service composition as the increase of candidate services.

VI. Conclusion

In this paper, we propose a declassification mechanism for secure service composition based on cryptographic operations and information flow security requirements. Considering the energy-limited characters of mobile network, a distributed secure service composition with declassification framework and approach are proposed to overcome the high-rate failure of composition 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 is affordable. More dynamic declassification policies for service composition will be considered in the future.

Acknowledgment

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, No.2015AA016007), the Fundamental Research Funds for the Central Universities (XJS14072, JB150308), Natural Science Basis Research Plan in Shaanxi Province of China (Grant No. 2016JM6034) and the Aviation Science Foundation of China (No.20141931001).

References

[1] N. Chen, N. Cardozo, and S. Clarke, "Goal-driven service composition in mobile and pervasive computing," *IEEE Transactions on Services Computing*, vol. PP, no. 99, pp. 1–1, 2016.

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

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

[4] R. Accorsi and C. Wonnemann, "Static information flow analysis of workflow models." *ISSS/BPSC*, vol. 2010, pp. 194–205, 2010.

[5] 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*. Springer, 2012, pp. 169–185.

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

[7] W. She, I. Yen, B. Thuraisingham, E. Bertino *et al.*, "The scifc model for information flow control in web service composition," in *Web Services, 2009. ICWS 2009. IEEE International Conference on*. IEEE, 2009, pp. 1–8.

[8] W. She, I. Yen, B. Thuraisingham, and E. Bertino, "Policy-driven service composition with information flow control," in *Web Services (ICWS), 2010 IEEE International Conference on*. IEEE, 2010, pp. 50–57.

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

[10] N. Xi, J. Ma, C. Sun, and T. Zhang, "Decentralized information flow verification framework for the service chain composition in mobile computing environments," in *Web Services (ICWS), 2013 IEEE 20th International Conference on*. IEEE, 2013, pp. 563–570.

[11] D. E. Denning, "A lattice model of secure information flow," *Communications of the ACM*, vol. 19, no. 5, pp. 236–243, 1976.

[12] S. Luo, J. Hu, and Z. Chen, "Implementing attribute-based encryption in web services," in *Web Services (ICWS), 2010 IEEE International Conference on*. IEEE, 2010, pp. 658–659.

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

[14] P. Chandra, M. Messier, and J. Viega, "Network security with openssl," *OReily, June*, 2002.