

# Pattern–Based Design and Verification of Secure Service Compositions

## ABSTRACT

Assurance is important for any software application but acutely so in the case of *service-based systems* (SBSs), i.e., systems composed of distributed software services, which can be deployed on different and heterogeneous infrastructures and operate without common ownership and centralized control. Assessing and providing assurance about the security of SBSs is a complex problem that has no comprehensive solution to the best of our knowledge.

## EXISTING SYSTEM

In Existing System, rely on different forms of model checking and theorem proving to verify security properties of service compositions. These approaches typically require the specification are:

- Behavioral models of the software services used by the SBS.
- The component that orchestrates them to provide the SBS functionality (i.e., the service orchestrator).
- The security properties that need to be guaranteed in some temporal logic language.

There are two main difficulties with such approaches. The first is that creating accurate specifications for realistic SBSs is a non-trivial and time-consuming task. The second is that, even if SBS specifications are available, performing automated static analysis might be computationally intractable.

## DIS ADVANTAGES

- It is a time-consuming task.
- The security properties that need to be guaranteed in some temporal logic language.

## PROPOSED SYSTEM

In Proposed System, we present an approach for verifying the security properties of SBS workflows and adapting them if such properties are not preserved. Our approach uses secure service composition patterns. These patterns encode proven dependencies between *service level*

---

and workflow level security properties. These dependencies are used in reasoning processes supporting the verification of SBS workflows with respect to workflow security properties and their adaptation in ways that guarantee the properties if necessary. Our approach has been implemented by extending the Eclipse BPEL Designer and validated experimentally. The experimental evaluation has produced positive results, indicating that even for complex workflows and large sets of secure service composition patterns verification can be performed efficiently.

## **ADVANTAGES**

- Guaranteed to preserve security properties.
- Allowed to access only public information, should not be able to determine anything about high-level (i.e., confidential) information.

## **SYSTEM REQUIREMENTS**

### **H/W System Configuration:-**

Processor	- Pentium –III
RAM	- 256 MB (min)
Hard Disk	- 20 GB
Key Board	- Standard Windows Keyboard
Mouse	- Two or Three Button Mouse
Monitor	- SVGA

### **S/W System Configuration:-**

Operating System	: Windows95/98/2000/XP
Application Server	: Tomcat5.0/6.X
Front End	: HTML, Jsp

Scripts : JavaScript.  
Server side Script : Java Server Pages.  
Database : MySQL 5.0  
Database Connectivity : JDBC