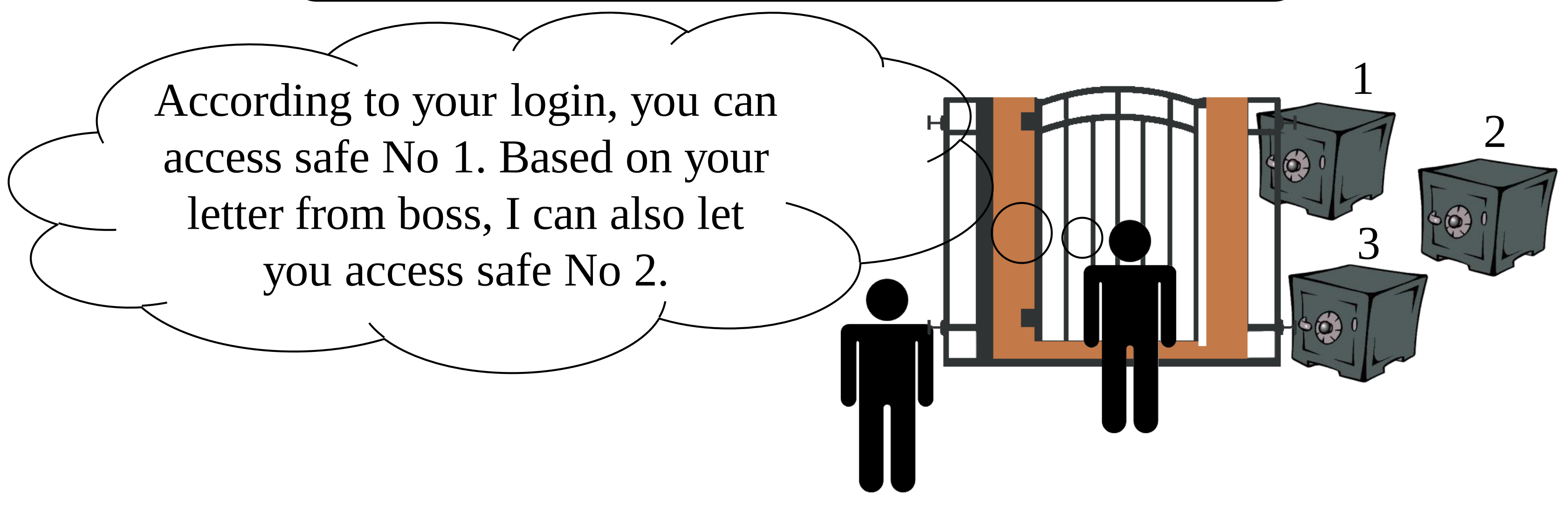
Verification of State Based Access Control

Masoud Koleini, Mark Ryan
M.Koleini@cs.bham.ac.uk, M.D.Ryan@cs.bham.ac.uk

University of Birmingham

Introduction


According to your login, you can access safe No 1. Based on your letter from boss, I can also let you access safe No 2.
1
2
3

Access Control Security Policy
Security policy is a high level rule that defines the (high-level) rules according to which access control must be regulated.

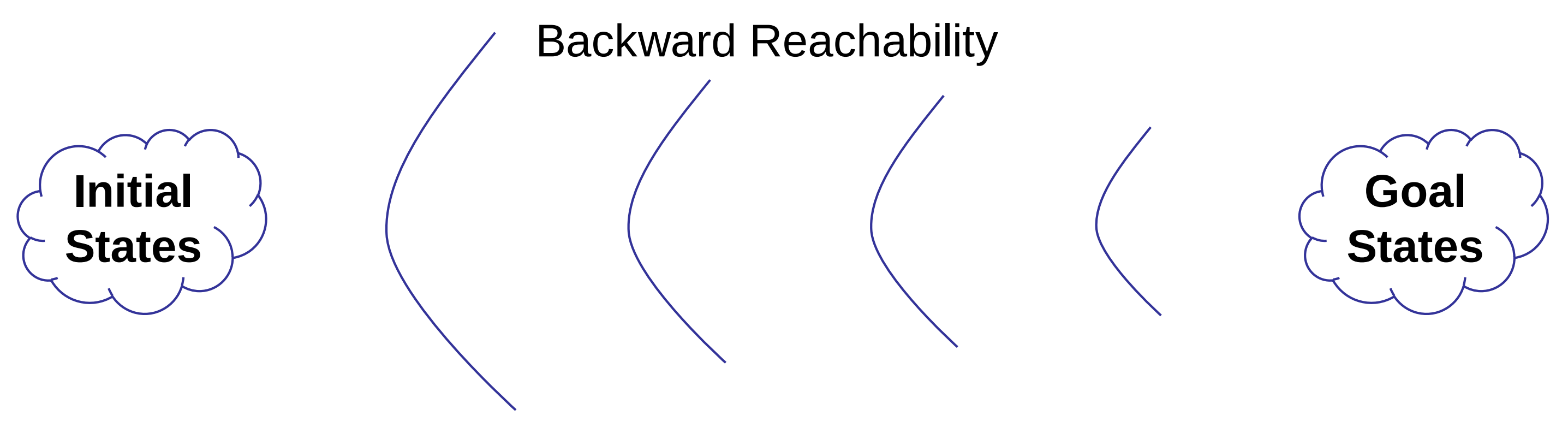
Access Control Security Model

DAC	MAC	RBAC
Discretionary Access Control	Mandatory Access Control	Role-Based Access Control

Access Control Security Mechanism
Low level (software and hardware) functions that implement the controls imposed by the policy and formally stated in the model

Motivation

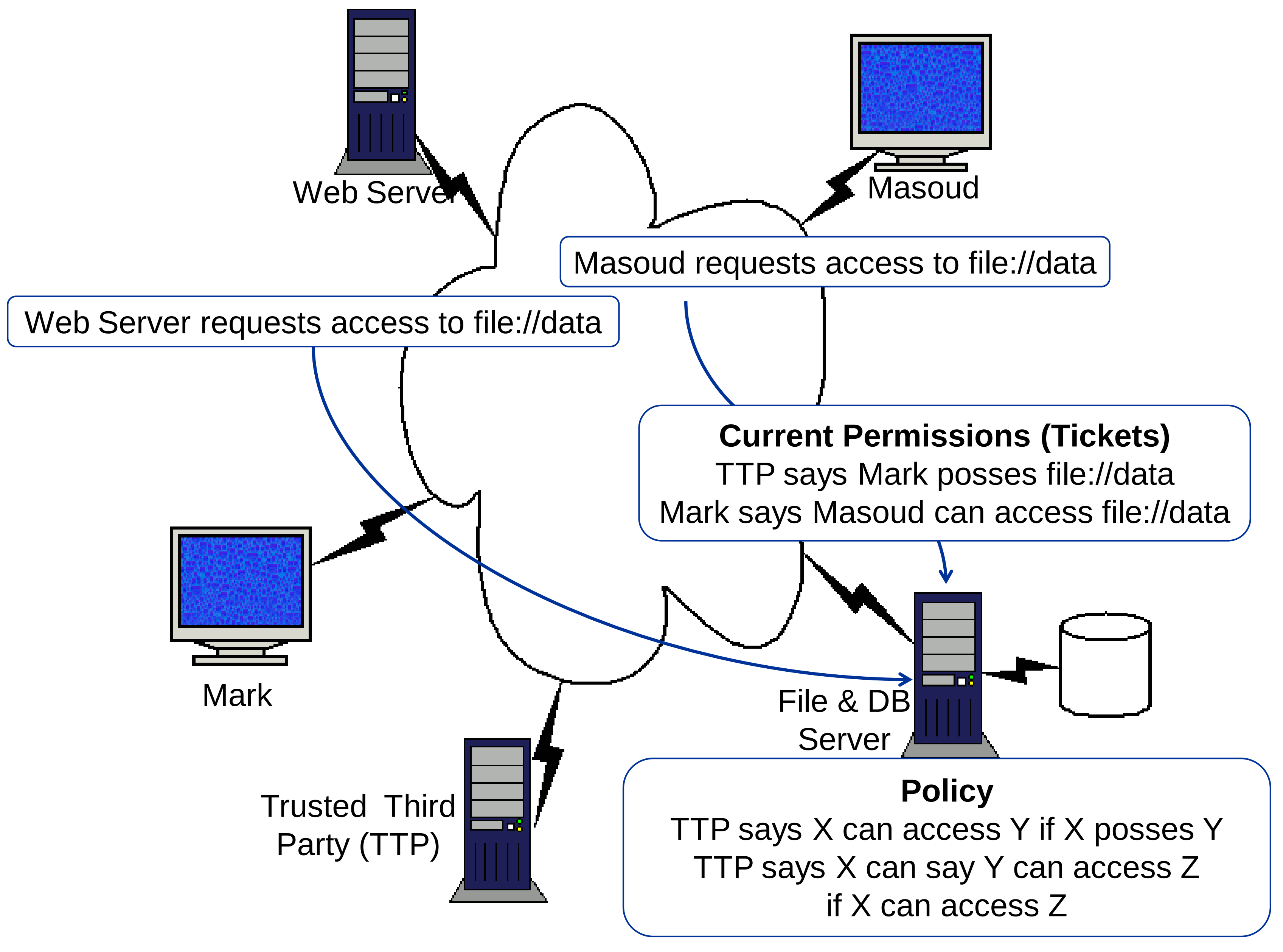
SecPAL authorization language is a stateless verification tool that is designed to support query verification upon local policy in distributed systems. RW is an stateful verification tool. It is based on model checking techniques. It can look for weaknesses of policy that is defined with RW syntax. Query that is going to be evaluated can be a situation that we don't like to happen in the system and RW checks if with some sequence of actions, it can be satisfied. So, we can change our security policy definition to a stronger one.



Current approaches have some weaknesses. One of the important problems is state-explosion problem that is more offensive in the way RW checks the model. Another one is the ability to check atomic read/write actions and also lack of appropriate delegation method. Because of that, there are some access control scenarios that can not be simulated by RW in an efficient way.

Recent Works

There are some tools related to simulation and verification of access control policies. Cassandra, SecPAL and RW are three important ones. SecPAL is an authorization policy language for distributed systems that based on the policy and tokens, finds out if a current access query is permitted or granted.


Web Server
Masoud
Masoud requests access to file://data
Web Server requests access to file://data
Current Permissions (Tickets)
TTP says Mark posses file://data
Mark says Masoud can access file://data
Mark
File & DB Server
Trusted Third Party (TTP)
Policy
TTP says X can access Y if X posses Y
TTP says X can say Y can access Z
if X can access Z

Consider the following query based on the above configuration:

Query: check if Masoud can access to file://data

SecPAL answers this query as YES. It means with current policy and tokens, Masoud is allowed to have access to file:///data. But if in the query, instead of Masoud we had Web Server, the answer would be NO. It means Web Server is not permitted to access the specific recourse.

RW is a stateful access control verification tool. It is different from SecPAL, because SecPAL is a stateless authorization language. RW looks for the sequence of actions that begin from initial state and end to the goal. It doesn't use current permission or ticket. For example, in the above configuration, RW will answer the following query (assume tickets as initial state):

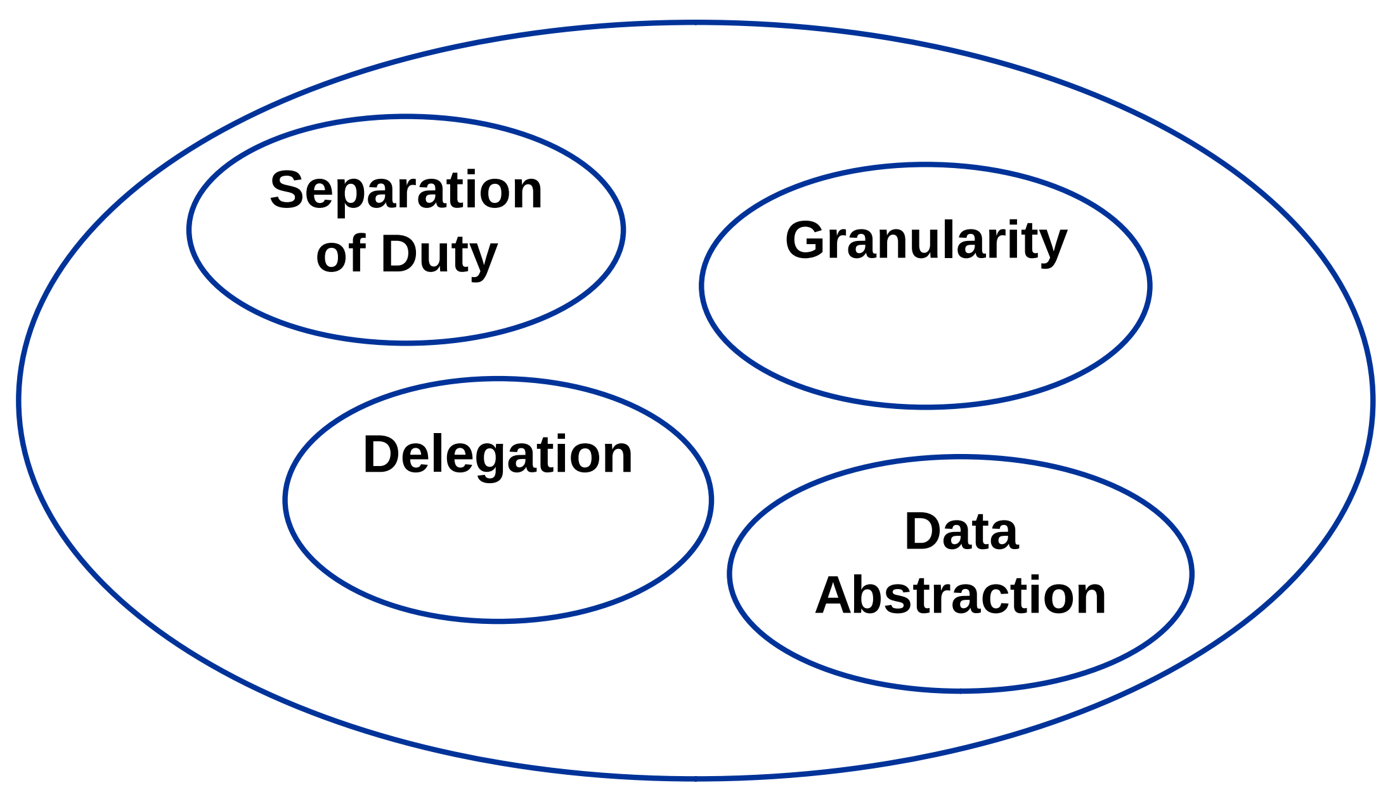
Query: check if Web Server can access to file://data

With the following strategy:

If Mark says Web Server can access file://data -> Web Server can access file://data

Our Approach

We will work on stateful access control policy verification. We are going to design a policy specification syntax that supports important specification of access control systems like separation of duty, granularity, delegation and data abstraction. . This framework should be capable of defining policy based on three different security models (DAC, MAC, RBAC). having a full notion of system state to allow state-dependent permissions is one of the goals in this research.



We will also provide a method which allows verification of ACSs expressed in the language against high-level properties, extending our techniques to support decentralisation and delegation. Finally, we aim to develop techniques to synthesise low-level access control rules from management-level properties.

One of the ways for verification of ACSs is model checking methods. But this method has the problem of state explosion. We are looking for a way to reduce the effect of state explosion and have the ability to synthesise complicated problems.

Conclusion

Access control policies may have weaknesses in their definition. Using verification tools can discover such a weaknesses and also ensure designers to have appropriate specifications in their policy.

It is important that policies can be defined easily in verification tool syntax based on different security models .

Also in the implementation, It is necessary to be able to simulate complex scenarios. Model checking based on BDDs is very important way of verifying stateful systems, but it suffers from state explosion problem. We will work on different ways to have more complicated higher level properties with more efficient verification algorithms.