# Practical Case: Secure Home Access

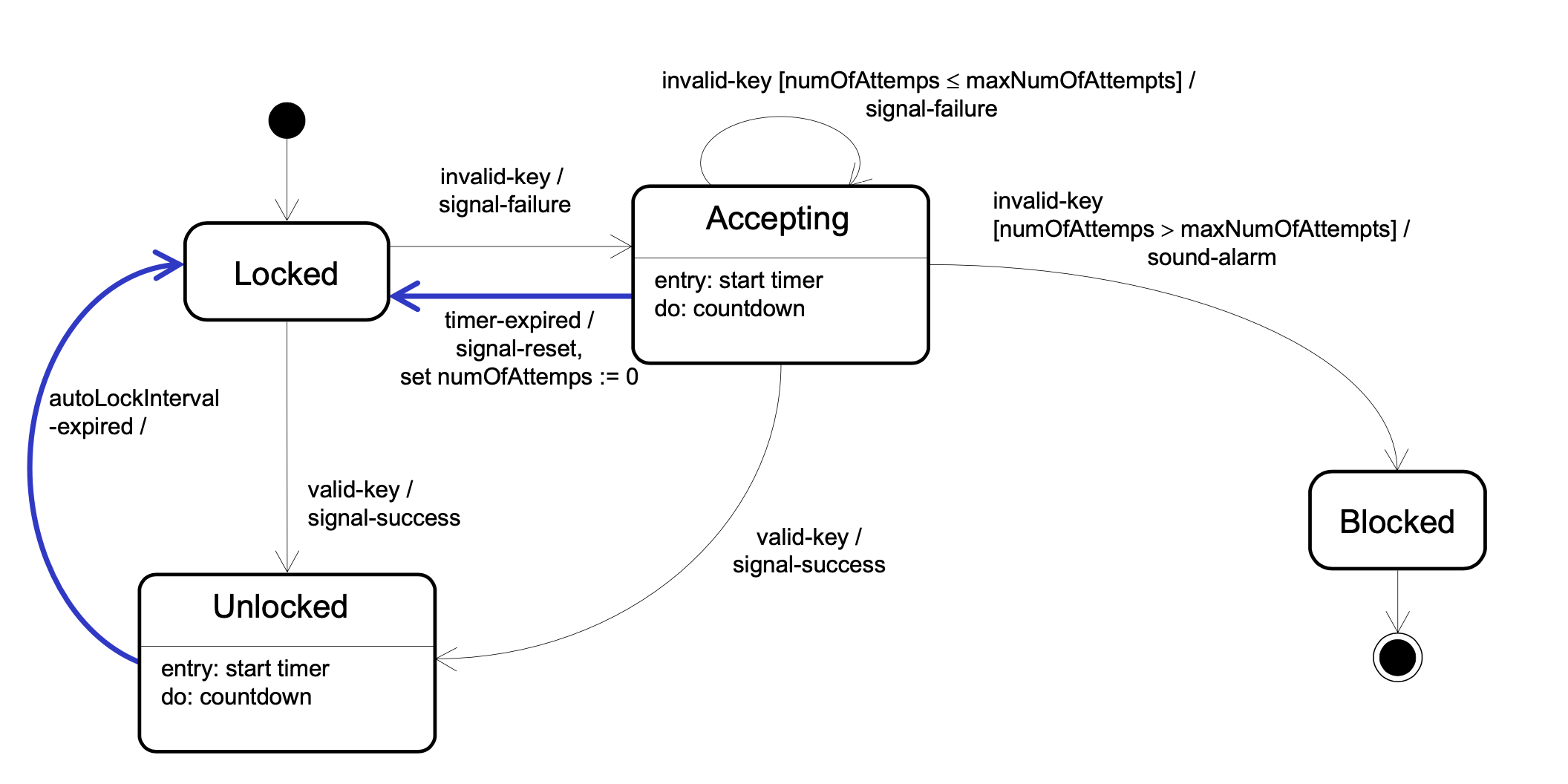
Consider a home access control system showed as followed. The system could be required to authenticate and validate people attempting to enter a building. Along with controlling the locks, the system may also control other household devices, such as the lighting, air conditioning, heating, alarms, etc.



To address the intolerable risk, we can design an automatic locking system which observes the lock state and auto-locks it after **autoLockInterval** seconds elapses. The auto-locking system could be made stand-alone, so its failure probability is independent of the main system.

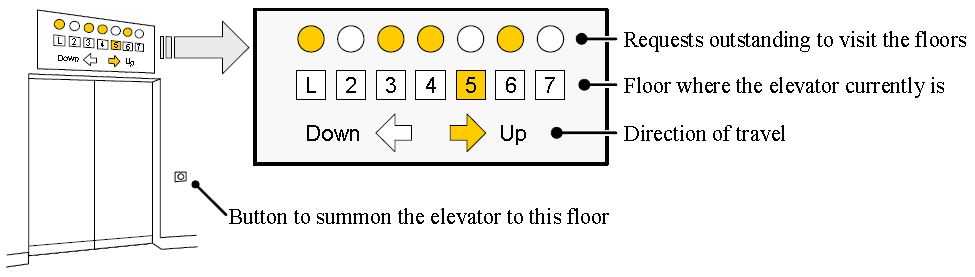
To prevent dictionary attacks, we need to count the unsuccessful attempts, but also need to reset the counter if the user leaves before providing a valid key or reaching **maxNumOfAttempts** (the maximum allowed number of unsuccessful attempts).

**UML State Diagram**



## Practical: Elevator Control

Consider developing a software system to control an elevator in a building. Assume that there will be a **button at each floor** to summon the elevator, and a set of **buttons inside** the **elevator car**—one button per floor to direct the elevator to the corresponding floor. Pressing a button will be detected as a pulse. When pressed, the button is illuminated. At each floor, there will be a **floor sensor** that is “on” when the elevator car is within 10 cm of the rest position at the **floor**.



The information panels will have two lamps representing each floor (see the figure above). A **square lamp** indicates that the car is at the corresponding floor, and a **round lamp** indicates that there is a request outstanding for the elevator to visit the corresponding floor. In addition, there will be two **arrow-shaped lamps** to indicate the current direction of travel. For example, in the figure above, the panel indicates that the elevator car is currently on the fifth floor, going up, and there are outstanding requests to visit the lobby, third, fourth, and sixth floor.

After the elevator visits a requested floor, the corresponding lamp on all information panels should be turned off. Also, the button that summoned the elevator to the floor should be turned off.

Let us assume that the outstanding requests are served so that the elevator will first visit all the requested floors in the direction to which it went first after the idle state. After this, it will serve the requests in the opposite direction, if any. When the elevator has no requests, it remains at its current floor with its doors closed.

**Practice:**

Design the UML State Diagram