Design and Implementation of Secure Control Architecture for Unmanned Aerial Vehicles

Hyeok-June Jeong[1] and Young-Guk Ha[2*]

[1]Department of Mechanical Engineering, Konkuk University, Neungdong-Ro, Gwangin-Gu, Seoul 143-701, Korea
[2]Department of Computer Science, Konkuk University, Neungdong-Ro, Gwangin-Gu, Seoul 143-701, Korea
amitajung@naver.com and ygha@konkuk.ac.kr

Abstract

This paper describes secure control architecture for UAV control system for multiple operator environments. Multiple access of UAV is necessary because UAV must capture and process various data and UAV's functions become more complex and diverse. From this perspective, UAV control system needs to be secure because new control system will be exposed to multiple operators. To discuss this issue, the paper proposes new control architecture for UAV using RBAC (Roll Based Access Control). Generally, RBAC is applied and contributes to the security of computer systems, thus, we modified RBAC to apply to UAV control system. In this paper, we present how the modified RBAC model can be applied to UAV control systems and implement a prototype UAV control system based on the modified RBAC model.

Keywords: UAV (Unmanned Aerial Vehicle), RBAC (Roll Based Access Control)

1. Introduction

Nowadays, there are many researches on UAV (Unmanned Aerial Vehicle) control systems for disaster monitoring, national defense and surveillance applications, which mainly focus on flight performance of UAV or multiple UAV control by single operator. Thanks to the excellent technology, we can use UAV safely and conveniently as our purpose. Despite these advances, complex and intelligent UAV control system is controlled by just single operator. So this paper approaches come from a slightly different view point. That is the control system makes it possible to do multiple accesses to one UAV to deal function and share information. This system has an advantage of increasing efficiency due to the sharing of limited resources and it has possibilities of distributed processing control. But indiscriminate access and control should not be allowed.

On the other hand, if there are some rules which are named permission regarding access and control of each operator, the system allows control the UAV which is perform more complex functions and information very effectively. RBAC is the most appropriate model for this system. RBAC is used in various fields of computer science, but it need to slight variation of the model in order to satisfy the demands of the system. So this paper presents a Modified RBAC model then, the modified model will be applied to the UAV control system. Then this system will be implemented as quad rotor control system for multiple operator environments and analyze the results.

This paper will describe five chapters. In Chapter 2 we will discuss the related research and in Chapter 3, architecture of the system will be discussed. Chapter 4 has contents about implementation and experiments. Finally in Chapter 5, the future research plan will be described.

* Corresponding Author

2. Related Work

In the field of computer security, RBAC is a useful concept that has been researched for a longtime. Ferraiolo, *et al.*, presents concept of national standards for RBAC model [1]. Barkley, *et al.*, interpreter RBAC model from a mathematical point of view and represented by the formula. However, this study was limited to the company's intranet [2].

For the last decade, UAV control system has received attention from many researches. Many systems were designed, for example Grzonka, *et al.*, research for indoor quad rotor flight control system that pilot quad rotor in the indoor space that is recognized by real-time image processing [3]. LihuiGu, *et al.*, focus on systems which are flying in formation, and each UAV is designed hierarchically depending on UAV's roles like flight leader or wingman [4]. Ly, *et al.*, design control system for hierarchical UAV by multi-layer communication to control effectively [5] by extending the concept of [4]. Gancet Ly, *et al.*, research is task planning for control multi-UAV. Each UAV carry out missions themselves and it also can make decisions by a particular algorithm [6].

Although there are many researches about the UAV control system, but most of all researches provide control system for Single-Access. So in this study, the control system will complement from a security point of view. Then we can make batter system by increasing the efficiency of information processing.

3. System Design

3.1. Modified RBAC

Ferraiolo, *et al.*, presents The NIST Model for Role-Based Access Control in field of computer security. According to this paper, RBAC is classified into two categories: Hierarchical RBAC, Flat RBAC. Those two RBAC model was distinguished by checking the presence of hierarchical roles in its structure. If there are no hierarchical roles, it will be flat RBAC.

In this system, it should be flat RBAC. Those settings for the following reasons. Suppose there are of two type role: Pilot, SensorMonitor. The role of authority is determined by only assigning permissions not by other roles. It should be perform independently to avoid confusion and to increase the efficiency of the authorization. So in this system has no Role Hierarchy.

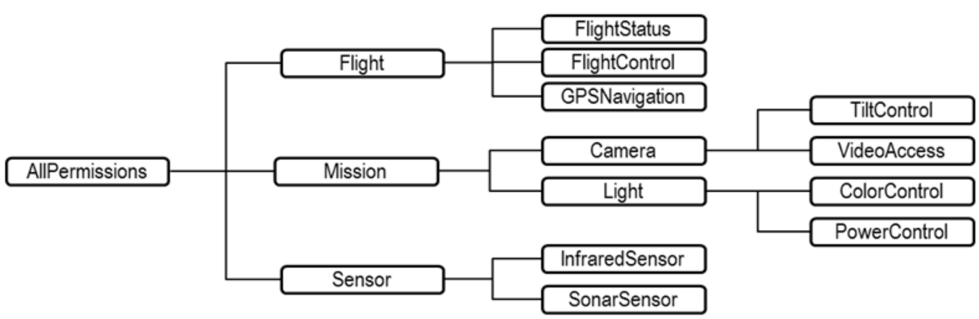

Figure 1. Permission hierarchy

In this system, it is more efficient to design hierarchical permissions. It makes simple to grant permissions to each role. For example pilot role will be just given permission Flight, then pilot can use the sub-permissions such as Flight Status, Flight Control and GPS Navigation. Suppose another situation, camera man role will be granted Camera permission, then camera man can control tilt and video options. Administrator is granted root permission named AllPermission then it can use all functions. Of course it is possible to get receiving other permissions, for example pilots can be granted Flight permission and Tilt control.

From this perspective, the existing flat RBAC can be more useful by adding permission hierarchy to flat RBAC. It named Modified RBAC and following figure shows the changes briefly.

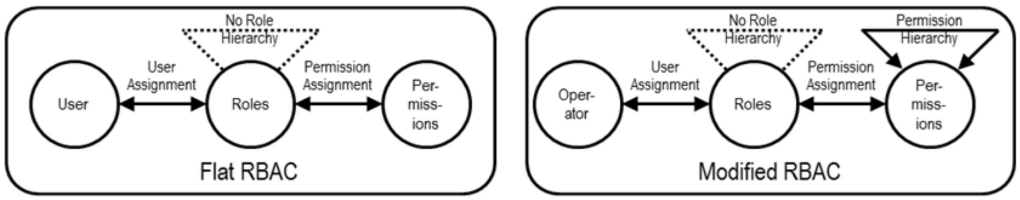

Figure 2. Flat RBAC and Modified RBAC

As shown in the figure above, the Modified RBAC as a slight variation of an original model structure. The modified RBAC satisfies all needs in this UAV control system and it is also include security and authorization features of the original model.

3.2. Overall Architecture

This system basically contains the algorithm for processing the data given in the UAV and it has the technology about the sending packet that contains the meaning of the command to the control UAV. And then, different kind of concept is grafted to the system. Because this system is a special case of control system, it has multi-operators.

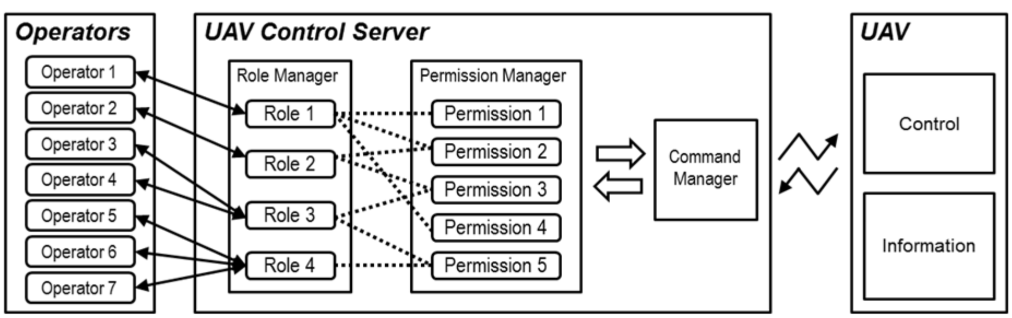

Figure 3. Architecture: Interactions between UAV, UAV Control Server, and access Operator

This system consists of three functional parts which are Operator, UAV Control Server and UAV. Each part is designed independently in order to effectively perform their functions. This part makes difference with existing UAV control system by designing associate with Role based access control. So this chapter will be discussed in detail.

'Operators' is a group of operators (users) who are connecting to the UAV control server to use the UAV. All operators in the Group should be assigned their role in order to control the UAV at the same time by unspecified number of users all. Because it is essential to limit the commanding rights and protect the information from UAV. This process is the same as User assignment in a generic RBAC model. And it can be represented by the following algebraic expressions.

$$\mathrm{OA(O,R)} = (\overset{\forall}{_{n}}\mathrm{O{:}\ Operator}) \rightarrow (\overset{\exists}{_{1}}\mathrm{R{:}\ Role})$$

Where O = the set of individuals Operator and R = the set of pre-defined Roles. OA = the assignment relationship between Operator and Role. ∀ = all of Operators must participate in OA relation and ∃ = all of Roles don't need to participate in OA relation. Expression 'n' means many Operators and '1' means just one Role. Expression '→' is assignment. This relation can be explained that all operators should be assigned just one Role.

Each role will be granted permission (In many cases, a role is granted multiple permissions.) by Role Manager depending on its mission then it can control UAV parts or get information according to their permissions. This step is named Permission assignment in the standard RBAC model and it is key concept because it makes a difference UAV control system from the past.

Roll Manager have a role to assign hierarchical permissions to each role. Some roles can be seen in the following table.

Table 1. Assign Permission

Role Name	Permissions
Administrator	AllPermissions
Pilot	Flight, VideoAccess
SensorMonitor	Sensors, FlightStatus
CameraMan	Camera, Light
Observer	VideoAccess

Administrator can do everything because it is assigned AllPermission by Permission Manger and AllPermission is the top level of authority because it can perform all of the lower level of permissions. Pilot assigned Flight and VideoAccess so it can use permission FlightControl, FlightStatus, GPSNavigation and VideoAccess, in other words it shows that one role can be granted various permission. It also means that Role Manager can define the number of Roles

Role Manager directly connects operators to each role and Role Manager can accept more than one operator per each role, of course each role which is connected with Operator will operate independently. So Role Manager can be represented by the following algebraic

$$\mathrm{PA(R,P^{+})} = (\overset{\forall}{_{n}}\mathrm{R{:}\ Role}) \rightarrow (\overset{\exists}{_{n}}\mathrm{P{:}\ Permission})$$

Where P = the set of pre-defined Permissions and R = the set of pre-defined Roles and PA = the assignment relationship between Operator and Role. ∀ = all of Roles must participate in OA relation and ∃ = all of Permissions don't need to participate in OA relation. Expression 'n' means many Operators and many Roles, '+' is same as the meaning of the regular expression. It means that one or more permissions can be assigned to a Role. Then expression '→' is assignment. So this relation means all Roles should be assigned one or more Permission.

As seen in the figure above, Modified RBAC is good for in the control system rather than flat RBAC. Because UAV has a lot of information and there are many elements to control. Above all, most permission has association with other permissions. For example, to flying

UAV, pilot should have the right to see flight status, GPS navigation As well as flight permission. Hierarchical permission makes simple Permission assignment so it increases the reliability of the system.

$$PH\big(P(p), P(c)^{+}\big) = (\,^{\exists}_{n}P(c): \text{Permission}) \subseteq (\,^{\exists}_{1}P(p): \text{Permission})$$

Where P = the set of pre-defined Permissions and Subscript (p) means parent permission and Subscript (c) means child permission. ∃ = all of Permissions don't need to participate in OA relation. Expression 'n' means many Permissions and '1' means just one Permission. '+' is same as the meaning of the regular expression. It means that one or more permissions can be related with their parent Role. PH is a relation of the hierarchical Permission relationship then expression '⊆' is hierarchical relation. So this algebraic expression can be explained that some permission may have child Permission one or more of the other Permissions.

These expressions above show that Modified RBAC model can be apply to the UAV control system by extending the original RBAC model. Now the UAV control system combined with RBAC and the system can absorb the advantages of original RBAC models. These three processes can be expressed in following single architecture.

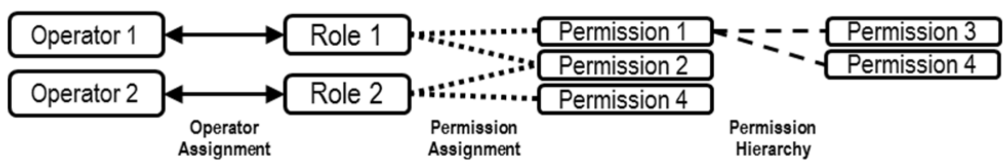

Figure 4. Apply Modified RBAC model in UAV control system

The architecture above shows Modified RBAC model which has many operators and permission hierarchy. In this system, however, the problem remains. It is about how to handle the concurrency control. The solution for this problem is presented in the following.

It is impossible to control multiple physical control parts at the same time. So permission manager grants exclusive rights to the each role.

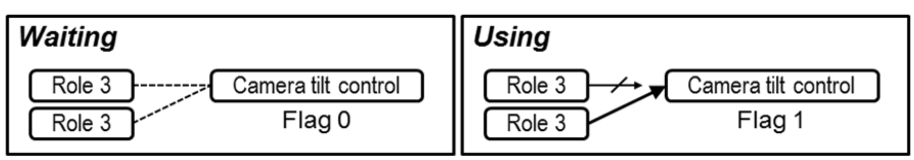

Figure 5. Semaphore on Permission Manager

Suppose situation that two users were granted same permission 'Role3'. If the Role3 has authority to control Camera tilt, two users control one camera tilt. But it cannot be controlled at the same time, one user who approach first have an exclusive license to control it by assigning flag.

4. Implementation and Experiments

4.1. Implementation

In order to implement the designed model software structure was devised, as shown in the following figure. This system uses optimization solution to operate their functions and run independently. In other words, this system`s modules have low coupling by designing them independently and high cohesion by exchanging packets that comply with protocol.

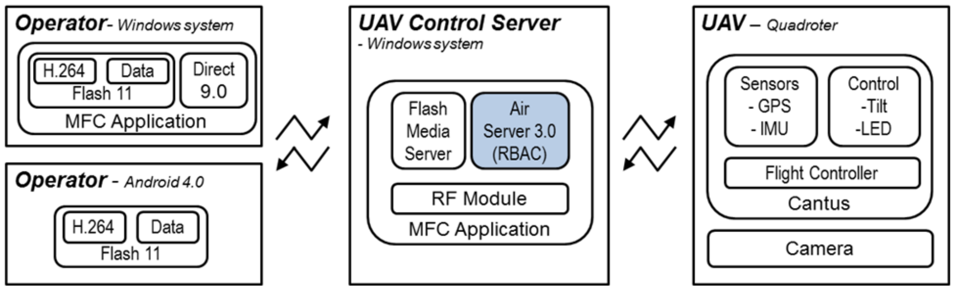

Figure 6. System implementation

In UAV, there is main CPU called Cantus that is 32bit microprocessor. It is very important in piloting UAV such as hovering by generating PWM signal to control each BLDC motor. Cantus controls LED light and camera tilt. In addition, Cantus's another important role is combining the information obtained from the UAV, transmitting it to server in the form of a packet through the RF module and, interpreting the packets coming from sever and performing the command.

UAV Control Server is a bridge that connects the UAV and client. In order to perform this function, RF-module is mounted on the server and then operating the RF module through serial communication. MFC (C + +) language is used in controlling module because it is very useful in using an external device. Also other programming languages, Flex (Action script3.0), java and C, are used because each language has different strength.

Then programs are combined into one executable program with nested programming technology. Flash media server provides an environment that can stream H.264 video in real-time. Air server3.0 technology gives a solution for constructing TCP/IP server simply that provides multi socket. Modified RBAC is implemented using Air server3.0 language by control command/status packet (JSON). Techniques mentioned above are specialized in operating special function and programs can run as a single executable file on any PC. So this server can be built easily and work lightly by executing applications.

Operator is designed for users to control in real-time. So it needs to access to Air server3.0 through TCP/IP for exchanging packets. Also it should display H.264 video through RTMP protocol. In addition, lots of information should be effectively displayed so that GUI (Graphic User Interface) client application could be adapt. Another important thing is Portability because it is used in many fields such as PC, laptop and smart devices. Flash11 can solve all these problems. Flash11 is specialized in multimedia like video stream and provides optimal environment to make GUI. Especially its compatibility is very good. It can be executed even on the web browser such as Explorer and Chrome.

4.2. Experimental Results

In this system, quad rotor platform is used as UAV. The object of first experiment is checking flight controllability because it is most important and basic in this system. The result is satisfying. Controllability is good and reaction time is fast. Also its flight was stable without any error. Thus, first experiment proves integrity of whole system which consists of many modules that are made on different of platforms and written by different programming languages. Then, the system functionality has been expanded.

Figure 7. Experimental result - Single operator (Operating on Windows7)

As shown in Figure 7, Operate application is developed to an integrated information control system. It can stream H.264 video in real time (delay within 0.5second) and can pilot quad rotor by using joystick on windows7 system or using G-sensor on Android 4.0 Smart phone. It can also control other parts such as Camera tile, Light. Then HUD-like UI is adopted to display IMU, GPS (with Google Map) and other sensors more effectively.

Figure 8. Experimental result - Multi operator (Operating on Windows7 and Android)

Multi-access environment is implemented as shown in Figure 8. Operator1 (left hand side) is an Administrator on windows7 system and operator2 (right hand side) is a Camera man on Android 4.0. In this situation operator1 pilots the quad rotor and operator 2 controls camera tilt at the same time. Another situation is applied to the system. It is about granting exclusive right while Operator 2 is controlling camera tilting. Of course, other operators (including Operator 1) should not be able to control the same time.

5. Conclusion and future works

This work presents requirement of a multi-access environment for UAV control system because UAV's functions will become more complex and diverse in the future. So it is need to apply security model which is named Roll Based Access Control to improve orderly and efficiency of multiple operators. Thus, Modified RBAC was applied to the flight system which has permission hierarchy by extending flat RBAC.

The control system needs to pilot UAV without delay while other parts such as light and camera tilting are being controlled by multi operators. For real-time implementation, the system used various programming languages which are specialized in special function. Some physical functions should be granted exclusive rights and there are effort to improve the reliability of the system by optimizing wireless communications and systemizing the packets.

Experimental results are successful. However, there is some limitation because this research only targeted the design and implementation of the system. It was time consuming to design and implementation for UAV embedded systems. So this research can’t focus on prove effectiveness of the system. Leaves this part as following research topics, proof excellence of UAV control system by expanding the system as a sort of cloud UAV control system to process effectively high-precision sensor data.

Acknowledgements

This research was supported by the Basic Science Research Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (grant number: 2012006817).

References

[1] R. Sandhu, D. Ferraiolo and D. R. Kuhn, "The NIST Model for Role-Based Access Control: Towards A Unified Standard", In: 5th ACM Workshop Role-Based Access Control, **(2000)**, pp. 47–63.

[2] J. Barkley, D. Ferraiolo and D. R. Kuhn, "A Role Based Access Control Model and Reference Implementation within a Corporate Intranet", In: ACM Transactions on Information and System Security (TISSEC), **(1999)**, pp. 34 - 64.

[3] S. Grzonka, G. Grisetti and W. Burgard, "A Fully Autonomous Indoor Quadrotor", In: IEEE Transaction on Robotics, vol. 28, no. 1, **(2012)**.

[4] D. LihuiGu, G. Pei, H. Ly, M. Gerla and X. Hong, "Hierarchical Routing for Multi-Layer Ad-Hoc Wireless Networks with UAVs", In: 21st Century Military Communications Conference Proceedings, MILCOM, **(2000)**.

[5] H. Ly, H. Hong, M. Gerla, G. Pei and Y. Lee, "C-ICAMA, a centralized intelligent channel assigned multiple access for multi-layer ad-hoc wireless networks with UAVs", In: Wireless Communications and Networking Confernce, WCNC 2000 IEEE, **(2000)**.

[6] J. Gancet, G. Hattenberger, R. Alami and S. Lacroix, "Task planning and control for a multi-UAV system: architecture and algorithms", In: Intelligent Robots and Systems, IEEE/RSJ International Conference on IROS, **(2005)**.

[7] G. Brookshear, "Computer Science An overview", 10th Edition, PEARSON, **(2010)**, pp. 135-141.

[8] Role based access control, http://en.wikipedia.org/wiki/Role-based_access_control.