Creating Multiple security layers for servers using security groups and NACL in AWS

[1] SK.Ayesha [2] Dr.P.Venkateswar lal, M.Tech, P.h.d.

[1] *CSE, NEC., Gudur*
[2] Professor, *CSE Department, NEC, Gudur*

Abstract: *In the realm of cloud computing, ensuring the security of servers is paramount. Amazon Web Services (AWS) offers robust tools to fortify server defenses, primarily through Security Groups (SGs) and Network Access Control Lists (NACLs). This paper explores the design and implementation of multiple security layers leveraging SGs and NACLs within AWS.*

Firstly, the paper delineates the conceptual framework of SGs and NACLs, elucidating their respective functionalities and differences. SGs operate at the instance level, acting as virtual firewalls that control inbound and outbound traffic. On the other hand, NACLs operate at the subnet level, governing traffic flow at a broader scope.

Subsequently, the paper presents a strategic approach to crafting multi-layered security architectures using these AWS constructs. By integrating SGs and NACLs, administrators can erect a formidable defense mechanism against diverse cyber threats. The strategy involves segmenting server resources into distinct security zones based on sensitivity levels and access requirements. Each zone is then fortified with tailored SG rules and NACL configurations to regulate traffic flows meticulously.

Furthermore, the paper elucidates best practices and considerations for optimizing security configurations within AWS. It underscores the significance of granular permissions, regular audits, and continuous monitoring to uphold the integrity of the security posture. Additionally, it delves into the role of AWS services like AWS Identity and Access Management (IAM) and AWS CloudTrail in bolstering security governance and compliance.

Keywords: *AWS, EC2 Instance, security groups , Network access control list.*

I. INTRODUCTION

In the landscape of cloud computing, securing resources within virtual environments is a fundamental concern. Amazon Web Services (AWS), as a leading cloud services provider, offers a suite of tools and features to fortify the security of infrastructure and applications. Two key components in AWS for managing network security are Security Groups (SGs) and Network Access Control Lists (NACLs).

Security Groups serve as the first line of defense in AWS network security. They act as virtual firewalls at the instance level, controlling inbound and outbound traffic based on defined rules. SGs are stateful, meaning they automatically allow return traffic for requests originating from the allowed sources. This feature simplifies network security management by reducing the need for explicit outbound rules.

Network Access Control Lists (NACLs):

Network Access Control Lists operate at a different layer compared to SGs. While SGs govern traffic at the instance level, NACLs provide control at the subnet level. NACLs function as stateless packet filters, enabling administrators to define rules for both inbound and outbound traffic. Unlike SGs, NACLs require explicit rules for return traffic, adding a layer of granularity and control over network traffic flows. The synergy between SGs and NACLs allows AWS users to implement multi-layered security architectures tailored to their specific

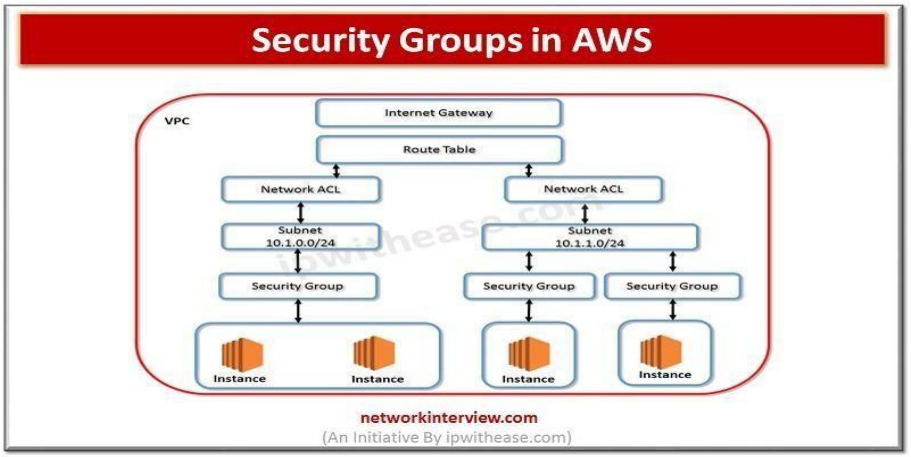
requirements. By strategically combining these constructs, organizations can segment their network infrastructure, enforce access controls, and mitigate various security threats effectively.

TYPES OF SECURITY GROUPS

In Amazon Web Services (AWS), security groups (SGs) are virtual firewalls that control inbound and outbound traffic for instances. There are primarily two types of security groups in AWS based on their scope and application:

1. **EC2-Classic Security Groups:** These security groups are associated with instances launched in the EC2-Classic network mode. In this networking mode, instances reside in the AWS data centers and communicate with the internet directly without VPC (Virtual Private Cloud) isolation. EC2-Classic Security Groups allow you to control traffic to and from instances within the EC2-Classic environment.

2. **VPC Security Groups:** VPC security groups are associated with instances launched within a Virtual Private Cloud (VPC). VPCs provide network isolation and segmentation within the AWS cloud, allowing you to create a logically isolated section of the AWS cloud where you can launch your resources. VPC security groups enable you to control traffic to and from instances within a VPC, offering more granular control over network traffic compared to EC2-Classic Security Groups.



II. APPLICATIONS OF SECURITY GROUPS

Instance Level Security: SGs are applied at the instance level, acting as virtual firewalls. They control inbound and outbound traffic based on user-defined rules. SGs are effective in limiting access to specific ports and protocols on individual instances, thereby reducing the attack surface and preventing unauthorized access.

Micro segmentation: SGs enable micro segmentation by grouping instances with similar roles or functions and applying specific security policies to each group. For example, web servers may have different SG rules than database servers, ensuring that only necessary traffic is allowed between them.

Dynamic Scaling: SGs seamlessly integrate with AWS services like Auto Scaling, allowing security policies to adapt dynamically as instances are launched or terminated. This ensures that newly launched instances inherit the appropriate security configurations automatically.

Integration with Load Balancers: SGs can be associated with Elastic Load Balancers (ELBs) to control traffic flow between clients and backend instances. This enables administrators to enforce security policies for incoming traffic to the load balancer and outgoing traffic from the backend instances.

APPLICATIONS OF NACL

Subnet Level Security: NACLs operate at the subnet level, providing an additional layer of security for traffic entering and leaving subnets within a Virtual Private Cloud (VPC). They complement SGs by filtering traffic before it reaches the instances, regardless of instance-level security settings.

Traffic Control Between Subnets: NACLs enable administrators to enforce network segmentation and control traffic flow between different subnets within a VPC. By defining ingress and egress rules based on IP addresses, protocols, and ports, NACLs prevent unauthorized communication between subnets.

Defense Against DDoS Attacks: NACLs can be configured to mitigate Distributed Denial of Service (DDoS) attacks by blocking or rate-limiting suspicious traffic at the subnet boundary. This helps in maintaining the availability and performance of critical applications during attacks.

Compliance Requirements: NACLs are often used to enforce network-level compliance requirements, such as restricting access to sensitive data or ensuring separation of development, testing, and production environments within a VPC.

III. IMPLEMENTATION

Following are the steps followed to implement the security groups and NACL

1. Create the VPC
2. ASSIGN the VPC to Security groups
3. Create the server EC2 instances
4. Config the result of the server
5. Create the another VPC for nacl
6. Creating two subnets
7. Configure the result

Create the VPC

Sign to the AWS console

Navigate to the VPC Dashboard: Once logged in, navigate to the "Services" dropdown menu at the top of the page and select "VPC" under the "Networking & Content Delivery" section. This will take you to the VPC Dashboard.

Start the VPC Creation Wizard: On the VPC Dashboard, locate the "Your VPCs" section and click on the "Create VPC" button.

Tenancy: Choose the tenancy option for your VPC. By default, it's set to "Default," which means instances launched in the VPC will run on shared hardware. You can also choose "Dedicated" if you need instances to run on dedicated hardware.

Configure Additional Settings (Optional): You can configure additional settings such as DNS resolution, DNS hostname, and tagging for your VPC. These settings are optional but can be useful for managing and identifying resources within your VPC.

Create the VPC: After configuring the VPC settings, review your choices and click on the "Create VPC" button to create the VPC.

Verify VPC Creation: Once the VPC creation process is complete, you'll see a confirmation message indicating that the VPC has been created successfully. You can also view the newly created VPC in the list of VPCs on the VPC Dashboard.

That's it! You've successfully created a Virtual Private Cloud (VPC) in your AWS account. You can now proceed to configure subnets, route tables, security groups, and other networking components within your VPC to build your desired network.

Creating the security groups

Creating a security group in AWS is a fundamental step in securing your resources within a Virtual Private Cloud (VPC). Here's a step-by-step guide on how to create a security group using the AWS Management Console:

Sign in to the AWS Management Console:

Visit the AWS Management Console at https://aws.amazon.com/console/ and sign in with your AWS account credentials.

Navigate to the EC2 Dashboard: Once logged in, navigate to the "Services" dropdown menu at the top of the page and select "EC2" under the "Compute" section. This will take you to the EC2 Dashboard.

Access the Security Groups Section: In the EC2 Dashboard, locate the "Network & Security" section in the navigation pane on the left-hand side. Click on "Security Groups" to access the list of existing security groups.

Create a New Security Group:On the Security Groups page, click on the "Create security group" button located at the top of the page.

Configure Security Group Settings:In the "Create security group" wizard, you'll need to provide the following information:

Security group name: Give your security group a descriptive name to identify its purpose.

Description: Optionally, provide a brief description of the security group's purpose.

VPC: Select the VPC where you want to create the security group.

Configure Inbound Rules:

Define the inbound rules to control the traffic allowed to reach instances associated with this security group. You can add rules to allow specific protocols (e.g., SSH, HTTP, HTTPS) from specific IP ranges or from other security groups within the same VPC.

Configure Outbound Rules:

Define the outbound rules to control the traffic allowed to leave instances associated with this security group. By default, all outbound traffic is allowed, but you can restrict it based on your requirements.

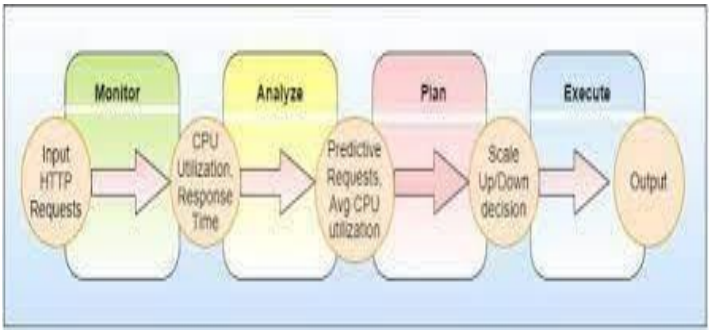
Review and Create the Security Group:

Review the configured settings to ensure they meet your requirements. Once you're satisfied, click on the "Create security group" button to create the security group.

Verify Security Group Creation:

After creating the security group, you'll see a confirmation message indicating that the security group has been created successfully. You can also view the newly created security group in the list of security groups on the Security Groups page

MAPE (Monitor, Analyze, Plan, Execute)



Amazon Elastic Compute Cloud (EC2) is a web service that provides resizable compute capacity in the cloud. To launch an EC2 instance (i.e., a virtual server), follow these steps:

- Navigate to the EC2 dashboard within the AWS Management Console.
- Click on the "Launch Instance" button to begin the instance creation process.
- Choose an Amazon Machine Image (AMI), which is a pre-configured template for your server. You can select from a variety of operating systems and software configurations.
- Select an instance type based on your resource requirements.
- Configure instance details such as the number of instances, network settings, and storage options.
- Optionally, add tags to your instance for easier management and organization.
- Configure security groups to control inbound and outbound traffic to your instance. Define rules for protocols, ports, and IP ranges.

Access Your Server:

Once your EC2 instance is running, you can access it using SSH (Secure Shell) for Linux instances or RDP (Remote Desktop Protocol) for Windows instances. Use the key pair you created during instance launch to authenticate the SSH or RDP session.

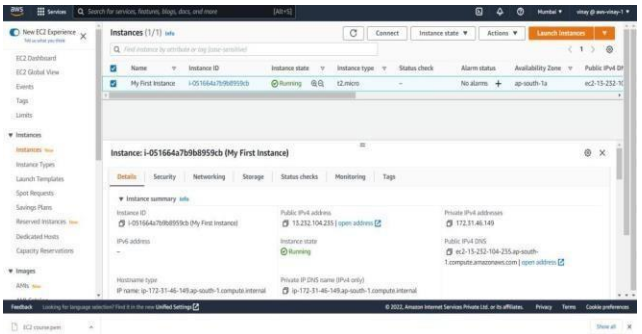


Fig-3 EC2 instance

Configure the result in mobeXterm:

1. **Connect to the EC2 instance:** Once the EC2 instance is created and running, you would use MobXterm to establish an SSH connection to the instance. To do this, open MobXterm and use the SSH option to connect to the public IP address or DNS name of your EC2 instance. You'll need the private key (.pem) file associated with the EC2 instance to authenticate.

2. **Enter credentials (if required):** Depending on the configuration of your EC2 instance, you may need to enter a username and/or password to authenticate. For Amazon Linux and most other Linux-based instances, the default username is typically "ec2-user" or "ubuntu", depending on the AMI used.

3. **Verify connectivity:** Once connected, you should see the command line interface of the EC2 instance within MobXterm. You can execute commands to verify the instance's status, check logs, install software, etc. For example, you can use commands like **ls** to list files, **sudo systemctl status** to check service status, or any other command relevant to your use case.

4. **View output:** Any output or result of commands executed on the EC2 instance will be displayed within the MobXterm terminal window. You can scroll through the output to see the results of your commands.

If you're looking to view the result of EC2 instance creation itself (e.g., status of the instance, public IP address, etc.), you would typically do that in the AWS Management Console or by using AWS CLI commands like **aws ec2 describe-instances**. MobXterm is used for accessing and interacting with existing instances rather than managing the infrastructure itself.

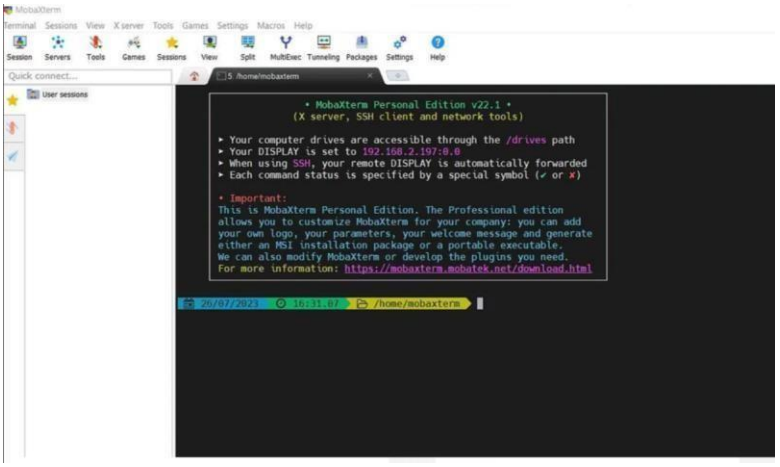


Fig-4 Ec2 instance with security group

Creating the NACL

Creating a Network Access Control List (NACL) in AWS involves several steps. Here's a guide on how to create a NACL using the AWS Management Console:

Sign in to the AWS Management Console: Visit the AWS Management Console at **https://aws.amazon.com/console/** and sign in with your AWS account credentials.

Navigate to the VPC Dashboard: Once logged in, navigate to the "Services" dropdown menu at the top of the page and select "VPC" under the "Networking & Content Delivery" section. This will take you to the VPC Dashboard.

Access the Network ACLs Section: In the VPC Dashboard, locate the "Security" section in the navigation pane on the left-hand side. Click on "Network ACLs" to access the list of existing NACLs.

Create a New Network ACL: On the Network ACLs page, click on the "Create network ACL" button located at the top of the page.

Configure Network ACL Settings: In the "Create network ACL" wizard, you'll need to provide the following information

Name: Give your NACL a descriptive name to identify its purpose.

VPC: Select the VPC where you want to create the NACL.

Create Inbound and Outbound Rules: Define the inbound and outbound rules for the NACL.

Each rule specifies a set of conditions (e.g., protocol, port range, source or destination IPaddress and an action (allow or deny). You can add multiple rules to control traffic flow in both directions.

Associate Subnets (Optional): After creating the NACL, you can optionally associate it with one or more subnets within the selected VPC. By default, a new NACL allows all traffic, but you can modify the rules as needed to restrict traffic based on your requirements.

Review and Create the Network ACL: Review the configured settings to ensure they meet your requirements. Once you're satisfied, click on the "Create network ACL" button to create the NACL.

Verify Network ACL Creation: After creating the NACL, you'll see a confirmation message indicating that the NACL has been created successfully. You can also view the newly created NACL in the list of NACLs on the Network ACLs page.

That's it! You've successfully created a Network Access Control List (NACL) in your AWS account. You can now associate this NACL with your subnets to control inbound and outbound traffic based on the defined rules.

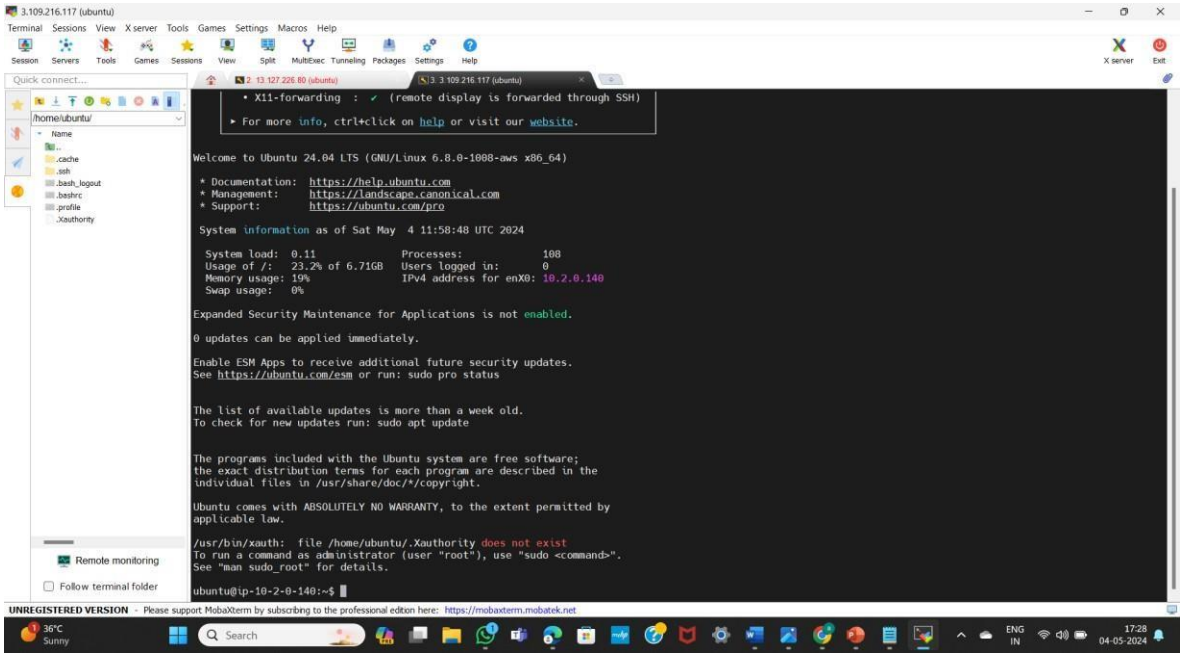
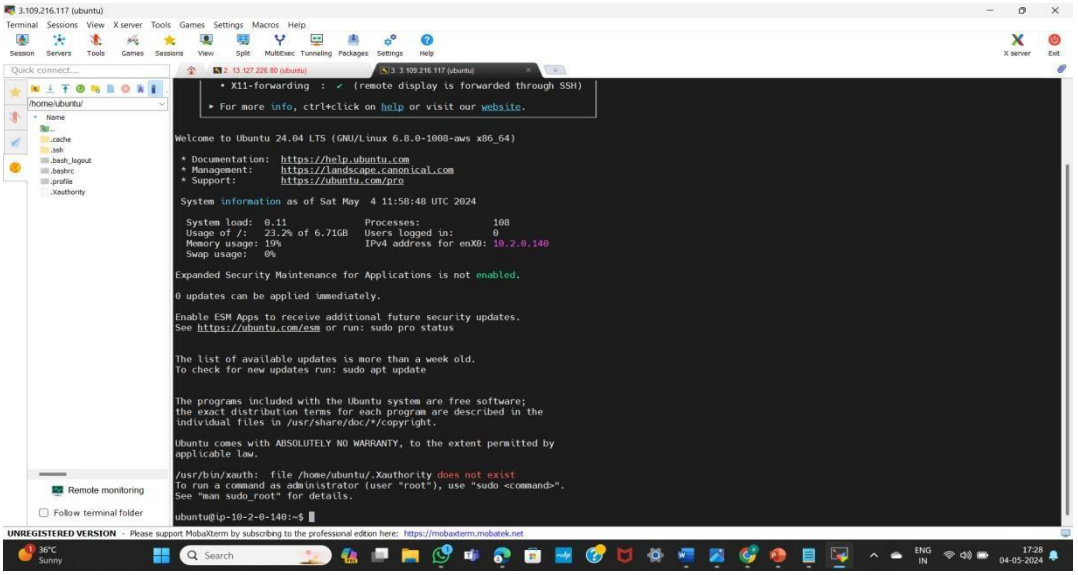


Fig -5

IV. RESULTS AND ANALYSIS

The server is successfully SGs act as a stateful firewall at the instance level within your VPC. They control inbound and outbound traffic based on security group rules. NACLs act as a stateless firewall at the subnet level within your Virtual Private Cloud (VPC). By default, they deny all inbound and outbound traffic unless explicitly allowed by rules



V. CONCLUSION

By creating NACLs and SGs, you establish a multi-layered defense strategy to protect your AWS resources. NACLs operate at the subnet level, controlling traffic flow between subnets, while SGs operate at the instance level, regulating inbound and outbound traffic for individual instances. NACLs and SGs offer granular control over network traffic based on various parameters such as IP addresses, ports, and protocols. This flexibility allows you to tailor security policies to meet specific requirements, ensuring that only authorized traffic is permitted while blocking unauthorized access.SGs provide stateful filtering, meaning they automatically allow return traffic related to established connections. On the other hand, NACLs offer stateless filtering, requiring explicit rules for both inbound and outbound traffic. Understanding these differences is crucial for designing effective security architectures.Implementing both NACLs and SGs enables a defense-in-depth approach to network security. NACLs serve as the first line of defense, controlling traffic at the subnet boundary, while SGs add an additional layer of security by filtering traffic at the instance level. This layered approach enhances overall security posture and mitigates the risk of unauthorized access or data breaches.In conclusion, creating NACLs and SGs in your AWS account is a fundamental step in establishing a secure and compliant cloud environment. By leveraging these AWS networking constructs, youcan implement robust security policies, protect sensitive data,mitigate risks effectively,

ensuring the integrity and availability of your resources in the cloud.

VI. REFERENCES

[1]E. Thomas, P. Ricardo, and M. Zaigham, *Cloud Computing: Concepts, Technology, and Architecture*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall, 2013.

[2]M. Peter and G. Timothy, ''The NIST definition of cloud computing recommendations of the national institute of standards and technology,'' NIST, Gaithersburg, MD, USA, Tech. Rep. 800-145, 2013.

[3]*Information Technology-Cloud Computing-Reference Architecture*, Standard ISO/IEC 17789:2014, 2014.

[4]*Information Technology-Cloud Computing-Overview and Vocabulary*, Standard ISO/IEC 17788:2014, 2014.

[5]L. Zhou, V. Varadharajan, and M. Hitchens, ''Achieving secure role-basedaccess control on encrypted data in cloud storage,'' *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 12, pp. 1947–1960, Dec. 2013.

[6]S. Kamara and K. Lauter, ''Cryptographic cloud storage,'' in *Proc. Int. Conf. Financial Cryptogr. Data Secur.*, Tenerife, Spain, 2010, pp. 136–149,doi: 10.1007/978-3-642-14992-4_13.

[7]X. Dong, J. Yu, Y. Luo, Y. Chen, G. Xue, and M. Li, ''Achieving secure and efficient data collaboration in cloud computing,'' in *Proc. IEEE/ACM 21st Int. Symp. Qual. Service (IWQoS)*, Jun. 2013, pp. 1–6, doi: 10.1109/IWQoS.2013.6550281.

[8]M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski,G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, ''A view of cloud computing,'' *Commun. ACM*, vol. 53, no. 4, pp. 50–58,

[9]L. Zhou, V. Varadharajan, and M. Hitchens, ''Trust enhanced cryptographic role-based access control for secure cloud data storage,'' *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 11, pp. 2381–2395, Nov. 2015.

[10]S. G. Akl and P. D. Taylor, ''Cryptographic solution to a problem of access control in a hierarchy,'' *ACM Trans. Comput. Syst.*, vol. 1, no. 3, pp. 239–248, 1983.