

Lab - Accessing Network Devices with SSH

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/1	192.168.1.1	255.255.255.0	N/A
S1	VLAN 1	192.168.1.11	255.255.255.0	192.168.1.1
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1

Objectives

Part 1: Configure Basic Device Settings

Part 2: Configure Basic Security Measures on the Router and Switch

Part 3: Configure the Router for SSH Access

Part 4: Configure the Switch for SSH Access

Part 5: SSH from the CLI on the Switch

Background / Scenario

In the past, Telnet was the most common network protocol used to remotely configure network devices. Telnet does not encrypt the information between the client and server. This allows a network sniffer to intercept passwords and configuration information.

Secure Shell (SSH) is a network protocol that establishes a secure terminal emulation connection to a router or other networking device. SSH encrypts all information that passes over the network link and provides authentication of the remote computer. SSH is rapidly replacing Telnet as the remote login tool of choice for network professionals. SSH is most often used to log in to a remote device and execute commands; however, it can also transfer files using the associated Secure FTP (SFTP) or Secure Copy (SCP) protocols.

The network devices that are communicating must be configured to support SSH in order for SSH to function. In this lab, you will enable the SSH server on a router and then connect to that router using a PC with an SSH client installed. On a local network, the connection is normally made using Ethernet and IP.

Note: The routers used with CCNA hands-on labs are Cisco 1941 Integrated Services Routers (ISRs) with Cisco IOS Release 15.2(4)M3 (universalk9 image). The switches used are Cisco Catalyst 2960s with Cisco IOS Release 15.0(2) (lanbasek9 image). Other routers, switches, and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and the output produced might vary from what is shown in the labs. Refer to the Router Interface Summary Table at the end of this lab for the correct interface identifiers.

Note: Make sure that the routers and switches have been erased and have no startup configurations. If you are unsure, contact your instructor.

Required Resources

- 1 Router (Cisco 1941 with Cisco IOS Release 15.2(4)M3 universal image or comparable)
- 1 Switch (Cisco 2960 with Cisco IOS Release 15.0(2) lanbasek9 image or comparable)
- 1 PC (Windows 7 or 8 with terminal emulation program, such as Tera Term, and Wireshark installed)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet cables as shown in the topology

Part 1: Configure Basic Device Settings

In Part 1, you will set up the network topology and configure basic settings, such as the interface IP addresses, device access, and passwords on the router.

Step 1: Cable the network as shown in the topology.

Step 2: Initialize and reload the router and switch.

Step 3: Configure the router.

- Console into the router and enable privileged EXEC mode.
- Enter configuration mode.
- Disable DNS lookup to prevent the router from attempting to translate incorrectly entered commands as though they were host names.
- Assign **class** as the privileged EXEC encrypted password.
- Assign **cisco** as the console password and enable login.
- Assign **cisco** as the VTY password and enable login.
- Encrypt the plaintext passwords.
- Create a banner that will warn anyone accessing the device that unauthorized access is prohibited.
- Configure and activate the G0/1 interface on the router using the information contained in the Addressing Table.
- Save the running configuration to the startup configuration file.

Step 4: Configure PC-A.

- Configure PC-A with an IP address and subnet mask.
- Configure a default gateway for PC-A.

Step 5: Verify network connectivity.

Ping R1 from PC-A. If the ping fails, troubleshoot the connection. – *Ping was Successful.*

Part 2: Configure Basic Security Measures on the Router

Step 1: Encrypt the clear text passwords.

```
R1(config)# service password-encryption
```

Step 2: Strengthen passwords.

An administrator should ensure that passwords meet the standard guidelines for strong passwords. These guidelines could include combining letters, numbers and special characters in the password and setting a minimum length.

Note: Best practice guidelines require the use of strong passwords, such as those shown here, in a production environment. However, the other labs in this course use the cisco and class passwords for ease in performing the labs.

- a. Change the privileged EXEC encrypted password to meet guidelines.

```
R1(config)# enable secret Enablep@55
```

(Note: password length must meet criteria being set later)

- b. Require that a minimum of 10 characters be used for all passwords.

```
R1(config)# security passwords min-length 10
```

Step 3: Secure the console and VTY lines.

- a. You can set the router to log out of a connection that has been idle for a specified time. If a network administrator was logged into a networking device and was suddenly called away, this command automatically logs the user out after the specified time. The following commands cause the line to log out after five minutes of inactivity.

```
R1(config)# line console 0
R1(config-line)# exec-timeout 5 0
R1(config-line)# line vty 0 4
R1(config-line)# exec-timeout 5 0
R1(config-line)# exit
R1(config)#
```

- b. The following command impedes brute force login attempts. The router blocks login attempts for 30 seconds if someone fails two attempts within 120 seconds. This timer is set especially low for the purpose of this lab.

```
R1(config)# login block-for 30 attempts 2 within 120
```

What does the **2 within 120** mean in the above command?

The two in the above command refers to the number of attempts the user can fail before being blocked for 30 seconds, where the two failed attempts are within 120 seconds of each other.

What does the **block-for 30** mean in the above command?

Block-for 30 refers to the number of seconds a user is blocked for until they are able to retry. As stated above, this timeout has been set for a short period for the lab. However, in a real-world application, the timeout would be higher to improve security.

Step 4: Verify that all unused ports are disabled.

Router ports are disabled by default, but it is always prudent to verify that all unused ports are in an administratively down state. This can be quickly checked by issuing the **show ip interface brief** command. Any unused ports that are not in an administratively down state should be disabled using the **shutdown** command in interface configuration mode.

```
R1# show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
Embedded-Service-Engine0/0	unassigned	YES	NVRAM	administratively down	down
GigabitEthernet0/0	unassigned	YES	NVRAM	administratively down	down
GigabitEthernet0/1	192.168.1.1	YES	manual	up	up
Serial0/0/0	unassigned	YES	NVRAM	administratively down	down
Serial0/0/1	unassigned	YES	NVRAM	administratively down	down

Part 3: Configure the Router for SSH Access

Using Telnet to connect to a network device is a security risk because all information is transmitted in a clear text format. SSH encrypts the session data and provides device authentication, which is why SSH is recommended for remote connections. In Part 2, you will configure the router to accept SSH connections over the VTY lines.

Step 1: Configure device authentication.

The device name and domain are used as part of the crypto key when it is generated. Therefore, these names must be entered prior to issuing the **crypto key** command.

- Configure the domain for the device.

```
R1(config)# ip domain-name ccna-lab.com
```

Step 2: Configure the encryption key method.

Command changed from "crypto key generate rsa modulus 1024" as this was incorrect.

```
R1(config)# crypto key generate rsa general-keys modulus 1024
```

The name for the keys will be: R1.ccna-lab.com

% The key modulus size is 1024 bits

% Generating 1024 bit RSA keys, keys will be non-exportable...

[OK] (elapsed time was 1 seconds)

```
R1(config)#
```

```
*Jan 28 21:09:29.867: %SSH-5-ENABLED: SSH 1.99 has been enable
```

Worked correctly.

Step 3: Configure a local database username.

```
R1(config)# username admin privilege 15 secret adminpass
```

The above command would not work, due to the earlier password constraint set, which limits passwords to be a minimum of 10 characters in length.

```
R1(config)# username admin privilege 15 secret Admin1p@55
```

Note: A privilege level of 15 gives the user administrator rights.

Step 4: Enable SSH on the VTY lines.

- a. Enable Telnet and SSH on the inbound VTY lines using the **transport input** command.

1) In a real-world application, Telnet should not be enabled as it is a security risk (as noted above).

```
R1(config)# line vty 0 4
```

```
R1(config-line)# transport input telnet ssh
```

The above command was incorrect, as it combined both telnet and ssh in one command. Instead, the command should be run separately for each, as below. However, due to the fact that telnet is not secure, I have not enabled it.

```
R1(config-line)# transport input ssh
```

```
R1(config-line)# transport input telnet
```

- b. Change the login method to use the local database for user verification.

```
R1(config-line)# login local
```

```
R1(config-line)# end
```

```
R1#
```

Step 5: Save the running configuration to the startup configuration file.

```
R1# copy running-config startup-config
```

```
Destination filename [startup-config]?
```

```
Building configuration...
```

```
[OK]
```

```
R1#
```

Step 6: Establish an SSH connection to the router.

- a. Start Tera Term from PC-A.

- b. Use Tera Term to telnet to R1.

Does R1 accept the Telnet connection? Explain.

R1 does not accept a Telnet connection as telnet has not been enabled. Telnet is not enabled as it is a security risk.

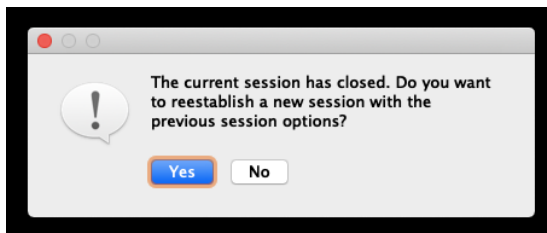


Figure 1. Screenshot showing failed telnet connection.

- c. Use Tera Term, establish an SSH session to R1. Use the username **admin** and password **Admin1p@55** (changed above to meet minimum criteria). You should be able to establish an SSH session with R1.

Does R1 accept the SSH connection?

The SSH connection was successful using the correct credentials, as shown below in the screenshot.

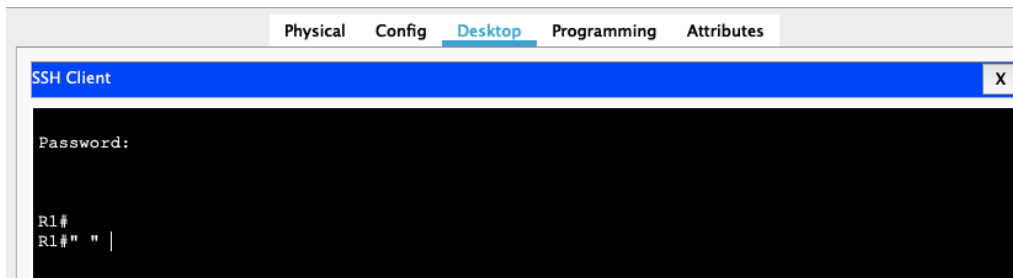


Figure 2. Screenshot showing successful SSH connection using credentials

- d. Intentionally mistype the user and password information to see if login access is blocked after two attempts.

What happened after you failed to login the second time?

After entering the incorrect credentials twice, the session was ended and the following alert was displayed.

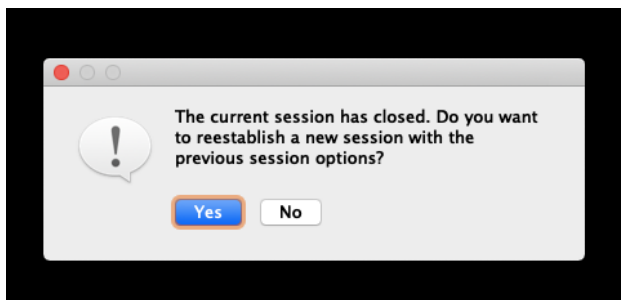


Figure 3. Screenshot of blocked session after two failed login attempts using incorrect credentials.

- e. From your console session on the router, issue the **show login** command to view the login status. In the example below, the **show login** command was issued within the 30 second login blocking period and shows that the router is in Quiet-Mode. The router will not accept any login attempts for 14 more seconds.

```
R1# show login
```

```
A default login delay of 1 second is applied.  
No Quiet-Mode access list has been configured.
```

```
Router enabled to watch for login Attacks.  
If more than 2 login failures occur in 120 seconds or less,  
logins will be disabled for 30 seconds.
```

```
Router presently in Quiet-Mode.  
Will remain in Quiet-Mode for 14 seconds.  
Denying logins from all sources.
```

R1#

- f. After the 30 seconds has expired, SSH to R1 again and login using the **admin** username and **Admin1p@55** for the password.

After you successfully logged in, what was displayed?

Experienced a bug during this section of the lab as packet tracer would not allow me to re-enter my correct password once the incorrect password had been blocked. Restarted packet tracer to continue the lab and the bug had gone.

When I tried again, the banner message set earlier in the lab was displayed as expected.

- g. Enter privileged EXEC mode and use **Enablep@55** for the password.

If you mistype this password, are you disconnected from your SSH session after two failed attempts within 120 seconds? Explain.

Once the password had been mistyped the session was disconnected. I had to wait 30 seconds before being able to re-enter my password due to the restrictions set earlier in the lab.

- h. Issue the **show running-config** command at the privileged EXEC prompt to view the security settings you have applied.

Part 4: Configure the Switch for SSH Access

In Part 3, you will configure the switch in the topology to accept SSH connections. After the switch has been configured, establish an SSH session using Tera Term.

Step 1: Configure the basic settings on the switch.

- Console into the switch and enable privileged EXEC mode.
- Enter configuration mode.
- Disable DNS lookup to prevent the router from attempting to translate incorrectly entered commands as though they were host names. (*no ip domain-lookup*)
- Assign **class** as the privileged EXEC encrypted password. (*enable secret class*)
- Assign **cisco** as the console password and enable login.
- Assign **cisco** as the VTY password and enable login.
- Encrypt the plain text passwords.
- Create a banner that will warn anyone accessing the device that unauthorized access is prohibited.
- Configure and activate the VLAN 1 interface on the switch according to the Addressing Table.
 - interface vlan 1*
 - ip address 192.168.1.11 255.255.255.0*
- Save the running configuration to the startup configuration file.

1) *copy running-config startup-config*

Step 2: Configure the switch for SSH connectivity.

Use the same commands that you used to configure SSH on the router in Part 2 to configure SSH for the switch.

- Configure the device name as listed in the Addressing Table.
- Configure the domain for the device.

```
S1(config)# ip domain-name ccna-lab.com
```

- Configure the encryption key method.

```
S1(config)# crypto key generate rsa modulus 1024
```

Incorrect command above, corrected to the following command below.

```
S1(config)# crypto key generate rsa general-keys modulus 1024
```

- Configure a local database username.

```
S1(config)# username admin privilege 15 secret adminpass
```

- Enable Telnet and SSH on the VTY lines.

```
S1(config)# line vty 0 15
```

```
S1(config-line)# transport input ssh - enables SSH ONLY (not Telnet)
```

- Change the login method to use the local database for user verification.

```
S1(config-line)# login local
```

```
S1(config-line)# end
```

Step 3: Establish an SSH connection to the switch.

Start Tera Term from PC-A, and then SSH to the SVI interface on S1.

Are you able to establish an SSH session with the switch?

Initially, I was unable to SSH to the switch. After using the show running-config command I was able to troubleshoot the issue, which was that the port was shut. After running the no shutdown command and retrying to SSH, it was successful and displayed the banner I had previously set.

Part 5: SSH From the CLI on the Switch

The SSH client is built into the Cisco IOS and can be run from the CLI. In Part 4, you will SSH to the router from the CLI on the switch.

Step 1: View the parameters available for the Cisco IOS SSH client.

Use the question mark (?) to display the parameter options available with the **ssh** command.

```
S1# ssh ?
  -c      Select encryption algorithm
  -l      Log in using this user name
  -m      Select HMAC algorithm
  -o      Specify options
```

```
-p      Connect to this port
-v      Specify SSH Protocol Version
-vrf    Specify vrf name
WORD    IP address or hostname of a remote system
```

Step 2: SSH to R1 from S1.

- a. You must use the **-l admin** option when you SSH to R1. This allows you to log in as user **admin**. When prompted, enter **adminpass - (Changed now to "Admin1p@55" to meet password criteria)** for the password.

```
S1# ssh -l admin 192.168.1.1
Password:
*****
Warning: Unauthorized Access is Prohibited!
*****
```

```
R1#
```

- b. You can return to S1 without closing the SSH session to R1 by pressing **Ctrl+Shift+6**. Release the **Ctrl+Shift+6** keys and press **x**. The switch privileged EXEC prompt displays.

```
R1#
S1#
```

- c. To return to the SSH session on R1, press Enter on a blank CLI line. You may need to press Enter a second time to see the router CLI prompt.

```
S1#
[Resuming connection 1 to 192.168.1.1 ... ]
```

```
R1#
```

- d. To end the SSH session on R1, type **exit** at the router prompt.

```
R1# exit

[Connection to 192.168.1.1 closed by foreign host]
S1#
```

What versions of SSH are supported from the CLI?

The switch supports SSH Protocol V1 and V2 as shown in the screenshot below, produced by running the **ssh -v** command on the switch.

```
S1#ssh -v ?
 1 Protocol Version 1
 2 Protocol Version 2
.. " " ..
```

Figure 4. Screenshot of SSH version from CLI on Switch 1

How would you provide multiple users, each with their own username, access to a network device?

In order to provide multiple users access to a device with their own username and password, the following command can be run “**username {username} privilege {level} secret {password}**”.

Lab Reflection:

During this lab, I encountered a few commands that did not work in packet tracer, such as the crypto key generate command. However, I was able to use the '?' help in order to find the correct command.

I also encountered a bug during the lab when testing my password timeouts. The bug caused me to be locked out of the router and would not accept my correct credentials. To overcome this I restarted packet tracer and was able to continue.

Overall, I learned many things in this lab, including how to set basic password requirements, and how to SSH into a router.

Report (max 800 words):

Write a research report comparing various Encryption standards, e.g AES, SHA, DES, RSA.

Please see report attached below titled 'Comparison of Common Encryption Standards'.



COMPARISON OF COMMON ENCRYPTION STANDARDS

University of Wales Trinity Saint David

BSc Computing Portfolio
Computer Architecture and Networks | ACCD4001

Siobhan Kelly Stephens
1800709

1. Introduction

Encryption is the process of taking plain text and turning it into an unrecognisable and unreadable format [1] for the purpose of providing confidentiality, integrity, non-repudiation and authentication of data [2]. Data Integrity and Confidentiality make up two sides of the CIA triad, which defines the basic aims of information security [3].

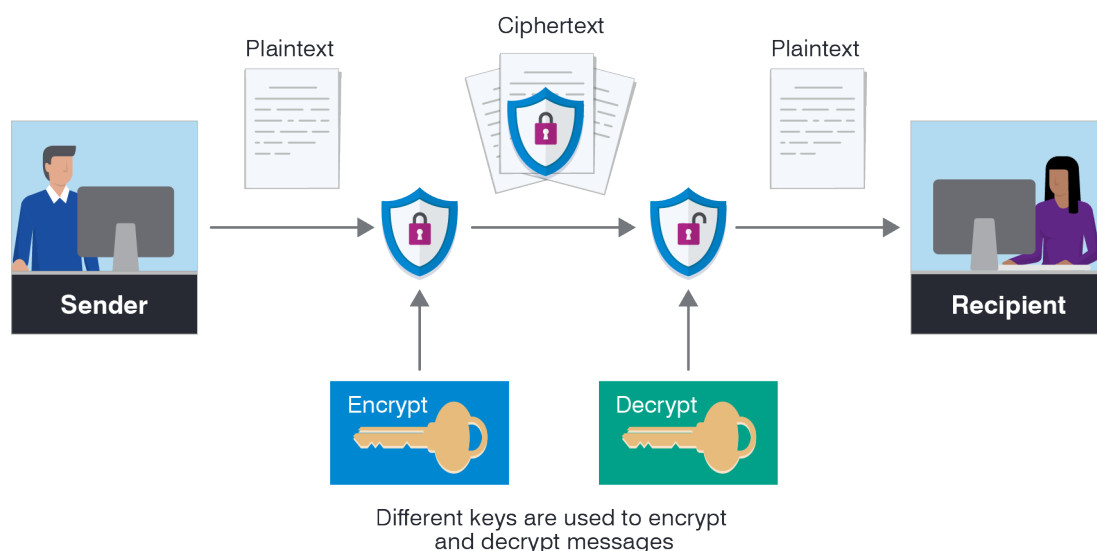


FIGURE 1. ENCRYPTION AND DECRYPTION [4].

A hashing algorithm is the process of applying a mathematical function to plain text and transforming it into a fixed-size unique hash value. As a result of this process, data is obscured. Hashing algorithms are irreversible, and therefore can be referred to as one-way encryption [5].

Encryption is highly important in modern technology and therefore different techniques and methods have been used to protect sensitive data. Organisations face fines of up to €20mil or 4% of turnover, if they do not have adequate security on data they hold [6].

2. Encryption Standards

There are many modern encryption standards that all vary in complexity and system requirements. Encryption can fall into one of two categories, asymmetric or symmetric (See Appendix A.1).

2.1 Advanced Encryption Standard (AES)

The AES algorithm was published by National Institute of Standards and technology (NIST) in 2000. AES was introduced to replace the DES algorithm and overcome many of its vulnerabilities. The AES algorithm is symmetric and takes a fixed 128-bit block and then uses either a 128-bit, 192-bit or 256-bit key [7].

The number of transformation rounds that are applied to the plain text is defined by the key size specified for the AES cipher. 128-bit keys undergo 10 rounds, 12 rounds for 192-bit and finally 14 rounds for 256-bit keys. Using the same encryption key, a set of reverse rounds are applied to transform the ciphertext into plaintext [7].

2.2 Secure Hashing Algorithm (SHA)

SHA was developed by the US National Security Agency (NSA) [5]. There are different versions of the Secure Hashing Algorithm such as SHA-1 and variations of SHA-2. SHA-2 is the successor to SHA-1 and contains many improvements over SHA-1. From 2016 onwards, SSL/TLS certificates must use SHA-256 as a minimum [8].

SHA-1 was first published in 1995 and uses a 160-bit hash, whereas SHA-2 comes in various bit-lengths, with 256-bit being the most common (known as SHA-256). A larger bit size provides more security as there are more possible combinations, which is important in order to prevent collisions, which are dangerous to security as they allow two files to produce the same hash – in the case of certificates, this could mean a certificate appears to be valid when it was never signed [8].

SHA has many uses along with certificates such as obscuring passwords in databases and identifying the tampering of files - due to the major difference in hash values as a result of an ‘avalanche’ effect [9].

2.3 Data Encryption Standard (DES) and Triple DES

DES was developed in 1970 at IBM by Horst Feistel [10]. It is a symmetric encryption that consists of a number of rounds, where each round contains bit-shuffling, non-linear substitutions and exclusive OR operations [10]. It expects a 64-bit input plus a 56-bit key to produce a 64-bit cipher/plaintext [11].

By 1997, DES was cracked by Deschall proving it to be insecure [12], and therefore a more secure encryption was required. In response, Triple DES or 3DES was introduced in 1998. 3DES applies the DES algorithm three times to each block by encrypting with a first key, decrypting with a second key, and encrypting with a third key. 3DES is embedded into payment systems, standards and technology in the finance industry, however in November 2017, NIST restricted the usage of 3DES so that it could no longer be used for TLS, IPsec or large file encryption [13].

2.4 RSA

RSA, or Rivest, Shamir, and Adelman, is an asymmetric public-key encryption algorithm based on prime factorization, developed in 1978. RSA is the standard encryption method for transmitting important data over the internet, and is built into many products such as Microsoft’s Internet Explorer [14].

RSA produces a variable block size cipher and works by finding the product of two large prime numbers and applying additional mathematical operations to create both the public and private keys [15].

RSA is used for SSL/TLS, S/MIME and Secure Shell encryption and digital signature functions [15].

3. Comparison of Encryption Standards

Figure 2 below outlines the main characteristics of the RSA, DES, 3DES and AES encryption algorithms. The SHA hashing algorithm will not be considered in this section.

Comparison of RSA, DES, 3DES and AES Encryption Algorithms				
	RSA	DES	3DES	AES
Creators	Rivest, Shamir, and Adelman in 1978	IBM in 1975	IBM in 1978	Vincent Rijem, John Daemon in 2001
Key Length	Depends on no of bits in the modulus n where $n = p * q$ (1024/2048 most common)	56 bits	168 bits	128, 192 or 256 bits
Round(s)	1	16	48	10 for 192-bit keys, 12 for 192-bit keys, 14 for 256bit keys
Block Size	Variable	64 bits	64 bits	128 bits
Cipher Type	Asymmetric Block Cipher	Symmetric Block Cipher	Symmetric Block Cipher	Symmetric Block Cipher
Speed of Algorithm (relative)	Slowest	Slow	Very Slow	Fast
Security (relative)	Least secure	Insecure	Adequate	Excellent
Storage Requirements	Large, due to the greater key size required for security	Minimal storage requirments	Minimal storage requirments	Minimal storage requirments
Resistant to Quantum Computing	No	No	No	Likely

FIGURE 2. COMPARISON OF ENCRYPTION STANDARDS FROM MULTIPLE SOURCES [16][17].

As shown above, AES provides the highest level of security, and is the only algorithm likely to remain secure against quantum computing. RSA has not yet been cracked, however would likely be easily brute-forced using quantum computing [16].

Unlike the other algorithms, RSA requires large storage requirements due to its greater key size required to maintain a level of security, and Parmar et al [16] stated that “to achieve the same level of security provided by 2048-bit RSA, AES requires only a 112-bit key”.

AES is the fastest algorithm for encryption and decryption, with RSA being the slowest due to the key generation time [16]. 3DES is more slow than DES due to it applying the same algorithm three times. AES is faster than DES and 3DES due to its greater block size, meaning it has to iterate the encryption less times [10].

4. Conclusion

Based on the above comparisons, AES offers the best memory and storage efficiency along with performance as an encryption algorithm, plus the highest security and is likely to be future proof against quantum computing.

However, if authentication is required, then RSA should be used. It does offer a reasonable level of security but requires greater key generation time [16] and will likely need to be updated in the coming years.

If only data obscuring is required, SHA-256 should be used as it is currently the most supported hashing algorithm which balances security, hash collisions and performance.

It must be noted that despite strong encryption on data, physical hardware must be adequately protected in order to keep data secure.

5. References

- [1] Kim Porter and Norton, "What is encryption and how does it protect your data?" [Online]. Available: <https://us.norton.com/internetsecurity-privacy-what-is-encryption.html>. [Accessed: 05-Dec-2019].
- [2] A. Muhammad and R. Hikmat, "New Approaches to Encrypt and Decrypt Data in Image using Cryptography and Steganography Algorithm," *Int. J. Comput. Appl.*, vol. 143, no. 4, pp. 11–17, Jun. 2016.
- [3] M. Cabric, "Confidentiality, Integrity, and Availability," in *Corporate Security Management*, Elsevier, 2015, pp. 185–200.
- [4] Proofpoint, "Encryption - What It Is, Types, Algorithms, & More." [Online]. Available: <https://www.proofpoint.com/us/glossary/encryption>. [Accessed: 05-Dec-2019].
- [5] R. Chaves, L. Sousa, N. Sklavos, G. Kalogerido, P. Kitsos, and F. Sheik, "Secure Hashing: SHA-1, SHA-2, and SHA-3," 2016.
- [6] "Fines / Penalties | General Data Protection Regulation (GDPR)." [Online]. Available: <https://gdpr-info.eu/issues/fines-penalties/>. [Accessed: 05-Dec-2019].
- [7] A. Muhammad Abdullah and A. Muhamad Abdullah, "Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data View project Application of Petri Nets in Computer Networks View project Advanced Encryption Standard (AES) A," 2017.
- [8] P. Nohe, "The Difference Between SHA-1, SHA-2 and SHA-256 Hash Algorithms." [Online]. Available: <https://www.thesslstore.com/blog/difference-sha-1-sha-2-sha-256-hash-algorithms/>. [Accessed: 05-Dec-2019].
- [9] P. Witoolkollachit, "The avalanche effect of various hash functions between encrypted raw images versus non-encrypted images: A comparison study," 2016.
- [10] M. I. Ali, Z. Ahmed, M. Tariq Mahmood, and S. Rahman, "AES, DES, Triple DES Encryption Efficiency Analysis."
- [11] "Difference Between DES and AES (with Comparison Chart) - Tech Differences," 2016. [Online]. Available: <https://techdifferences.com/difference-between-des-and-aes.html>. [Accessed: 06-Dec-2019].
- [12] "Brute Force Cracking the Data Encryption Standard | RSA Conference." [Online]. Available: <https://www.rsaconference.com/industry-topics/blog/brute-force-cracking-the-data-encryption-standard>. [Accessed: 06-Dec-2019].
- [13] J. Henry, "3DES is Officially Being Retired." [Online]. Available: <https://www.cryptomathic.com/news-events/blog/3des-is-officially-being-retired>. [Accessed: 06-Dec-2019].
- [14] "What is RSA Encryption?" [Online]. Available: <https://www.techopedia.com/definition/21852/rsa-encryption>. [Accessed: 06-Dec-2019].
- [15] M. Rouse, "RSA algorithm (Rivest-Shamir-Adleman)," 2018. [Online]. Available: <https://searchsecurity.techtarget.com/definition/RSA>. [Accessed: 05-Dec-2019].
- [16] N. J. Parmar and P. K. Verma, "A comparative evaluation of algorithms in the implementation of an ultra-secure router-to-router key exchange system," *Secur. Commun. Networks*, vol. 2017, 2017.
- [17] M. B. Yassein, S. Aljawarneh, E. Qawasmeh, W. Mardini, and Y. Khamayseh, "Comprehensive study of symmetric key and asymmetric key encryption algorithms," in *Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017*, 2018, vol. 2018-Janua, pp. 1–7.
- [18] Phillip A Laplante, *Encyclopedia of Computer Science and Technology Volume II*, II. CRC Press Taylor & Francis Group, 2017.

6. Appendix A

Appendix A.1

Symmetric encryption uses a single key for both encryption and decryption and is much faster than asymmetric, whereas asymmetric encryption requires two separate keys, a public key and a private key known together as a key pair, for encryption and decryption [18].