2021

CtrlS Datacenters Ltd / Cloud4C Services Pvt Ltd

Cryptographic Implementation and Key Management Policy

Cryptographic Key Management Policy

Purpose

This policy establishes requirements for cryptographic key management.

Scope

This policy applies to all Cloud4C cryptographic controls that are used to protect Cloud4C Company Confidential Information (CCI), employee personal information, and the employees / third parties responsible for the cryptographic controls.

Policy

Cryptographic keys must be properly maintained to ensure the integrity of cryptographic controls.

1 Document Revision History

Revision	Date	Changes
1.0	14/01/2019	New Document
1.1	21/02/2019	Updated the Policy with Crypto Details and Cloud4C strategy to address delinquent ISM Controls.

1.1 Change History

This policy is a living document that will be reviewed and updated annually, or more often if the need arises, based on changes in technology, applications, procedures, business needs, or threats.

Version:	Changed By:	Change Date:
1.0	Vineet Bulbule	12/01/2019
1.1	Vineet Bulbule	21/02/2019
1.1	Ashrith Karru	01/01/2021

1.2 Approved By

Approved By:	Approval Date:
Binu Chacko - CISO	14/01/2019
Binu Chacko – CISO	22/02/2019
Sreeram Chilakamarri – AVP - SOC	02/01/2021

2 Key Management

Secure methods for key management shall be in place to support the integrity of cryptographic controls. Cryptographic keys shall be protected against modification, unauthorized disclosure and destruction.

2.1 Generating Cryptographic Keys

Information Custodians shall develop and follow secure procedures for:

- Generating keys for different cryptographic systems and applications
- Generating and obtaining public key certificates

2.2 Protection of Cryptographic Keys

Information Custodians shall ensure the protection of cryptographic keys entrusted to them as detailed in the Cryptographic Key Management Standards document

2.3 Changing and Revoking Cryptographic Keys

Information Custodians must follow documented standards for changing and revoking cryptographic keys.

2.4 Key Archive

Cryptographic keys that are no longer in active use shall be securely archived. Key access shall require dual-party authentication and all access controls shall be in place to ensure protection of archived keys.

2.5 Key Recovery

Information Custodians shall ensure that the capability to recover encrypted information exists.

This recovery capability shall be in place prior to authorizing the encryption of any Cloud4C information.

2.6 Key Destruction

Cryptographic keys shall be destroyed in accordance with specified destruction method procedures.

3 Approved Cryptography Techniques

Sensitive and restricted data shall be encrypted with approved cryptographic techniques when transmitted over an unsecured path, as appropriate and feasible.

Cloud4C's CIO Committee made a decision to implement Cryptographic Controls of the products independently by their respective OEM technical features.

Cloud4C's CIO Committee also made a decision to only use products that have approved AACA Cryptographic Ciphers deployed for our Infrastructure and for our customers.

As such below are the major points:
Cloud4C has deployed:

- Bitlocker Mobile Workstation Disk Encryption
- SAP HANA Native Encryption
- Azure Blob Storage Encryption
- Fortinet SecureVPN IPSec Encryption
- CommVault Backup Encryption

The above solutions barring SAP HANA implemented AES-256 Bit SHA-2 Cipher Encryption that is approved for use by ASD and AACA Guidelines.

3.1 Key Management of the Solution

Cloud4C made a conscious decision to let the built-in Key Management Features of the above solutions manage the Encryption Keys.

- The Bitlocker Mobile Workstation Encryption Key is managed by Trusted Product Module TPM 2.0
- SAP HANA has a native Encryption feature that is patented and the key is managed by SAP SSFS
- Azure Blob Storage Encryption keys are managed by Microsoft Azure.
- Fortinet IPSec tunnelling has been implemented with AES-256 Encryption and the key are managed by the Forticlient
- CommVault manages AES-256 bit ciphers natively.

3.2 Choice between AES and Elliptic Curve Cryptography (EC, ECDH, ECDSA)

Cloud4C made a conscious decision to opt for AES-256 bit ciphers for encryption which is approved by AACA. The option to use 3DES, DSA and Other Digital Signatures for implementation to reduce the technical complexity and standardize the Cryptographic Implementation firm wide and for its customers.

- AACA and NSA's FIPs guidelines suggest that the use of AES-256 is equally effective in implementing non-ECC based Cryptographic Solutions.
- As such, ISM Controls relevant to ECC were rendered NOT APPLICABLE as Cloud4C opted for usage of AES-256 Encryption Ciphers as an effective alternate mitigating control.
- AES has been adopted by the U.S. government and is now used worldwide. It supersedes the Data Encryption Standard (DES),[7] which was published in 1977. The algorithm described by AES is a symmetric-key algorithm, meaning the same key is used for both encrypting and decrypting the data.
- Cloud4C made a conscious choice to deploye AES-256 SHA-2 Encryption Ciphers to comply with AACA, ASD and ISM Guidelines.

3.3 Detailed Information on Advanced Encryption Standard and Elliptic Curve Cryptography.

More detailed information regarding AES, ECC can be found at the below knowledge bases:

- https://en.wikipedia.org/wiki/Advanced_Encryption_Standard
- https://crypto.stackexchange.com/questions/61248/aes-and-ecdh-key
- https://www.quora.com/Which-one-is-better-elliptic-curve-cryptography-or-RSA-algorithm-and-why

4 ASD’s ISM Cryptographic Controls in-depth analysis

4.1 Diffie-Hellman (DH):

Diffie–Hellman key exchange (DH) is a method of securely exchanging cryptographic keys over a public channel. DH is one of the earliest practical examples of public key exchange implemented within the field of cryptography. The Diffie–Hellman exchange by itself does not provide authentication of the communicating parties and is thus vulnerable to a man-in-the-middle attack. To avoid these vulnerabilities, it is recommended to use elliptic curve cryptography, for which no similar attack is known. Failing that, it is recommended that the order, of the Diffie–Hellman group should be at least 2048 bits. They estimate that the pre-computation required for a 2048-bit prime is 109 more difficult than for 1024-bit primes.

To setup 1024-bit and 2048-bit DH encryption, it would require very significant amount of resources and the resulting costs are prohibitive. Hence, Cloud4C decided to make a choice between Elliptic Curve Cryptography (ECC) - ECDH, ECDSA or Symmetric algorithms such as AES-256 bit.

4.2 Elliptic Key Cryptography (ECC):

Elliptic-curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. ECC requires smaller keys compared to non-EC cryptography (based on plain Galois fields) to provide equivalent security.[1]

Elliptic curves are applicable for key agreement, digital signatures, pseudo-random generators and other tasks. Indirectly, they can be used for encryption by combining the key agreement with a symmetric encryption scheme. They are also used in several integer factorization algorithms based on elliptic curves that have applications in cryptography, such as Lenstra elliptic-curve factorization.

4.3 Popular ECC-based cryptographic algorithms: ECDH & ECDSA

4.3.1 Elliptic-Curve Diffie-Hellman (ECDH):

Elliptic-curve Diffie–Hellman (ECDH) is an anonymous key agreement protocol that allows two parties, each having an elliptic-curve public–private key pair, to establish a shared secret over an insecure channel. This shared secret may be directly used as a key, or to derive another key. The key, or the derived key, can then be used to encrypt subsequent communications using a symmetric-key cipher. It is a variant of the Diffie–Hellman protocol using elliptic-curve cryptography.

4.3.2 Elliptic Curve Digital Signature Algorithm (ECDSA):

In cryptography, the Elliptic Curve Digital Signature Algorithm (ECDSA) offers a variant of the Digital Signature Algorithm (DSA) which uses elliptic curve cryptography.

4.3.3 Technical concerns surrounding ECDSA and ECDH:

The difficulty of properly implementing the standard, its slowness, and design flaws which reduce security in insufficiently defensive implementations of the Dual EC DRBG random number generator.

5 Comparison of RSA, AES & ECC Encryption Standards:

Abstract

This summary will compare the RSA, AES, and ECC encryption algorithms. RSA certificates are widely used within the industry but require a trusted key generation and distribution architecture. AES and ECC provide advantages in key length, processing requirements, and storage space, also maintaining an arbitrarily high level of security. This summary modifies each of the four algorithms for use within the self-contained router-to-router environment system and then compares them in terms of features offered, storage space and data transmission needed, encryption/decryption efficiency, and key generation requirements.

1. Introduction

With the rise of globalization, microelectronics, and the information age, the need for rapid, long-distance transmission of unconditionally secure information has never been greater. Whether dealing with military intelligence, corporate secrets shared between two (or more) company offices, remote control of vital national infrastructure components such as power and traffic control systems, or mechanical instructions transmitted to off-site medical devices for telesurgery, device updates, and health reports, there are many situations where the rapid, accurate, and secure transmission of information between two parties is a basic necessity. In extreme cases, alteration or even decryption of this information by unauthorized parties may result in damages of billions of dollars and the lives of others.

While unconditional security may be an unachievable goal, it may be realized to an arbitrarily high level via existing symmetric and asymmetric encryption systems. Currently, the most widely used form of global network communication between two distant parties relies on public key, asymmetric key cryptography such as RSA for transferring symmetric keys. Symmetric encryption systems then use these keys to encrypt the information being transferred.

Although presenting a viable and widely used solution to secure communication, allowing for message encryption and authentication, the security certificate system requires the presence of a trusted third party for the verification of the identity and legitimacy of certificate owners. The compromise of or loss of trust in such a third party, or the inability to contact the distribution network at need, may result in a large-scale breakdown of reliable and secure communications. Furthermore, the increasingly large RSA key length requirements of public certificates to guarantee secure communication may be a barrier to practical implementation on limited-resource devices.

This summary first examines the originally proposed discrete logarithm-based encryption system and then proposes and compares other more commonly used encryption systems which may be used in this entirely self-contained environment, including RSA, ECC, and AES based encryption.

2. Related Work: Discrete Logarithm

The encryption system initially proposed is a variant of the discrete logarithm problem. This problem states that for the equation if a user knows, computing is computationally trivial. If, however, only and are known (and), then there is no efficient algorithm to compute.

If the key transfer protocol is not completed successfully, whether due to data loss or due to malicious interference, it may be necessary to reinitialize the system via use of another preshared secret.

Storage requirements for this system involve a preshared secret of length. Although no minimum length is required for, for increased security, it should be assumed that is relatively large, at a minimum approaching the approximate length of itself should be a large prime, in order to deter brute force attacks. Processing time for this encryption system for both encryption and decryption is relatively trivial, involving multiple multiplication, exponentiation, and mod operations. As both endpoints share a common key, this system does not allow for external message authentication or differentiation between messages originating from Alice or Bob.

The most efficient attack currently used on the general case of the discrete logarithm problem is the number field sieve, arriving at a solution for a prime number in (this is approximately). The security provided may thus be directly compared to that of RSA, which also may be most efficiently defeated via the general number field sieve, although discrete logarithms offer slightly more protection for a given key size. A quantum system, once it exists, may use Shor's algorithm to solve this problem in polynomial time.

3. Alternative I: RSA

The RSA algorithm has the advantage of being one of the most widely used and studied encryption methods today and is extremely elegant, simple, and well-tested. As the default algorithm used by many SSL providers, as well as the basic public key encryption scheme most others are compared to, RSA is used here as a baseline for the comparison of other encryption methods, even though it is not as storage-efficient or processing-efficient as other algorithms studied and requires the use of longer key lengths for equivalent security. Current commonly used RSA key lengths include 1024 and 2048 bits.

Typically, as the sending party must know the recipient's public key, as well as their own private key, RSA is not used within a self-contained system. Key generation for large primes may also be time consuming and resource intensive. Instead, third-party organizations must exist and are trusted to verify that a given public key corresponds to the stated owner's private key. Issued certificates linking a public key and verification of its owner's identity are generally valid for a set length of time, after which a new key must be generated and a new certificate request verifying the key's owner must be submitted to the central verification authority.

Storage requirements for an -bit RSA system are comparatively large, as larger key lengths are needed to assure equivalent security. Specifically, each router using this -bit RSA algorithm will need to store 1 public and 1 private key, each consisting of an -bit modulus and a smaller exponent (also of maximum length about) for maximum total requirement of bits per router. Processing time for RSA is also comparatively long, due to the larger key lengths and exponentiation operations required. The security of RSA is based upon the difficulty of the factorization problem.

Although it is obvious that RSA offers several disadvantages when compared to other symmetric and asymmetric ciphers, it also offers at least one key advantage when compared to the other algorithms herein: message authentication. Unlike discrete logarithm, ECC, or AES encryption, it would be possible for a third-party external audit, given hardware access to both router keys and all traffic sent, to determine the sender of all encrypted data.

4. Alternative II: AES

AES, based upon the Rijndael cipher, was announced by the National Institute of Standards and Technology in 2001 and was shortly thereafter approved as an accepted encryption standard by the United States Federal Government. AES, similar to its predecessor, DES, is a symmetric block cipher, using a shared secret key to encrypt a data stream one block at a time. In AES, each 128-bit data block undergoes 10–14 rounds (depending on key length) of permutations, substitutions, and additions [1]. AES is an extensively used and studied algorithm and like most symmetric ciphers offers advantages in terms of required processing power, processing time, and key length when compared to asymmetric ciphers such as RSA and ECC. The simplicity of each round enables simple and rapid implementation on any 8-bit processor, while the chaining of multiple rounds per block provides excellent security. The AES algorithm itself is quite straightforward to implement within hardware, and hardware AES optimization is currently already present in many modern, commercially available processors, including current processors from Intel, AMD, and Qualcomm, making this an excellent algorithm choice for use with existing components.

As mentioned earlier, AES offers efficient processing time, and the storage requirements for this system are minimal, requiring a single preshared key to be saved on each of the two end routers, much shorter than a security-equivalent RSA key pair. No effective cryptanalytic attacks are currently known against AES, with the current best attacks only a few orders of magnitude above the worst-case brute force scenario and requiring infeasibly large amounts of storage space. Unlike asymmetric encryption algorithms, AES is likely resistant to attacks by theoretical future quantum computers. In the event of a communication failure due to data loss or malicious action, it maybe necessary to switch to a new preshared key and begin the process again.

5. Alternative III: ECC

Elliptic Curve Cryptography (ECC) is an asymmetric cryptographic system, which uses a variant of the discrete logarithm problem as applied to points in an elliptic curve group as the core of its security. Many consumers have recently begun adopting ECC as an alternative to RSA, due to its efficiency in both key size and processing requirements. Careful choice of the ECC curve is necessary to avoid potential security hazards.

In Elliptic Curve Cryptography, first a curve is chosen, with variables and coefficients restricted over either the finite field GF(2m) of the form or a prime curve over and modulo where variables and coefficients range from 0 to () of the form .

In the prime curve case, there are a limited number of nonnegative integer points between and ,) which satisfy any given elliptic curve values for and . Similarly, for the finite field case, there will be a limited number of integer values that lie on the curve for any given values of and .

These points are used to define a finite abelian group, with rules for addition defined specifically for the abelian group, similar to modular multiplication in conventional algorithms. Likewise, multiple additions are preformed similarly to modular exponentiation. Using abelian group rules, given two points and , is easily calculated given and but difficult to calculate given and , forming the one-way trapdoor function at the basis of elliptic cryptography.

Encryption and decryption function as standard ECC operations. After a data threshold is exceeded, choosing new secret integers, and encrypt and send each other their new public keys using their old private keys. Once both parties have received the new keys, all data will be transmitted using these. This system would allow for the use of ECC indefinitely, with rapid key updates, without the necessity of a third party. In the event of a communication failure due to data loss or malicious action, it may be necessary to switch to a new preshared certificate pair and begin the process again. Unlike in RSA, the use of a common secret key prevents message authentication via external audit.

Storage requirements for ECC involve two large integers of size or smaller, corresponding to the public and private keys, for a total maximum storage capacity of per shared secret per router. Key lengths used are much shorter than those needed for equivalent RSA or discrete logarithm security levels, about double the size of that found in symmetric encryption systems. Likewise, while not quite as processing-efficient as a symmetric cryptosystem, ECC offers large performance gains when compared to RSA. The best known attack to ECC is Pollard's Rho [14] which may be paralyzed and needs relatively little memory but is nevertheless not computationally feasible for currently used curve parameters. As with other public key protocols, ECC is expected to be vulnerable to attack by quantum computers, once such exist.

6. Algorithm Comparison

The RSA, ECC, AES, and discrete logarithm protocols may each provide an arbitrary level of security, determined by the length of the encryption keys used for each algorithm [8]. Figure 1 visually illustrates the required key length needed by various encryption algorithms in order to achieve a level of security comparable to a specified RSA key length (e.g., to achieve the same level of security provided by 2048-bit RSA encryption, AES requires only a 112-bit key). In the case of the discrete logarithm method, the equivalent key length of the prime used was determined using the general number field algorithm as compared to RSA key lengths and was found to be approximately equal in requirement with RSA key equivalent to a discrete log key 0.84

(less than one-bit difference). ECC and AES hold clear advantages here over RSA and discrete log methods, as key sizes for the latter two increase rapidly as increased security is needed, while the key length : security ratio remains relatively linear for ECC and AES. The longer key lengths of RSA and discrete log will also require additional bandwidth for public key transfer, compared to shorter ECC public keys, and no additional bandwidth overhead is required for AES.

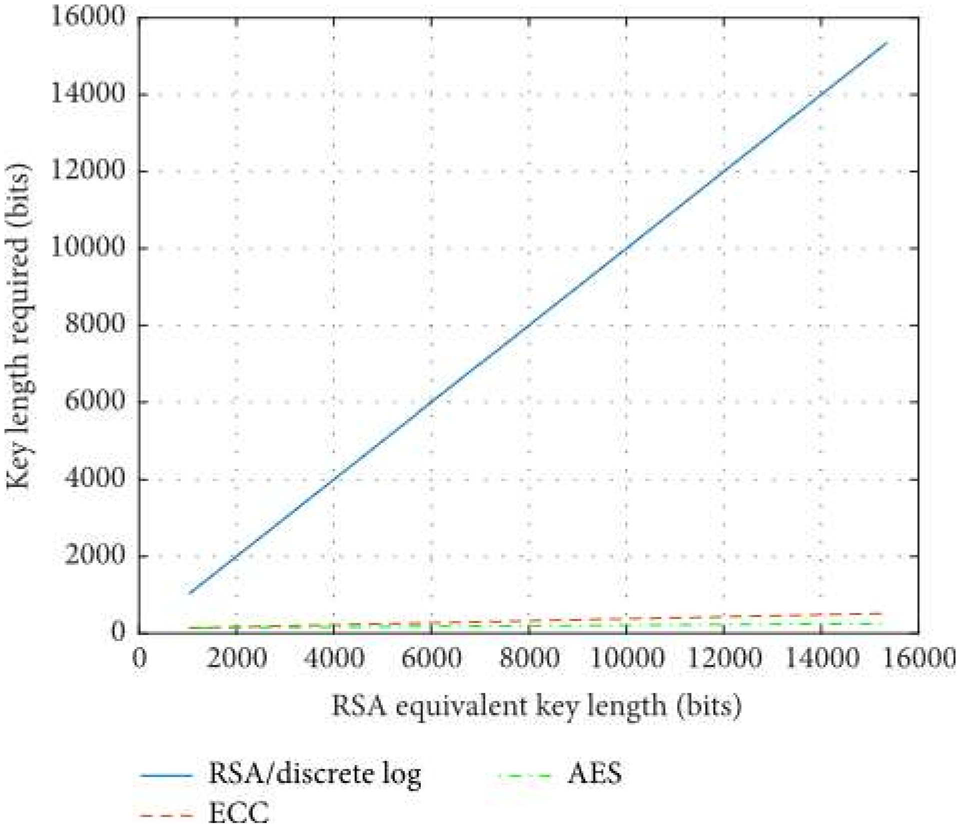

Figure 1: Key length versus security for AES, ECC, RSA, and discrete log. Data source: National Security Agency, Central Security Service.

Storage requirements for preshared secret data per router (ignoring overhead and indexing values), as outlined by the modified algorithms described earlier, are as follows:(1)-bit RSA requires a maximum of bits per secret.(2)-bit ECC requires a total of bits per secret.(3)-bit AES requires a single stored -bit key.(4)-bit discrete log method involves a preshared secret , assumed to be of maximum length .

Using these values, in combination with the key length requirements illustrated in Figure 1, it is possible to calculate the minimum storage requirements of each router for preshared secret data. For example, from Figure 1, we see that a 2048-bit RSA or discrete logarithm key is the equivalent of a 224-bit ECC key, or a 112-bit AES key. Each shared secret stored by the router at this security level would thus require a maximum of bits for RSA and 2048 bits for discrete log but only bits for ECC, or 112 bits for AES. Using these calculations, Figure 2 illustrates the total number of preshared secrets which may be stored per gigabyte of memory for any given security level and encryption algorithm (e.g., 8,000,000,000/8192 = 976,562 shared secrets per GB for 2048-bit RSA, or over 70 million shared secrets per GB for the equivalent 112-bit AES).

Encryption and decryption performance for the various algorithms are difficult to measure and are heavily influenced by system architecture and software/hardware optimizations. Generally, however, symmetric key ciphers such as AES will offer the fastest encryption and decryption times. ECC offers dramatically superior key pair generation performance compared to RSA, with the large primes generated for RSA requiring several orders of magnitude more time when compared to a much smaller ECC key, especially at RSA bit lengths of 2048 and above. In router systems with frequent key refreshes this could be a potential issue. Additionally, manufacturing hardware may struggle to fill even a modestly sized storage chip with unique preshared RSA keys (even a 1GB sized chip may be able to hold hundreds of thousands of preshared RSA certificates!), while even millions of shared symmetric encryption keys would simply involve filling the same chip pair with identical random data. RSA encryption is generally slightly faster than ECC, while ECC decryption may be several times faster than RSA, although both are generally efficient enough not to provide a practical system bottleneck. The discrete log method is assumed to offer a similar processing time as RSA due to similarities in algorithm implementation butwill likely take longer due to the multiple exchanges involved.

7. Conclusion and Future Work

Ultimately, algorithm choice will likely be determined by system needs and the availability of supporting hardware. Whatever algorithm is chosen, it will be necessary to provide preshared secret data to factory-paired communication devices, either built directly into each router pair or provided as paired insertable expansion chips with pregenerated shared encryption keys.

While discrete logarithm, RSA, ECC, and AES may each be used to provide the necessary nonlinearity for the establishment of a self-contained secure communication channel between two paired hardware devices, RSA and AES offer the most features and most efficient functionality, respectively. If authentication is needed, RSA, the weakest algorithm in terms of key generation and processing efficiency, is the clear choice. The use of RSA will, however, require a great deal of additional key generation time on the router manufacturing end. If, however, authentication is not needed, then symmetric key systems such as the AES exchange proposed offer the most efficient alternative and the only choice which offers more resistance to quantum computing attacks. AES hardware optimization is both extremely efficient and widely available in many currently used commercial processors, resulting in superior encryption, decryption, and processing times. AES key pair data, consisting effectively of a random bitstream, may be much more rapidly generated and preloaded onto devices than RSA, ECC, or discrete logarithm key pairs and provide greater security than equivalent-length asymmetric ciphers. Alternatively, a hybrid of both systems may be used, offering on-demand authentication when needed and efficient non-authenticated secure communication otherwise.

Cloud4C's decision making is derived scientifically based on the above details and descriptions and highlighting each encryption's advantages and disadvantages. It is clear that AES-256 bit encryption is far superior, secure, reliable and faster compared to both RSA & ECC based algorithms.

6 ASD's ISM Cryptographic Controls and Cloud4C's Mitigating Statements:

Control No.	Control Description	Cloud4C Risk Assessment
0472	When using DH for agreeing on encryption session keys, a modulus of at least 1024 bits, preferably 2048 bits, is used.	Cloud4C has a very effective Mitigating control in place and renders this control NOT APPLICABLE. As detailed and clearly summarized in the policy document, DH Algorithm is not amongst the latest and secure algorithms. As ASD suggests, DH should be used alteast with a 1024 bit modulus. This results in unnecessary overhead in terms of resources and costs and also adds to the latency. Use of a more secure and faster Algorithm is advised. Hence Cloud4C has deployed AES-256 bit encryption for storage, backup and IPSec VPN Tunnel. Kindly find the evidences and policy attached to review the same.

1446	When using elliptic curve cryptography, a curve from FIPS 186-4 is used.	Cloud4C has a very effective Mitigating control in place and renders this control NOT APPLICABLE. As detailed and clearly summarized in the policy document, ECC based Algorithm have certain advantages over DH and RSA. As ASD suggests, ECC should be used alteast with a 256 bit modulus. This results in unnecessary overhead in terms of resources and costs and also adds to the latency. An ECC- based 256 bit modulus is less secure than AES-256 which is comparable to an equivalent ECC based algorithm to be of 512-bit modulus atleast. Use of a more secure and faster Algorithm is advised. Hence Cloud4C has deployed AES-256 bit encryption for storage, backup and IPSec VPN Tunnel. Kindly find the evidences and policy attached to review the same.

0474	When using ECDH for agreeing on encryption session keys, a base point order and key size of at least 224 bits is used.	Cloud4C has a very effective Mitigating control in place and renders this control NOT APPLICABLE. As detailed and clearly summarized in the policy document, ECC based Algorithm have certain advantages over DH and RSA. As ASD suggests, ECC should be used alteast with a 224 bit modulus (ECDH). This results in unnecessary overhead in terms of resources and costs and also adds to the latency. An ECC-based 224 bit modulus is less secure than AES-256 which is comparable to an equivalent ECC based algorithm to be of 512-bit modulus atleast. Use of a more secure and faster Algorithm is advised. Hence Cloud4C has deployed AES-256 bit encryption for storage, backup and IPSec VPN Tunnel. Kindly find the evidences and policy attached to review the same.

0475	When using ECDSA for digital signatures, a base point order and key size of at least 224 bits is used.	Cloud4C has a very effective Mitigating control in place and renders this control NOT APPLICABLE. As detailed and clearly summarized in the policy document, ECC based Algorithm have certain advantages over DH and RSA. As ASD suggests, ECC should be used alteast with a 224 bit modulus (ECDSA). This results in unnecessary overhead in terms of resources and costs and also adds to the latency. An ECC-based 224 bit modulus is less secure than AES-256 which is comparable to an equivalent ECC based algorithm to be of 512-bit modulus atleast. Use of a more secure and faster Algorithm is advised. Hence Cloud4C has deployed AES-256 bit encryption for storage, backup and IPSec VPN Tunnel. Kindly find the evidences and policy attached to review the same.

0476	When using RSA for digital signatures, and passing encryption session keys or similar keys, a modulus of at least 1024 bits, preferably 2048 bits, is used.	Cloud4C has a very effective Mitigating control in place and renders this control NOT APPLICABLE. Digital Signatures are primarily aimed at Web Applications that are internet facing. Digital Signatures for the current IRAP Setup is not required, as there are no web applications deployed within that environment that are Internet facing. As detailed and clearly summarized in the policy document, RSA for Digital Signatures is not amongst the latest and secure algorithms. As ASD suggests, DH should be used alteast with a 1024 bit modulus. This results in unnecessary overhead in terms of resources and costs and also adds to the latency. Use of a more secure and faster Algorithm is advised. Hence Cloud4C has deployed AES-256 bit encryption for storage, backup and IPSec VPN Tunnel. Kindly find the evidences and policy attached to review the same.

0477	When using RSA for digital signatures, and for passing encryption session keys or similar keys, a key pair for passing encrypted session keys that is different from the key pair used for digital signatures is used.	Cloud4C has a very effective Mitigating control in place and renders this control NOT APPLICABLE. Digital Signatures are primarily aimed at Web Applications that are internet facing. Digital Signatures for the current IRAP Setup is not required, as there are no web applications deployed within that environment that are Internet facing. As detailed and clearly summarized in the policy document, RSA for Digital Signatures is not amongst the latest and secure algorithms. As ASD suggests, DH should be used alteast with a 1024 bit modulus. This results in unnecessary overhead in terms of resources and costs and also adds to the latency. Use of a more secure and faster Algorithm is advised. Hence Cloud4C has deployed AES-256 bit encryption for storage, backup and IPSec VPN Tunnel. Kindly find the evidences and policy attached to review the same.
1054	A hashing algorithm from the SHA-2 family is used instead of SHA-1.	Evidence has been attached that showcases that SHA-2 is implemented with AES-256 Encryption.
0479	Symmetric cryptographic algorithms are not used in Electronic Codebook Mode.	This control is assessed to be compliant / NA. Cloud4C has deployed AES-256 bit encryption in the Galcois Mode. The Fortinet Literature on IPSec VPN deployment clearly states that the Galcois mode is enabled instead of Electronic Codebook Mode.

0481	If using cryptographic equipment or software that implements an AACP, only AACAs can be used.	This control is assessed to be compliant. Cloud4C has deployed AES-256 bit encryption which is AACA approved.
455	Where practical, cryptographic equipment and encryption software provides a means of data recovery to allow for circumstances where the encryption key is unavailable due to loss, damage or failure.	Cloud4C has implemented a mitigating control by setting up Backup for the IRAP Protected Customer's datasources. The data can be recovered from the backup when encryption key is unavailable.
1369	AES in Galois Counter Mode is used for symmetric encryption when available.	This control is assessed to be compliant / NA. Cloud4C has deployed AES-256 bit encryption in the Galcois Mode. The Fortinet Literature on IPSec VPN deployment clearly states that the Galcois mode is enabled instead of Electronic Codebook Mode.

1372	DH or ECDH is used for key establishment.	Cloud4C has a very effective Mitigating control in place and renders this control NOT APPLICABLE. As detailed and clearly summarized in the policy document, DH or ECDH Algorithm is not amongst the latest and secure algorithms. As ASD suggests, DH should be used alteast with a 1024 bit modulus. This results in unnecessary overhead in terms of resources and costs and also adds to the latency. Use of a more secure and faster Algorithm is advised. Hence Cloud4C has deployed AES-256 bit encryption for storage, backup and IPSec VPN Tunnel. Kindly find the evidences and policy attached to review the same.

1448	When using DH or ECDH for key establishment, the ephemeral variant is used.	Cloud4C has a very effective Mitigating control in place and renders this control NOT APPLICABLE. As detailed and clearly summarized in the policy document, DH or ECDH Algorithm is not amongst the latest and secure algorithms. As ASD suggests, DH should be used alteast with a 1024 bit modulus. This results in unnecessary overhead in terms of resources and costs and also adds to the latency. Use of a more secure and faster Algorithm is advised. Hence Cloud4C has deployed AES-256 bit encryption for storage, backup and IPSec VPN Tunnel. Kindly find the evidences and policy attached to review the same.