Communications Security Establishment Canada
Centre de la sécurité des télécommunications Canada

CANADIAN CENTRE FOR CYBER SECURITY

Guidance on securely configuring network protocols

(Version 3)

Practitioner

Canada

Foreword

This is an UNCLASSIFIED publication issued by the Canadian Centre for Cyber Security (Cyber Centre) and provides an update to the previously published version.

We recommend that you also read Cryptographic Algorithms for UNCLASSIFIED, PROTECTED A, and PROTECTED B Information (ITSP.40.111) [1]. The configurations in this publication comply with the cryptographic requirements in ITSP.40.111.

Effective date

This publication takes effect January 2025.

Revision history

Revision	Amendments	Date
1	First release	August 2, 2016
2	Updated version (version 2)	October 13, 2020
3	Updated version (version 3)	January 2025

D97-3/40-062-2024E-PDF
978-0-660-70989-5

Overview

This publication identifies and describes acceptable security protocols, and their appropriate methods of use, that organizations can implement to protect sensitive information. For Government of Canada (GC) departments and agencies, the guidance in this publication applies to UNCLASSIFIED, PROTECTED A, and PROTECTED B information.

Your organization's ability to securely transmit sensitive data and information is fundamental to the delivery of your programs and services. Using cryptographic security protocols ensures the confidentiality, integrity, and availability of information and helps provide protection against certain cyber intrusion threats.

Data confidentiality, integrity, availability, stakeholder authentication and accountability, and non-repudiation are all benefits of properly configured security protocols. Various protocols may be required to satisfy your organization's specific security requirements, and each protocol should be selected and implemented to ensure all requirements are met.

For more information on securely configuring network protocols, contact our Contact Centre by email at contact@cyber.gc.ca or by phone at (613) 949-7048 or 1-833-CYBER-88.

Table of contents

List of figures

List of tables

1 Introduction

Organizations rely on information technology (IT) systems to achieve business objectives. These interconnected systems can be the targets of serious threats and cyber attacks that jeopardize the availability, the confidentiality, and the integrity of information assets. Compromised networks, systems, or information can have adverse effects on business activities and may result in data breaches and financial loss.

This publication provides guidance on the following topics:

- Securely configuring network protocols to protect sensitive information[a]
- Approved algorithms that the Cyber Centre recommends for use with these network protocols
- Standards and National Institute of Standards and Technology (NIST) special publications that provide additional information on these network protocols

This publication aids technology practitioners in choosing and using appropriate security protocols for protecting sensitive information (UNCLASSIFIED, PROTECTED A, and PROTECTED B information) and complements the Treasury Board of Canada Secretariat (TBS) Guideline on Defining Authentication Requirements [2]. This publication also provides cryptographic guidance for IT solutions at the UNCLASSIFIED, PROTECTED A, and PROTECTED B levels.[b] Organizations are responsible for determining their security objectives and requirements as part of their risk management framework.

1.1 IT security risk management process

When implementing security protocols, practitioners should consider the IT security risk management activities described in IT Security Risk Management: A Lifecycle Approach (ITSG-33) [3]. ITSG-33 addresses two levels of IT security risk management activities: departmental-level activities and information system-level activities. It also includes a catalogue of security controls (for example, standardized security requirements to protect the confidentiality, integrity, and availability of IT assets). See Figure 1: for an overview of the IT security risk management activity levels.

Additionally, organizations should consider the following activity areas: define, deploy, monitor, and assess. See Annex 1 of ITSG-33 [3] for more information on these activities.

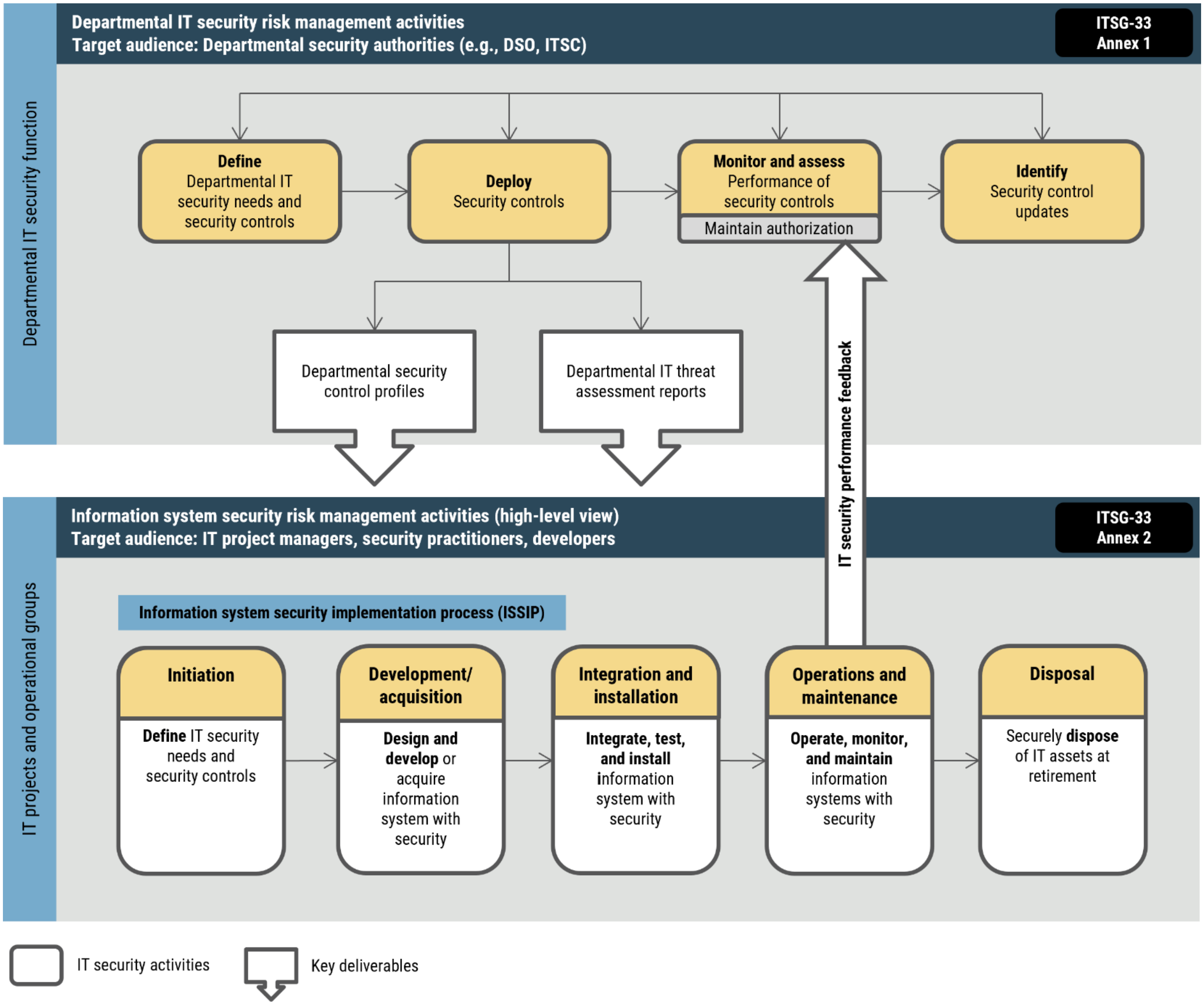
Departmental-level activities (or organizational-level activities for non-GC organizations) are included in departmental or organizational security programs to plan, manage, assess, and improve the management of IT security risks.

Information system-level activities are included in an information system's lifecycle through the information system security implementation process (ISSIP). When implementing network security protocols, you should consider all the steps in the ISSIP. See Annex 2 of ITSG-33 [3] for more information.

[a] For a GC department or agency, this guidance can be applied to UNCLASSIFIED, PROTECTED A, and PROTECTED B systems and information. Systems operating in PROTECTED C or classified domains may require additional design considerations that are not within the scope of this publication.

[b] Systems operating in PROTECTED C or classified domains may require additional design considerations that are not within the scope of this publication.

Figure 1: IT security risk management process


Departmental IT security function
Departmental IT security risk management activities
Target audience: Departmental security authorities (e.g., DSO, ITSC)
ITSG-33 Annex 1
Define Departmental IT security needs and security controls
Deploy Security controls
Monitor and assess Performance of security controls
Maintain authorization
Identify Security control updates
Departmental security control profiles
Departmental IT threat assessment reports
IT security performance feedback
IT projects and operational groups
Information system security risk management activities (high-level view)
Target audience: IT project managers, security practitioners, developers
ITSG-33 Annex 2
Information system security implementation process (ISSIP)
Initiation
Define IT security needs and security controls
Development/ acquisition
Design and develop or acquire information system with security
Integration and installation
Integrate, test, and install information system with security
Operations and maintenance
Operate, monitor, and maintain information systems with security
Disposal
Securely dispose of IT assets at retirement
IT security activities
Key deliverables

Long description: IT security risk management process

This figure describes the high-level departmental IT security risk management activities within the Departmental IT security function. The figure also describes the information system security risk management activities within IT projects and operational groups. It also highlights how the IT security risk management activities as both levels act together in a continuous cycle to efficiently maintain and improve the security posture of departmental information systems.

The departmental IT security risk management activities are targeted at an audience of departmental security authorities and encompass the following steps:

1. Define the business needs for security and the security controls
2. Deploy the security controls, including developing department security control profiles and department IT threat assessment reports
3. Continuously monitor and assess the performance of the security controls and maintain authorization

4. Identify security control updates

The information security risk management activities are aimed at IT project managers, security practitioners and developers. They encompass the following steps:

1. Initiation: Define IT security needs and security controls
2. Development/acquisition: Design and develop or acquire information system with security
3. Integration and installation: Integrate, test, and install information system with security
4. Operation and maintenance: Operate, monitor, and maintain
5. Disposal: Dispose of IT assets securely at retirement

1.2 Recommendations

Throughout this publication, we make recommendations that fall within three categories: recommended, sufficient, and phase out.

1.2.1 Recommended

Configurations listed in the Recommended column have advantages over those in the Sufficient column. Recommended configurations should always be implemented if allowed by the remote connection profile.

1.2.2 Sufficient

Configurations listed in the Sufficient column are appropriate to be used as deemed necessary to support the profile of remote connections. Sufficient configurations should be applied when it is not possible to implement a Recommended profile.

1.2.3 Phase out

Configurations listed in the Phase Out column are marked for transition according to guidance in ITSP.40.111 [1] or due to protocol-specific concerns.

If you have systems that use Phase Out selections, we recommend that you transition to Recommended or Sufficient alternatives as soon as possible.

Note: Systems do not need to be configured with all the selections listed in the recommended or sufficient columns. The chosen configurations will depend on an organization’s remote connection profile. The protocol selections should be implemented to limit the network attack surface.

2 Public Key Infrastructure

Public Key Infrastructures (PKIs) support the management of public keys for security services in PKI-enabled protocols, including Transport Layer Security (TLS), Internet Protocol Security (IPsec), and Secure/Multipurpose Internet Mail Extensions (S/MIME).

PKI key management guidance is provided in NIST Special Publication (SP) 800-57 Part 3 Revision 1 - Recommendation for Key Management, Part 3: Application-Specific Key Management Guidance [4]. We recommend that you refer to section 2 of NIST SP 800-57 Part 3 Rev. 1 [4] for the guidance on installing and administering PKI.

Your implementations must not reuse public key pairs across multiple protocols within the PKI. For example, key pairs used in IKEv2 must not be reused for Secure Shell (SSH).

You should format public key certificates in the X.509 version 3 certificate format, as specified in Request for Comments (RFC) 5280 Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile [5].

To support algorithm and key size agility, protocol implementations should support multiple certificates with their associated private keys. Public key certificates used for signing, key agreement, or key encipherment should be distinguished by the key usage extension, asserting one of the following bit-valued flags:

- digitalSignature
- keyAgreement
- keyEncipherment

To satisfy the cryptographic guidance provided in Cryptographic Algorithms for UNCLASSIFIED, PROTECTED A, and PROTECTED B Information (ITSP.40.111) [1], SHA-1 should not be used to generate or verify public key certificate digital signatures.

3 Transport Layer Security

Transport Layer Security (TLS) is a protocol developed to protect the confidentiality, integrity, and availability of Internet communications between server and client applications.

As specified in RFC 8446 The Transport Layer Security (TLS) Protocol Version 1.3 [6], we **recommend** configuring TLS servers and clients to use TLS 1.3. Using TLS version 1.2, updated in RFC 8446 [6], is **sufficient** if it is required for wider compatibility, internal audit compliance, or threat monitoring systems. You should phase out versions of TLS older than 1.2 or any versions of Secure Sockets Layer (SSL).

Servers that use TLS to protect HTTP traffic (i.e. HTTPS) should support HTTP Strict Transport Security (HSTS), as specified in RFC 6797 HTTP Strict Transport Security (HSTS) [7].

An email server acting as a Message Transfer Agent (MTA) for Simple Mail Transfer Protocol (SMTP) should support the negotiation of TLS with other MTAs. SMTP traffic can be upgraded to TLS using STARTTLS, as specified in RFC 3207 SMTP Service Extension for Secure SMTP over Transport Layer Security [8]. To ensure the use of TLS for SMTP traffic, MTAs should either support RFC 8461 SMTP MTA Strict Transport Security (MTA-STS) [9] and be configured to use the "enforce" policy mode or support RFC 7672 SMTP Security via Opportunistic DNS Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) [10].

Note: These opportunistic encryption techniques are only supported on a hop-by-hop basis. End-to-end message protection is provided by S/MIME. For more information, see Section 7 - Secure/multi-purpose Internet mail extensions.

When TLS is used to protect the confidentiality or integrity of PROTECTED A or PROTECTED B information you should use X.509 version 3 certificates to mutually authenticate between the server and the client.

3.1 TLS cipher suites

If the server or the client is configured to support TLS version 1.3, then the server or the client should be configured to support only the cipher suites listed in Table 1: .

Table 1: Recommended cipher suites for TLS version 1.3

Recommended	Sufficient
TLS_AES_256_GCM_SHA384 TLS_AES_128_GCM_SHA256 TLS_AES_128_CCM_SHA256	TLS_AES_128_CCM_8_SHA256

If TLS 1.2 support is required, a TLS server or client should be configured to support only the TLS 1.2 cipher suites listed in Table 2: .

Table 2: Recommended cipher suites for TLS 1.2

Recommended	Sufficient	Phase out
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_ECDSA_WITH_AES_256_CCM TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_AES_128_CCM TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8 TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 TLS_DHE_DSS_WITH_AES_256_GCM_SHA384 TLS_DHE_RSA_WITH_AES_256_CCM TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 TLS_DHE_DSS_WITH_AES_128_GCM_SHA256 TLS_DHE_RSA_WITH_AES_128_CCM TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 TLS_RSA_WITH_AES_256_GCM_SHA384 TLS_RSA_WITH_AES_128_GCM_SHA256 TLS_RSA_WITH_AES_256_CBC_SHA256 TLS_RSA_WITH_AES_256_CBC_SHA TLS_RSA_WITH_AES_128_CBC_SHA256 TLS_RSA_WITH_AES_128_CBC_SHA TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA TLS_DHE_RSA_WITH_AES_256_CBC_SHA TLS_DHE_RSA_WITH_AES_128_CBC_SHA

TLS servers and clients may use any or all the listed cipher suites according to the deployment profile. However, if an Internet-facing deployment requires cipher suites listed in the Phase Out column, we recommend you transition away from these as soon as possible. Your internal enterprise or data centre deployments of TLS may continue to use cipher suites with RSA key transport if required for audit compliance or threat monitoring systems, but this guidance may change in the future.

Cipher suites do not specify a key size for the public key algorithm. TLS servers and clients should ensure that the server and client ephemeral key pairs that are used to establish the main secret[c] satisfy the key length requirements specified in ITSP.40.111 [1]. Table 3: lists the Supported Groups that conform to ITSP.40.111 [1].

[c] The "main secret" may be called the "master secret" in some TLS references

Table 3: TLS supported groups that conform to ITSP.40.111

Recommended	Sufficient	Phase out
secp256r1	ffdhe3072	secp224r1
secp384r1	ffdhe4096	sect233r1
secp521r1	ffdhe6144	sect233k1
	ffdhe8192	sect283k1
		sect283r1
		sect409k1
		sect409r1
		sect571k1
		sect571r1
		ffdhe2048

Table 4: lists the Signature Algorithms that comply with ITSP.40.111 [1].

Table 4: TLS signature algorithms that comply with ITSP.40.111

Recommended	Sufficient	Phase out
ecdsa_secp256r1_sha256	rsa_pkcs1_sha256	ecdsa_secp224r1_sha224
ecdsa_secp384r1_sha384	rsa_pkcs1_sha384	rsa_pkcs1_sha224
ecdsa_secp521r1_sha512	rsa_pkcs1_sha512	dsa_sha224
ed25519		dsa_sha256
ed448		dsa_sha384
rsa_pss_pss_sha256		dsa_sha512
rsa_pss_pss_sha384		
rsa_pss_pss_sha512		
rsa_pss_rsae_sha256		
rsa_pss_rsae_sha384		
rsa_pss_rsae_sha512		

3.2 TLS extensions

We recommend that TLS servers and clients support the extensions listed in Table 5: .

Table 5: Recommended TLS extensions

Extension name	Extension code point	References	Notes
Application-Layer Protocol Negotiation	application_layer_protocol_negotiation	RFC 7301 [11]	
Certificate Signature Algorithms	signature_algorithms_cert	RFC 8446 [6] sec 4.2.3	
Certificate Status Request	status_request	RFC 6066 [12] sec 8	

Cookie	cookie	RFC 8446 [6] sec 4.2.2	TLS 1.3 only
Encrypt-then-MAC	encrypt_then_mac	RFC 7366 [13]	TLS 1.2 only
Extended Main Secret	extended_main_secret / extended_master secret	RFC 7627 [14]	TLS 1.2 only
Key Share	key_share	RFC 8446 [6] sec 4.2.8	TLS 1.3 only
Multiple Certificate Status	status_request_v2	RFC 6961 [15]	TLS 1.2 only
Pre-Shared Key	pre_shared_key	RFC 8446 [6] sec 4.2.11	TLS 1.3 only
Pre-shared Key Exchange Modes	psk_key_exchange_modes	RFC 8446 [6] sec 4.2.9	TLS 1.3 only
Renegotiation Indication	renegotiation_info	RFC 5746 [16]	TLS 1.2 only
Server Name Indication	server_name	RFC 6066 [12] sec 3	
Signature Algorithms	signature_algorithms	RFC 8446 [6] sec 4.2.3	
Supported Groups	supported_groups	RFC 8422 [17] sec 5.1.1, RFC 7919 [18]	Renamed from "elliptic_curves"
Supported Point Formats	ec_point_formats	RFC 8422 [17] sec 5.1.2	TLS 1.2 only
Supported Versions	supported_versions	RFC 8446 [6] sec 4.2.1	TLS 1.3 only
Transparency Information	transparency_info	RFC 9162 [19] sec 6.5	
Trusted CA Indication	trusted_ca_keys	RFC 6066 [12] sec 6	TLS 1.2 only

Note: Do not enable extensions in your configurations that are not listed above.

We recommend that TLS servers and clients Phase Out the extensions listed in Table 6: .

Table 6: TLS extensions to phase out

Extension name	Extension code point	References	Rationale
Signed Certificate Timestamp	signed_certificate_timestamp	RFC 6962 [20]	Made obsolete by "transparency_info" extension.

3.3 Client and server authentication

The client must validate the server certificate according to RFCs 5280 [5] and 8446 [6]. The revocation status of the certificate must be checked using a certificate revocation list (CRL) or the Online Certificate Status Protocol (OCSP)and the client should verify that the certificate appears in a certificate transparency log according to RFC 9162 Certificate Transparency Info Version 2.0 [19]. The client must check that the certificate contains a value in the Subject Alternative Name extension or in the Subject Distinguished Name field that matches the DNS or IP address requested.

If the client included the certificate signature algorithms extension, the client should verify that the certificate signature algorithm matches one of the proposed values. Otherwise, the client should verify that the certificate signature algorithm matches one of the proposed values in the signature algorithms extension.

Finally, the client should verify the public key length in the certificate satisfies the key length requirements specified in ITSP.40.111 [1].

If client authentication (also referred to as mutual authentication) is configured, the server must validate the client certificate according to RFCs 5280 [5] and 8446 [6]. The server must verify that the certificate validation path chains to a certificate authority (CA) that is trusted by the server to validate access to the requested resource. The revocation status of the certificate must be checked using a CRL or the OCSP. The server should check that the certificate contains a value in the Subject Alternative Name extension or in the Subject Distinguished Name field that matches an authorized client.

Finally, the server should verify that the public key length in the certificate satisfies the key length requirements specified in ITSP.40.111 [1].

3.4 Other TLS configuration guidelines

TLS clients and servers must be configured to disable TLS compression, which is done by negotiating the null compression method.

Due to the complication of mitigating replay attacks, we recommend that configurations do not support the 0-RTT mode of TLS version 1.3.

TLS 1.2 renegotiation without the Renegotiation Indication extension (see RFC 5746 Transport Layer Security [TLS] Renegotiation Indication Extension [16]) must be disabled. Furthermore, we recommend that TLS servers are configured to not accept client-initiated renegotiation at all in favour of establishing a new TLS connection.

If support for session resumption is desired, we recommend that you use the session identifier method in TLS 1.2 or session resumption via pre-shared keys (PSKs) in TLS 1.3. You should use PSKs with an ECDHE/DHE key exchange to provide forward secrecy.

4 Internet Protocol Security

You can use a combination of the protocol pair Internet Key Exchange Protocol Version 2 (IKEv2) and Internet protocol security (IPsec) to create secure data tunneling at the network layer. The IKEv2 protocol establishes secure key material that can be used in the IPsec protocol to secure the data that is exchanged between peers.

4.1 Internet key exchange protocol version 2

Internet key exchange protocol version 2 (IKEv2) is specified in RFC 7296 Internet Key Exchange Protocol Version 2 (IKEv2) [21].

Note: IKEv1 should no longer be used.

4.1.1 Authentication

When IKEv2 is used to set up an IPsec security association (SA) to protect the confidentiality of PROTECTED A or PROTECTED B information or the integrity of UNCLASSIFIED, PROTECTED A, or PROTECTED B information, digital signatures should be used for authentication. Pre-shared keys should not be used for authentication.

Table 7: lists the authentication schemes that comply with ITSP.40.111 [1].

Table 7: Recommended IKEv2 authentication schemes

Recommended	Sufficient	Phase out
ECDSA with SHA-256 on the P-256 curve ECDSA with SHA-384 on the P-384 curve ECDSA with SHA-512 on the P-521 curve Ed25519 with the identity hash ED448 with the identity hash RSASSA-PSS with bit length 3072 and SHA-384 RSASSA-PSS with bit length 4096 and SHA-384	RSASSA-PKCS1-v1.5 with bit length 3072 and SHA-384 RSASSA-PKCS1-v1.5 with bit length 4096 and SHA-384	RSASSA-PSS with bit length 2048 and SHA-256 RSASSA-PKCS1-v1.5 with bit length 2048 and SHA-256

4.1.2 Message encryption

Table 8: lists the IKEv2 message encryption algorithms that comply with ITSP.40.111 [1] when used with a key length of 128, 192, or 256 bits.

Table 8: Recommended IKEv2 Message Encryption Algorithms

Recommended	Sufficient	Phase out
ENCR_AES_GCM_16 ENCR_AES_CCM_16	ENCR_AES_GCM_12 ENCR_AES_CCM_12 ENCR_AES_CBC ENCR_AES_CTR	ENCR_3DES ENCR_CAST

We recommend using Advanced Encryption Standard (AES) in Galois/Counter Mode (GCM) to encrypt IKEv2 messages. If GCM or CCM is not supported, use an integrity protection mechanism from subsection 4.1.5.

4.1.3 Key exchange

Table 9: lists the IKEv2 key exchange groups that comply with ITSP.40.111 [1].

Table 9: Recommended IKEv2 Key Exchange Groups

Recommended	Sufficient	Phase out
256-bit Random ECP Group 384-bit Random ECP Group 521-bit Random ECP Group	3072-bit MODP Group 4096-bit MODP Group 6144-bit MODP Group 8192-bit MODP Group	2048-bit MODP Group 2048-bit MODP Group with 224-bit Prime Order Subgroup 2048-bit MODP Group with 256-bit Prime Order Subgroup 224-bit Random ECP Group

Implementations must check that received public values are between 1 and p-1 and, in the case of Elliptic-Curve Diffie-Hellman (ECDH), satisfy the elliptic curve equation.

We recommend that every key exchange uses a freshly generated ephemeral ECDH/DH key pair.

4.1.4 Pseudo-random functions for key generation

IKEv2 uses a pseudo-random function (PRF) to generate key material. Table 10: lists PRFs that comply with ITSP.40.111 [1].

Table 10: Sufficient PRF for IKEv2 key generation

Sufficient
PRF_HMAC_SHA2_256 PRF_HMAC_SHA2_384 PRF_HMAC_SHA2_512 PRF_AES128_CMAC

4.1.5 Integrity protection

When not using an authenticated encryption (AEAD) algorithm (such as AES GCM) for message encryption, an additional integrity protection mechanism is required. Table 11: lists the integrity protection mechanisms that comply with ITSP.40.111 [1].

Table 11: Sufficient and phase out integrity protection mechanisms for IKEv2

Sufficient	Phase out
AUTH_HMAC_SHA2_256_128	AUTH_HMAC_SHA1_160
AUTH_HMAC_SHA2_384_192	
AUTH_HMAC_SHA2_512_256	
AUTH_AES_128_GMAC	
AUTH_AES_192_GMAC	
AUTH_AES_256_GMAC	
AUTH_AES_CMAC_96	

4.1.6 Extensible Authentication Protocol

RFC 7396 JSON Merge Patch [22] specifies that Extensible Authentication Protocol (EAP) in IKEv2 can be used if it is used with the IKEv2 responder public key-based authentication. RFC 5998 An Extension for EAP-Only Authentication in IKEv2 [23] lists the methods that can be used in IKEv2 to provide mutual authentication and that do not require responder public key-based authentication.

While many authentication methods are listed as safe EAP methods in RFC 5998 [23], we recommend that you use methods that support channel binding. We also recommend that you maintain the use of responder public key-based authentication.

4.1.7 Distributed denial-of-service protection

IKEv2 is prone to DDoS attacks. In a DDoS attack, a threat actor overwhelms a responder with a huge number of SA requests that are sent from spoofed IP addresses, creating half-open SAs.

You should implement the DDoS protection mechanisms described in RFC 8019 Protecting IKEv2 Implementations from DDoS Attacks [24]. In particular, we recommend reducing the lifetime of half-open SAs as well as setting a limit on the number of half-open SAs allowed for any given IP address and introducing additional protective measures once that limit is met.

You should not use IP fragmentation, as it is prone to DDoS attacks. Instead, use IKEv2 fragmentation and configure the size of the IKEv2 fragments. RFC 7383 Internet Key Exchange Protocol Version 2 (IKEv2) Message Fragmentation [25] recommends selecting an IKEv2 fragment size which results in a maximum datagram size of 1280 bytes for IPv6 traffic and 576 bytes for IPv4 traffic.

4.1.8 Key and authentication lifetimes

In the context of IKEv2, re-keying creates new key material for the IKE SA or a CHILD SA via the CREATE_CHILD_SA exchange. Re-authentication requires the creation of a new IKE SA. In this case, the old SAs are deleted.

We recommend that you ensure that the re-key period or key lifetime of a CHILD SA (including the Encapsulating Security Payload [ESP] SA) does not exceed 8 hours. The re-authentication period or authentication lifetime of the IKE SA should not exceed 24 hours.

4.1.9 Session resumption

RFC 5723 Internet Key Exchange Protocol Version 2 (IKEv2) Session Resumption [26] offers a means for peers to reconnect a broken connection by using a previously established IKE SA.

If session resumption is used, the ticket-by-reference method is recommended, under the condition that the peers can be trusted to maintain the security of stored SA information. We also recommend that you limit the lifetime of a ticket to no more than the re-keying time.

4.2 Internet Protocol Security

Internet Protocol Security (IPsec) is a suite of network protocols developed to protect the confidentiality, integrity, and availability of Internet communications between network hosts, gateways, and devices. IPsec also provides access control, replay protection, and traffic analysis protection.

IPsec hosts, gateways, and devices should be configured as specified in:

- RFC 4301 Security Architecture for the Internet Protocol [27]
- RFC 4303 IP Encapsulating Security Payload (ESP) [28]
- RFC 7321 Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH) [29]

IPsec key management guidance is provided in NIST SP 800-57 Part 3 Rev. 1 [4]. Refer to section 3 of that publication for further guidance on installing and administering IPsec.

4.2.1 Key generation

An IPsec SA specifies the key material used to encrypt and provide integrity protection for the traffic protected under a specific IPsec session. An IPsec SA must be established by a prior IKEv2 exchange as specified above.

4.2.2 Data and integrity protection

You should use digital signatures for authentication when IPsec is used to protect the confidentiality of PROTECTED A or PROTECTED B information or the integrity of UNCLASSIFIED, PROTECTED A, or PROTECTED B information. You should not use PSKs for authentication.

IPsec should use ESP protocol in tunnel mode to protect the confidentiality, integrity, and availability of the packets and packet headers. Do not use the Authentication Header (AH) protocol. AH protocol cannot protect confidentiality.

Table 12: lists the ESP packet encryption algorithms that comply with ITSP.40.111 [1] when used with a key length of 128, 192, or 256 bits.

Table 12: Recommended ESP packet encryption algorithms

Recommended	Sufficient	Phase out
ENCR_AES_GCM_16 ENCR_AES_CCM_16	ENCR_AES_GCM_12 ENCR_AES_CCM_12 ENCR_AES_CBC ENCR_AES_CTR	ENCR_3DES ENCR_CAST

We recommend that you use AES in GCM for the encryption of ESP packets, as described in RFC 4106 The Use of Galois/Counter Mode (GCM) in IPSec Encapsulating Security Payload [30]. If GCM or CCM is not supported, an integrity protection mechanism must be configured. Table 13: lists the integrity protection mechanisms that comply with ITSP.40.111 [1].

Table 13: Sufficient and phase out integrity protection mechanisms for ESP

Sufficient	Phase out
AUTH_HMAC_SHA2_256_128 AUTH_HMAC_SHA2_384_192 AUTH_HMAC_SHA2_512_256 AUTH_AES_128_GMAC AUTH_AES_192_GMAC AUTH_AES_256_GMAC AUTH_AES_CMAC_96	AUTH_HMAC_SHA1_160

4.2.3 Replay protection

Replay protection for IPsec implementations should be used. If performance allows, use the recommended anti-replay window size of 128.

5 Secure Shell

Secure Shell (SSH) is a protocol developed to protect the confidentiality, integrity, and availability of remote access, file transfer, and point-to-point tunneling over the Internet.

SSH servers and clients should be configured to use SSH protocol version 2.0. SSH is a family of protocols that is specified in:

- RFC 4251 The Secure Shell (SSH) Protocol Architecture [31]
- RFC 4252 The Secure Shell (SSH) Authentication Protocol [32]
- RFC 4253 The Secure Shell (SSH) Transport Layer Protocol [33]
- RFC 4254 The Secure Shell (SSH) Connection Protocol [34]

SSH protocol version 1.0 has serious vulnerabilities. Administrators should verify that it is not running on their systems.

NIST SP 800-57 Part 3 Rev. 1 [4] provides SSH key management guidance. Refer to section 10 of NIST SP 800-57 Rev. 1 for further guidance on installing and administering SSH.

5.1 SSH authentication

SSH offers both server-only and server-client mutual authentication.

You should use server-client mutual authentication. In this case, the server is first authenticated via the SSH Transport Layer Protocol, followed by client authentication via the SSH Authentication Protocol.

Server authentication is performed with public key cryptography. Client authentication to the server can use various mechanisms. Client authentication that is based on public keys or Kerberos is preferred rather than the various forms of password authentication. You should not use SSH host-based authentication as it is vulnerable to IP address spoofing.

If using public key authentication, you should use public key certificates that are managed by a PKI framework for both server and client authentication.

A PKI framework provides digital signing of keys by a trusted source. The framework also provides key management functions, such as revocation CRLs, key lifetime controls, and key usage restrictions. RFC 6187 x509.v3 Certificates for Secure Shell Authentication [35] specifies the use of x509.v3 certificates in SSH.

Since SSH keys are typically system-level keys, keys should be generated upon session initialization to ensure uniqueness across devices and virtual machine images.

5.2 SSH port forwarding

With SSH port forwarding, a host can access an insecure network service on a machine residing behind a server that acts as an SSH VPN gateway. Port forwarding should be disabled for interactive user accounts. For devices that require SSH tunneling, the traffic should be secured with a second tunnel, for example by using IPsec.

5.3 SSH root access

You should disable remote root user account logins.

5.4 SSH parameter selection

This section details the cryptographic algorithms recommend for SSH that satisfy the cryptographic guidance of ITSP.40.111 [1] and align with NIST SP 800-57 Part 3 Rev. 1 [4]. We recommend that you refer to subsection 10.2.1 of NIST SP 800-57 Part 3 Rev. 1 for cryptographic guidance on the SSH Transport Layer Protocol.

5.4.1 Encryption algorithm selection

Do not use Cipher Block Chaining (CBC) mode in SSH. CBC mode is vulnerable to plaintext recovery attacks. RFC 4344 The Secure Shell (SSH) Transport Layer Encryption Modes [36] recommends using Counter (CTR) mode in SSH in place of CBC mode. Even better, authenticated encryption with associated data (AEAD) algorithms (such as AES GCM) protect both authenticity and confidentiality. Therefore, when you use AEAD algorithms, you do not need to use a separate message authentication code (MAC) algorithm.

Table 14: lists the SSH encryption algorithms that satisfy the cryptographic guidance provided in ITSP.40.111 [1].

Table 14: Recommended SSH encryption algorithms

Recommended	Sufficient	Phase out
AEAD_AES_128_GCM AEAD_AES_256_GCM	aes128-ctr aes192-ctr aes256-ctr	cast128-ctr 3des-ctr

The AEAD GCM encryption algorithms are vulnerable to nonce reuse. Implementations should ensure that the (key, nonce) pair is unique for each encrypted message.

5.4.2 MAC algorithm selection

In addition to the AEAD algorithms specified above, Table 15: lists the SSH MAC algorithms that satisfy the cryptographic guidance provided in ITSP.40.111 [1].

Table 15: Sufficient and phase out SSH MAC algorithms

Sufficient	Phase out
hmac-sha2-256 hmac-sha2-512	hmac-sha1

5.4.3 Key exchange algorithm

Table 16: lists the SSH key exchange algorithms that satisfy the cryptographic guidance provided in ITSP.40.111 [1].

Table 16: Recommended SSH key exchange algorithms

Recommended	Sufficient	Phase out
ecdh-sha2-nistp256	diffie-hellman-group15-sha512	rsa2048-sha256
ecdh-sha2-nistp384	diffie-hellman-group16-sha512	diffie-hellman-group14-sha256
ecdh-sha2-nistp521	diffie-hellman-group17-sha512	gss-group14-sha256-*
ecmqv-sha2	diffie-hellman-group18-sha512	
gss-nistp256-sha256-*	gss-group15-sha512-*	
gss-nistp384-sha384-*	gss-group16-sha512-*	
gss-nistp521-sha512-*	gss-group17-sha512-*	
	gss-group18-sha512-*	

The SSH protocol allows the session keys to be renewed by either the client or the server. Re-keying schedules are based on a time limit or a data volume, as described in RFC 4344 [36].

To avoid MAC collisions, RFC 4344 [36] recommends re-keying after receiving 2^{32} packets when a 32-bit sequence number is used.

5.4.4 Public key algorithm

SSH optionally allows for authentication using public keys. Table 17: lists the SSH public key algorithms that satisfy the cryptographic guidance provided in ITSP.40.111 [1].

Table 17: Recommended SSH public key algorithms

Recommended	Sufficient	Phase out
ecdsa-sha2-nistp256	rsa-sha2-256	x509v3-ecdsa-sha2-nistp224
ecdsa-sha2-nistp384	rsa-sha2-512	
ecdsa-sha2-nistp521	x509v3-rsa2048-sha256	
ssh-ed25519		
ssh-ed448		
x509v3-ecdsa-sha2-nistp256		
x509v3-ecdsa-sha2-nistp384		
x509v3-ecdsa-sha2-nistp521		

6 Simple Network Management Protocol

Simple Network Management Protocol (SNMP) is an IETF protocol designed for managing and monitoring devices on a computer network. The latest version, SNMPv3, is specified in:

- An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks. Request for Comments (RFC) 3411 [37]
- Message Processing and Dispatching for the Simple Network Management Protocol (SNMP). Request for Comments (RFC) 3412 [38]
- Simple Network Management Protocol (SNMP) Applications. Request for Comments (RFC) 3413 [39]
- User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3). Request for Comments (RFC) 3414 [40]
- View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP). Request for Comments (RFC) 3415 [41]
- Protocol Operations for the Simple Network Management Protocol (SNMP). Request for Comments (RFC) 3416 [42]
- Transport Mappings for the Simple Network Management Protocol (SNMP). Request for Comments (RFC) 3417 [43]
- Management Information Base (MIB) for the Simple Network Management Protocol (SNMP). Request for Comments (RFC) 3418 [44]

It provides security improvements to previous versions, such as SNMPv1, SNMPv2, SNMPv2c, SNMPv2u and SNMPv2*. All SNMP versions older than SNMPv3 should be phased out.

SNMPv3 adds security and remote configuration capabilities to the previous versions and a new access control model. The SNMPv3 architecture introduces two security mechanisms: the User-based Security Model (USM) and the Transport Security Model (TSM).

The use of USM is defined in RFC 3414 User-based Security Model (USM) for version 3 of the Simple Network Management Protocol [40]. The use of TSM is defined for TLS/DTLS in RFC 6353 Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP) [45] and for SSH in RFC 5592 Secure Shell Transport Model for the Simple Network Management Protocol (SNMP) [46]. TSM allows for a similar user-based access control. RFC 3415 View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP) [41] defines a process for controlling access to management information based on groups with specific access rights and security.

6.1 SNMPv3 interfaces and access control

We recommend that SNMP interfaces be accessible only from within an internal, corporate network and additionally on a specific management LAN or VLAN that is separate from regular network traffic. If external access is required, we recommend that access to the interface only be allowed within an IPsec tunnel.

We recommend that SNMP be disabled on any device which is not being actively managed. Special care must be taken with new devices which may have SNMP enabled by default.

Implementations can offer two security models: USM and TSM. There are 3 security levels for messages:

- without authentication and without privacy (noAuthNoPriv)
- with authentication but without privacy (authNoPriv)
- with authentication and with privacy (authPriv)

We recommend the use of TSM as the security model when available. Specific recommendations for TSM are described in section 6.3 TSM security model. The use of USM is sufficient at the authPriv security level if it is configured following recommendations in section 6.2 SNMPv3 USM security model.

We recommend that read-write access to management information be restricted and only granted to a limited number of administrative groups. We also recommend that accessible information for each user be explicitly specified in the configuration. We do not recommend relying solely on globally denying access to restricted information. Configurations should be reviewed periodically to determine if the assigned levels of access are still appropriate for groups and users.

6.2 SNMPv3 USM security model

The SNMPv3 User-Based Security Model (USM) described in RFC 3414 [40] offers both authentication and privacy for SNMPv3 messages. The USM was designed to function independently of other existing security infrastructures and can function when other network security infrastructures are unavailable. When configuring groups and users, we recommend that the required security level be set at the authPriv level to guarantee that both authentication and privacy will be applied. Recommended algorithms for authentication and privacy are defined in 6.2.1 SNMPv3 USM authentication algorithms and 6.2.2 SNMPv3 USM privacy algorithms respectively.

6.2.1 SNMPv3 USM authentication algorithms

SNMPv3 offers message authentication based on secure message digest (HMAC). Table 18: lists the algorithms that satisfy the cryptographic guidance provided in ITSP.40.111 [1]. The option usmNoAuthProtocol provides no authentication and should not be used.

Table 18: Sufficient and phase out authentication algorithms for SNMPv3 USM

Sufficient	Phase out
usmHMAC192SHA256AuthProtocol usmHMAC256SHA384AuthProtocol usmHMAC384SHA512AuthProtocol	usmHMAC128SHA224AuthProtocol usmHMACSHAAuthProtocol usmHMACMD5AuthProtocol

6.2.2 SNMPv3 USM privacy algorithms

SNMPv3 USM offers privacy protection via message encryption. Table 19: lists the algorithms that satisfy the cryptographic guidance provided in ITSP.40.111 [1]. The option usmNoPrivProtocol provides no privacy protection and should not be used.

If not using the recommended method of salt generation for IV formation in RFC 3826 The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model [47], implementations should use a method to ensure IVs are unpredictable in accordance with ITSP.40.111 [1].

Table 19: Sufficient and phase out privacy protection for SNMPv3 USM

Sufficient	Phase out
usmAesCfb128Protocol	usmNoPrivProtocol usmDESPrivProtocol

6.2.3 USM authentication and privacy secrets

USM authentication and privacy rely on choosing strong user secrets and protecting them from disclosure.

We recommend that the USM authentication and privacy secrets be based on a randomly generated pre-shared key rather than derived from a user-generated password. If user-generated passwords are used, passwords are recommended to satisfy requirements set in User authentication guidance for information technology systems (ITSP.30.031 v3) [48] and Best practices for passphrases and passwords (ITSAP.30.032) [49]. We strongly recommend not using the same secrets for privacy and authentication.

We recommend that user secrets never be stored on any device performing SNMP encryption and authentication services, rather only the localized keys derived from the user secrets as defined in RFC 3414 [40] should be kept on such devices. We recommend that organizations create a system for securely managing user secrets.

We recommend that a user's localized keys be derived and configured locally on each managed device through the configuration interface. The localized keys should be set immediately after the new user is created as no default keys should be used to encrypt or authenticate messages.

If a user's localized keys are compromised on a device, we recommend that the user not directly change the compromised keys, but that the user's secrets be changed, and new keys generated through the configuration interface.

If new users are created by cloning a pre-existing user as defined in RFC 3414 [40], we recommend that the user being cloned should have minimal access to management information and should not send messages on the network.

We recommend that user secrets be periodically updated following recommendations set down in ITSP.30.031 [48] or organizational password policy.

6.3 TSM security model

The SNMPv3 Transport Security Model described in RFC 5591 Transport Security Model for the Simple Network Management Protocol (SNMP) [50] relies on the use of other specific secure transport protocols for mutual authentication, binding of keys, confidentiality, and integrity. RFC 5591 mandates how transport security protocols such as SSH, DTLS and TLS can be used to secure SNMPv3 traffic to meet one of the security levels mentioned in section 6.1 SNMPv3 interfaces and access control.

TSM is a good choice for organizations that already have or are planning to deploy an X.509 PKI. The use of TSM precludes the necessity of managing SNMP USM private keys. Organizations using TSM may consider maintaining an equivalent USM configuration as backup, particularly if there is any concern about stressed or unavailable networks rendering the secure transport protocol inoperable. If USM is allowed as a fallback, its use should be logged and reported immediately to administrators as suspicious behaviour.

6.3.1 SNMPv3 over TLS/DTLS

RFC 6353 [45] spells out the general recommendations for configuring SNMPv3 in the TSM security model using TLS or DTLS. When using TLS or DTLS, the recommendations from Section 3 Transport Layer Security should be followed to achieve the equivalent of authPriv security and ensure that sufficient encryption and authentication services are applied. Acceptable cipher suites are listed in Table 1: Recommended cipher suites for TLS 1.2 and Table 2: TLS supported groups that conform to ITSP.40.111.

As stated in RFC 6353 [45], a certificate's subjectAltName should be used to map certificates to SNMP security names.

The choice of hash algorithm used in a certificate's SnmpTLSFingerprint should be a collision resistant algorithm that follows the guidance from ITSP.40.111 [1].

6.3.2 SNMPv3 over SSH

The use of SSH with SNMPv3 is specified in RFC 5592 [46] and we recommend that the guidelines for establishing an SSH tunnel detailed in Section 5 Secure Shell be followed to ensure confidentiality and integrity. Acceptable mechanisms for client authentication are set out in Table 17: of this publication.

Ensure that SSH is not configured to skip public-key verification.

6.4 SNMPv3 over an IPsec tunnel

An IPsec tunnel can be used to protect the SNMPv3 traffic already configured under the USM or the TSM as recommended above. Recommendations for the establishment of an IPsec tunnel are defined in section 4 Internet Protocol Security .

6.5 SNMPv3 notifications: Traps and informs

Trap and inform notifications should be transmitted securely. Implementations may use a separate configuration for notifications. When configuring the USM security level for notifications, we strongly recommend using the same security level as was used for SNMPv3 commands but protected by a different set of keys.

6.6 SNMPv3 discovery process

The SNMPv3 discovery process consists of one or more requests which allows an SNMP entity to obtain another entity's configured identity when communicating with it for the first time.

The second discovery request to determine the clock of the managed entity is authenticated and can be performed as often as needed to maintain time synchronisation (even if the initial request was not performed).

When using USM, the discovery's initial request and response are not authenticated or encrypted. This means the response which contains the entity's identity could be spoofed or modified by a malicious agent. When using TSM, all discovery messages are authenticated and encrypted.

SNMP entities making discovery requests should either:

- maintain a list of identities with their network addresses to avoid having to make the initial request altogether
- make the initial request in an IPsec tunnel so that it is cryptographically protected

7 Secure/Multipurpose Internet Mail Extensions

Secure/Multipurpose Internet Mail Extensions (S/MIME) is a standard developed to protect the confidentiality, integrity, and availability of electronic messages over the Internet.

S/MIME 4.0 as specified in RFC 8551 Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification [51] and RFC 8550 Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Certificate Handling [52] should be used. S/MIME 4.0 includes support for AES-GCM.

RFC 5753 Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS) [53] provides guidance on the use of elliptic curve cryptography (ECC) in Cryptographic Message Syntax (CMS) for generating digital signatures and exchanging keys to encrypt or authenticate messages.

Software vendors should implement multi-part isolation with security considerations for dealing with HTML and multi-part/mixed messages, as discussed in RFC 8551 [51]. Until such multi-part isolation is supported, S/MIME clients must be configured to disable the loading of remote content or only display messages in plain text.

7.1 Digest algorithms

Digest algorithms are used in S/MIME for digesting the body of a message or as part of a signature algorithm. Table 20: lists the digest algorithms that satisfy the cryptographic guidance provided in ITSP.40.111 [1].

Table 20: Sufficient and phase out S/MIME digest algorithms

Sufficient	Phase out
SHA-256 SHA-384 SHA-512 SHA3-256 SHA3-384 SHA3-512	SHA-224 SHA3-224

Using SHA-1 to generate digital signatures does not satisfy the cryptographic guidance provided in ITSP.40.111 [1]. For S/MIME 3.2 or earlier versions, SHA-1 should not be used as a digest algorithm to sign messages.

7.2 Signature algorithms

Signature algorithms should be used with a digest algorithm. Table 21: lists the signature algorithms, which are paired with a digest algorithm from Section 7.1, that satisfy the cryptographic guidance provided in ITSP.40.111 [1].

Table 21: Recommended S/MIME signature algorithms

Recommended	Sufficient	Phase out
ECDSA with NIST P-256 curve ECDSA with NIST P-384 curve ECDSA with NIST P-521 curve EdDSA with curve25519 RSASSA PSS with 3072-bit or larger modulus	RSASSA PKCS1v1.5 with 3072-bit or larger modulus	ECDSA with NIST P-224 curve RSASSA PSS with 2048-bit modulus RSASSA PKCS1v1.5 with 2048-bit modulus DSA with any group size

We recommend using RSASSA-PSS (instead of PKCS #1 v1.5) as the encoding mechanism for RSA digital signatures. This applies to both X.509 certificates, as specified in RFC 5756 Updates for RSAES-OAEP and RSASSA-PSS Algorithm Parameters [54], and signed-data content types, as specified in RFC 4056 Use of the RSASSA-PSS Signature Algorithm in Cryptographic Message Syntax (CMS) [55]. If signing with multiple signature algorithms, implementations should use the multipleSignatures CMS attribute as specified in RFC 5752 Multiple Signatures in Cryptographic Message Syntax (CMS) [56].

Implementations of RSASSA-PSS should protect against possible hash algorithm substitution attacks. Implementations should check that the hash algorithm used to compute the digest of the message content is the same as the hash algorithm used to compute the digest of signed attributes.

7.3 Key encryption algorithms

Most key encryption algorithms for S/MIME require a key wrap algorithm to be specified as a parameter. Acceptable key wrap algorithms are specified in subsection 7.3.1 Key wrap algorithms of this document. Table 22: lists the key encryption algorithms that satisfy the cryptographic guidance provided in ITSP.40.111 [1].

Table 22: Recommended S/MIME key encryption algorithms

Recommended	Sufficient	Phase out
dhSinglePass stdDH SHA256 KDF with the NIST P-256 curve dhSinglePass stdDH SHA384 KDF with the NIST P-384 curve dhSinglePass stdDH SHA512 KDF with the NIST P-521 curve	RSAES OAEP with a 3072-bit or larger modulus dhSinglePass cofactorDH SHA256 KDF with the NIST P-256 curve dhSinglePass cofactorDH SHA384 KDF with the NIST P-384 curve dhSinglePass cofactorDH SHA512 KDF with the NIST P-521 curve mqvSinglePass SHA256 KDF with the NIST P-256 curve mqvSinglePass SHA384 KDF with the NIST P-384 curve mqvSinglePass SHA512 KDF with the NIST P-521 curve	dhSinglePass stdDH SHA224 KDF with the NIST P-224 curve dhSinglePass cofactorDH SHA224 KDF with the NIST P-224 curve RSA KEM with a 2048-bit modulus or larger RSAES OAEP with a 2048-bit modulus RSAES PKCS1v1.5 with a 2048-bit or larger modulus

We recommend the use of standard Elliptic Curve Diffie-Hellman, as specified in RFC 5753 Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS) [53].

If you are using RSA encryption, RSAES-OAEP should be implemented, as specified in RFC 3560 Use of the RSAES-OAEP Key Transport Algorithm in Cryptographic Message Syntax (CMS) [57] and RFC 5756 [54], to meet the cryptographic guidance of ITSP.40.111 [1].

Mitigations like careful checking or random filling should be implemented, as described in RFC 3218 Preventing the Million Message Attack on Cryptographic Message Syntax [58], if you have S/MIME implementations that allow the decryption of PKCS #1 v1.5 encoding.

7.3.1 Key wrap algorithms

Table 23: lists the key wrap algorithms that can be used with an appropriate key encryption algorithm to satisfy the cryptographic guidance provided in ITSP.40.111 [1].

Table 23: Recommended S/MIME key wrap algorithms

Recommended	Phase out
AES-128 Wrap AES-192 Wrap AES-256 Wrap AES-128 Wrap Pad AES-192 Wrap Pad AES-256 Wrap Pad	3DES Wrap CAST5 CMS Key Wrap with a key length of 128 bits

7.4 Content encryption algorithms

The S/MIME content encryption algorithms listed in Table 24: satisfy the cryptographic guidance provided in ITSP.40.111 [1].

Table 24: S/MIME content encryption algorithms

Recommended	Phase out
AES-128 GCM AES-192 GCM AES-256 GCM	AES-128 CBC AES-192 CBC AES-256 CBC

8 Commercial technologies assurance programs

When implementing PKI, TLS, IPsec, SSH and S/MIME, the implementation assurance guidance in Section 11 of Cryptographic Algorithms for UNCLASSIFIED, PROTECTED A, and PROTECTED B Information (ITSP.40.111) [1] should be followed.

9 Preparing for post quantum cryptography

Quantum computers threaten to break the public key cryptosystems and weaken the symmetric cryptosystems that we currently use. Although quantum technologies are not yet powerful enough to break the cryptography recommended in this publication, there is significant research in the area. . In August 2024, NIST published standards for post-quantum cryptography that is designed to be resistant to the advantages of future quantum computers. Organizations that maintain the protocol standards listed in this publication are currently working on revising the protocols to integrate post-quantum cryptography. We expect to include recommendations for the configuration of post-quantum cryptography in an update to this publication once the protocol standards are finalized.

In the meantime, we recommend the following high-level steps:

- Evaluate the sensitivity of your organization's information and determine its lifespan to identify information that may be at risk (for example, as part of on-going risk assessment processes)
- Review your IT lifecycle management plan and budget for potentially significant software and hardware updates
- Educate your workforce on the quantum threat

For more detailed information on how to prepare, consult Preparing your organization for the quantum threat to cryptography (ITSAP.00.017) [59].

Organizations should wait until the standards for using post-quantum cryptography in protocols are finalized before revising configurations to protect information or systems.

10 Summary

Your organization can use cryptographic security protocols to provide the security mechanisms to protect the confidentiality, integrity, and availability of information. As a first step, you should determine your organizational security requirements before choosing which protocols to use. Your organization may require the use of multiple protocols to satisfy a particular security requirement. You should select and implement each protocol in a manner that supports and meets your specific organizational requirements.

11 Supporting content

11.1 List of abbreviations

Term	Definition
AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard
AH	Authentication Header
CA	Certificate Authority
CBC	Cipher Block Chaining
CMS	Cryptographic Message Syntax
CMVP	Cryptographic Module Validation Program
CRL	Certificate Revocation List
DANE	DNS-Based Authentication of Named Entities
DDoS	Distributed Denial of Service
DH	Diffie-Hellman
DNS	Domain Name System
DTLS	Datagram Transport Layer Security
ECC	Elliptic Curve Cryptography
ECDH	Elliptic-Curve Diffie-Hellman
ECDHE	Ephemeral Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
ECP	Elliptic Curve Groups modulo a Prime
ESP	Encapsulating Security Payload
GC	Government of Canada
GCM	Galois/Counter Mode
HMAC	Keyed-Hash Message Authentication Code
HSTS	HTTP Strict Transport Security
IKE	Internet Key Exchange
IPsec	Internet Protocol Security
IT	Information Technology
MAC	Message Authentication Code
MTA	Message Transfer Agent
NIST	National Institute of Standards and Technology

Term	Definition
PFS	Perfect Forward Secrecy
PKI	Public Key Infrastructure
PRF	Pseudo-Random Function
PSK	Pre-shared Key
RFC	Request for Comments
RSA	Rivest Shamir Adleman
SA	Security Association
SHA	Secure Hash Algorithm
SSH	Secure Shell
S/MIME	Secure/Multipurpose Internet Mail Extensions
SMTP	Simple Mail Transfer Protocol
SP	Special Publication
SSL	Secure Socket Layer
TBS	Treasury Board of Canada Secretariat
TLS	Transport Layer Security
CSE	Communications Security Establishment
GC	Government of Canada
IT	Information Technology
ITS	Information Technology Security
SSH	Secure Shell
TLS	Transport Layer Protocol

11.2 Glossary

Term	Definition
Authentication	A process or measure used to verify a user's identity.
Authenticity	The state of being genuine and being able to be verified and trusted; confidence in the validity of a transmission, a message, or message originator.
Availability	The ability for the right people to access the right information or systems when needed. Availability is applied to information assets, software, and hardware (infrastructure and its components).
Classified information	A Government of Canada label for specific types of sensitive data that, if compromised, could cause harm to the national interest (e.g. national defence, relationships with other countries, economic interests).
Confidentiality	The ability to protect sensitive information from being accessed by unauthorized people.

Term	Definition
Cryptography	The study of techniques used to make plain information unreadable, as well as to convert it back to a readable form.
Distributed denial of service (DDoS) Attack	An attack in which multiple compromised systems are used to attack a single target. The flood of incoming messages to the target system forces it to shut down and denies service to legitimate users.
Decryption	A process that converts encrypted voice or data information into plain form by reversing the encryption process.
Digital signature	A cryptologic mechanism used to validate an item's (e.g. document, software) authenticity and integrity.
Encryption	Converting information from one form to another to hide its content and prevent unauthorized access.
Forward secrecy	A property of key establishment protocols where the compromise of the long-term private key will not allow an adversary to re-compute previously derived keys or sessions.
Integrity	The ability to protect information from being modified or deleted unintentionally when it's not supposed to be. Integrity helps determine that information is what it claims to be. Integrity also applies to business processes, software application logic, hardware, and personnel.
Key management	The procedures and mechanisms for generating, disseminating, replacing, storing, archiving, and destroying cryptographic keys.
Replay attack	A form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed.

11.3 References

Number	Reference
1	Canadian Centre for Cyber Security. Cryptographic Algorithms for UNCLASSIFIED, PROTECTED A, and PROTECTED B Information (ITSP.40.111). September 2023.
2	Treasury Board of Canada Secretariat. Guideline on Defining Authentication Requirements. November 2012.
3	Canadian Centre for Cyber Security. IT Security Risk Management: A Lifecycle Approach (ITSG-33). November 2012.
4	National Institute of Standards and Technology. Recommendation for Key Management Part 3: Application-Specific Key Management Guidance Special Publication 800-57 Part 3 Rev 1. January 2015.
5	Cooper, D., et al. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. Request for Comments (RFC) 5280. Internet Engineering Task Force (IETF). May 2008.
6	Rescorla, E. The Transport Layer Security (TLS) Protocol Version 1.3. Request for Comments (RFC) 8446. Internet Engineering Task Force (IETF). August 2018.
7	Hodges, J., Jackson, C. and Barth, A. HTTP Strict Transport Security (HSTS). Request for Comments (RFC) 6797. Internet Engineering Task Force (IETF). November 2012.

Number	Reference
8	Hoffman, P. SMTP Service Extension for Secure SMTP over Transport Layer Security. Request for Comments (RFC) 3207. Internet Engineering Task Force (IETF). February 2002.
9	Margolis, D., et al. SMTP MTA Strict Transport Security (MTA-STS). Request for Comments (RFC) 8461. Internet Engineering Task Force (IETF). September 2018.
10	Dukhovni, V. and Hardaker, W. SMTP Security via Opportunistic DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS). Request for Comments (RFC) 7672. Internet Engineering Task Force (IETF). October 2015.
11	Friedl, S., et al. Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension. Request for Comments (RFC) 7301. Internet Engineering Task Force (IETF). July 2014.
12	Eastlake, D. and 3rd. Transport Layer Security (TLS) Extensions: Extension Definitions. Request for Comments (RFC) 6066. Internet Engineering Task Force (IETF). January 2011.
13	Gutmann, P. Encrypt-then-MAC for Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS). Request for Comments (RFC) 7366. Internet Engineering Task Force (IETF). September 2014.
14	Bhargavan Ed, K., et al. Transport Layer Security (TLS) Session Hash and Extended Master Secret Extension. Request for Comments (RFC) 7627. Internet Engineering Task Force (IETF). September 2015.
15	Pettersen, Y. The Transport Layer Security (TLS) Multiple Certificate Status Request Extension. Request for Comments (RFC) 6961. Internet Engineering Task Force (IETF). June 2013.
16	Ray, M. and Dispensa, S. Transport Layer Security (TLS) Renegotiation Indication Extension. Request for Comments (RFC) 5746. Internet Engineering Task Force (IETF). February 2010.
17	Nir, Y., Josefsson, S. and Pegourie-Gonnard, M. Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS) Versions 1.2 and Earlier. Request for Comments (RFC) 8422. Internet Engineering Task Force (IETF). August 2018.
18	Gillmor, D. Negotiated Finite Field Diffie-Hellman Ephemeral Parameters for Transport Layer Security (TLS). Request for Comments (RFC) 7919. Internet Engineering Task Force (IETF). August 2016.
19	Laurie, B., Messeri, E. and Stradling, R. Certificate Transparency Version 2.0. Request for Comments (RFC) 9162. Internet Engineering Task Force (IETF). December 2021.
20	Laurie, B., Langley, A. and Kasper, E. Certificate Transparency. Request for Comments (RFC) 6962. Internet Engineering Task Force (IETF). June 2013.
21	Kaufman, C. and et al. Internet Key Exchange Protocol Version 2 (IKEv2). Request for Comments (RFC) 7296. Internet Engineering Task Force (IETF). October 2014.
22	Hoffman, P. and Snell, J. JSON Merge Patch. Request for Comments (RFC) 7396. Internet Engineering Task Force (IETF). October 2014.
23	Eronen, P. and Tschofenig, H. An Extension for EAP-Only Authentication in IKEv2. Request for Comments (RFC) 5998. Internet Engineering Task Force (IETF). September 2010.
24	Nir, Y. and Smyslov, V. Protecting IKEv2 Implementations from Distributed Denial-of-Service Attacks. Request for Comments (RFC) 8019. Internet Engineering Task Force (IETF). November 2016.
25	Smyslov, V. Internet Key Exchange Protocol Version 2 (IKEv2) Message Fragmentation. Request for Comments (RFC) 7383. Internet Engineering Task Force (IETF). November 2014.
26	Sheffer, Y. and Tschofenig, H. Internet Key Exchange Protocol Version 2 (IKEv2) Session Resumption. Request for Comments (RFC) 5723. Internet Engineering Task Force (IETF). January 2010.

Number	Reference
27	Kent, S. and Seo, K. Security Architecture for the Internet Protocol. Request for Comments (RFC) 4301. Internet Engineering Task Force (IETF). December 2005.
28	Kent, S. IP Encapsulating Security Payload (ESP). Request for Comments (RFC) 4303. Internet Engineering Task Force (IETF). December 2005.
29	McGrew, D. and Hoffman, P. Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH). Request for Comments (RFC) 7321. Internet Engineering Task Force (IETF). August 2014.
30	Viega, J. and McGrew, D. The Use of Galois Counter Mode (GCM) in IPSec Encapsulating Security Payload. Request for Comments (RFC) 4106. Internet Engineering Task Force (IETF). June 2005.
31	Ylonen, T. and Lonvick, C., Ed. The Secure Shell (SSH) Protocol Architecture. Request for Comments (RFC) 4251. Internet Engineering Task Force (IETF). January 2006.
32	Ylonen, T. and Lonvick, C., Ed. The Secure Shell (SSH) Authentication Protocol. Request for Comments (RFC) 4252. Internet Engineering Task Force (IETF). January 2006.
33	Ylonen, T. and Lonvick, C., Ed. The Secure Shell (SSH) Transport Layer Protocol. Request for Comments (RFC) 4253. Internet Engineering Task Force (IETF). January 2006.
34	Ylonen, T. and Lonvick, C., Ed. The Secure Shell (SSH) Connection Protocol. Request for Comments (RFC) 4254. Internet Engineering Task Force (IETF). January 2006.
35	Igoe, K. and Stebila, D. x509.v3 certificates for Secure Shell Authentication. Request for Comments (RFC) 6187. Internet Engineering Task Force (IETF). March 2011.
36	Bellare, M., Kohno, T. and Namprempre, C. The Secure Shell (SSH) Transport Layer Encryption Modes. Request for Comments (RFC) 4344. Internet Engineering Task Force (IETF). January 2006.
37	Harrington, D., Presuhn, R. and Wijnen, B. An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks. Request for Comments (RFC) 3411. Internet Engineering Task Force (IETF). December 2002.
38	Case, J., et al. Message Processing and Dispatching for the Simple Network Management Protocol (SNMP). Request for Comments (RFC) 3412. Internet Engineering Task Force (IETF), December 2002.
39	Levi, D., Meyer, P. and Stewart, B. Simple Network Management Protocol (SNMP) Applications. Request for Comments (RFC) 3413. Internet Engineering Task Force (IETF). December 2002.
40	Blumenthal, U. and Wijnen, B. User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3). Request for Comments (RFC) 3414. Internet Engineering Task Force (IETF). December 2002.
41	Wijnen, B., Presuhn, R. and McCloghrie, K. View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP). Request for Comments (RFC) 3415. Internet Engineering Task Force (IETF), December 2002.
42	Presuhn, R., et al. Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP). Request for Comments (RFC) 3416. Internet Engineering Task Force (IETF). December 2002.
43	Presuhn, R., et al. Transport Mappings for the Simple Network Management Protocol (SNMP). Request for Comments (RFC) 3417. Internet Engineering Task Force (IETF). December 2002.
44	Presuhn, R., et al. Management Information Base (MIB) for the Simple Network Management Protocol (SNMP). Request for Comments (RFC) 3418. Internet Engineering Task Force (IETF). December 2002.

Number	Reference
45	Hardaker, W. Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP). Request for Comments (RFC) 6353. Internet Engineering Task Force (IETF). July 2011.
46	Harrington, D., Salowey, J. and Hardaker, W. Secure Shell Transport Model for the Simple Network Management Protocol (SNMP). Request for Comments (RFC) 5592. Internet Engineering Task Force (IETF). June 2009.
47	Blumenthal, U., Maino, F. and McCloghrie, K. The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model. Request for Comments (RFC) 3826. Internet Engineering Task Force (IETF). June 2004.
48	Canadian Centre for Cyber Security. User authentication guidance for information technology systems (ITSP.30.031) v3. April 2018.
49	Canadian Centre for Cyber Security. Best practices for passphrases and passwords (ITSAP.30.032). September 2019.
50	Harrington, D. and Hardaker, W. Transport Security Model for the Simple Network Management Protocol (SNMP). Request for Comments (RFC) 5591. Internet Engineering Task Force (IETF). June 2009.
51	Schaad, J., Ramsdell, B. and Turner, S. Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification. Request for Comments (RFC) 8551. Internet Engineering Task Force (IETF). April 2019.
52	Schaad, J., Ramsdell, B. and Turner, S. Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Certificate Handling. Request for Comments (RFC) 8550. Internet Engineering Task Force (IETF). April 2019.
53	Turner, S. and Brown, D. Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS). Request for Comments (RFC) 5753. Internet Engineering Task Force (IETF). January 2010.
54	Turner, S., et al. Updates for RSAES-OAEP and RSASSA-PSS Algorithm Parameters. Request for Comments (RFC) 5756. Internet Engineering Task Force (IETF). January 2010.
55	Schaad, J. Use of the RSASSA-PSS Signature Algorithm in Cryptographic Message Syntax (CMS). Request for Comments (RFC) 4056. Internet Engineering Task Force (IETF). June 2005.
56	Turner, S. and Schaad, J. Multiple Signatures in Cryptographic Message Syntax (CMS). Request for Comments (RFC) 5752. Internet Engineering Task Force (IETF). January 2010.
57	Housley, R. Use of the RSAES-OAEP Key Transport Algorithm in Cryptographic Message Syntax (CMS). Request for Comments (RFC) 3560. Internet Engineering Task Force (IETF). July 2003.
58	Rescorla, E. Preventing the Million Message Attack on Cryptographic Message Syntax. Request for Comments (RFC) 3218. Internet Engineering Task Force (IETF). January 2002.
59	Canadian Centre for Cyber Security. Preparing your organization for the quantum threat to cryptography (ITSAP.00.017). February 2021.