Anvisning för kryptering

Anvisning för kryptering gällande transport och lagrad information för tjänster som Inera tillhandahåller

Version 3.2

Godkänt av Caroline Hagberg

Datum
2020-05-15

inera
LANDSTING OCH REGIONER
I SAMVERKAN FÖR E-HÄLSA

Innehåll

1. Dokumentinformation

1.1 Revisionshistorik

Version	Datum	Kommentar	Författare
1.0			Björn Skeppner
2.0 RC1	2017-04-19	Hela dokumentet reviderat Ändrat Camellia från röd till gul. Ändringar för att tydliggöra att anvisningen gäller tjänster som tillhandahålls av Inera.	Bengt-Göran Andersson, Björn Gustavsson, Christoffer Johansson
2.1	2018-01-30	Reviderat av Christoffer Johansson, Björn Gustavsson och Roberth Lundin Inkonsistens i beskrivna Cipher Suites borttagen. Godkända Cipher Suites uppdaterats och flyttat till Appendix Beslut på Informations och IT Säkerhetsrådet att godkänna uppdateringen till v2.1	Bengt-Göran Andersson, Inera
3.0	2018-10-01	Anvisningen uppdateras att gälla både transportkryptering IPSec och TLS samt kryptering av lagrad information. Ny Lila rekommendationsnivå skapad plus uppdatering av text för IPsec och filkryptering. Förkortningar flyttat till kap 1 och uppdaterat. Referenser uppdaterade. Information och Cipher Suites för TLS 1.3 adderat. Cipher Suite för IPsec adderat. Omstrukturerat för att tydliggöra rekommenderade algoritmer. TLS 1.0 nedgraderats till Röd nivå. Ny Informativ rubrik för att tydliggöra vad som är en rekommendation Korrigering av smärre skrivfel	Bengt-Göran Andersson, Inera
3.1	2019-08-21	Ny hänvisning kap 7.3.1 som hänvisar till Instruktion för nyckelhantering	Bengt-Göran Andersson, Inera

3.2	2020-05-13	TLS 1.1 ändras röd nivå dvs. till icke acceptabel. TLS Cipher Suites med CBC nedgraderas till gul nivå SHA-1 nedgraderas till röd nivå Cipher Suites uppdaterat för TLS 1.2 och 1.3. Externa referenser omarbetade Alla korsreferenser i dokumentet ska vara länkar till kapitel. Nya IETF referenser tillagda för standarddokument från till exempel NIST och IETF. Nya Förkortningar tillagda. Nya rubriker och omfördelning av kapitel Nytt kapitel 1 Dokumentinformation och 1.1 Revidering Nytt kapitel 3, Ineras rekommendationer Nytt kapitel 5, Nyheter inom krypteringsområdet Kapitel 6, Cipher Suites uppdaterad, Kapitel 12.1, Grundkonfiguration av webbservrar, uppdaterad och verifierad Kapitel 12.2, Godkända och accepterade Cipher Suites för TLS uppdaterad	Bengt-Göran Andersson, Inera

1.2 Revidering

Anvisning för kryptering ska revideras årligen utgående från det godkännandedatum som anges på första sidan av dokumentet eller när skäl finns att uppdatera hela eller delar av dokumentet. Revisionsinformation dvs. nuvarande status på anvisningen finns under rubriken Revisionshistorik.

Ineras Informations- och IT-säkerhetsfunktion är ägare av denna Anvisning.

2. Förkortningar-förklaringar

Ett urval av använda förkortningar, beteckningar och förklaringar som berör kryptering.

0-RTT	Zero Round Trip Time Resumption
AES	Advanced Encryption Standard, är en specifikation för the kryptering av elektroniska data etablerad av U.S. National Institute of Standards and Technology (NIST) 2001. AES har en blockstorlek på 128 bitar, men är definierad med tre olika nyckellängder: 128, 192 och 256 bitar. [R18]
AEAD	Authenticated Encryption (AE) and Authenticated Encryption with Associated Data (AEAD) mode [R13]
AH	Authentication Header (IPSec) [R6b]
Camellia	En symmetrisk blockkryptoalgoritm med en blockstorlek av 128 bitar och en nyckellängd av 128, 192 och and 256 bits.
CBC	Cipher Block Chaining, i CBC mode blir varje block av klartext XOR:ad med det tidigare krypterade blocket innan den krypteras. Ett block har alltid fast längd.
DES	Data Encryption Standard, är en symmetrisk blockkryptoalgoritm med på 56 bitars nyckellängd för kryptering av elektroniska data.
3DES	”Triple DES” (3DES), är ett symmetriskt blockkrypto som applicerar DES kryptoalgoritmen tre gånger för varje datablock.
DH	Diffie–Hellman, en metod för att säkert utbyta kryptografiska nycklar över en publik kanal, baserat på den matematiska ”diskreta logaritmproblemet”.
DHE	DH Ephemeral, DH metoden men kortlivade nycklar, se punkten 11.1.2.
DSS	Digital Signature Standard, inkluderar DSA som signeringsalgoritm. [R15]
DSA	Digital Signature Algorithm är en ”Federal Information Processing Standard” för digitala signaturer. Se DSS
EC	Elliptic Curve (elliptiska kurvor), är en variant på publiknyckel kryptografi, ett slags asymmetrisk kryptering baserad på de matematiska egenskaperna hos elliptiska kurvor.
ECDH	DH med elliptiska kurvor.
ECDHE	EC, DH med Ephemeral dvs. kortlivade nycklar
ECDSA	Elliptic Curve Digital Signature Algorithm, erbjuder en variant av Digital Signature Algorithm (DSA) som använder kryptografi med elliptiska kurvor. (Beskrivs i dokument ANSI X9.62:2005 Public Key Cryptography)
EdDSA	Edwards-curve Digital Signature Algorithm (EdDSA)

Ephemeral	Kortlivad
ESP	Encapsulating Security Payload (IPSec) [R6a]
GCM	Galois/Counter Mode, “The operation is an Authenticated Encryption algorithm designed to provide both data authenticity (integrity) and confidentiality”, se AEAD
IDEA	International Data Encryption Algorithm, En symmetrisk blockkryptoalgoritm designat av James Massey och Xuejia Lai år 1991
IKEv2	Internet Key Exchange version 2 [R6]
IPsec	Internet Protocol Security, är kryptografiska transmissionsprotokoll som tillhandahåller kommunikationssäkerhet över datanätverk på nätverkslagret
ISAKMP	Internet Security Association and Key Management Protocol, kallas numera IKE (se ovan). Tidigare förekom även benämningen ISAKMP/Oakley.
MAC	Beskrivs i punkt 11.3.2.
MD5	Beskrivs i punkt 11.3.2.
NIST	National Institute for Standards and Technology
RC4	En symmetrisk blockkryptoalgoritm för designat av Ron Rivest in 1987
P-box	”Permutation-box”, är tillsammans med S-box en grundläggande komponent i symmetrisk kryptering som hanterar utbytet av indata med nyckel som ska dölja relationen mellan nyckeln och den krypterade texten.
PFS	Perfect Forward Secrecy. Beskrivs i punkt 11.2.
PRF	Pseudo-Random Function. The PRF is used for the construction of keying material for all of the cryptographic algorithms used in both the IKE SA and the Child SAs., se [R6] (IKE)
PSK	Pre-Shared Key, en, mellan kommunicerande parter, sedan tidigare delad nyckel.
RSA	Rivest Shamir Adleman (uppkallat efter upphovsmännen), är ett av det första kryptosystemet med publika och privat nyckel som används för säker datatransmission.
S-box	”Substitution-box”, är tillsammans med P-box en grundläggande komponent i symmetrisk kryptering som hanterar utbytet av indata med nyckel som ska dölja relationen mellan nyckeln och den krypterade texten.
SHA-1	Beskrivs i punkt 11.3.2.

SHA-2/SHA256	Beskrivs i punkt 11.3.2.
SRP	Secure Remote Password, är en lösenordbaserad autentiseringsfunktion med ett utökat nyckelförhandlingsprotokoll.
SSL/TLS	Secure Sockel Layer och Transport Layer Security är ett kryptografiskt transmissionsprotokoll som tillhandahåller kommunikationssäkerhet över datanätverk på transportlagret. [R9]
VPN	Virtuella Privata Nätverk (Virtual Private Network)

3. Bakgrund och syfte

Denna anvisning är en del av Ineras Riktlinje för informationssäkerhet [R1] och behandlar tekniska detaljer kring konfigurationen av krypterad kommunikation mha VPN (IPSec) eller TLS och kryptering av lagrade data s.k. filkryptering för tjänster som tillhandahålls av Inera.

Syftet är att beskriva de krav som Inera ställer på kryptering av lagrade data, IPSec och TLS-protokollet och tillhörande Cipher Suites för att upprätthålla en tillfredställande informationssäkerhet vid

- webbaserad klientaccess när en användare nyttjar en webbaserad e-hälsotjänst
- Web Services anrop mellan tjänster t.ex. när e-hälsotjänst kommunicerar via en Tjänsteplattform.
- site to sitekommunikation över ett VPN mellan e-hälsotjänster
- datalagring som, pga. informationssäkerhetsklassning, måste skyddas mot obehörig åtkomst.

Kompatibla inställningar mellan kommunicerande tjänster är en förutsättning för en säker och väl fungerande miljö där parter använder enhetlig, säker och etablerad krypteringsteknik.

3.1 Målgrupp

Målgrupper för denna anvisning är utvecklings- och förvaltningsteam för tjänster som tillhandahålls av Inera.

4. Ineras rekommendationer

Följande markeringar visar Ineras rekommendationer i detta dokument:

Grönt är godkänd nivå för transportkryptering och lagring.

Gult är acceptabel nivå för transportkryptering och lagring men det ska finnas en i tiden rimlig avvecklingsplan.

Rött är en icke acceptabel nivå för transportkryptering och lagring och ska Ej användas utan avvecklas snarast.

Lila är en ny och helt acceptabel nivå för transportkryptering och lagring **men** kan innebära kompatibilitetsproblem från kommunicerande tjänster och klienter som kanske ännu inte implementerat stöd för denna algoritm.

Målet är att alla tjänster ska avveckla icke acceptabla nivåer för transportkryptering dvs. de Rödmarkerade samt att tjänster tar fram en avvecklingsplan för Gulmarkerade nivåer för transportkryptering.

5. Inledning

När man väljer att använda transportkryptering i sin tjänst eller att lagrad data ska skyddas finns det vissa saker man bör tänka på. Denna anvisning syftar till att ge ramar och rekommendationer på området, men respektive tjänst måste ansvara för att inventera behov/möjligheter och att göra ett val som både har en acceptabel säkerhet och användarvänlighet.

5.1 Referensmaterial

För kryptografiska algoritmer och gällande nyckellängder: ”Transitioning the Use of Cryptographic Algorithms and Key Lengths, NIST Special Publication 800-131A Revision 2” [R4]

För nyckelhantering: “Recommendation for Key Management Part 1: General ”NIST Special Publication 800–57 Part 1, Revision 4” [R3] samt ”Ineras Instruktion för nyckelhantering till lagrade krypterade data” [R8]

5.2 Autentisering

Autentisering behandlas i sin helhet i Anvisning Autentisering [R2].

6. Nyheter inom krypteringsområdet

Detta kapitel behandlar nyheter gällande Cipher Suites, Cipher modes och kommande VPN protokoll, som är bra att känna till.

6.1 Krypteringsalgoritmer

En relativt ny krypteringsalgoritmen ChaCha20 [R12] har bubblat upp anses vara cirka tre gånger så snabb som AES i en ren mjukvaruimplementation på servrar utan behov av speciella AES kryptoaccelerator-kort. Google har till exempel valt ChaCha20 tillsammans med Poly1305 (Message Authentication Code) som en ersättare för RC4 i TLS, för sin Internetsäkerhet. Google har implementerat säker HTTPS-kommunikation mellan Chrome webbläsare på Androidmobiler och till Google’s webbsajter.

ChaCha20 finns nu med som en rekommenderad krypteringsalgoritm.

6.2 Dataintegritet och autenticitet

En ren kryptering av data skyddar mot insyn dvs. konfidentialitet men den skyddar inte mot att innehållet förvanskas. Bra informationssäkerhet kräver att både Konfidentialitet och Riktighet (Integritet) kan hanteras och att man kan avgöra äktheten (Autenticitet) i överförda data.

Inom säker transportkryptering används begreppet **AEAD** (Authenticated Encryption with Associated Data) [R13] som tillhandahåller att både konfidentialitet och riktighet säkerställs i en

skyddad överföring av data, se kapitel 11.3.2 om HMAC (Hashed Message Authentication Code).

GCM är, till skillnad mot CBC, en Cipher mode som stöder autentiserad kryptering, AE.

En nykomling inom HMAC är Poly1305 för dataautenticitet [R12b] som nu också finns med som en rekommenderad HMAC algoritm.

6.3 Nyckelutbyte (Key Exchange)

Curve25519 är en "Elliptic Curve" som erbjuder 128 bitars säkerhet och är designad att användas för elliptiska kurvor till Diffie–Hellman (ECDH) nyckelutbytesscheman. Det är en av de snabbaste ECC kurvorna och omfattas inte av några patent. Referensimplementationer är s.k. "public domain software". Curve25519 är en del av ECDHE and ECDSA och finns beskrivet tillsammans med TLS 1.2 [R10a].

6.4 WireGuard, en uppstickare inom VPN

WireGuard [R20] är en fri "open-source" mjukvara för applikationer och kommunikation som implementerar en ny VPN teknik som skapar en säker punkt till punktanslutning i routade eller bryggade konfigurationer. Den körs som en modul i Linux-kärnan och har som mål att erbjuda bättre prestanda och ska vara betydligt enklare att konfigurera än tunnelprotokollen IPSec och OpenVPN.
WireGuard använder Curve25519 för nyckelutbyte, ChaCha20 som kryptering, Poly1305 för dataautenticitet, SipHash för hash-tabellnycklar och BLAKE2s som kryptografisk hashfunktion. Den appliceras på lager 3 dvs. på nätverkslagret för både IPv4 och IPv6 och kan kapsla in v4-i-v6 och vice versa.
Som information kan även nämnas att Linus Torvald (skaparen av operativsystemskärnan Linux) tar med WireGuard VPN in i kommande version 5.6 av Linux kärnans "source tree".

7. Cipher Suites

Ett begrepp man bör känna, till när man jobbar med transportkryptering och i viss mån kryptering av lagrade data, är Cipher Suites. Cipher Suites är ett samlingsbegrepp för de komponenter som ingår när en krypterad session förhandlas inom IPSec och TLS. En Cipher Suite består av följande delkomponenter, där Ineras rekommendationer är Grönmarkerade.

En komplett lista över godkända och accepterade Cipher Suites för TLS och IPsec finns under kapitel 13 Appendix.

OBS Cipher Suites är i alla delar inte applicerbart på lagrade data OBS:

- Key Exchange Algorithm (**se kapitel 11.1**):
  - RSA med DHE eller ECDHE
  - DHE-DSS
  - ECDH-ECDSA
  - RSA med DH, ECDH eller SRP

- RSA, PSK
- Encryption Algorithm (**se kapitel 11.3**)
  - AES (AES128-GCM, AES256-GCM)
  - ChaCha20 (AEAD_CHACHA20_POLY1305)
  - AES (AES128-CBC, AES256-CBC), Camellia
  - RC4, Triple DES, IDEA, DES,
- Message Authentication Code (MAC) (**se kapitel 11.3.2**)
  - SHA-3
  - Poly1305
  - SHA-2
  - SHA-1
  - MD5

I praktiken fungerar valet av vilken Cipher Suite som ska användas så här:

1. Klienten eller den som initierar kommunikationen skickar en lista över vilka Cipher Suites den har stöd för i prioriteringsordning
2. Servern/eller mottagande part väljer en av dessa, alternativt nekar anslutningen.

Rekommendationen för att öka säkerheten är således att man begränsar servern till att bara tillåta ett visst urval av Cipher Suites för att undvika klienter som inte har stöd för eller som vill göra en förhandling som innebär dålig eller icke accepterad säkerhetsnivå.

8. Protokoll för transportkryptering med TLS

Utöver Cipher Suites behöver man för transportkryptering med TLS också välja vilket/vilka protokollversioner som e-tjänsten ska stödja. Valet av protokoll påverkar också vilka Cipher Suites som stöds.

8.1 Rekommenderade protokoll för transportkryptering med TLS

- **TLS 1.3** – Version 1.3 är from augusti 2018 godkänd av IETF standard [R9]. Många Webbläsare har redan idag, i senaste versioner, stöd för TLS 1.3 men alla kommunicerande tjänster och klienter kanske ännu inte implementerat stöd för detta protokoll. Se punkten Informativt om TLS 1.3.
- **TLS 1.2** – [R10] Aktiverat som standard i de senaste versionerna av alla webbläsare, kan manuellt aktiveras på vissa äldre OS/Webbläsare, se punkt 8.2, men är dock aktiverat som standard i de versioner som stöds inom eKlient i Samverkan
- **TLS 1.1** – Rekommenderas INTE, men kan i enstaka fall behöva aktiveras och bibehållas för bakåtkompatibilitet.
- **TLS 1.0** – Rekommenderas INTE, men kan i enstaka fall behöva aktiveras och bibehållas för bakåtkompatibilitet.
- **SSL 3.0** – Rekommenderas INTE. Vid en ev. användning skall man vara extra noga att klienter och server är patchade mot kända attacker och svagheter. Relativt liten skillnad mot TLS 1.0.

- **SSL 2.0** – Rekommenderas INTE, då den har säkerhetsbrister som inte kan åtgärdas genom val av Cipher Suite och fixar på klienter.

Av ovanstående protokoll rekommenderar vi att endast TLS 1.2 är aktiverat om man inte har kontroll på sin klientmiljö. Tjänster ska avveckla TLS 1.0 och TLS 1.1.

Kan man ställa krav på klienterna är rekommendationen alltid att stänga av så många äldre protokoll som möjligt.

8.1.1 Informativt om TLS 1.3

TLS 1.3 är nu en godkänd RFC 8446 och det innebär fler väsentliga säkerhetshöjande egenskaper i protokollet:

- Stöder kryptering som simultant hanterar konfidentialitet, riktighet (integritet) och autenticitet av innehållet s.k. Authenticated Encryption with Associated Data (AEAD).
- Stöder endast 5 Cipher Suites, se kap. 13.2.1
- Stöd för Elliptiska kurvor redan i basstandarden.
- Stöder endast 5 ECDHE grupper och 5 DHE grupper för nyckelutbyte
- Stöder även för PSK och PSK med (EC)DHE för nyckelutbyte
- Autentisering stöds endast med RSA, ECDSA och EdDSA
- Optimerad handskakning för att minimera antal initiala paket för att etablera en TLS session.
- En ny optimeringsteknik för återupptagna anslutningar kallad 0-RTT som medför att färre initiala paket behöver sändas mellan klient och server för att etablera en TLS session.
- Krypteringsalgoritmer baserad på CBC stöds alltså ej längre
- Enbart RSA för nyckelutbyte stöds ej längre.
- Krypteringsalgoritmerna RC4, 3DES och Camellia samt hashingalgoritmerna MD5 och SHA1 stöds ej längre
- Svaga DH (Diffie-Hellman) grupper stöds ej längre.
- Alla publika nyckelchiffer ska stödja PFS.

8.2 OS/Browser stöd för olika krypteringsprotokoll (Wikipedia)

https://en.wikipedia.org/wiki/Transport_Layer_Security#Web_browsers

Med reservation för att länken kan ha flyttats. Sök på TLS och Web browsers.

9. Protokoll för transportkryptering med IKE/IPsec

Protokollet beskriv mer ingående under punkten Översiktlig beskrivning av IKE/IPsec, nedan.

9.1 IKE, Internet Key Exchange

IKE protokollet finns i två versioner, IKEv1 och IKEv2 [R6]. IKEv2 protokollet har genomgått ett antal säkerhetsförbättringar som förklaras i "IKEv2 Clarifications and Implementation Guidelines" [R6c].

9.1.1 Rekommenderat IKE protokoll:

- IKEv2 – Den rekommenderade versionen.
- IKEv1 – Main mode. Kan användas där kompatibilitet krävs mot utrustning som inte stöder IKEv2
- IKE v1 – Aggressive mode. Sårbarheter är identifierade som gör att Aggressive mode ska undvikas

9.1.2 Rekommenderad autentiseringsmetod

Det finns ett flertal autentiseringsmetoder beskrivna för IKEv2 men i huvudsak används tre olika sätt:

1. RSA och DSS, digitala signaturer, certifikatsbaserad autentisering.
2. EAP, Extensible Authentication Protocol, som i första hand används för att autentisera IPsec baserade fjärranklienter mot en autentiseringsserver typ Radius eller AD.
3. Shared key (Preshared key), en, mellan kommunicerande parter, delad nyckel.

Rekommenderad autentiseringsmetod är den certifikatsbaserade med t.ex. SITHS.

OBS. Pre-Shared key ska endast användas i lab- eller tidiga testmiljöer och får inte förekomma i en produktionsmiljö då den relativt enkelt kan delas och användas av obehöriga.

9.1.3 Rekommenderade krypteringsalgoritmer

Krypterings- och nyckelutbytesalgoritmer för IPsec baserad VPN ska följa den rekommendation som finns i kapitel 11 Parametrar för transportkryptering.

9.1.4 Rekommenderad Key Lifetime

Rekommenderad Key Lifetime för en IKE SA:

8 timmar

9.1.5 Informativt om Key Lifetime

En väsentlig skillnad mellan TLS baserad transportkryptering och IPsec baserad är att man med IPsec och IKE på ett tydligt sätt kan sätta livslängden på en session (Key Lifetime) innan nya nycklar ska omförhandlas. Nycklar ska omförhandlas med jämna mellanrum beroende på hur mycket information som skickas mellan två parter. Möjligheten för en attackerare, som kan avlyssna och spela in en krypterad session, att dekryptera en skyddad och krypterad session ökar betydligt med mängden data som skickats med samma krypteringsnyckel.

En relativt gammal Best practice från NIST ”Guide to IPsec VPNs” [R5] är att IKE SA Key Lifetime aldrig ska överstiga 24 timmar (86 400 sekunder) men bör sättas kortare om stora datamängder skickas över IPsec SA som gör att dessa nycklar kommer att omförhandlas ofta.

9.2 IPsec

9.2.1 Rekommenderat IPsec protokoll

IPsec ”Phase 2” tunnlar finns i två protokollvarianter ESP och AH, (se Förkortningar kap 1) och i två olika moder, Transport och Tunnel mode.

ESP - Här krypteras och signeras hela original IP paketet in i ett nytt IP paket.

AH – Endast signering av IPsec paketet dvs. ingen kryptering av original IP paketet (payload).

Tunnel mode – Tillsammans med ESP krypteras hela originalpaketet och den inre IP adressen (läs IP header) döljs med VPN gatewayens externa IP adress.

Transport mode – Originalpaketets IP adress (läs IP header) speglas ut till den yttre IPsec headern.

För att erhålla full konfidentialitet, dvs. att dölja hela trafiken mellan två kommunicerande parter, krävs ESP i Tunnel mode.

9.2.2 Rekommenderade krypteringsalgoritmer

Krypterings- och nyckelutbytesalgoritmer för IPsec baserad VPN ska följa den rekommendation som finns i kapitel 11 Parametrar för transportkryptering.

9.2.3 Rekommenderad Key Lifetime

Rekommenderad Key Lifetime för en IPsec SA:

1 timma (alternativt kan Key Lifetime sättas utifrån överförd datamängd t.ex. 100 MB).

Best practice från NIST ”Guide to IPsec VPNs” [R5] säger att en IPsec SA Key Lifetime aldrig ska överstiga 8 timmar (28 880 sekunder) men bör sättas kortare om stora datamängder skickas över IPsec SA. IPsec Key Lifetime bör kunna sätta baserat på både tid och datamängd.

9.3 Informativt om IKE/IPsec

Internet Protocol Security, IPSec är en protokollsvit för att i nätverk säkra kommunikationen av paket som skickas över ett IPv4 eller IPv6 nätverk.

IPsec tillsammans med IKE protokollen är framtagna för att på ett säkert och tillfredsställande sätt hantera långfristiga VPN förbindelser över publika nätverk.

Ett VPN byggs normalt sett upp i två typer av topologier, ”Hub and Spoke” eller ett ”Meshat” nät. I en Hub and Spoke topologi kommunicerar alla ingående VPN noder genom en central Hub till skillnad mot en ”Meshat” topologi där alla VPN noder kan prata direkt med varandra.

IPSec består, oftast men inte alltid, av två olika tunnelprotokoll dvs. ”Phase 1”, som är den initiala fasen för autentisering och nyckelutbyte mellan två kommunicerande parter mha. protokollet Internet Key Exchange, IKE (tidigare kallad ISAKMP). Efter godkänd autentisering, förhandling om krypteringsalgoritmer och nyckelutbyte etableras en egen ”Phase 1” dubbelriktad VPN tunnel som identifieras med en s.k. Security Association, SA.

Själva datatransporten mellan två kommunicerande parter sker i de tunnlar som förhandlas fram för ”Phase 2” där själva IPsec protokollet, AH eller ESP arbetar (två tunnlar, en i vardera riktningen), Förhandlingen för ”Phase 2”, mellan kommunicerande parter, sker i IKE tunneln. För ”Phase 2” sker en ny förhandling av krypteringsalgoritmer och nyckelutbyte.

Nyckelförhandlingen kan också, precis som med TLS protokollet, förstärkas med att förhandlingen av nya nycklar sker oberoende av tidigare nycklar, se rubriken nedan Perfect Forward Secrecy, PFS, för att minimera eventuella möjligheter, för attackerare som kan avlyssna en kommunikation, att kunna dekryptera inspelade data.

IPSec används i första hand för site- to sitekommunikation där trafik, mellan två eller flera kommunicerande servermiljöer, paketeras i krypterade tunnlar mellan ingående VPN Gateways.

I de flesta implementationer av IPsec hanteras ”Phase 1” av en IKE demon som hanterar autentisering, nyckelutbyte och som initierar IPsec tunnlarna. I de vanligaste Linux distributionerna är IPsec en del av kärnan och där kan man med IP transformer policy sätta upp en VPN policy som etablerar en VPN tunnel utan att blanda in en IKE demon men det underlättar betydligt att föra in IKE för ”Phase 1”. Hur detta går till är utanför scopet för den här anvisningen.

För transporten av data mellan kommunicerande parter hanterar IPsec följande:

- Konfidentialitet –En IPSec-avsändaren kan kryptera packet innan det skickas ut på ett öppet nätverk.
- Integritet—En IPsec-mottagare verifierar ett paket från en IPsec avsändare och försäkrar att paketet inte har förvanskats under en överföring.
- Originalitet (Autenticitet) –En IPsec-mottagare kan verifiera källan av ett skickat IPsec packet. OBS Autenticitet i IPsec protokollet är att verifiera ett pakets källa. Det ska inte likställas med att autentisera en kommunicerande part som sker i t.ex. IKE protokollet.
- Återuppspelningsskydd – En IPsec-mottagare kan detektera och förhindra att ett paket mottaget paket bearbetas igen.

10. Kryptering av lagrad information (filkryptering)

10.1.1 Rekommenderade krypteringsalgoritmer

Krypteringsalgoritmer för lagrad information och filer följa relevanta delar av den rekommendation som finns i kapitel 11 Parametrar för transportkryptering.

Rekommenderade krypteringsalgoritmer för att skydda lagrade data är:

AES 256, AES algoritmen med 256 bitars krypteringsnyckel

AES 128, AES algoritmen med 128 bitars krypteringsnyckel

10.2 Informativt om kryptoalgoritmens nyckellängd

I t.ex. AES 256 använder krypteringsalgoritmen en 256 bitars nyckel i S-boxen men det är också lösenordets (se citat nedan) kvalitet som sedan avgör säkerheten. Kryptonycklar för lagrad information måste på motsvarande sätt som i trafikkryptering, genereras slumpmässigt. Kryptonycklar som används för kryptering av lagrad information hanteras inte direkt av användarens lösenord, utan krypteringsapplikationen kan använda ett flertal slumpmässigt genererade nycklar som sparas i ett nyckellager som måste krypteras med ett användarlösenord eller så skapas själva krypteringsnyckeln genom att användarlösenordet hashas i flera steg t.ex. med SHA-256 så som t.ex. 7-Zip fungerar.

Ett annat rekommenderat sätt är att använda en asymmetrisk krypteringsalgoritm som t.ex. RSA och kryptera kryptonycklarna med sin egen eller mottagarens publika nyckel.

“Need for secrecy
In designing security systems, it is wise to assume that the details of the cryptographic algorithm are already available to the attacker. This is known as Kerckhoffs' principle — "only secrecy of the key provides security", or, reformulated as Shannon's maxim, "the enemy knows the system".”
(Källa: www.wikepedia.org)

10.3 Lösenord för skydd av nyckellager och krypterade filer

10.3.1 Skydd av krypteringsnycklar

Hänvisning till Instruktion för nyckelhantering till lagrade krypterade data [R8]

10.3.2 Rekommenderad lösenordslängd

Ett lösenord ska vara minst 16 tecken långt med komplexitetskrav.

16 tecken x 8 bitar = 128 bitar (minst, se nedan) som i så fall står i paritet med krypteringsalgoritmens nyckellängd.

Med Unicode-baserad teckentabell representeras varje tecken av en till tre bytes, således blir, beroende på valda tecken, lösenordet med 16 tecken minst 128 bitar.

10.4 Informativt om lösenord för skydd av lagrade data

Om man inte använder RSA för att skydda sitt nyckellager behöver man antingen använda en symmetrisk krypteringsalgoritm för att kryptera nyckellager eller som med 7-Zip att skapa krypteringsnyckel från lösenordet med en hash-rutin. Ett lösenord kan attackeras med en Brute-force attack [R7] vilket då ställer krav på ett lösenord av tillräckligt hög kvalitet som står i paritet med använd en krypteringsalgoritm som t.ex. AES 256. AES-256 utesluter en Brute-force attack direkt mot den krypterade informationen.

“Key sizes

For the one-time pad system, the key must be at least as long as the message. In encryption systems that use a cipher algorithm, messages can be much longer than the key. The key must, however, be long enough so that an attacker cannot try all possible combinations.
A key length of 80 bits is generally considered the minimum for strong security with symmetric encryption algorithms. 128-bit keys are commonly used and considered very strong."
(Källa: www.wikepedia.org)

11. Parametrar för transportkryptering

11.1 Rekommenderade algoritmer för nyckelutbyte (Key Exchange)

Rekommenderade kombinationer av algoritmer för nyckelutbyte listas nedan i prioriteringsordning:

Med stöd för RSA:

ECDHE-RSA, DHE-RSA

Övriga alternativ ställer vissa krav på servercertifikatet som idag inte kan uppfyllas av SITHS.

ECDHE-ECDSA, **DHE-DSS**

ECDH-ECDSA, **DH-DSS, SRP**

NIST (National Institute of Standards and Technology) har tagit fram en standard för signaturer DSS (Digital Signature Standard) DSA (Digital Standard Algorithm) [R15].

11.1.1 Rekommenderade DH grupper för nyckelutbyte

Rekommendationen är att man använder dem enligt följande i prioriteringsordning [R17]:

Cipher_Suite_P521-bit, DH group 21, Elliptic Curve Groups (ECP groups) algorithm

Cipher_Suite_P384-bit, DH group 20, (ECP groups) algorithm

Cipher_Suite_P256-bit, DH group 19, (ECP groups) algorithm

Cipher_Suite_4096-bit, DH group 16, Modular Exponential (MODP) algorithm

Cipher_Suite_3072-bit, DH group 15, (MODP) algorithm

Cipher_Suite_2048-bit, DH group 14, (MODP) algorithm

Cipher_Suite_1536-bit, DH group 5, (MODP) algorithm

Om elliptiska kurvor (EC) inte kan användas, är vår starka rekommendation att generera nya egna Diffie-Hellman grupper (MODP) med minst 2048-bits gruppstorlek, se Not.

Not. Att generera en ny Diffie-Hellman grupp oavsett serverprogramvara som används. Inera rekommenderar att minst en 2048-bit grupp genereras. Enklaste sättet är att generera en ny grupp är genom Openssl tool.

```
"openssl dhparam -out dhparams.pem 2048"
```

11.1.2 Rekommenderade algoritmer för PFS (ephemerala)

De algoritmer för nyckelutbyte som stöder PFS är de som använder så kalla ephemerala (kortlivade) Diffie-Hellman nycklar som:
ECDHE-RSA, DHE-RSA, för RSA (som bl.a. innefattar SITHS):
ECDHE-ECDSA, DHE-DSS, för DSA/DSS

11.2 Informativt om Perfect Forward Secrecy

Valet av Key Exchange algoritm påverkar även möjligheten att ha stöd för något som kallas för **Forward Secrecy** eller Perfect Forward Secrecy**, PFS.**

PFS **(**även kallat **Ephemeral mode)** bildar tillsammans med ECDH (Eliptic Curve Diffie-Hellman) ECDHE.

Enkelt beskrivet så är fördelen med denna funktionalitet att varje session mellan server och klient krypteras med egna unika nycklar som är oberoende av de certifikat som servern och klientens använde vid nyckelutbytet. Nycklarna sparas inte och används endast till just denna session och kastas sedan. Rent säkerhetsmässigt innebär detta att även om en attackerare sitter och sparar sessioner under 1 år och så småningom kommer över serverns privata nycklar, så kommer dessa sessioner inte kunna avkrypteras med hjälp av dessa privata nycklar.

OBS DSA och ECDSA har en svaghet/sårbarhet i att de kräver ett nytt slumptal för varje signeringstillfälle, annars kan signaturen röja den privata nyckeln. Det finns förutsägbara implementationer för digitala signaturer som inte har ordentliga krav på slumptalshantering för signering.

11.2.1 Standardgrupper för PFS

Om man använder en Key Exchange Algoritm som stödjer PFS finns det också ett antal standarder för grupper och nyckellängd. Dessa brukar anges med ett E (Ephemeral **)** som tillägg bakom vald Cipher Suite och baseras på den nyckellängd som används i handskakningen av Diffie-Hellman med antingen elliptiska kurvor (ECDH**E**) eller modulo (DH**E**) inom själva TLS- eller IKE/IPsec-sessionen.

11.2.2 Nackdelar med PFS

- PFS skyddar inte mot metoder som försöker avkryptera meddelanden utan nyckeln t.ex. Brute Force
- PFS ställer något högre krav på systemresurser hos klient/server, bör vara försumbart på nyare system.

11.3 Krypteringsalgoritmer

Valet av krypteringsalgoritm eller chiffer påverkar till stor del säkerheten i vald Cipher Suite. Inera rekommenderar att man använder någon av de symmetriska krypteringsalgoritmerna: AES256, AES128 tillsammans med GCM eller med stor tvekan CBC. GCM är idag den

rekommenderade blockkrypteringsformen (Block Cipher mode se [R13]). För TLS krävs att man använder TLS 1.2 alternativt TLS 1.3.

Motsvande val av krypteringsalgoritm som ovan gäller även för IKE/IPsec.

TLS 1.0/1.1, oavsett ”Block Cipher” mode ska helt undvikas.

Referenslitteratur för val av krypteringsalgoritmer FMV, 188 Scheme Crypto Policy [R16] samt BlueKrypt, Cryptographic Key Length Recommendation [R17].

11.3.1 Rekommenderade Cipher Suites för TLS och IPsec

Godkända och accepterade Cipher Suites finns under Appendix 13 nedan

11.3.2 Informativt om Hashed Message Authentication Code, HMAC

HMAC används för att verifiera integritet och autenticitet av ett meddelande som indikerar för mottagande part att meddelandet levererats intakt, alltså inte har manipulerats på vägen från sändare till mottagare. Den, inom kryptografin, rekommenderade hashingalgoritmen som används för att kalkylera en HMAC är SHA (Secure Hash Algorithm). En stark rekommendation är att man använder **minst** SHA-2 256. SHA-1 (160 bitars) anses, i takt med allt högra beräkningskapacitet i moderna datorer, vara sårbar för ”kollisionsattacker” där man kan kalkylera fram ett nytt identiskt meddelande för en viss given checksumma. Hashfunktionen MD5 (Message Digest 5) anses vara kompromitterad och ska inte användas.

SHA är en grupp av hashingalgoritmer (SHA-1, SHA-2, SHA-3) som ofta anges med ett tillägg om hur många bitar som används per skapad ”hash” som t.ex. SHA-512 som egentligen ingår i SHA-2. Generellt gäller att ju nyare grupp av SHA och ju större antal bitar som används desto säkrare. Dock ökar också prestandakraven på de enheter som ska utföra beräkningar därefter.

National Institute of Standards and Technology (NIST) släppte 2015 Secure Hash Algorithm-3 (SHA-3) [R14]. När den här anvisning av version 3.2 uppdaterades, fanns ännu inga tydligt beskrivna standards eller rekommendationer, gällande Cipher Suites, där SHA-3 ingår. Således avses att följa upp SHA-3 i kommande version.

En nykomling som har tagit plats som en vedertagen HMAC är Poly1305 som, tillsammans med den relativt nya krypteringsalgoritmen ChaCha20, också definierats som IETF standarden [R12]

12. Referenslista

OBS (alla referenser nedan kan vara föremål för uppdateringar)

Ref	Dokumentnamn	Länk till dokument
R1	Riktlinje för informationssäkerhet (Gäller inom Inera AB)	https://intra.inera.se/Vart-regelverk/Informationssakerhet/Riktlinjer-informationssakerhet/ OBS intern Inera-länk
R2	Anvisning Autentisering	https://intra.inera.se/Vart-regelverk/Informationssakerhet/Riktlinjer-informationssakerhet/ OBS intern Inera-länk Alternativt RIV Tekniska Anvisningar Säkerhet: http://rivta.se/documents.html
R3	NIST SP 800–57 Rev. 4, Recommendation for Key Management	https://dx.doi.org/10.6028/NIST.SP.800-57pt1r4
R4	NIST.SP.800-131Ar2, Transitioning the Use of Cryptographic Algorithms and Key Lengths	https://csrc.nist.gov/publications/detail/sp/800-131a/rev-2/final
R5	NIST, Guide to IPsec VPNs	https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-77.pdf
R6	Internet Key Exchange Protocol Version 2 (IKEv2)	https://tools.ietf.org/html/rfc7296 Updated by: rfc7427, 7670, 8247
R6a	IP Encapsulating Security Payload (ESP)	https://tools.ietf.org/html/rfc4303
R6b	IP Authentication Header (AH)	https://tools.ietf.org/html/rfc4302
R6c	IKEv2 Clarifications and Implementation Guidelines	http://www.rfc-editor.org/info/rfc4718 (Informativ RFC)
R7	Brute-force attack	https://en.wikipedia.org/wiki/Brute-force_attack
R8	Instruktion, nyckelhantering för lagrade krypterade data	https://rivta.se/documents
R9	The Transport Layer Security (TLS) Protocol Version 1.3	https://tools.ietf.org/html/rfc8446

R10	The Transport Layer Security (TLS) Protocol Version 1.2	https://tools.ietf.org/html/rfc5246 (Updated by: 5746, 5878, 6176, 7465, 7507, 7568, 7627, 7685, 7905, 7919, 8447)
R10a	Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS) Versions 1.2 and Earlier	https://tools.ietf.org/html/rfc8422
R11	TLS 1.3 Cipher Suites	https://tools.ietf.org/html/rfc8446#appendix-B.4
R12	ChaCha20-Poly1305 Cipher Suites for Transport Layer Security (TLS)	https://tools.ietf.org/html/rfc7905
R12a	ChaCha20 and Poly1305 for IETF Protocols	https://tools.ietf.org/html/rfc8439
R12b	Poly1305	https://en.m.wikipedia.org/wiki/Poly1305
R13	An Interface and Algorithms for Authenticated Encryption	https://tools.ietf.org/html/rfc5116 https://en.wikipedia.org/wiki/Authenticated_encryption Se också ”Block cipher mode of operation”, https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation
R14	NIST SHA-3 Secure Hash Algorithm-3	https://www.nist.gov/publications/sha-3-standard-permutation-based-hash-and-extendable-output-functions https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf
R15	FIPS PUB 186–4 Digital Signature Standard (DSS)	https://dx.doi.org/10.6028/NIST.FIPS.186-4
R16	FMV, 188 Scheme Crypto Policy Issue: 7.0 ,2017-04-04	https://www.fmv.se/Global/Dokument/Verksamhet/CSEC/188%20Scheme%20Crypto%20Policy%20Issue%202.0,%202012-09-27.pdf
R17	BlueKrypt Cryptographic Key Length Recommendation	https://www.keylength.com/
R18	Advanced Encryption Standard (AES)	http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf
R19	The Poly1305-AES message-authentication code	http://cr.yp.to/mac/poly1305-20050329.pdf

R20	WireGuard, fast modern, secure VPN Tunnel	https://www.wireguard.com/ https://en.m.wikipedia.org/wiki/WireGuard

13. Appendix

13.1 Grundkonfiguration av webbservrar

13.1.1 Apache - Cipher Suite

Cipher Suites för Apache HTTP server konfigureras enligt de rekommendationer som finns i dokumentationen på Apache hemsida: http://httpd.apache.org/

Under menyvalet **Documentation** och "**Version xx**":
Välj under rubriken **Users Guide** "**SSL/TLS Encryption**",
Välj under rubriken **Documentaton** "**mod_ssl Configuration How-To**".

I avsnittet "**SSL/TLS Strong Encryption: How-To**" finns specifikt "**Cipher Suites and Enforcing Strong Security**"

OBS! Att inställningarna i exemplet ovan inte nödvändigtvis ger de nivåer som rekommenderas i detta dokument dvs. punkt 13.1.3 **Övriga inställningar för webbservrar** måste också följas.

13.1.2 Internet Information Services - Cipher Suite

Cipher Suites för Microsoft IIS konfigureras enligt de rekommendationer som finns i på sajten https://www.ssl.com/howto/
Sök upp "**Require Strong Ciphers**"
Välj " **Require Strong Ciphers in Windows IIS 7.5 and 8**"

OBS! Att inställningarna i exemplet ovan kanske inte nödvändigtvis ger de nivåer vi rekommenderar i detta dokument dvs. punkt 13.1.3 **Övriga inställningar för webbservrar** måste också följas.

13.1.3 Övriga inställningar för webbservrar

Efter ovanstående grundkonfiguration för respektive Apache och IIS servrar skall följande anpassningar göras:

13.1.4 Allmänt

- Använd alltid senaste versionen av OS/Servermjukvarans TLS-bibliotek Se till att kontinuerligt uppdatera både operativsystem och de komponenter som används av webbservern.
- Avaktivera SSL och osäkra TLS dvs. v1.0/1.1 protokollen enligt punkt 8.1
- Avaktivera Client-Initiated Renegotiation - mot DoS attacker.
- Avaktivera TLS-kompression - mot CRIME och TIME-attacker.

- Avaktivera TLS_RSA mot DROWN och ROBOT
- Avaktivera HTTP-kompression, eller vidta åtgärder mot CSRF - mot BREACH-attacker
- Avaktivera **alltid** cachning av HTTP-respons - motverkar att information lagras lokalt på klienten som kan bli tillgänglig efter utloggning.
- Aktivera HTTP Strict Transport Security - tillåt Inte ”mixed-content” på en webbsida dvs. att man ska köra hela sessionen över TLS när man använder TLS och inte blanda krypterad och okrypterad data
- Aktivera Forward Secrecy (PFS) så att ECDHE och DHE används vid förhandling av sessionsnycklar. Sker automatiskt på de flesta webbservrar vid rätt val av Cipher Suite - se rubrik 11.2 (Perfect) Forward Secrecy och nästa punkt nedan.
- Avaktivera DHE_EXPORT - tillåt inte svaga Cipher Suites som bl.a. innehåller 512 bits Diffie-Hellman grupper som anses vara komprometterad - mot Logjam attacker.
- Avaktivera TLS_FALLBACK_SCSV för att säkerställa att ingen omförhandling av kommunikationen till att lägre kryptoprotokollversioner tillåts - mot POODLE
- Maximal TLS-sessionslängd utan omförhandling är 12h.
- Maximal inaktiv TLS-sessionslängd är 35 minuter

13.2 Godkända och accepterade Cipher Suites för TLS protokollen

Presenteras i prioriteringsordning som de ska föredras av servrar för tjänster inom Inera

OBS TLS 1.0/1.1 är nedgraderat till Röd nivå och avveckling ska planeras snarast OBS

13.2.1 Rekommenderade Cipher Suites i TLS 1.3 (med AEAD)

Även om TLS 1.3 [R11] använder samma ”Cipher Suite” rymd som tidigare versioner av TLS så definieras TLS 1.3 annorlunda, dvs. endast det symmetriska kryptot specificeras, och kan **inte** användas för TLS 1.2. På liknande sätt kan ”Cipher Suites” definierade för TLS 1.2 **inte** användas för TLS 1.3.

1. TLS_AES_256_GCM_SHA384
2. TLS_AES_128_GCM_SHA256
3. TLS_AES_128_CCM_SHA256
4. TLS_AES_128_CCM_8_SHA256
5. TLS_CHACHA20_POLY1305_SHA256

13.2.2 Rekommenderade Cipher Suites i TLS 1.2 (i prioriteringsordning)

AEAD_CHACHA20_POLY1305 (GCM)

6. TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
7. TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256
8. TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256

AES, SHA-2 och GCM med PFS (RSA/DSA/DSS)

9. TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (SHA-2)
10. TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (SHA-2)
11. TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (SHA-2)
12. TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (SHA-2)
13. TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (SHA-2)

14. TLS_DHE_DSS_WITH_AES_256_GCM_SHA384 (SHA-2)
15. TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (SHA-2)
16. TLS_DHE_DSS_WITH_AES_128_GCM_SHA256 (SHA-2)

AES, SHA-2 och GCM utan PFS (RSA/DSA/DSS)

17. TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384 (SHA-2)
18. TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384 (SHA-2)
19. TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256 (SHA-2)
20. TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256 (SHA-2)
21. TLS_DH_RSA_WITH_AES_256_GCM_SHA384 (SHA-2)
22. TLS_DH_DSS_WITH_AES_256_GCM_SHA384 (SHA-2)
23. TLS_DH_RSA_WITH_AES_128_GCM_SHA256 (SHA-2)
24. TLS_DH_DSS_WITH_AES_128_GCM_SHA256 (SHA-2)

AES, SHA-2 och CBC med PFS (RSA/DSA/DSS)

25. TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (SHA-2)
26. TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (SHA-2)
27. TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (SHA-2)
28. TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (SHA-2)
29. TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (SHA-2)
30. TLS_DHE_DSS_WITH_AES_256_CBC_SHA256 (SHA-2)
31. TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (SHA-2)
32. TLS_DHE_DSS_WITH_AES_128_CBC_SHA256 (SHA-2)

AES, SHA-2 och CBC utan PFS (RSA/DSA/DSS)

33. TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384 (SHA-2)
34. TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384 (SHA-2)
35. TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256 (SHA-2)
36. TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256 (SHA-2)
37. TLS_DH_RSA_WITH_AES_256_CBC_SHA256 (SHA-2)
38. TLS_DH_DSS_WITH_AES_256_CBC_SHA256 (SHA-2)
39. TLS_DH_RSA_WITH_AES_128_CBC_SHA256 (SHA-2)
40. TLS_DH_DSS_WITH_AES_128_CBC_SHA256 (SHA-2)

AES och SHA-1 med PFS (RSA/DSA/DSS)

41. TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (SHA-1)
42. TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (SHA-1)
43. TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (SHA-1)
44. TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (SHA-1)
45. TLS_DHE_DSS_WITH_AES_256_CBC_SHA (SHA-1)
46. TLS_DHE_RSA_WITH_AES_256_CBC_SHA (SHA-1)
47. TLS_DHE_RSA_WITH_AES_128_CBC_SHA (SHA-1)
48. TLS_DHE_RSA_WITH_AES_128_CBC_SHA (SHA-1)
49. TLS_DHE_DSS_WITH_AES_128_CBC_SHA (SHA-1)

AES och SHA-1 utan PFS (RSA/DSA/DSS)

50. TLS_ECDH_RSA_WITH_AES_256_CBC_SHA (SHA-1)
51. TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA (SHA-1)
52. TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA (SHA-1)
53. TLS_ECDH_RSA_WITH_AES_128_CBC_SHA (SHA-1)
54. TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA (SHA-1)
55. TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA (SHA-1)

56. TLS_DH_RSA_WITH_AES_256_CBC_SHA (SHA-1)
57. TLS_DH_DSS_WITH_AES_256_CBC_SHA (SHA-1)
58. TLS_DH_RSA_WITH_AES_128_CBC_SHA (SHA-1)
59. TLS_DH_DSS_WITH_AES_128_CBC_SHA (SHA-1)

CAMELLIA och SHA-1 med PFS (RSA/DSS)

60. TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (SHA-1)
61. TLS_DHE_DSS_WITH_CAMELLIA_256_CBC_SHA (SHA-1)
62. TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (SHA-1)
63. TLS_DHE_DSS_WITH_CAMELLIA_128_CBC_SHA (SHA-1)

CAMELLIA och SHA-1 utan PFS (RSA/DSS)

64. TLS_DH_RSA_WITH_CAMELLIA_256_CBC_SHA (SHA-1)
65. TLS_DH_DSS_WITH_CAMELLIA_256_CBC_SHA (SHA-1)
66. TLS_DH_RSA_WITH_CAMELLIA_128_CBC_SHA (SHA-1)
67. TLS_DH_DSS_WITH_CAMELLIA_128_CBC_SHA (SHA-1)

AES och SHA-1 utan PFS (SRP)

68. TLS_SRP_SHA_DSS_WITH_AES_256_CBC_SHA (SHA-1)
69. TLS_SRP_SHA_RSA_WITH_AES_256_CBC_SHA (SHA-1)
70. TLS_SRP_SHA_RSA_WITH_AES_128_CBC_SHA (SHA-1)
71. TLS_SRP_SHA_DSS_WITH_AES_128_CBC_SHA (SHA-1)

13.2.3 TLS 1.0 och 1.1

TLS 1.0 och TLS 1.1 är icke acceptabla transportprotokoll.
Alla varianter av TLS 1.0 och 1,1 ska helt undvikas.

13.3 Godkända och accepterade Cipher Suites för IKE och IPsec protokollen

Presenteras i den prioriteringsordning som föredras för servrar och tjänster inom Inera.

13.3.1 Rekommenderade Cipher Suites i IKE och IPsec

Rekommenderas i första hand:

AES256GCM16-PRFSHA384-ECP384, (AES-GCM-256 AEAD, SHA-384 as PRF and ECDH) key exchange with 384-bit key length)
AES128GCM16-PRFSHA256-ECP256, (AES-GCM-128 AEAD, SHA-256 as PRF and ECDH) key exchange with 256-bit key length)

13.3.2 Krypterings-, hashing- och nyckelutbytesalgoritmer

Övriga rekommenderade krypterings-, hashing- och nyckelutbytesalgoritmer grupperade i prioritetsordning.

Encryption Algorithms

Keyword	Description

1. chacha20poly1305	256-bit ChaCha20/Poly1305 with 128-bit ICV
2. aes256gcm8 or aes256gcm64	256-bit AES-GCM with 64-bit ICV
3. aes256gcm12 or aes256gcm96	256-bit AES-GCM with 96-bit ICV
4. aes256gcm16 or aes256gcm128	256-bit AES-GCM with 128-bit ICV
5. aes256ccm8 or aes256ccm64	256-bit AES-CCM with 64-bit ICV
6. aes256ccm12 or aes256ccm96	256-bit AES-CCM with 96-bit ICV
7. aes256ccm16 or aes256ccm128	256-bit AES-CCM with 128-bit ICV
8. aes192gcm8 or aes192gcm64	192-bit AES-GCM with 64-bit ICV
9. aes192gcm12 or aes192gcm96	192-bit AES-GCM with 96-bit ICV
10. aes192gcm16 or aes192gcm128	192-bit AES-GCM with 128-bit ICV
11. aes192ccm8 or aes192ccm64	192-bit AES-CCM with 64-bit ICV
12. aes192ccm16 or aes192ccm128	192-bit AES-CCM with 128 bit ICV
13. aes192ccm12 or aes192ccm96	192-bit AES-CCM with 96-bit ICV
14. aes128gcm16 or aes128gcm128	128-bit AES-GCM with 128-bit ICV
15. aes128gcm12 or aes128gcm96	128-bit AES-GCM with 96-bit ICV
16. aes128gcm8 or aes128gcm64	128-bit AES-GCM with 64-bit ICV
17. aes128ccm16 or aes128ccm128	128-bit AES-CCM with 128-bit ICV
18. aes128ccm12 or aes128ccm96	128-bit AES-CCM with 96-bit ICV
19. aes128ccm8 or aes128ccm64	128-bit AES-CCM with 64-bit ICV
20. aes128	128 bit AES-CBC
21. aes192	192 bit AES-CBC
22. aes256	256 bit AES-CBC
23. camellia256ccm16 or camellia256ccm128	256-bit Camellia-CCM with 128-bit ICV
24. camellia256ccm12 or camellia256ccm96	256-bit Camellia-CCM with 96-bit ICV
25. camellia256ccm8 or camellia256ccm64	256-bit Camellia-CCM with 64-bit ICV
26. camellia192ccm16 or camellia192ccm128	192-bit Camellia-CCM with 128-bit ICV
27. camellia192ccm12 or camellia192ccm96	192-bit Camellia-CCM with 96-bit ICV
28. camellia192ccm8 or camellia192ccm64	192-bit Camellia-CCM with 64-bit ICV
29. camellia128ccm16 or camellia128ccm128	128-bit Camellia-CCM with 128-bit ICV
30. camellia128ccm12 or camellia128ccm96	128-bit Camellia-CCM with 96-bit ICV
31. camellia128ccm8 or camellia128ccm64	128-bit Camellia-CCM with 64-bit ICV
32. camellia256	256 bit Camellia-CBC
33. camellia192	192 bit Camellia-CBC
34. camellia128	128 bit Camellia-CBC

Integrity Algorithms

Keyword	Description
1. sha512 or sha2_512	SHA2_512_256 HMAC

2. sha384 or sha2_384	SHA2_384_192 HMAC
3. sha256 or sha2_256	SHA2_256_128 HMAC
4. sha256_96 or sha2_256_96	SHA2_256_96 HMAC
5. sha1_160	SHA1_160 HMAC

Diffie Hellman Groups

Keyword	DH Group
NIST Elliptic Curve Groups	
1. ecp521	21
2. ecp384	20
3. ecp256	19
4. ecp224	26
5. ecp192	25
Regular Groups	
6. modp8192	18
7. modp6144	17
8. modp4096	16
9. modp3072	15
10. modp2048	14
11. modp1536	5