Anvisning för kryptering

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

V3.0

Reviderat och godkänt av Petter Könberg

Datum
2018-10-15

inera
LANDSTING OCH REGIONER
I SAMVERKAN FÖR E-HÄLSA

Innehåll

inera
LANDSTING OCH REGIONER
I SAMVERKAN FÖR E-HÄLSA

Revisionshistorik

Version	Datum	Författare	Kommentar
1.0		Björn Skeppner	
2.0 RC1	2016-09-06	Bengt-Göran Andersson, Björn Gustavsson, Christoffer Johansson	Hela dokumentet reviderat
2.0 RC2	2016-09-06	Bengt-Göran Andersson, Björn Gustavsson, Christoffer Johansson	
2.0 RC3	2017-04-19	Bengt-Göran Andersson	Ändrat Camellia från röd till gul. Ändringar för att tydliggöra att anvisningen gäller tjänster som tillhandahålls av Inera.
ua 2.1 a/b	2018-01-15 2018-01-17	Bengt-Göran Andersson	Reviderat av Christoffer Johansson, Björn Gustavsson och Roberth Lundin Inkonsistens i beskrivna Cipher Suites borttagen. Godkända Cipher Suites uppdaterats och flyttat till Appendix
2.1	2018-01-30	Bengt-Göran Andersson	Beslut på Informations och IT Säkerhetsrådet att godkänna uppdateringen till v2.1
3.0d-f, RC1/2	2018-06-12 - 2018-10-01	Bengt-Göran Andersson	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

1. Förkortningar

Ett urval av använda förkortningar/beteckningar.

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. http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf
AH	Authentication Header
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.
DH	Diffie–Hellman, en metod för att säkert utbyta kryptografiska nycklar över en publik kanal, baserat på den matematiska ”diskreta logaritm problemet”.
DHE	DH Ephemeral, DH metoden men kortlivade nycklar, se punkten 8.1.2.
DSS	Digital Signature Standard, inkluderar DSA som signeringsalgoritm. https://dx.doi.org/10.6028/NIST.FIPS.186-4
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
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.
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)
ESP	Encapsulating Security Payload
GCM	Galois/Counter Mode, ett operationsmode för symmetriska kryptografiska blockkrypton t.ex. AES eller DES.
IDEA	International Data Encryption Algorithm, En symmetrisk blockkryptoalgoritm designat av James Massey och Xuejia Lai år 1991
IKE	Internet Key Exchange
IPsec	Internet Protocol Security
ISAKMP	Internet Security Association and Key Management Protocol, kallas numera IKE. Tidigare förekom även benämningen ISAKMP/Oakley.
MAC	Beskrivs i punkt 8.3.2.
MD5	Beskrivs i punkt 8.3.2.
NIST	National Institute for Standards and Technology
RC4	En symmetrisk blockkryptoalgoritm för designat av Ron Rivest in 1987
PFS	Beskrivs i punkt 8.2.
PRF	
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	”Substituition-box”, är 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 8.3.2.
SHA-2, SHA256	Beskrivs i punkt 8.3.2.
SRP	Secure Remote Password, är ett utökat lösenord-autentisering nyckelförhandlings protokoll.
SSL/TLS	Secure Sockel Layer och Transport Layer Security är kryptografiska transmissionsprotokoll som tillhandahåller kommunikationssäkerhet över datanätverk.

2. Bakgrund och syfte

Denna anvisning är en del av Ineras Riktlinje för informationssäkerhet [R1] och behandlar tekniska detaljer kring konfigurationen av krypterad kommunikationen 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
- När data/information som lagras pga. Informationssäkerhetsklassning måste skyddas mot obehörig åtkomst.

Kompatibla inställningar är en förutsättning för en säker och väl fungerande miljö med en homogen Cipher hantering.

Följande markeringar visar Ineras riktlinjer och 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.

Målgruppen för detta dokument är utvecklings- och förvaltningsteam för tjänster som tillhandahålls av Inera.

2.1 Autentisering

Autentisering behandlas i sin helhet i Anvisning Autentisering.

3. 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. Referens material ” NIST Special Publication 800-57 Part 1, Revision 4” [R2]

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.

4. 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

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

- Key Exchange Algorithm (**se punkt 8**):
  - RSA med DHE eller ECDHE
  - DHE-DSS
  - ECDH-ECDSA
  - RSA med DH, ECDH eller SRP
  - RSA, PSK
- Encryption Algorithm/Cipher Suites (**se punkt 8.3**)
  - AES (AES128-CBC el. GCM, AES256CBC el. GCM)
  - Camellia
  - RC4, Triple DES, IDEA, DES,
- Message Authentication Code (MAC) (**se punkt 8.3.2**)
  - SHA-3
  - 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.

5. 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.

5.1 Rekommenderade protokoll för transportkryptering med TLS

- **TLS 1.3** – Version 1.3 är from augusti 2018 godkänd av IETF som RFC 8446. 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** – Aktiverat som standard i de senaste versionerna av alla webbläsare, kan manuellt aktiveras på vissa äldre OS/Webbläsare, se punkt 5.2, men är dock aktiverat som standard i de versioner som stöds inom eKlient i Samverkan
- **TLS 1.1** – Kan aktiveras i syfte att ge bakåtkompatibilitet för lite äldre versioner av Safari och Chrome. Kräver manuell aktivering på många andra operativsystem/browser, se punkt 5.2. Många klienter som stödjer TLS 1.1 stödjer förmodligen även TLS 1.2.
- **TLS 1.0** – Rekommenderas INTE, men kan behöva aktiveras i enstaka fall bibehållas för bakåtkompatibilitet.
- **SSL 3.0** – Rekommenderas INTE, men kan behöva aktiveras i enstaka fall. 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 planera för en avveckling av TLS 1.0 och bör även ta fram en avvecklingsplan för TLS 1.1.

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

5.1.1 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. 11.2.1
- 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.

5.2 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.

6. Protokoll för transportkryptering med IKE/IPsec

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

6.1 IKE, Internet Key Exchange

IKE protokollet finns idag i två versioner IKEv1 och IKEv2. IKEv2. I IKEv2 protokollet har genomgått ett antal säkerhetsförbättringar som förklaras i " IKEv2 Clarifications and Implementation Guidelines" [R3].

6.1.1 Rekommenderat IKE protokoll:

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

6.1.2 Rekommenderad autentiseringsmetod

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

1. RSA/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 eller kommande Efos funktionscertifikat.

OBS. 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.

6.1.3 Rekommenderade krypteringsalgoritmer

Krypterings- och nyckelutbytesalgoritmer för IPsec baserad VPN ska följa den rekommendation som finns i kapitel 8 Parametrar för transportkryptering.

6.1.4 Rekommenderad Key Lifetime

Rekommenderad Key Lifetime för en IKE SA:

8 timmar

6.1.5 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” [R4] 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.

6.2 IPsec

6.2.1 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.

6.2.2 Rekommenderade krypteringsalgoritmer

Krypterings- och nyckelutbytesalgoritmer för IPsec baserad VPN ska följa den rekommendation som finns i kapitel 8 Parametrar för transportkryptering.

6.2.3 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” [R4] 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.

6.3 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 det 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.

7. Kryptering av lagrad information (filkryptering)

7.1.1 Rekommenderade krypteringsalgoritmer

Krypteringsalgoritmer för lagrad information och filer följa den rekommendation som finns i kapitel 6 Parametrar för kryptering.

Rekommenderade krypteringsalgoritmer för att skydda lagrade data är:

AES 256, AES algoritmen med 256 bitars krypteringsnyckel

AES 128, AES algoritmen med 128 bitars krypteringsnyckel

7.2 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)

7.3 Lösenord för skydd av nyckellager och krypterade filer

7.3.1 Rekommenderad lösenordslängd

Ett lösenord ska vara minst 16 tecken långt med komplexitetskrav.

16 tecken x 8 bitar = 128 bitar som i så fall står i paritet med krypteringsalgoritmens nyckellängd.

7.4 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 [R5] 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)

8. Parametrar för transportkryptering

8.1 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 ett SITHS- eller kommande Efos-certifikat

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).

8.1.1 Rekommenderade DH grupper för nyckelutbyte

Rekommendationen är att man använder dem enligt följande i prioriteringsordning:

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 toolet.

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

8.1.2 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

8.2 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)**.

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.

8.2.1 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 som tillägg bakom vald Cipher Suite och baseras på den nyckellängd som används i handskakningen av Diffie-Hellman med antingen elliptiska kurvor (ECDHE) eller modulo (DHE) inom själva TLS- eller IKE/IPsec-sessionen.

8.2.2 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.

8.3 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 CBC. GCM är att föredra men ställer vissa prestandakrav på både klient och server. 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 med ”Block Cipher” baserat på CBC ska undvikas då sårbarheter är identifierade

8.3.1 Rekommenderade Cipher Suites för TLS och IPsec

Godkända och accepterade Cipher Suites finns under Appendix nedan

8.3.2 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). Rekommendationen att man använder minst SHA-2 256, då SHA-1 (160 bitars) anses, i takt med allt högra beräkningskapacitet i moderna datorer, vara sårbar för ”kollisionsattacker” där man kan skapa ett meddelande för en viss 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 en uppdatering av hash-algoritmen som kallas SHA-3. I skrivande stund hittas inga tydliga beskrivningar över Cipher Suites där SHA-3 ingår, således avses att följa upp denna anvisning gällande SHA-3 i kommande version.

9. Referenslista

Ref	Dokumentnamn	Dokument

R1	Riktlinje för informationssäkerhet	https://intra.inera.se/Vart-regelverk/Informationssakerhet/Riktlinjer-informationssakerhet/
R2	NIST SP 800-57 Rev. 4	https://dx.doi.org/10.6028/NIST.SP.800-57pt1r4
R3	IKEv2 Clarifications and Implementation Guidelines	http://www.ietf.org/rfc/rfc4718.txt
R4	NIST, Guide to IPsec VPNs	https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-77.pdf
R5	Brute-force attack	https://en.wikipedia.org/wiki/Brute-force_attack

10. Appendix

10.1 Grundkonfiguration av webbservrar

10.1.1 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 avsnitet "**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 **10.1.3 Övriga inställningar för webbservrar** måste också följas.

10.1.2 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 **10.1.3 Övriga inställningar för webbservrar** måste också följas.

10.1.3 Övriga inställningar för webbservrar

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

10.1.4 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 osäkra SSL och TLS protokoll enligt punkt 4.1
- Avaktivera Client-Initiated Renegotiation - mot DoS attacker.
- Avaktivera TLS-kompression - mot CRIME och TIME-attacker.
- Avvaktivera 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 7.1.1 (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 lägre cipher protokollversioner tillåts - mot POODLE
- Maximal TLS-sessionslängd utan omförhandling är 12h.
- Maximal inaktiv TLS-sessionslängd är 35 minuter
- Vid användningen av SITHS certifikat ska HCC Funktion utfärdas av SITHS Type 3 CA v1.

10.2 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 är nedgraderat till Röd nivå och avveckling ska planeras snarast OBS

10.2.1 Rekommenderade Cipher Suites i TLS 1.3

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

10.2.2 Rekommenderade Cipher Suites i TLS 1.1 - 1.2 AES och SHA-2 med PFS (RSA/DSA/DSS)

6. TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (SHA-2)
7. TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (SHA-2)
8. TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (SHA-2)
9. TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (SHA-2)
10. TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (SHA-2)
11. TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (SHA-2)
12. TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (SHA-2)
13. TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (SHA-2)
14. TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (SHA-2)
15. TLS_DHE_DSS_WITH_AES_256_GCM_SHA384 (SHA-2)
16. TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (SHA-2)
17. TLS_DHE_DSS_WITH_AES_256_CBC_SHA256 (SHA-2)
18. TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (SHA-2)
19. TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (SHA-2)
20. TLS_DHE_DSS_WITH_AES_128_CBC_SHA256 (SHA-2)

21. TLS_DHE_DSS_WITH_AES_128_GCM_SHA256 (SHA-2)

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

22. TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384 (SHA-2)
23. TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384 (SHA-2)
24. TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384 (SHA-2)
25. TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384 (SHA-2)
26. TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256 (SHA-2)
27. TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256 (SHA-2)
28. TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256 (SHA-2)
29. TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256 (SHA-2)
30. TLS_DH_RSA_WITH_AES_256_GCM_SHA384 (SHA-2)
31. TLS_DH_DSS_WITH_AES_256_GCM_SHA384 (SHA-2)
32. TLS_DH_RSA_WITH_AES_256_CBC_SHA256 (SHA-2)
33. TLS_DH_DSS_WITH_AES_256_CBC_SHA256 (SHA-2)
34. TLS_DH_RSA_WITH_AES_128_GCM_SHA256 (SHA-2)
35. TLS_DH_RSA_WITH_AES_128_CBC_SHA256 (SHA-2)
36. TLS_DH_DSS_WITH_AES_128_CBC_SHA256 (SHA-2)
37. TLS_DH_DSS_WITH_AES_128_GCM_SHA256 (SHA-2)

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

38. TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (SHA-1)
39. TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (SHA-1)
40. TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (SHA-1)
41. TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (SHA-1)
42. TLS_DHE_DSS_WITH_AES_256_CBC_SHA (SHA-1)
43. TLS_DHE_RSA_WITH_AES_256_CBC_SHA (SHA-1)
44. TLS_DHE_RSA_WITH_AES_128_CBC_SHA (SHA-1)
45. TLS_DHE_RSA_WITH_AES_128_CBC_SHA (SHA-1)
46. TLS_DHE_DSS_WITH_AES_128_CBC_SHA (SHA-1)

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

47. TLS_ECDH_RSA_WITH_AES_256_CBC_SHA (SHA-1)
48. TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA (SHA-1)
49. TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA (SHA-1)
50. TLS_ECDH_RSA_WITH_AES_128_CBC_SHA (SHA-1)
51. TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA (SHA-1)
52. TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA (SHA-1)
53. TLS_DH_RSA_WITH_AES_256_CBC_SHA (SHA-1)
54. TLS_DH_DSS_WITH_AES_256_CBC_SHA (SHA-1)
55. TLS_DH_RSA_WITH_AES_128_CBC_SHA (SHA-1)
56. TLS_DH_DSS_WITH_AES_128_CBC_SHA (SHA-1)

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

57. TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (SHA-1)
58. TLS_DHE_DSS_WITH_CAMELLIA_256_CBC_SHA (SHA-1)
59. TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (SHA-1)
60. TLS_DHE_DSS_WITH_CAMELLIA_128_CBC_SHA (SHA-1)

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

61. TLS_DH_RSA_WITH_CAMELLIA_256_CBC_SHA (SHA-1)
62. TLS_DH_DSS_WITH_CAMELLIA_256_CBC_SHA (SHA-1)

63. TLS_DH_RSA_WITH_CAMELLIA_128_CBC_SHA (SHA-1)
64. TLS_DH_DSS_WITH_CAMELLIA_128_CBC_SHA (SHA-1)

AES och SHA-1 utan PFS (SRP)

65. TLS_SRP_SHA_DSS_WITH_AES_256_CBC_SHA (SHA-1)
66. TLS_SRP_SHA_RSA_WITH_AES_256_CBC_SHA (SHA-1)
67. TLS_SRP_SHA_RSA_WITH_AES_128_CBC_SHA (SHA-1)
68. TLS_SRP_SHA_DSS_WITH_AES_128_CBC_SHA (SHA-1)

10.2.3 TLS 1.0

TLS 1.0 Är ett icke acceptabelt transportprotokoll.
Alla varianter av TLS 1.0 och framför allt tillsammans med CBC ska helt undvikas.

10.3 Godkända och accepterade Cipher Suites för IKE och IPsec protokollen

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

10.3.1 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)

10.3.2 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