|  |  |
| --- | --- |
| Druh dokumentu: | Software Dokument/ Software Document |
| Název dokumentu: | Zásady použití kryptografie pro dodávané elektronické komponenty/ Cryptography Policy for Supplied Electronic Components |
| Projekt: | Kybernetická bezpečnost / Cybernetic Security |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Číslo dokumentu | EY15743P | Revize | a |
|  | Jméno | Datum | Podpis |
| Vypracoval | Toncar Vladimír | 5.6.2024 |  |
| Změnil | Toncar Vladimír | 19.7.2024 |  |
| Přezkoušel | Baroň Pavel | 19.7.2024 |  |
| Přezkoušel za HW | - | - |  |
| Přezkoušel za Projekty | - | - |  |
| Schválil | Straka František | 24.7.2024 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Revize** | **Jméno** | **Datum** | **Podpis** |
| **Popis změny** | | | |
| a | Vladimír Toncar | 18.7.2024 |  |
| Included English version of the cryptography policy. Added random number generator requirements. | | | |
| b |  |  |  |
|  | | | |
| c |  |  |  |
|  | | | |
| d |  |  |  |
|  | | | |
| e |  |  |  |
|  | | | |
| f |  |  |  |
|  | | | |
| g |  |  |  |
|  | | | |
| h |  |  |  |
|  | | | |
| i |  |  |  |
|  | | | |
| j |  |  |  |

OBSAH

[1. Zásady použití kryptografie pro dodávané elektronické komponenty 4](#_Toc172206395)

[2. související dokumentace 6](#_Toc172206396)

[3. přílohy 7](#_Toc172206397)

[4. Policy for the use of cryptography for supplied electronic components 8](#_Toc172206398)

[5. related documentation 10](#_Toc172206399)

[6. ANNEXES 11](#_Toc172206400)

This document is in Czech and English. See chapter 4 for the English version.

# Zásady použití kryptografie pro dodávané elektronické komponenty

1. Následující pravidla definují zakázané a doporučené kryptografické protokoly, šifry a hashovací funkce. Zakázány jsou šifry a algoritmy, které mají známé zranitelnosti nebo již zastaraly. Doporučeny jsou zejména algoritmy s menšími nároky a procesor a paměť při velmi dobré kryptografické odolnosti. Akceptovány jsou také algoritmy označené v tomto textu jako alternativní, při dodržení uvedených podmínek (zejména délky klíčů).
2. Při potřebě použít jiné, než zde uvedené algoritmy je třeba toto použití konzultovat se ŠELC.
3. Seznamy zde doporučených či zakázaných položek budou přezkoumány alespoň jednou ročně a aktualizovány dle doporučení příslušných autorit (NÚKIB, NIST) a stavu poznání kryptografie. Přezkoumání se obzvláště zaměří na algoritmy, které zde budou označeny jako algoritmy akceptované, avšak dosluhující.
4. **Kryptografické protokoly**
   1. Zakázáno: SSL 2.0, SSL 3.0, TLS 1.0, TLS 1.1, DTLS 1.0
   2. Doporučeno: TLS 1.2, TLS 1.3, DTLS 1.2, DTLS 1.3.
5. **Symetrické šifry**
   1. Zakázáno: DES, GOST, RC2, RC4, CBC mód různých šifer, 3-DES
   2. Doporučeno: ChaCha20 s délkou klíče 256 bitů a zatížením klíče max. 256 GB (klíč se použije v rámci spojení pro nejvýše toto množství přenesených dat, vhodné řešení je klíč dostatečně často měnit)
   3. Alternativní algoritmy: je povoleno použití šifry AES s délkou klíčů 128 bitů a vyšší, Camellia s délkou klíče 128 bitů a vyšší, Serpent s délkou klíče 128 bitů a vyšší.
6. **Asymetrické šifry**
   1. Zakázáno: DSA (Digital Signature Algorithm) s délkou klíče 2048 bitů, EC-DSA (Elliptic Curve Digital Signature Algorithm) s délkou klíče kratší než 256 bitů, RSA-PSS s délkou klíče menší než 2048 bitů, EC-Schnorr s délkou klíče 224 bitů.
   2. Doporučeno: šifry typu EC-DSA s klíčem délky alespoň 256 bitů (konkrétně šifry prime256v1 a silnější)
   3. Alternativní algoritmy: je povoleno použití RSA-PSS s délkou klíče alespoň 3072 bitů a vyšší, EC-Schnorr s délkou klíče 256 bitů a vyšší. Na přechodnou dobu se akceptuje použití algoritmu RSA s klíčem délky 2048 bitů, považuje se však za algoritmus dosluhující (bude přezkoumáno v příští revizi tohoto dokumentu).
7. **Algoritmy pro vyjednání klíče a šifrování klíčů**
   1. Zakázáno: DSA (Digital Signature Algorithm) s délkou klíče 2048 bitů, EC-DSA (Elliptic Curve Digital Signature Algorithm) s délkou klíče kratší než 256 bitů, RSA-PSS s délkou klíče menší než 2048 bitů, EC-Schnorr s délkou klíče 224 bitů.
   2. Doporučeno: ECDHE (Elliptic Curve Diffie-Hellman Ephemeral) s délkou klíče 256 bitů a vyšší, DHE (Diffie-Hellman Ephemeral) s délkou klíče 3072 bitů a vyšší.
   3. Alternativní algoritmy: je povoleno použití ostatních algoritmů na bázi eliptických křivek (ESIC-KEM, PSEC-KEM, ACE-KEM) s délkou klíče alespoň 256 bitů, algoritmy skupiny RSA (Rivest Shamir Adleman), jmenovitě RSA-OAEP, RSA-KEM jsou povoleny s délkou klíče alespoň 3072 bitů. Na přechodnou dobu se akceptuje použití algoritmů skupiny RSA s klíčem délky 2048 bitů, považují se však za algoritmy dosluhující (bude přezkoumáno v příští revizi tohoto dokumentu).
8. **Hashovací algoritmy**
   1. Zakázáno: MD4, MD5, SHA-1 (s výjimkou níže), MDC2
   2. Doporučeno: SHA-256, SHA-384, SHA-512, SHA3-256, SHA3-384, SHA3-512
   3. Alternativní algoritmy: SHAKE128, SHAKE256.
   4. Specializované algoritmy pro hashování hesel (např. PBKDF2) použijí jako vnitřní hashovací funkci alespoň SHA-256 nebo vyšší a dostatečný počet iterací (PBKDF2 100 000 iteraci), pro každé heslo se použije individuální tzv. „sůl“.
9. **Generátory náhodných čísel**

Pro kryptografické účely je nezbytné použít kryptograficky bezpečné generátory náhodných čísel (pro které se obvykle užívá zkratka CSPRNG, tj. Cryptographically Secure Pseudo-Random Number Generator).

* 1. Pokud používáte kryptografickou knihovnu nebo framework, ověřte (např. v dokumentaci), že použité generátory náhodných čísel jsou CSPRNG.
  2. Pokud provádíte implementaci sami a vaše platforma disponuje hardwarovým generátorem náhodných čísel, který je kryptograficky bezpečný (certifikát výrobce, případně vyplývá z dokumentace), je preferováno použití tohoto hardwarového generátoru náhodných čísel (obvykle označováno jako TRNG, tj. True Random Number Generator).
  3. V ostatních případech při vlastní implementaci důsledně rozlišujte mezi běžnými pseudo-náhodnými generátory (vhodnými např. pro účely simulace) a funkcemi či knihovnami poskytujícími CSPRNG. Funkce/knihovny poskytující CSPRNG obvykle využívají vhodný zdroj entropie z operačního systému a jsou často založeny na moderních šifrách (např. ChaCha, AES, Serpent). Např. v jazyce Java poskytuje CSPRNG knihovna java.security.SecureRadom, v jazyce Python je to knihovna secrets.

# související dokumentace

|  |  |  |
| --- | --- | --- |
|  |  | Neobsahuje související dokumentaci |
|  |  |  |

# přílohy

|  |  |  |
| --- | --- | --- |
|  |  | Neobsahuje přílohy |
|  |  |  |

# Policy for the use of cryptography for supplied electronic components

1. The following rules define prohibited and recommended cryptographic protocols, ciphers, and hash functions. Ciphers and algorithms that have known vulnerabilities or are outdated are prohibited. The algorithms marked as recommended are those with lower processor and memory requirements, yet with a very good cryptographic strength. Algorithms marked in this text as alternative are also accepted, provided that the stated conditions are met (especially the length of the keys).
2. If it is necessary to use algorithms other than those listed here, this use must be consulted with ŠELC.
3. The lists of items recommended or prohibited here will be reviewed at least once a year and updated according to the recommendations of the relevant authorities (NÚKIB, NIST) and the state of knowledge of cryptography. The review will particularly focus on algorithms designated herein as deprecated.
4. **Cryptographic protocols**
   1. Prohibited: SSL 2.0, SSL 3.0, TLS 1.0, TLS 1.1, DTLS 1.0
   2. Recommended: TLS 1.2, TLS 1.3, DTLS 1.2, DTLS 1.3.
5. **Symmetric ciphers**
   1. Prohibited: DES, GOST, RC2, RC4, CBC mode of various ciphers, 3-DES
   2. Recommended: ChaCha20 with a key length of 256 bits and a key load of max. 256 GB (the key is used within the connection for a maximum of this amount of transferred data, a suitable solution is to change the key often enough)
   3. Alternative algorithms: AES cipher with key length 128 bits and higher, Camellia with key length 128 bits and higher, Serpent with key length 128 bits and higher are allowed.
6. **Asymmetric ciphers**
   1. Prohibited: DSA (Digital Signature Algorithm) with a key length of 2048 bits, EC-DSA (Elliptic Curve Digital Signature Algorithm) with a key length of less than 256 bits, RSA-PSS with a key length of less than 2048 bits, EC-Schnorr with a key length of 224 bits.
   2. Recommended: EC-DSA type ciphers with a key length of at least 256 bits (specifically prime256v1 ciphers and stronger)
   3. Alternative algorithms: the use of RSA-PSS with a key length of at least 3072 bits and higher, EC-Schnorr with a key length of 256 bits and higher is allowed. The use of the RSA algorithm with a key length of 2048 bits is accepted for a temporary period, but it is considered to be a deprecated algorithm (it will be reviewed in the next revision of this document).
7. **Algorithms for key negotiation and key encryption**
   1. Prohibited: DSA (Digital Signature Algorithm) with a key length of 2048 bits, EC-DSA (Elliptic Curve Digital Signature Algorithm) with a key length of less than 256 bits, RSA-PSS with a key length of less than 2048 bits, EC-Schnorr with a key length of 224 bits.
   2. Recommended: ECDHE (Elliptic Curve Diffie-Hellman Ephemeral) with a key length of 256 bits and above, DHE (Diffie-Hellman Ephemeral) with a key length of 3072 bits and above.
   3. Alternative algorithms: the use of other algorithms based on elliptic curves (ESIC-KEM, PSEC-KEM, ACE-KEM) with a key length of at least 256 bits is allowed, the algorithms of the RSA group (Rivest Shamir Adleman), namely RSA-OAEP, RSA-KEM are allowed with a key length of at least 3072 bits. The use of RSA group algorithms with a key length of 2048 bits is accepted for a temporary period, but they are considered to be deprecated algorithms (will be reviewed in the next revision of this document).
8. **Hashing algorithms**
   1. Forbidden: MD4, MD5, SHA-1, MDC2
   2. Recommended: SHA-256, SHA-384, SHA-512, SHA3-256, SHA3-384, SHA3-512
   3. Alternative algorithms: SHAKE128, SHAKE256.
   4. Specialized password hashing algorithms (e.g. PBKDF2) will use at least SHA-256 or higher as an internal hash function and a sufficient number of iterations (PBKDF2 100,000 iterations), an individual so-called "salt" will be used for each password.
9. **Random number generators**

For cryptographic purposes, it is necessary to use cryptographically secure random number generators (usually abbreviated as CSPRNG, i.e. Cryptographically Secure Pseudo-Random Number Generator ).

* 1. If you are using a cryptographic library or framework, verify (e.g. in the documentation) that the random number generators used are CSPRNG.
  2. If you are doing the implementation yourself and your platform has a hardware random number generator that is cryptographically secure (check manufacturer's certificate or documentation), it is preferred to use this hardware random number generator (usually referred to as TRNG, i.e. True Random Number Generator ).
  3. In other cases, during your own implementation, do distinguish between common pseudo-random generators (suitable, for example, for simulation purposes) and functions or libraries providing CSPRNG. The functions/libraries providing CSPRNG usually use a suitable source of entropy from the operating system and are often based on modern ciphers (e.g. ChaCha, AES, Serpent). As an example, in Java, CSPRNG is provided by the java.security .SecureRadom library, in Python it is the “secrets” library.

# related documentation

|  |  |  |
| --- | --- | --- |
|  |  | Does not include related documentation |
|  |  |  |

# ANNEXES

|  |  |  |
| --- | --- | --- |
|  |  | Does not contain attachments |
|  |  |  |



Veškerá práva k tomuto dokumentu přísluší ŠKODA ELECTRIC a.s.!  
Bez souhlasu této společnosti nesmí být dokument kopírován, rozmnožován a není povoleno postoupit jej třetím osobám!

ŠKODA ELECTRIC a.s.  
Průmyslová 610/2a  
301 00 Plzeň  
Czech Republic