**Payconiq**

**Use of Encryption Standards Guideline**

**Document Information**

|  |  |
| --- | --- |
| **Document ID** | PQI-SEC-PRO-011 |
| **Classification** | Internal Use |
| **Registered on** | 03-09-2018 |
| **Last Modified on** | 25-03-2024 |
| **Business Owner** | CTO |
| **Author** | Jeffry Laan |
| **Contact** | security@payconiq.com |
| **Version** | 2.0 |
| **Status** | Final |

**Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Change | Author |
|  | 1.0 | Draft | Mustafa Kucukaytekin |
|  | 1.1 | Added hashing requirements | Konstantin Pavlov |
|  | 1.2 | Added IV and cipher modes | Pieter Van Prooijen |
|  | 1.3 | Review | Nitin R. Deshmukh |
|  | 1.4 | Added algorithms, password hashing | Remco Nabuurs |
|  | 1.5 | Review | Luis Charruadas |
|  | 1.6 | Review | Murat Cetinpolat |
|  | 1.6 | Review | Mustafa Kucukaytekin |
|  | 1.7 | Review | Safet Acifovic |
|  | 1.7 | Review | Reinier Vegter |
|  | 1.8 | Review | Nishta Wadhawan |
|  | 1.9 | Final Draft | Nishta Wadhawan |
|  | 2.0 | Approved by Campfire | Jeffry Laan |
|  | 2.1 | Review | Michael Oki |

**Documented changes**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Change | Author |
| 25-03-2024 | 2.1 | Updated sections 5.5, 5.6 | Michael Oki |

**Table of Contents**

[1. Overview 2](#_Toc347290746)

[2. Purpose 3](#_Toc1825391593)

[3. Scope 3](#_Toc1662497658)

[4. Governance 3](#_Toc1543451401)

[5. Standard 3](#_Toc819906519)

[5.1 Principles 3](#_Toc801342656)

[5.2 Encryption Controls 3](#_Toc2142494022)

5.3 Key Size Requirements 5

5.4 Hashing Requirements 5

5.5 Encryption Requirements 5

5.6 Authentication Requirements 5

5.7 Block Cipher Modes 6

6. Control 6

6.1 Control Measurement 6

6.2 Exceptions 6

6.3 Non-compliance 6

7. Related Standards, Policies and Processes 6

# Overview

# Purpose

The purpose of this document is to define minimum technical requirements for the use of encryption controls, as well as setting the governance for the use of cryptographic keys.

# Scope

This guideline is intended to protect the confidentiality, integrity, authenticity, and non-repudiation of information assets for Payconiq’s production environment, applied to all Payconiq employees, vendors, contractors, consultants and temporary employees of Payconiq International S.A. and Payconiq Services B.V. entities.

# Governance

**All** **developers** are responsible to implement their solutions according to this Standard.

**SRE** provides the technical solution to enable requirements of this Standard.

The **Security team** validates the effectiveness of the controls.

**PQI and PQS Management Boards** approves this Standard.

The **Head of** Security authorizes the exceptions to this Standard.

# Standard

**5.1 Principles**

* The list of approved cryptographic algorithms and key sizes shall be reviewed at least annually.
* Where multiple layers of encryption are available (e.g., media-level and database field level), each layer shall be applied proportionally to mitigate risks identified during the risk assessment process.

**5.2 Encryption Controls**

According to information classification control in *Information Security Policy,* as well as legal and contractual obligations, Payconiq shall protect the systems, and information according to one of the controls listed in the below table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Control / Classification** | **Internal Use** | **Confidential** | **Sensitive Payment Data[1]** |
| **Encryption at Rest** | **-** | **Required** | **Required** |
| **Encryption in Transit** | **Required** | **Required** | **Required** |
| **Digital Signature** | **-** | **Required\*\*** | **Required** |
| **Message Authentication\*** | **-** | **Required\*\*\*** | **Required** |

* This requirement is relaxed if an authenticated encryption mode is used.

\*\*   Digital signatures shall be used at least if the information is crossing the trust boundaries and risk calculation, consult with the Security Team.

\*\*\* Message authentication shall be used in case of the integrity and authenticity of the information are required based on risk calculation, consult with the Security Team.

**5.3 Key Size Requirements**

For each of the specific cryptographic function used, the following algorithms and key sizes are to be used. All key sizes are provided in bits. These are the **minimal sizes** for the desired security level.

|  |  |  |  |
| --- | --- | --- | --- |
| **Control** | **Algorithm** | **Key size (bits)** | **Remark** |
| Encryption | AES | 256 | Symmetric |
| Encryption | RSA | 2048 | Asymmetric, ECC preferred |
| ECC | 384 | Any curve known to be safe unless required for integration or compatibility purposes |
| Hash | SHA2, SHA3 | 256 |  |
| Password hash | Argon2id/i | No minimum key length required | Preferred[AH1] [VK2] |
| scrypt | Preferred if Argon2 is not available |
| bcrypt | Legacy use only |
| PKBDF2 | Legacy use only |
| SHA-256 | Allowed for API keys only, due to higher DDoS risk |

**5.4 Hashing Requirements**

The salt must be long and from a cryptographically secure source, (e.g., java.security.SecureRandom). Use a salt that is the same size as the output of the hash function[2].

Cryptographic hashing algorithms shall be used as the basis for:

* Creating message digests,
* Generating digital signatures,
* Message Authentication Codes (MACs / HMACs),
* Pseudorandom Generation Functions,
* Key Derivation Functions (KDFs).

**5.5 Encryption Requirements**

Encrypted communications channels shall be protected – at minimum - using one of the following methods:

* At the application layer, using Transport Layer Security (TLS)
* At the network layer, using VPN;
* Secure Shell (SSH) for remote administration of systems only.

Based on the risk assessment, additional encryption layers might be added.

Minimum TLS[3] version is 1.2, an exception should only be granted if there’s an approved migration plan for the service or application under review.

**5.6 Authentication Requirements**

Authentication information which grants authorized access to assets, e.g., passwords, API keys:

* Not be stored in plain text or in any reversible format.
* Be salted with at least 64 bits of pseudorandom data.
* Be hashed using a method described in the section 4.4.
* User IDs or usernames shouldn’t be used during salting.
* Salts should be rotated periodically.

Due to the different nature of SRP evidence, they are excluded from this requirement.

**5.7 Block Cipher Modes**

Authenticated encryption shall be preferred over unauthenticated encryption modes, as defence against chosen-ciphertext attacks. AES-GCM or AES-CCM can be used for authenticated encryption. When using AES-GCM, a 96-bit IV and the full 128 bits of the authentication tag shall be used.

When authenticated encryption mode implementations are not available, an Encrypt-then-MAC construct shall be used instead. A combination of AES-CTR or AES-CBC with HMAC or CMAC can be used for this purpose. The message authentication code must authenticate both the ciphertext and the IV.

The use of unauthenticated encryption or any block cipher modes other than the ones specified in this section is only allowed for legacy purposes.

* Do not use ECB mode.
* Do not use CBC MAC for variable length data.

# Control

**6.1 Control Measurement**

The Security team will verify compliance with this policy through various methods, including but not limited to, periodic system analysis, internal and external audits, red teaming exercises, penetration tests, and source code reviews.

**6.2 Exceptions**

Any exception to the Standard shall be approved by the Head of Security.

**6.3 Non-Compliance**

An employee found to have violated this Standard may be subject to disciplinary action, up to and including termination of employment.

# Related Standards, Policies and Processes

* OWASP: [Cryptographic Storage Cheat Sheet](https://www.owasp.org/index.php/Cryptographic_Storage_Cheat_Sheet)
* Key size comparison:

<https://www.keylength.com/en/compare/>

* [Cryptographic Algorithm Validation Program](https://csrc.nist.gov/Projects/Cryptographic-Algorithm-Validation-Program/CAVP-TESTING-BLOCK-CIPHER-MODES#GCMVS)
* [SafeCurves: choosing safe curves for elliptic-curve cryptography](https://safecurves.cr.yp.to/)
* ISO/IEC 27001 standard, clause 10.1.1

[1] As of today, Payconiq stores 3 instances of sensitive payment data: SCT Access token, SCT Refresh token, SRP PIN evidence.

[2] For example, the output of SHA-256 is 256 bits (32 bytes), so the salt should be at least 32 random bytes.

[3] Due to compatibility issues with Internet Explorer browser as of today.