Cryptonics

ENTERPRISE BLOCKCHAIN

EBSS COMPLIANT

SECURITY SPECIFICATION

Compliance Report for **XM Colombia**

Enterprise Blockchain Security Specification

August 26, 2020

Table of Contents

Disclaimer

THE CONTENT OF THIS AUDIT REPORT IS PROVIDED “AS IS”, WITHOUT REPRESENTATIONS AND WARRANTIES OF ANY KIND.

THE AUTHOR AND HIS EMPLOYER DISCLAIM ANY LIABILITY FOR DAMAGE ARISING OUT OF, OR IN CONNECTION WITH, THIS COMPLIANCE REPORT.

This compliance audit has been performed by

Cryptonics Consulting S.L.
Ramiro de Maeztu 7
46022 Valencia
SPAIN

https://cryptonics.consulting/
info@cryptonics.consulting

Statement of Compliance

The processes and policies put in place by XM Colombia have been found in compliance with the Enterprise Blockchain Security Specification v 1.0 with minor observations.

A full list of the requirements analyzed and their description can be found in the Detailed Findings section of this document.

The observations that have been identified and addressed during the audit process are detailed in the Observations section of this document.

ENTERPRISE BLOCKCHAIN

EBSS COMPLIANT

SECURITY SPECIFICATION

Introduction

Purpose of this Report

Cryptonics Consulting has been engaged to perform an Enterprise Blockchain Security Specification (EBSS) compliance audit for three blockchain applications developed by **XM Colombia** and **Latin Checkout**:

- **ECOREGISTRY:** Private blockchain application aimed at the validation, verification, emission, and withdrawal of CO2 certificates.
- **ECOGOX:** Tokenization solution for generated energy
- **SICEP:** Permission management and notarization solution based on a public blockchain

The objectives of the audit are as follows:

1. Determine compliance with the Enterprise Blockchain Security Specification.
2. Identify potential security improvements.
3. Issue a statement of compliance.

As with any compliance audit, there is a limit to which the information supplied by the company can be verified. The author of this report does not guarantee complete coverage (see disclaimer).

About EBSS

The **Enterprise Blockchain Security Specification (EBSS)** is a specification aimed at fomenting a minimum standard of security for enterprise applications that make use of / or interact with a distributed ledger system, which may include both public and permissioned blockchains. The specification includes a list of requirements that MUST or SHOULD be fulfilled by such an enterprise blockchain application.

The EBSS should be seen as complementary to existing information security standards, such as **ISO/IEC 27001:2013**. It covers best practices and organizational security policies in the design and deployment of enterprise blockchain applications, in areas, such as key generation and storage, user privacy, and administrative policies.

However, the specification does not cover details of cryptographic implementations, secure coding patterns, and specific node configurations. It also does not consider the caveats of specific distributed ledger platforms and the details of their configuration.

The full specification can be found at https://ebsec.github.io/ebss/.

Methodology

The audit has been performed in the following steps:

1. Gaining an understanding of the application's purpose and functionality through initial scoping meetings.
2. Review of technical design documentation provided by the development team.
3. Review of smart contract code and configuration files supplied by the development team.
4. Question and answer sessions with the development team.
5. Report preparation and iterative improvement process.
6. Emission of statement of approval.

Detailed Findings

Domain 1: Key Management

Key Generation

R1.1.1	**Cryptographically Secure Randomness**	All keys MUST be created from a cryptographically secure source of randomness according to common cryptographic standards.	✓
R1.1.2	**Trusted Key Creation Environment**	Keys MUST be created in a trusted environment. In the context of normal user keys, this means they MUST be created either on the machine controlled by the user of an environment local to the key creator and storage process. Alternatively, they can be generated within a trusted execution environment (TEE) or a special purpose hardware security module (HSM).	✓
R1.1.3	**Key Derivation Functions**	Keys derived from a password MUST use a cryptographically secure and preferably standardized key derivation function configured to use standard recommended parameters. For instance, in the case of PBKDF2HMAC, the NIST recommendation of using 128-bit long salt values SHOULD be used.	✓ Observation 1
R1.1.4	**Secure Key Delivery**	Keys generated on behalf of a user MUST be delivered over a secure channel or offline. This means, they may be sent over an encrypted communication channel with secure authentication, but not via standard email or similar unencrypted channels.	✓

Key Storage

R1.2.1	**Encrypted Key Storage**	All keys MUST be encrypted using a standard encryption algorithm when stored on a digital storage medium.	✓
R1.2.2	**Disconnected Storage for**	Keys that control significant assets or critical operations and are to be used by an operator	(✓)

	Critical Interactive Keys	interactively, MUST be in cold storage when not used, preferably in a hardware wallet that has to be connected for specific uses.	Observation 2
R1.2.3	**Node Software Key Store**	Keys SHOULD not be stored in the blockchain platform's in-node wallet. Instead, keys SHOULD be stored in encrypted format in a separate location and raw transactions SHOULD be processed by the node, meaning the transactions are not signed by the node software itself.	(✓) Observation 3

Key Usage Protocol

R1.3.1	**Automated Key Use**	All keys used for automated processes in so-called hot wallets MUST be stored in a trusted system, in encrypted form. These "hot wallets" SHOULD be covered by real-time monitorization, if significant funds are involved.	N/A
R1.3.2	**Administrative Keys**	All administrative keys MUST be issued to trusted personnel only and issuance MUST be accounted for (recorded).	✓
R1.3.3	**Key Usage Logging**	Every use of a critical key (administrative key) MUST be logged.	✓
R1.3.4	**Multi-sig**	In the case of keys dealing with highly sensitive operations, such as upgrading a smart contract or modifying important parameters, a multi-signature scheme SHOULD be considered.	N/A

Key Recovery

R1.4.1	**HD-Wallets**	In the case of using hierarchical deterministic key generation (for example BIP 32) a clear process MUST be defined on how seed words are stored offline for internal users. Clear guidance MUST be given to external users on seed word security.	✓
R1.4.2	**Seed Backups**	Seed words, passphrases used for key recovery, and any other material that may serve to recover keys MUST be backed up securely offline.	✓

R1.4.3	**Key Compromise Protocol**	A document describing the process executed in the case a key is compromised MUST exist and cover: compromise communication procedure, key revocation, issuance of replacement keys	✓ Observation 4

Domain 2: Operational Policy

Node Operation

R2.1.1	**Backup Nodes**	In the case of the blockchain application interfacing with a public blockchain, the company SHOULD run a full node of the blockchain in question, in order to have a full local backup of the data.	✓
R2.1.2	**Gateway Redundancy**	In the case of interfacing with a public blockchain, at least one backup access MUST be used. For example, the own node a company operates SHOULD be backed up by a public access gateway, to ensure continuous operation.	(✓) Observation 5

IT Security

R2.2.1	**Data Sanitization Policy**	The company MUST have a clearly defined policy on how sensitive data is removed from storage media when equipment is retired or leaves the company.	✓
R2.2.2	**Credentials Management**	A clear policy MUST exist on how credentials are issued and revoked in the company.	✓

Domain 3: User Privacy

R3.1.1	**Address Re-use**	Addresses on public blockchains for users MUST not be re-used between different applications, unless unavoidable because of the application logic. The purpose of this is to avoid correlation. All external users SHOULD	✓

		be advised not to use their addresses for other applications.	
R3.1.2	**Personal Data**	No personal data MUST be stored directly on a public or private blockchain if the data allows identifying the user. Instead, this type of data SHOULD be stored off-chain and be cryptographically linked (hash-timestamped) onto the blockchain.	✓

Domain 4: Smart Contract Security

R4.1.1	**External Audits**	All smart contracts that are deployed on a public blockchain MUST be audited by an external security firm.	✓
R4.1.2	**Contract Monitorization**	Smart contracts that hold significant value SHOULD be monitored in real-time.	N/A

Domain 5: Supply Chain

R5.1.1	**Third-party software**	Third-party software, such as libraries used MUST be vetted for known security issues and if critical components SHOULD be audited. In the case of a library being deployed in the context, in which private keys are accessible, it MUST be supplied by a trusted party.	✓
R5.1.2	**Sourcing and Integration**	DIstribution channels of third party software MUST be reliably vetted. In the case of packet downloaded packages, integration checksums SHOULD be verified. Whenever possible, reliable package and dependency management software SHOULD be used. Content distribution networks SHOULD be avoided for integration.	✓

Observations

No	Description	Severity	Status
1	During the audit, a key derivation function with a salt value of a lesser than recommended length was encountered. The team has applied the recommendation to lengthen this value.	Minor	Resolved
2	A single admin for smart contract control was found to be stored on a cloud-based key vault. Given the scope of permissions assign to this key, this is considered acceptable.	Minor	Acknowledged
3	In the current version, some keys are stored on a firewall-protected internal Multichain node. Migration to a key management solution is scheduled for the near future.	Minor	Acknowledged
4	A key compromise protocol was not in existence but has now been added to the company's internal policy.	Minor	Resolved
5	Gateway redundancy was found partially implemented, allowing manual switch-over within a reasonable timeframe.	Minor	Acknowledged