# Protobs Firmware V1.0

## Introduction:

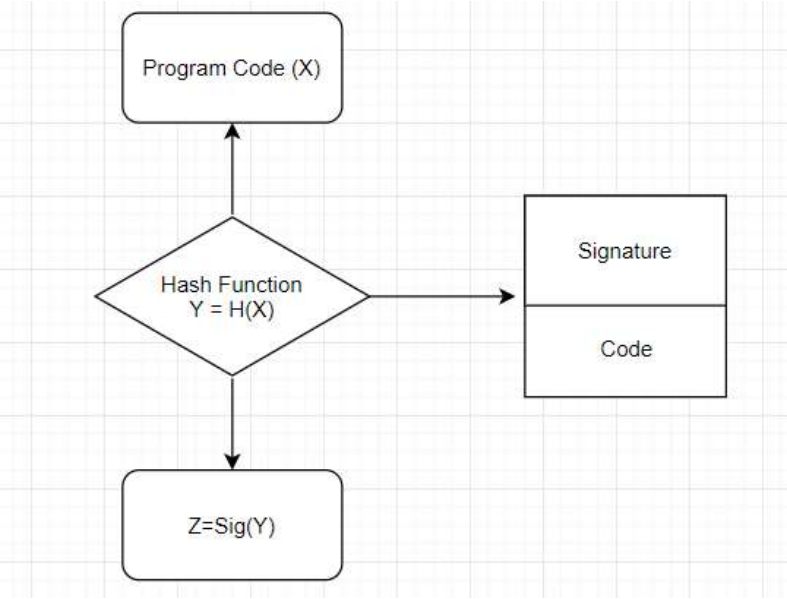
This document details about provisioning and digital signing of Protobs firmware.

To prevent counterfeiting or unauthorized access, software – which is typically stored in reprogrammable flash memory – must be updated securely. During the boot process – where the software is typically signed at a secure back end server and then installed using a boot loader – the system must verify the authenticity of the new firmware by checking the digital signature. The new firmware must be executed by the device only if this verification is successful.

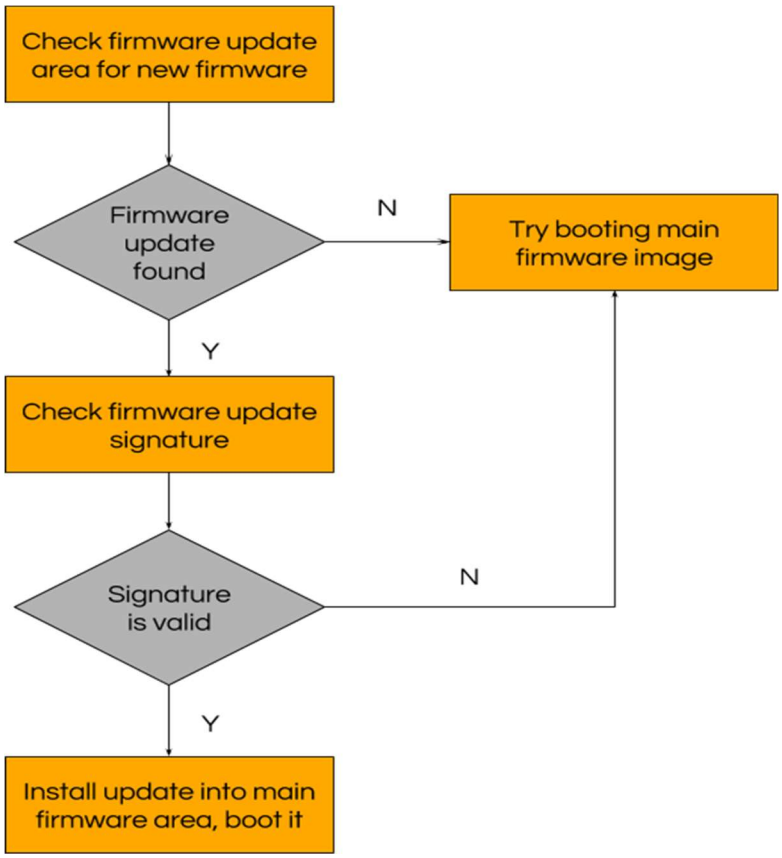
## Authenticity via Digital Signatures:

A digital signature provides integrity and authenticity; data that is digitally signed cannot be altered by a malicious third party without being detected by the receiver.

The image depicted below explains the protobs firmware digital signing process.



Our firmware provisioning process is as below.



## Our Security Considerations:

**Theft of private signing key:** Private signing keys that are not properly protected are at risk of theft, allowing a successful attacker to sign arbitrary code. Limited revocation mechanisms in some systems that rely on code signing exacerbate this threat.

**Issuance of unauthorized code signing certificates:** Weak protections on CA private keys used to issue code signing certificates, or weak vetting procedures used to issue those certificates, could allow an attacker to obtain one or more unauthorized code signing certificates.

**Misplaced trust in certificates or keys:** Verifiers could trust certificates or keys for code signing that were never intended to be used for code signing or that otherwise should not be trusted to sign code from a particular source. In some cases, verifiers may allow users to extend trust to untrustworthy certificates or signers.

**Signing of unauthorized or malicious code:** Code signing procedures could allow malicious or unauthorized code to be inadvertently signed, either as the result of a legitimate mistake, poor governance controls, an insider attack or a successful intrusion into software/firmware development or code signing systems. Similarly, intrusions into development systems or the code signing infrastructure itself, could result in malicious code being signed.

**Use of insecure cryptography:** Use of weak or insecure cryptographic algorithms or key generation methods could allow cryptanalytic or brute-force attacks to recover private keys or obtain fraudulent certificates. Future developments, such as new cryptanalysis, implementation vulnerabilities, or the development of a cryptographically-significant quantum computer, could render a previously-deployed system insecure

Our trusted developers and customers can upgrade device firmware from our cloud portal. You can download protobs firmware from (http://product.player2.htb/protobs/protobs_firmware_v1.0.tar). We recommend our dev team to do a sanity check at http://product.player2.htb/protobs/ before provisioning or pushing updates of the firmware.