Finding Report:
SwissPost Voting System - Signature Verification

Thomas Haines

November 2021

Executive Summary

When verifying signatures the SwissPost Voting system[1] failed to check that the signatures came from the party it expected to be corresponding with. This potentially allowed attacks on integrity by spoofing the input of honest parties. These attacks could be caught by the verifier, but since the relevant parts of the verifier were not published at the point the bug was submitted (March 2021), it was not possible to verify this. SwissPost has now confirmed how they intend to resolve this issue and pending some slight updates to the documentation and code it should indeed be fixed.

Key recommendations:

Check identity The signature verification should check that the corresponding party is correct. This could be done by checking that the X.509 certificate's subject field contains the expected name.

Check key usage All certificates in the chain should be checked to verify that they are being used for a valid purpose (using the attributes provided in RFC 5280).

Secure initialisation It is crucially important that the root certificates are correctly loaded. The documentation should clearly describe how this is accomplished.

Details

This section of the report describes the problem as it existed in March of 2021. The current public version includes several improvements which partially address this issue; SwissPost has confirmed they intended to update the documentation to completely address the attacks raised.

Many of the authentication checks in the system verify that the input is signed but not who it is signed by. Since the adversary has valid signing keys it can then impersonate honest parties. Examples appear to include validateChoiceCodesEncryptionKey in VotingCardSet-DataGeneratorServiceImpl and validateSignature in ChoiceCodesGenerationServiceImpl.

[1] This vulnerability was detected in version 0.7

For example, this could allow the adversary to impersonate the one honest return code control component until the logs of the control components are examined in 12.2.3 VerifyVotingPhase.

The key issue here is that the system, when verifying signatures, does not check that the attached X.509 certificate's subject field matches the expected party or that the keys are being used for a purpose which the signer of the key's certificate intended. No check has been found which prevents the control components from impersonating the one honest control component. This would allow the one honest control component to be bypassed, which breaks cast-as-intended verification.

No audit of the config phase described in the computational proof or system specification, at the time this issue was reported, would catch this attack on cast-as-intended. Nor was the verifier for the config phase in the repository. However, it was an open question if the attack (or a similar attack) would go undetected by the verifier spec and implementation that were (and to a significant extent are) unreleased and under development.

In conclusion, the identified vulnerability did appear to lead to manipulation that goes undetected by the voter but not by the system based on the then released material. However, the attack was caught by then unreleased checks.

Resolution

SwissPost has prevented the attack detailed in this report by a manual process which checks that the certificates used in the verification are the correct certificates. This certainly prevents the specific attack detailed in this report. More details on the resolutions should appear soon when SwissPost posts an issue on their GitLab repo related to this finding.

Related issue

A related issue stemming from the vulnerability first disclosed in this report is documented on the SwissPost voting gitlab.[2] In that case, the ability to impersonate honest components allowed the adversary to change the public key used to encrypt votes, which allowed attacks on individual verifiability. This also allowed attacks on privacy but those attacks are expressly allowed by the Swiss Ordinance which does not require privacy to hold when the voting server sends incorrect parameters to the voting device (Explanatory report 5.2.2 No 3.1).

Summary

The underlying vulnerabilities described here are still present in the SignatureChecker class in the verifier and the various signature verification implementations in the voting system. While there are no currently known attacks which exploit the vulnerabilities, we nevertheless strongly encourage SwissPost to patch the underlying vulnerabilities by implementing the key recommendations of this report.

Future versions of the SwissPost Voting system aiming for higher levels of assurance may wish to dispense with certificate chains entirely and load all certificates through a manual process; this would eliminate the need to trust any root certificate authority.

[2] `https://gitlab.com/swisspost-evoting/e-voting/e-voting/-/issues/1`