Review of code changes in e-voting-1.3.2.4

CLIENT	Cantons of Basel, St-Gallen and Thurgau	CLASSIFICATION	Public
DATE	October 22, 2023	AUTHOR	Philippe Oechslin
VERSION	1.0	DISTRIBUTION	Barbara Erni, Désirée Harmuth, Marius Kobi, Fabienne Mäder
GIT COMMIT	a3522a8		
STATUS	Final	MODIFICATIONS	

OS OBJECTIF SÉCURITÉ
Architecte de la sécurité informatique

1 Context

In national elections of October 2023 a software error in version 1.3.2.2 of the software prevented the mixing and decryption of the votes on the week-end of October 21st.

Swiss Post issued a patch to solve the issue.

Objectif Sécurité was tasked to confirm that the change had no impact on the correctness of the results, the secrecy of the votes or other requirements of the e-voting system set out in the Swiss ordinance on e-voting.

2 Changes in the code

The code that was adapted is related to the control components. The corresponding executable file is `control-component-runnable.jar`

The new version of the code is labelled 1.3.2.4 and was published in the afternoon of october 21st 2023 on Gitlab[1].

A build ceremony was carried out and the hashes of the resulting executable were also published on Gitlab. According to the protocol of the ceremony[2], the SHA-256 hash of the new version is the following:

```
control-component-runnable.jar  a3d1c230ed62bab284cb873266f59c53b37d5e667950746153884d35704bf020
```

This new version of the control component software was deployed in Swiss Post's infrastructure for all participating cantons (Basel, St-Gallen and Thurgau). No other elements of the system was redeployed.

The previous version that was deployed was v1.3.2.2. We studied the differences between the code of v1.3.2.2 and v1.3.2.4 and found the following changes:

2.1 Code changes in the control component

Except for changes where the version was adapted (1.3.2.2 replaced by 1.3.2.4) there is one change in the code of the control component:

control-component/src/main/java/ch/post/it/evoting/controlcomponent/tally/mixonline/MixDecryptProcessor.java — View file @ 4e2b1acf

```
@@ -7,6 +7,7 @@ import static ch.post.it.evoting.controlcomponent.ControlComponentsApplicationBo
 import static ch.post.it.evoting.controlcomponent.tally.mixonline.MixDecryptService.MixDecryptServiceOutput;
 import static ch.post.it.evoting.domain.SharedQueue.MIX_DEC_ONLINE_REQUEST_PATTERN;
 import static ch.post.it.evoting.domain.SharedQueue.MIX_DEC_ONLINE_RESPONSE_PATTERN;
+import static com.google.common.base.Preconditions.checkArgument;
 import static com.google.common.base.Preconditions.checkNotNull;
 import static com.google.common.base.Preconditions.checkState;

@@ -89,6 +90,10 @@ class MixDecryptProcessor {
                 final MixDecryptOnlineRequestPayload mixDecryptOnlineRequestPayload = objectMapper.readValue(messageBytes,
                         MixDecryptOnlineRequestPayload.class);

+                checkArgument(mixDecryptOnlineRequestPayload.nodeId() == nodeId,
+                        "The mix decrypt online request payload is not addressed to the current control component. [nodeId: %s, payloadNodeId: %s]", nodeId,
+                        mixDecryptOnlineRequestPayload.nodeId());
+
                 mixDecryptOnlineRequestPayload.controlComponentShufflePayloads().forEach(this::verifySignature);

                 final String electionEventId = mixDecryptOnlineRequestPayload.electionEventId();
```

Figure 1 Code changes in the control component

[1] https://gitlab.com/swisspost-evoting/e-voting/e-voting/-/tree/e-voting-1.3.2.4?ref_type=tags

[2] https://gitlab.com/swisspost-evoting/e-voting/e-voting-documentation/-/blob/master/Trusted-Build/E-Voting/Release-1.3.2/Protocols/231021_E-Voting-System_Post_1.3.2.4-control-component-runnable.jar_Trusted_Build_Bericht_TG_SG_BS.pdf

The code introduces a new argument check in the MixDecrypt processor. It validates that the nodeId of a new MixDecrypt payload is the same as the id of the node processing the request. This code is situated in the `onMessage` method of the MixDecrypt processor, which is called when a new message arrives.

The code appears almost at the top of the method. The only code that appears before the check reads the incoming bytes and casts them into a `MixDecryptionOnlineRequestPayload` object.

2.2 Code changes related to the authetication challenge

The new version of the code (v1.3.2.2.4) also contains changes in the voter portal, the voter client and the voting server. These changes are related to the authentication challenge algorithm, which is used when a voter submits a vote.

The new version of the voting client, portal and server has not been deployed. The changes in their code thus has no impact on the software used for mixing and decryption of the national elections on the weekend of October 21st.

2.3 Other changes in the code

The only other changes found in the code are related to changing the version number from 1.3.2.2 to 1.3.2.4.

3 Analysis and Conclusions

Without analysing the quality of the code or whether the code actually fixes the issue that prevented mixing and decryption, we can clearly see that it is executed before any operation is carried out on the votes or any cryptographic material that are stored in the control component.

The only impact that the code can have is that some payloads would not be processed that would otherwise have been processed. The same impact could also happen if a message was lost in transmission before it reached the control components.

The trust model of the cryptographic protocol does not imply that the communication channels between the control components are lossless. Lost messages are already taken into account in proofs, audits and other tests.

We thus conclude that the analysis and audits done on the previous code base are still valid after this change. The only thing that could happen would be that a bug in this code would make mixing and decryption impossible. It actually seems to be the opposite.