Data managment on the Belenios voting platform

July 25, 2022

Document written by Véronique Cortier, Pierrick Gaudry, and Stéphane Glondu.

Abstract

The goal of this document is to list the personal data stored on the Belenios server, at each step of an election, and to explain how they are managed (e.g. their life cycle). Personal data are used only to conduct the corresponding election. The use of email addresses is restricted to sending voting material to voters. The information used for statistical purposes, in order to report on Belenios activity, is described in Section 3.3.

This document corresponds to version 1.19 of the Belenios software, that implements version 1.17 of the specification.

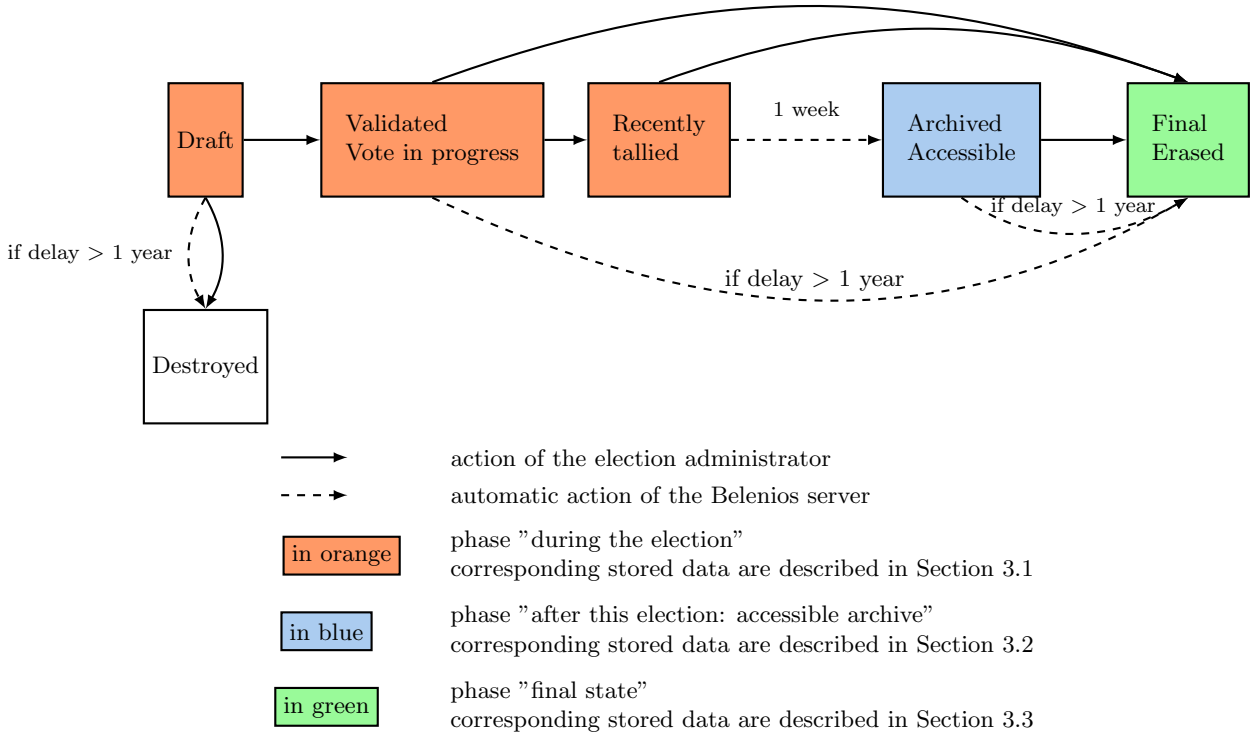
1 Preliminaries

Elections run through Belenios involve several entities, played either by persons or servers. The central point of interaction is the Belenios server, accessible through Internet, that stores various pieces of information. This document describes how data are managed by the server, exclusively. Third parties (administrator of the election, decryption authorities, credential authorities) are responsible of the data they manage. In what follows, when we describe data storage or data erasure, this only concerns the part that we control, namely the central Belenios server.

Data stored on Belenios are automatically backed up in order to be able to conduct an election even in the eventuality of some data corruption due to some physical issue or some software bug. Backups are stored encrypted on another computer, not accessible from Internet, and are erased after 1 month.

We use cookies to register language preference (if the voter choses a language different from the default language), to register the voter's consent to use cookies, and to link the requests made by a voter during the voting phase. Once a voter is done with voting, all identifying information are erased. In particular, we do not use cookies for statistical nor marketing purposes.

2 Typical evolution of an election



The election administrator decides on the progress of an election: draft, validated, recently tallied, archived and accessible (after the election), final and erased (final state). At any time, s.he may suppress the election data by moving to the final state. If an election is not yet validated (thus the election could not start), all data are purely and simply erased, after one year at most, or before if the administrator decides so. Once an election is tallied, the administrator may download an archive that contains:

- public data (cf Section 3),
- administrator data (cf Section 3).

This archive may be kept by the administrator while suppressing the data from the server (by moving to the final state). In any case, the Belenios server suppresses election data (moving to final state) after one year without any change in the election status. Moreover, one week after the election, the server moves a tallied election in the state "after election".

3 Details of data managed by the Belenios server

3.1 During the election

Before, during, and one week after the end of the election (tally), the following data are stored on the server.

public data (accessible by voters)

- title of the election
- questions of the election
- result of the election
- public keys of the decryption authorities
- public part of the voting credentials
- encrypted ballots

- proof of correct decryption

administrator data (accessible by the election administrator)
- list of eligible voters (email and login)
- list of voters who already voted, with timestamps
- in case the server generates the credentials itself (no external credential authority) and before the election starts (state "Draft"), the administrator may download the private part of the voters credentials. This is useful in case some voters lose their credential. This does not diminish the security if the server is trusted. In case the server is not fully trusted, an external credential authority mus be used.

server data (accessible by the administrators of the Belenios server)
- voters' passwords (salted and hashed), when the password mode is chosen. NB: by default, one-time passwords are used (OTP), that are not stored on the server
- general logs of the connections to the server
- file security.log (contains failed and suspicious revoting attempts)
- validation and tally date
- decryption key if no decryption authority has been chosen, or if the server is one of the decryption authorities
- internal database: register the link between voters and their public voting credential
- in the "threshold decryption mode" (k out of n keys are sufficient to decrypt the election): decryption keys of the authorities (trustees) are stored *encrypted* by the public key of the corresponding authority
- in case the server generates the credentials itself (no external credential authority), the private parts of the voters credentials are stored on the server before the election starts (state "Draft").

Server's administrators can also access the public and election administrator data.

Note: general logs of the connections to the server and the file security.log are stored outside the Belenios database. Therefore, they are managed differently: independently of the choices made by the administrator, these files are stored during 2 weeks and then erased.

3.2 After the election: accessible archive

After the tally (and during 1 year), the election is archived. Voters may access the election result and check its consistency w.r.t. the encrypted ballots of the election. The election administrator may also access the voter list, with the date of vote, when applicable. Data stored on the server correspond to the data previously listed (Section 3.1), **except**:
- the decryption key (if the server owned one of the decryption keys of the election),
- the link between voters and their public voting credential (stored in the internal database),
- in the "threshold decryption mode" (k out of n keys are sufficient to decrypt the election): decryption keys of the authorities (trustees) were stored *encrypted* by the public key of the corresponding authority. These encrypted keys are now erased.

3.3 Final state

1 year after tally (or if no tally, 1 year after the creation of the election), election data are erased. Only some information are kept for statistical purposes, in order to report on how the Belenios server is used. The election administrator may chose to erase an election (move to final state) before this delay of one year.

The following data are kept for statistical purposes.
- title of the election
- shape of the questions (type of the election, number of answers and number of choices), all strings of characters are replaced by the empty string. The questions and the possible answers are therefore erased.

- number of voters and number of received ballots
- identifier of the election administrator. Depending on the authentication mode, it can be her.his email address or a login.
- tally date (or validation date, if the election has not been tallied)
- choice of the authentication mode, and the corresponding address of the CAS server (if any)
- choice of the management mode for vote credentials
- regarding decryption authorities: total number and whether the server is one of them
- has the election been tallied?

4 Sensitivity of the data

We provide here our (subjective) analysis of the sensitivity of the data stored on the Belenios server, together with a brief analysis of the associated risks.

Extremely sensitive

- The stored decryption key, when no other decryption authority has been selected (mode without decryption authority).
  With the data available on the server (in particular the encrypted ballots), it is possible to read who has voted what. The mode is meant for test elections or for elections with absolutely no stake (e.g. choice of the pizza for tonight). It should not be used for any serious election.
  attack scenarios: The server is hacked (physically or remotely) or the server administrators are corrupted.

Very sensitive

- Internal database, that can be used to establish a link between a voter and her.his encrypted ballot.
  There is a risk against vote privacy in case encryption is not secure.
  attack scenarios: intrusion on the server AND the decryption keys leak (the corresponding risk has to be evaluated depending on the decryption authorities) or encryption is broken (quantum computer?).
- The decryption key when it is stored on the server and other decryption authorities have been selected (reminder: the case where the server is the only decryption authority is an extremely sensitive case, as explained above).
  attack scenarios: intrusion on the server AND the decryption keys leak (the corresponding risk has to be evaluated depending on the trustees) or encryption is broken (quantum computer?).
- The private parts of the voters credentials stored on the server before the election, when there is no external credential authority. There is a risk that some ballots are added (ballot stuffing).
  attack scenario: intrusion on the server before the election to retrieve the private credentials AND during the election to add ballots. This attack can be detected if the attacker does not update correctly the votig list. It may be detected by an attentive look at the voter list (voters that did not vote appear on the list).

Sensitive

- Voter list with timestamps: it can be used to establish a link between a voter and her.his encrypted ballot *provided that* the attacker has monitored arrival time of the ballots to the ballot box. It also tells who participated to the election and at which time.
- General logs of the connections to the server (ip, user-agent, date, url): quite similar to the voter list with timestamps.
  The risk is again the loss of vote privacy in case encryption is not secure.

attack scenarios: leakage of the voter list (or of the general logs) AND active monitoring of the ballot box AND the decryption keys leak (the corresponding risk has to be evaluated depending on the trustees) or encryption is broken (quantum computer?).

- Decryption keys of the decryption authorities (trustees), stored *encrypted* by the public key of the corresponding authority (in the "threshold encryption" mode only): it is possible to decrypt all the ballots IF the attacker can also learn the private keys of sufficiently many decryption authorities.
  attack scenarios: leakage of the voter list (or of the general logs) AND active monitoring of the ballot box AND leakage of sufficiently many private keys from the authorities (that should properly store their own keys) or encryption is broken (quantum computer?).

Moderately sensitive

- List of eligible voters (= list of emails). The risk is that emails are used in another context (e.g. spam, targeted phishing).
- List of hashed passwords (contains also the email list), when password-based authentication is used, which is not the default mode.
  The passwords are generated by the server and not by the user thus the risk associated to the loss of this table is minimal.