Dokumentation Fachschaftswahl und Urabstimmung 2021

Table of Contents

Dieses Dokument beschreibt die Durchführung der Fachschaftswahl und Urabstimmung 2021 an der Bauhaus-Universität Weimar. Es wird die Wahlsoftware Belenios in der Version 1.15 genutzt.

Dieses Dokument wird aus einem Git-Repository generiert und kann dort editiert werden. PDF-Version dieses Dokumentes

Überblick

- Zuerst wird eine kurze Einführung in die genutzte Wahlsoftware Belenios gegeben.
- Es folgt eine ausführliche Dokumentation des Ablaufs der Wahl für alle Beteiligten.
- Ein weiteres Kapitel widmet sich der Konformität der Wahl mit der aktuellen Wahlordnung des StuKo sowie den Sicherheitsanforderungen an Online-Wahlsoftware des BSI.

Wie funktioniert Belenios?

Bei diesem Dokument handelt es sich um eine Zusammenschrift die größtenteils auf Übersetzungen basiert, welche an unseren speziellen Anwendungsfall angepasst wurde. Die orginalen Dokumente können unter [belenios-howitworks], [inria-belenios-intro], [belenios-whodoeswhat] und [belenios-2019] abgerufen werden. Insbesondere wird auf die Veröffentlichung [belenios-2019] Bezug genommen, welche den Ablauf einer Wahl mit Belenios detailliert beschreibt.

Belenios

Belenios ist ein peer-reviewtes elektronisches Wahlprotokoll mit einer quelloffenen Implementierung in OCaml und JavaScript. Es wird aktiv von der französischen Forschungseinrichtung INRIA und der Université de Lorraine entwickelt und wird u.a. von der *International Association for Cryptologic Research* für Gremienwahlen genutzt.

Eignung für Fachschaftsratswahlen/Urabstimmung

Belenios ist interessant für uns weil es durch Nutzung von kryptographischen Verfahren eine sichere und transparente Wahl ermöglichen kann, bei der dem Wahlserver nicht in jedem Fall vertraut werden muss.

Das Kernprinzip basiert auf Verschlüsselung mit mehreren Schlüsseln: Jede Stimme einer Wahl wird mit dem öffentlichen Schlüssel der Wahl verschlüsselt, aber eine Entschlüsselung benötigt einen privaten Schlüssel der zwischen verschiedenen Autoritäten (Trustees) aufgeteilt ist. Um die Ergebnisse zu entschlüsseln müssen alle Trustees das Ergebnis zusammen entschlüsseln. Damit kann keine einzelne Person das finale Wahlergebnis entschlüsseln.

Auf einem ähnlichen Weg wird für jeden Wähler ein anonymer privater Schlüssel erstellt (Credential, Wahlschlüssel) dieser wird *nie* auf dem Wahlserver gespeichert, der nur den öffentlichen Teil des Schlüssels enthält. Damit kann jeder Wähler überprüfen ob seine Stimme im Endergebnis enthalten ist (individuelle Überprüfbarkeit). Darüber hinaus ist es jedem Wähler möglich zu überprüfen ob das Endergebnis auch den abgegeben Stimmen entspricht (universelle Überprüfbarkeit).

Im Falle der Fachschaftswahlen wird die Sicherheit der Wahl mit Hilfe von Aufgabenteilung

zwischen verschiedenen Autoritäten an der Universität sichergestellt:

- Der Wahlserver wird vom Referat-Digitales des StuKo betrieben, kennt aber den privaten Schlüssel der Wahl nicht und hat auch keine Kenntnis über die anonymen privaten Schlüssel der Wähler.
- Der private Schlüssel der Wahl wird auf mehrere Personen aus dem Umfeld der Universität aufgeteilt z.B. der Datenschutzbeauftragte oder Professoren/Mitarbeiter (Trustees). Um das Wahlergebnis zu bestimmen müssen sich alle zusammenfinden, eine Person alleine kann nichts entschlüsseln. Die einzelnen Stimmen der Wähler bleiben verschlüsselt, selbst für die Auszählung siehe: Auszählung.
- Die anonymen privaten Schlüssel der Wähler werden vom SCC generiert und per Mail verschickt. Weder die Trustees noch der Wahlserver kennt diese.
- Wählende können mit ihren anonymen privaten Schlüssel und den öffentlichen Schlüsseln auf dem Wahlserver kontrollieren, dass sowohl ihre Stimme gezählt wurde und auch alle Stimmen in das Endergebnis eingeflossen sind.

	Number of dishonest authorities							
Decryption trustees	$\leq t$	$\leq t$	$\leq t$	$\leq t$	$> t$	$> t$	$> t$	$> t$
Registrar	0	0	1	1	0	0	1	1
Voting Server	0	1	0	1	0	1	0	1
Verifiability	✓	✓	✓	✗	✓	✓	✓	✗
Privacy	✓	.	.	.	✗	✗	✗	✗

✓ indicates that the property is satisfied. ✗ indicates that the property is not satisfied. . indicates that there is no formal proof, yet no attack is known. As in Section 2.3, t is the threshold decryption parameter, *i.e.* at least $t + 1$ contributions are required to be able to decrypt.

Fig. 5. Trust assumptions for Belenios.

Figure 1. Belenios Sicherheitsgarantien

Details zum Wahlprozess sind unter [belenios-2019] nachlesbar.

Wählende

Um zu wählen, braucht ein Wählender:

- *einen Zugangscode* (Credential, Wahlschlüssel) (per E-Mail erhalten);
- *Universitäts-Anmeldedaten*

Über die Weboberfläche gibt der Wählende seinen Zugangscode ein und gibt seine Stimme ab.

Sein Computer berechnet dann lokal aus der abgegebenen Stimme, dem Credential und dem öffentlichen Schlüssel der Wahl den digitalen Stimmzettel, welcher der verschlüsselten Stimme entspricht.

Um die Überprüfung zu erleichtern, wird nur die Kontrollnummer (der Hash des Stimmzettels) angezeigt. Sobald der Stimmzettel abgeschickt wurde, kann der Wählende überprüfen, ob seine Stimme eingegangen ist, indem er überprüft, ob seine Kontrollnummer auf dem öffentlichen Stimmzettel erscheint (auf den man über die Webseite der Wahl zugreifen kann). Der Wähler kann auch den gesamten Stimmzettel sehen.

Ein Wähler kann mehrmals wählen, solange die Wahl geöffnet ist. Es wird nur der letzte Stimmzettel gezählt.

Der Zugangscode wird zum Signieren des Stimmzettels verwendet. Dies ist ein Schutz gegen *"ballot stuffing"*. Selbst wenn der Wahlserver kompromittiert ist, ist es unmöglich, gültige Stimmzettel hinzuzufügen. Der Server kennt zwar die (öffentlichen) Verifikationsschlüssel, aber niemand kann die entsprechenden privaten Signierschlüssel berechnen, die den Wählern per E-Mail zugeschickt werden.

Der Stimmzettel wird mit ElGamal-Verschlüsselung verschlüsselt. Der Chiffretext wird zusammen mit einem Zero-Knowledge-Beweis berechnet, der garantiert, dass eine gültige Wahl verschlüsselt wurde (z. B. nicht mehr als eine Stimme pro Kandidat). Mehr Informationen über Zero-Knowledge-Beweise gibt es unter [zk-slides]. Die genaue Spezifikation der Zero-Knowledge-Beweise von Belenios ist in der [belenios-spec] verfügbar und ein begleitender Entwurf mit den Sicherheitsbeweisen ist unter [belenios-zk-proof] verfügbar.

Wahlserver

Der Wahlserver authentifiziert die Wähler durch deren Universitäts-Anmeldedaten - dafür wird Shibboleth genutzt, so dass der Wahlserver auch keine Kenntnis über Uni-Login Passwörter der Wählenden hat. Der Login findet auf einem Server der Universität statt, lediglich eine kryptografisch signierte Nachricht identifiziert den Nutzer. Weiterhin führt der Wahlserver das Wählerverzeichnis und zeigt die eingegangenen Stimmzettel auf der öffentlichen Webseite der Wahl an, sofern die Stimmzettel gültig sind.

Der Wahlserver prüft die Gültigkeit der Zero-Knowledge-Proofs und verifiziert, dass die Stimmzettel korrekt signiert sind, wobei ein Schlüssel verwendet wird, der den anfänglichen öffentlichen Schlüsseln (Credentials) entspricht.

Generierung von Zugangscodes

Der Wahladministrator beauftragt das SCC, die Credentials zu generieren und zu versenden. Der Code zum Generieren von Credentials ist im [belenios-source] verfügbar und kann auch Offline mit Hilfe des `belenios-tool` ausgeführt werden.

Ein Credential ist eine reine Zufallszeichenfolge, aus der ein Signaturschlüsselpaar abgeleitet wird. Auf dem Wahlserver sind nur die entsprechenden Prüfschlüssel gespeichert. Auf diese Weise kann der Wahlserver prüfen, ob ein Stimmzettel von einem wahlberechtigten Wähler stammt (d. h. der Server prüft, ob der Stimmzettel mit einem Schlüssel signiert ist, der einem der Verifikationsschlüssel entspricht), aber selbst wenn der Server kompromittiert ist, kann kein Stimmzettel hinzugefügt werden (ohne entdeckt zu werden).

Auszählung

Standardmäßig speichert der Wahlserver den Entschlüsselungsschlüssel. Das Verschlüsselungsschema hat eine besondere Eigenschaft, die als Homomorphismus bezeichnet wird: Aus den verschlüsselten Stimmzetteln kann jeder die Verschlüsselung des Ergebnisses (die Summe der Stimmen für jeden Kandidaten) berechnen, indem er die Chiffretexte (ohne Verwendung eines Schlüssels) kombiniert. Auf diese Weise muss nur das verschlüsselte Endergebnis entschlüsselt werden, was die Privatsphäre der Wähler garantiert: **Der Stimmzettel eines einzelnen Wählers wird niemals entschlüsselt.**

Alternativ und von uns genutzt um das Vertrauen zu verteilen, kann der Wahladministrator den privaten Schlüssel auf mehrere Entschlüsselungsstellen (Trustees) aufteilen und an jeden Trustee einfach einen privaten Link senden, den er von der Belenios-Weboberfläche erhält. Wenn man auf diesen Link klickt, generiert der Trustee lokal seinen privaten Schlüssel - siehe Trustee Anleitung - und sendet den entsprechenden öffentlichen Schlüssel an den Wahlserver. Alle Autoritäten müssen zusammenarbeiten, um zu entschlüsseln, was bessere Datenschutzgarantien bietet (sie müssten sich alle absprechen, um die Stimmzettel zu entschlüsseln).

Die Entschlüsselungsinstanzen (d.h. der Wahlserver oder die vom Wahlleiter gewählten Trustees) erstellen zusätzlich einen Zero-Knowledge-Beweis der korrekten Entschlüsselung. Auf diese Weise kann jeder überprüfen, ob das Ergebnis der Wahl mit den auf dem Wahlserver gespeicherten Stimmzetteln übereinstimmt.

Wahlprüfer

Belenios ist Ende-zu-Ende-verifizierbar (end-to-end verifiable).

- Die Wähler können überprüfen, ob ihre Stimme gezählt wurde, indem sie überprüfen, ob ihr Stimmzettel auf dem Wahlserver erscheint (über eine öffentliche Webseite).
- Jeder kann überprüfen, dass das Ergebnis mit den auf der Wahlurne angezeigten Stimmzetteln übereinstimmt und dass dank der Zugangscodes kein Stimmzettel hinzugefügt wurde

Um die Gültigkeit einer Wahl mit dem [belenios-source] zu überprüfen siehe Audit der Wahldaten. Die genaue [belenios-spec] sowie [belenios-2019] und [belenios-whodoeswhat] sollte alle notwendigen Details liefern (aber natürlich sind Fragen willkommen).

Ablauf

Folgend ist beschrieben, wie die Fachschaftwahl mittels Belenios veranstaltet wird. Die beschriebenen Schritte werden jeweils für 5 unterschiedliche Wahlen durchgeführt. Jede Fachschaft und die Urabstimmung wird mit Belenios *als eigene Wahl* durchgeführt.

Rollen

- **Credential Authority** (SCC)
  - Diese Instanz übernimmt das Generieren und Versenden der Credentials.
- **Trustees** (Vertrauenswürdige Autoritäten der Universität)
  - Generieren die privaten Schlüssel der Wahl. Sie entschlüsseln und verifizieren nach der Wahl das Ergebnis.
- **Administrator** (Referat Digitales StuKo / Wahlkommission StuKo)
  - Konfiguriert die Wahl und infomiert Credential Authority und Trustees über notwendige Schritte
- **Wählende** (Studierende)
  - Wird von der Credential Authority über die Wahl informiert und wählt mit ihrem privaten Credential.

Terminologie

- **Credential** (private Credential, Wahlschlüssel)
  - Der private Teil des Schlüssels der an jeden Wählenden verschickt wird.
- **Public Credential**
  - Der öffentliche des Schlüssels der auf dem Wahlserver gespeichert wird.

Daten

- `voters.txt`
  - Einer Liste der E-Mail Adressen aller für eine Wahl wahlberechtigten Wählenden.
- `group.json`
  - Gruppen-Parameter für die Erzeugung der Schlüssel: [group] - siehe auch Kapitel 5 in [belenios-spec].

Ablauf *vor* der Wahl

Administrator

- Erstellt die Wahl unter https://wahl.m18.uni-weimar.de/admin

- Er konfiguriert die Fragen, welche auf dem Wahlzettel stehen sollen.
- Er lädt die `voters.txt` auf den Wahlserver hoch.
- Er trägt den Namen der Credential Authority ein.
- Er informiert die Credential Authority und sendet ihr folgende Informationen:
  - Link über den die öffentlichen Credentials hochgeladen werden
  - *UUID* der Wahl
  - `voters.txt`
- Er trägt die Namen und E-Mail Adressen der Trustees ein.
- Er informiert die Trustees und sendet ihnen folgende Informationen:
  - Link über den die öffentlichen Wahl-Schlüssel hochgeladen werden.

Credential Authority

- Mittels der *UUID*, der `voters.txt` und der `group.json` generiert sie für jeden Wählenden ein privates und öffentliches Credential.
- Sie speichert die öffentlichen Credential auf dem Wahlserver.
- Sie versendet die privaten Credentials an die Wählenden.
- Sie speichert die privaten Credentials während der Wahl an einem sicheren Ort ab.

Trustees

- Sie generieren jeweils ihren Teil des Wahlschlüssels.
- Sie speichern den öffentlichen Teil ihres Schlüssels auf dem Wahlserver.
- Sie speichern den privaten Teil des Schlüssels an einem sicheren Ort.

Ablauf *während* der Wahl

Administrator

- Er eröffnet die Wahl

Trustees

- Überprüfen ob die auf der Wahlseite angezeigten Fingerabdrücke ihrer Schlüssel korrekt sind. Siehe Überprüfung des Public-Key Fingerabdrucks.

Credential Authority

- Überprüft ob der auf der Wahlseite angezeigte Fingerabdruck des Wählerverzeichnisses korrekt ist.
- Sollte ein Wählender sein privates Credential verloren haben, so meldet er sich bei der Wahlkomission welche die Credential Authority informiert. Diese kann dann das jeweilige

private Credential erneut an den Wählenden versenden.

Wähler:in

- Sie gibt auf der Wahlseite das von der Credential Authority übermittelte Credential ein.
- Sie beantwortet die Fragen.
- Sie verschlüsselt ihre Stimme lokal mittels ihres privaten Schlüssels und der öffentlichen Schlüssel der Trustees.
- Sie speichert ihre verschlüsselte Stimme auf dem Wahlserver.
- Sie authentifiziert sich mit ihrem Universitäts-Login.
- Sie bestätigt ihre Stimme.

Ablauf *nach* der Wahl

Zu diesem Zeitpunkt ist für den Administrator nur einsehbar welche Wähler ihre Stimme abgegeben haben. Das verschlüsselte Ergebnis lässt keine Rückschlüsse auf die jeweilige Stimme der Wähler:in zu.

Credential Authority

- Vernichtet die privaten Credentials.

Administrator

- Er startet die Auszählung der Stimmen.
- Er informiert die Trustees und sendet ihnen folgende Informationen:
  - Link über den sie die Teilentschlüsselung des Wahlergebnisses durchführen können.

Trustees

- Sie öffnen den Link, den der Administrator an sie geschickt hat
- Auf der Seite wählen sie ihren privaten Schlüssel aus und führen die Entschlüsselung durch.

Administrator

- Auf der Wahladministrationsseite klickt er auf die Schaltfläche “Compute the result”
- Er validiert die Vollständikeit des Wahlergebnisses
- Zusammen mit der Wahlkomission veröffentlicht er das Wahlergebnis

Jeder

- Mit Hilfe des `belenios-tool` kann jeder die Vollständigkeit der Wahl verifizieren:

```
$ belenios-tool verify --url https://wahl.m18.uni-weimar.de/elections/<UUID>/
```

Wahladmin Checkliste

Es müssen 5 Wahlen erstellt werden. Eine pro Fakultät und eine für die Urabstimmung.

Erstellen einer neuen Wahl:

- ☐ Manuelles Credential-Management ausgewählt
- ☐ Titel der Wahl in Deutsch und Englisch
- ☐ Beschreibung benutzt dieses Format
- ☐ Sprachen sind “de en”
- ☐ Kontakt ist wahl@m18.uni-weimar.de
- ☐ 5 Fragen für die jeweils 5 Stimmen
  - ☐ Titel der Fragen “Erstimme”, “Zweitstimme”, …
  - ☐ 1 von 1 Antworten ist eingestellt
  - ☐ Ungültige Stimme (blank vote) ist erlaubt
  - ☐ Die Antworten aller 5 Fragen sind identisch
- ☐ Wahlkabine über Vorschaufunktion nochmals überprüft
- ☐ E-Mail-Adressen der Wähler sind eingetragen
- ☐ Anzahl der eingetragenen Wähler stimmt mit dem Wählerverzeichnis überein (“` voter(s) registered `”)
- ☐ Alle Trustees sind mit E-Mail-Addresse und Name eingetragen
- ☐ Der Name der Credential-Authority ist gesetzt (SCC)

Validierung des Wahl-Entwurfs:

- Draft herunterladen:
  - `$ wget https://wahl.m18.uni-weimar.de/draft/preview/<UUID>/election.json`
- Validierungs-Script ausführen:
  - `$ ./validate-studentcouncil-election.py /path/to/election.json`

Trustees

- Jede:r Trustee erhält für jede der 5 Wahlen eine E-Mail. Diese können wenn die Schritte oben erledigt sind im Abschnitt “Trustees” verschickt werden.
- Dazu bei jedem Trustee auf den “E-Mail”-Link klicken. In den Betreff der E-Mail sollte der Name der Wahl eingetragen werden. (z.B.: “Wahl FsR M - Link to generate the decryption key”)
- Da es für die Trustee-Emails keine deutsche Übersetzung gibt ist der Inhalt doppelt vorhanden.

Bitte einfach die Dopplung entfernen um Verwirrung zu vermeiden.

Credential Authority

- Für jede Wahl sollten der Credential Authority bis spätestens einen Tag vor Beginn der Wahl eine E-Mail mit folgenden Daten geschickt werden:
  - ▢ Wählerverzeichnis als Textdatei (`.txt`) (pro Zeile eine Wähler:in, nur die E-Mail-Adresse)
  - ▢ Hash des Wählerverzeichnisses
  - ▢ E-Mail-Vorlage
  - ▢ Link unter dem die Credential-Authority die öffentlichen Credentials hochladen kann
  - ▢ UUID der Wahl (steht im Link für die Credential-Authority)
- Nachdem die Credential-Authority die Wahlbenachrichtigungen verschickt hat sollte die Wahl so schnell wie möglich erstellt/eröffnet werden um Verwirrung bei den Wählenden zu vermeiden.

Bevor die Wahl eröffnet werden kann

- ▢ Hat die Credential-Authority die öffentlichen Credentials hochgeladen? (Indiz: Das eintragen von Wählenden ist nicht mehr möglich.)
- ▢ Haben alle Trustees ihre Schlüssel erstellt? (Indiz: Auf der Trustee-Seite steht in der "DONE"-Spalte "Yes".)

Eröffnen der Wahl:

- Auf der Wahladmin-Seite durck klicken auf "Create election" kann die Wahl eröffent werden.

Achtung: Nach diesem Schritt können keine Änderungen mehr an den Wahldaten vorgenommen werden.

- Ab jetzt könnte jede:r mit einem Credential wählen. Deshalb vorerst auf "Close election" klicken.
- Datum für automatisches eröffnen und beenden eintragen (und speichern). Zeit bis zur Wahl kann auf der jeweiligen Wahlseite überprüft werden.
- Zeitangaben sind in UTC deshalb müssten folgende Angaben verwendet werden
  - Eröffnung: `2021-05-17 22:00:00`
  - Schließung: `2021-05-20 22:00:00`

Nach der Wahl:

- Auf der Wahl-Adminseite auf "Proceed to vote counting" klicken. (Dafür muss die Wahl geschlossen sein.)
- E-Mails auf die oben beschriebene Art (mit Betreff) an die Trustees verschicken.

Der folgende Teil sollte als protokollierte (öffentliche) Sitzung der Wahlkomission stattfinden.

- Nachdem alle Trustees das Ergebnis entschlüsselt haben kann das Ergebnis berechnet werden. Mittels des Auszählungs-Scripts und der *UUID* der jeweiligen Wahl kann das finale Ergebnis ausgezählt werden.

```
$ ./tally.py <UUID>
```

Trustee Anleitung

Eine kurze Erklärung zur Funktionsweise von Belenios finden Sie hier.

Wenn Sie Ihre Schlüssel selbst ohne Webbrowser generieren möchten benötigen Sie das `belenios-tool`.

Vor der Wahl

Vor der Wahl erhalten Sie vom Wahladministrator eine E-Mail mit der Bitte ihren Wahlschlüssel zu generieren. Öffnen Sie den in der E-Mail enthaltenen Link.

Standard-Modus

In diesem Modus müssen alle Trustees an der Entschlüsselung des Wahlergebnisses teilnehmen.

Sie werden nun aufgefordert Ihren Wahlschlüssel zu generieren.

Schlüsselgenerierung

- Klicken Sie auf die Schaltfläche “Generate private key”.

Optional: Manuelles generieren von `*.privkey` und `*.pubkey`

```
$ wget -O group.json https://wahl.m18.uni-
weimar.de/static/groups/default.json ①
$ belenios-tool trustee-keygen --group group.json ②
```

① Die `group.json` herunterladen

② Mit `belenios-tool` den Schlüssel erzeugen

③ Den Inhalt von `*.pubkey` in das "Data"-Textfeld kopieren und mit "Submit" bestätigen.

- Folgen Sie den Anweisungen auf der Website unter "Instructions"

1. Laden Sie Ihren privaten Schlüssel herunter und speichern Sie ihn an einem sicheren Ort. (**Verliert ein Trustee ihren Schlüssel, dann kann das Wahlergebenis nicht mehr entschlüsselt werden.**)
2. Speichern Sie den Fingerabdruck ihres öffentlichen Schlüssels. (Mittels dieses Fingerabdrucks können Sie später auf der Wahlseite überprüfen, ob der Wahlserver den korrekten Schlüssel verwendet.)
3. Senden Sie Ihren öffentlichen Schlüssel an den Wahlserver indem Sie auf die Schaltfläche “Submit” klicken.
4. Ihnen sollte nun eine Bestätigung angezeigt werden.

Threshold-Modus

In diesem Modus muss nur ein Teil der Trustees an der Entschlüsselung des Wahlergebnisses teilnehmen.

- Sie werden nun aufgefordert gemeinsam mit den anderen Trustees den Wahlschlüssel zu generieren.

Schritt 1

- Klicken Sie auf die Schaltfläche “Generate private key”.

Optional: Manuelles generieren von `*.cert` und `*.key`

```
$ wget -O group.json https://wahl.m18.uni-
weimar.de/static/groups/default.json ①
$ belenios-tool threshold-trustee-keygen --step 1 --group group.json ②
```

① Die `group.json` herunterladen

② Den privaten Schlüssel generieren

③ Den Inhalt von `*.cert` in das "Data"-Textfeld kopieren und mit "Submit" bestätigen.

- Folgen Sie den Anweisungen auf der Website unter "Instructions"
  1. Laden Sie Ihren privaten Schlüssel herunter und speichern Sie ihn an einem sicheren Ort. (Verlieren mehrere Trustees ihren Schlüssel, dann kann auch im Threshold-Modus das Wahlergebenis nicht mehr entschlüsselt werden.)
  2. Speichern Sie den Fingerabdruck ihres öffentlichen Schlüssels. (Mittels dieses Fingerabdrucks können Sie später auf der Wahlseite überprüfen, ob der Wahlserver den korrekten Schlüssel verwendet.)
  3. Senden Sie Ihren öffentlichen Schlüssel an den Wahlserver indem Sie auf die Schaltfläche “Submit” klicken.

Für den nächsten Schritt warten Sie bitte bis alle Trustees den ersten Schritt durchgeführt haben.

Schritt 2

Nachdem alle Trustees ihre Schlüsselpaare erzeugt haben, können Sie nun Ihren Teil des Entschlüsselungs-Schlüssel generieren.

- Folgen Sie den Anweisungen auf der Website unter "Instructions"
    1. Geben Sie Ihren privaten Schlüssel in erste Textfeld ein und bestätigen Sie mit der Schaltfläche “Proceed”.
    2. Überprüfen Sie, dass das zweite Textfeld nun nicht mehr leer ist.
    3. Senden Sie die generierten Daten an den Wahlserver indem Sie auf die Schaltfläche “Submit” klicken.

Für den nächsten Schritt warten Sie bitte bis alle Trustees den zweiten Schritt durchgeführt haben.

Schritt 3

Nun haben alle Trustees ihre geheimen Anteile generiert. Fahren Sie mit den abschließenden Überprüfungen fort, damit die Wahl validiert werden kann.

- Folgen Sie den Anweisungen auf der Website unter "Instructions"
    1. Geben Sie erneut Ihren privaten Schlüssel in das erste Textfeld ein und bestätigen Sie mit der Schaltfläche “Proceed”.
    2. Überprüfen Sie, dass das zweite Textfeld nun nicht mehr leer ist.
    3. Senden Sie die generierten Daten an den Wahlserver indem Sie auf die Schaltfläche “Submit” klicken.
    4. Überprüfen Sie ob nun ob folgende Nachricht angezeigt wird: “Your job in the key establishment protocol is done!” Wenn ja folgend Sie den angezeigten Anweisungen.

Ihr Teil in der Vorbereitung der Wahl ist nun erledigt. Der Wahladministrator wird Sie kontaktieren sobald weitere Schritte notwendig sind.

Nach der Wahl

Vor der Wahl erhalten Sie vom Wahladministrator eine E-Mail mit der Bitte an der Entschlüsselung des Wahlergebnisses teilzunehmen. Öffnen Sie den in der E-Mail enthaltenen Link.

- Geben Sie Ihren privaten Schlüssel in das dafür vorgesehene Textfeld ein oder wählen Sie ihn über die Schaltfläche “Browse…” aus.
- Klicken Sie auf die Schaltfläche “Generate your contribution to decryption”
- Überprüfen Sie, dass das “Data”-Textfeld nun nicht mehr lehr ist.
- Senden Sie die generierten Daten an den Wahlserver indem Sie auf die Schaltfläche “Submit” klicken.

Überprüfung des Public-Key Fingerabdrucks

Auf der Wahlseite sind die Figerabdrücke der öffentlichen Schlüssel aller Trustees gelistet.

Um zu überprüfen ob Ihr Fingerabdruck stimmt können Sie den angezeigten Fingerabdruck mit dem den Sie während der Schlüsselgenerierung abgespeichert haben vergleichen.

Falls Sie Ihre Schlüssel mit dem Belenios-Tool generiert haben können Sie den Publickey-Hash mit folgendem Befehl berechnen:

```
$ cut --delimiter=":" --fields=5 <FILENAME>.pubkey | sed -e 's/"}/"/g'| tr -d '\n'|
openssl dgst -binary -sha256 | openssl base64 | tr -d "=" ①
```

① Der Schlüssel liegt im JSON-Format vor, die Kommandozeile extrahiert den Schlüssel und berechnet mit `openssl` einen SHA256-Hash und gibt ihm in Base64-Format zurück.

oder

```
$ cut --delimiter=":" --fields=5 <FILENAME>.pubkey | sed -e 's/"}/"/g'| tr -d '\n'|
belenios-tool sha256-b64 ①
```

① Wie oben - hier wird anstatt `openssl` das `belenios-tool` genutzt um den Hash zu berechnen.

Audit der Wahldaten

Mit `belenios-tool`: automatisch

```
$ belenios-tool verify --url https://wahl.m18.uni-weimar.de/elections/<UUID> ①
```

① In dieser Form lädt das Belenios-Tool alle nötigen Daten herunter und überprüft, dass keine Unstimmigkeiten auftreten.

Mit `belenios-tool`: manuell

- Sie können sich die folgenden Daten von der Wahlseite manuell herunterladen und die Verifikation so ausführen:
  - `election.json` (Wahlparameter)
  - `trustees.json` (öffentliche Schlüssel der Trustees)
  - `public_creds.txt` (öffentlicher Teil der Credential)
  - `ballots.json` (verschlüsselte Stimmzettel)
- Wenn sich alle genannten Daten im Ordner `./wahldaten` befinden ist eine manuelle Überprüfung so möglich:

```
$ belenios-tool verify --dir ./wahldaten
```

Mit `belenios-tool`: manuelle Differenz zur Überprüfung während der Wahl

```
$ belenios-tool verify-diff --dir1 ./alte-wahldaten --dir2 ./neue-wahldaten ①
```

① Mit diesem Befehl können Sie während der Wahl überprüfen, dass sich die abgebenen Stimmzettel nicht auf ungewöhnliche Art und Weise verändern. Dafür laden Sie sich einfach die zuvor beschrieben Daten zu Beginn der Wahl (oder zu Ihrem gewählten Stichzeitpunkt) herunter und zu einem späteren Zeitpunkt dann erneut. Dann können Sie die Entwicklung des Wahlergebenisses zwischen diesen zwei Zeitpunkten überprüfen.

belenios-tool

`belenios-tool` ist ein CLI-Tool mit dem alle wesentlichen Schritte für eine Wahl auch ohne die Website ausgeführt werden können.

Es gibt mehrere Varianten zur Installation. Da die Wahl mit der aktuellen Version 1.15 von Belenios durchgeführt wird sind die Versionen in den Paketquellen von Debian und Ubuntu leider nicht nutzbar, da diese derzeit noch ältere Versionen beinhalten.

`belenios-tool verify --url` ist aus derzeit noch unklaren Gründen sehr langsam. Es ist besser die benötigten Daten vorher in einem Ordner zu speichern und die manuelle Verifikation zu nutzen.

Selbst kompilieren

Zum Installieren des Belenios-Tools können Sie diese Anleitung befolgen.

Kurze Zusammenfassung zum selbst kompilieren unter Debian/Ubuntu:

```
$ wget https://gitlab.inria.fr/belenios/belenios/-/archive/1.15/belenios-1.15.zip ①
$ unzip belenios-1.15.zip ②
$ cd belenios-1.15 ③
$ sudo apt install bubblewrap build-essential libgmp-dev libpcre3-dev pkg-config m4
libssl-dev libsqlite3-dev wget ca-certificates zip unzip libncurses-dev zlib1g-dev
libgd-securityimage-perl cracklib-runtime jq ④
$ ./opam-bootstrap.sh ⑤
$ make ⑥
```

① Herunterladen

② Entpacken

③ In das eben entpackte Verzeichnis wechseln

④ Notwendige Abhängigkeiten zum kompilieren installieren

⑤ Die Ocaml-Paketverwaltung (`opam`) initial einrichten

⑥ Belenios kompilieren - unter `./_build/install/default/bin/belenios-tool` im entpackten Ordner

ist das `belenios-tool` dann verfügbar.

Selbst kompilieren mit Docker

Folgenden Source als `Dockerfile` speichern und mit `docker build . -t belenios` wird ein Docker-Image gebaut welches `belenios-tool` aus dem Sourcen in der Version 1.15 baut.

```
FROM archlinux:base-devel-20210523.0.23638

RUN set -x \
  && pacman-db-upgrade \
  && pacman -Syyyu opam git curl --noconfirm

RUN useradd --create-home belenios

USER belenios
WORKDIR /home/belenios
RUN set -x \
        && git clone -b 1.15 https://gitlab.inria.fr/belenios/belenios \
        && cd belenios \
        && cp .opamrc-nosandbox /home/belenios/.opamrc \
        && opam init -y \
        && eval $(opam env) \
        && opam install -y dune atdgen zarith cryptokit cmdliner \
        && make

ENV PATH $PATH:/home/belenios/belenios/_build/install/default/bin
```

Das Image kann z.B. folgermaßen genutzt werden:

```
$ docker run -it belenios belenios-tool verify --url https://wahl.m18.uni-
weimar.de/elections/<UUID>
```

Vorkompilierte Binary für Ubuntu 20.04

Alternativ gibt es hier eine bereits kompilierte Version. (Ubuntu 20.04)

Konformität mit der Wahlordnung des StudierendenKonvents

Die folgenden Diskussionen zur Konformität der Belenios-Wahlsoftware mitsamt Wahlablauf wurden zwar nach besten Wissen erstellt aber es nicht auszuschließen, dass sich Missverständnisse und Fehler eingeschlichen haben. Fehler und Ungereimtheiten bitte an digitales@m18.uni-weimar.de senden oder im Git-Repository dieses Dokumentes ein Issue erstellen und/oder einen Merge-Request mit Korrekten.

Für die Fachschaftswahlen und Urabstimmung ist die Wahlordnung des StudierendenKonvents [stuko-wahlordnung] die Grundlage. In einer überarbeiteten Fassung (MdU steht noch aus, aktuelle Version: [stuko-wahlordnung-neu]) sind Online-Wahlen zulässig und es wird auf § 29 der Wahlordnung der Bauhaus-Universität verwiesen: [buw-wahlordnung-neu], welcher im Zuge der Corona-Pandemie im April 2020 eingefügt wurde und Bedingungen für Elektronische Wahlen beschreibt.

Auszug: Erste Änderung Wahlordnung der Studierendenschaft der Bauhaus-Universität Weimar

§ 8a Onlinewahl

Sofern diese Wahlordnung keine abweichenden Regelungen vorsieht, kommen im Falle elektronischer Wahlen die Bestimmungen der §§ 29a – 29e der Ersten Änderung der Wahlordnung der Bauhaus-Universität Weimar (MdU 17/2020 vom 30. April 2020) entsprechend zur Anwendung.

In diesen Zusammenhang wurden Mindestanforderungen an die Durchführung einer elektronischen Wahl formuliert. Im folgenden wird erläutert wie die mit Belenios durchgeführte Fachschaftswahl die gestellten Anforderungen zu lösen versucht - siehe auch: Wie funktioniert Belenios?

§ 29a Stimmabgabe bei der Elektronischen Wahl

(1) Die Wahlberechtigten erhalten durch das Wahlamt das Wahlschreiben mit den Zugangsdaten sowie Informationen zur Durchführung der Wahl und der Nutzung des Wahlportals. Das Wahlportal ermöglicht die Stimmabgabe mittels Aufruf eines elektronischen Stimmzettels.

Für jeden Wahlberechtigten generiert die Credential Authority (SCC) einen personalisierten Wahlschlüssel (Credential), der Zusammen mit Informationen zur Wahl und dem Link zum jeweiligen Wahlportal per E-Mail verschickt wird. Der Wahlberechtigte muss sich weiterhin während der Wahl durch seinen Universitätslogin legitimieren - eine Stimmabgabe ist nur möglich wenn der Wählende sowohl über den korrekten Wahlschlüssel sowie gültige Uni-Login Zugangsdaten verfügt.

(2) Die Stimmabgabe erfolgt persönlich und unbeobachtet in elektronischer Form. Die Authentifizierung des Wahlberechtigten erfolgt durch die im Wahlschreiben genannten Zugangsdaten am Wahlportal. Der elektronische Stimmzettel ist entsprechend den im Wahlschreiben und im Wahlportal enthaltenen Anleitungen elektronisch auszufüllen und abzusenden…

Eine persönliche und unbeobachtete Stimmabgabe ist nicht kontrollierbar bei einer elektronischen Wahl. Die Authentifizierung des Wahlberechtigten basiert auf dem geheimeinen Wahlschlüssel der an die persönliche Uni-Mail Addresse verschickt wird, sowie der Überprüfung des Universitätslogins - also nur mit beiden Informationen ist eine Wahl möglich.

(...) (Dabei ist durch das verwendete elektronische Wahlsystem sicherzustellen, dass das Stimmrecht nicht mehrfach ausgeübt werden kann. Die Speicherung der abgesandten Stimmen muss anonymisiert und so erfolgen, dass die Reihenfolge des Stimmeingangs nicht nachvollzogen werden kann. Die Wahlberechtigten müssen bis zur endgültigen Stimmabgabe die Möglichkeit haben, ihre Eingabe zu korrigieren oder die Wahl abzubrechen.

Das genutzte Wahlsystem Belenios stellt kryptographisch sicher, dass das Stimmrecht nicht mehrfach ausgeübt werden kann. Der personalisierte Wahlschlüssel signiert die Stimmabgabe und wird zur Überprüfung ob mehrfach gewählt wurde genutzt. Siehe [belenios-2019] Kapitel 3.1. Die abgesandten Stimmen werden mit dem öffentlichen Schlüssel der Wahl verschlüsselt und mit Hilfe des privaten Credentials des Wählenden signiert und sind durch dritte ohne Kenntnis des privaten Credentials nicht zuordbar. Eine Entschlüsselung ist nur für das Endergebnis, nicht für einzelne Stimmen notwendig - siehe [belenios-2019] Kapitel 2. Weiterhin wird durch die Aufteilung des zur Entschlüsselung der Wahl notwendigen privaten Schlüssels auf mehrere Trustees sichergestellt, dass alle Trustees zusammenarbeiten müssen für die Entschlüsselung des Wahlergebnisses. Keine einzelne Person ist so in der Lage das Ergebnis zu entschlüsseln. Als Trustees wurden unabhängige Mitarbeiter der Universität ausgewählt.
Die Reihenfolge des Stimmeingangs ist durch die Verschlüsselung der Stimmen nicht nachvollziehbar.
Die Wahlberechtigten haben bis zum Ende der Wahl die Möglichkeit ihre Stimme zu korrigieren bzw. ungültig zu wählen. Der letzte abgegebene Stimmzettel fleißt in die Auszählung mit ein.

(...) Ein Absenden der Stimme ist erst auf der Grundlage einer elektronischen Bestätigung durch den Wähler zu ermöglichen. Die Übermittlung muss für den Wähler am Bildschirm erkennbar sein. Mit dem Hinweis über die erfolgreiche Stimmabgabe gilt diese als vollzogen.

Die engültige Übermittlung der Stimme ist klar durch einen Dialog auf der Wahlseite dargestellt. Nach erfolgreicher Stimmabgabe wird weiterhin eine E-Mail an den Wählenden versandt mit Informationen zur Überprüfung, ob die Stimme auch mitgezählt wird.

(3) Bei der Stimmeingabe darf es durch das verwendete elektronische Wahlsystem zu keiner Speicherung der Stimme des Wählers in dem von ihm hierzu verwendeten Computer kommen. Es muss gewährleistet sein, dass unbemerkte Veränderungen der Stimmeingabe durch Dritte ausgeschlossen sind.

Belenios speichert die Stimme zu keiner Zeit auf dem Computer des Wählers. Unbemerkte Veränderungen der Stimmeingabe durch dritte sind durch die Signierung der abgegeben Stimme durch den Wahlschlüssel des Wählers sowie der der Verschlüsselung mit dem öffentlichen Schlüssels der Wahl und der notwendigen Legitimation durch den Uni-Login ausgeschlossen bzw. würden bei einer Prüfung der Wahl auffallen. Hinzufügen von Stimmen ("Ballot Stuffing") wird verhindert in dem die Credential Authority (SCC) unabhängig von der Wahladministration/Wahlserver die Wahlschlüssel für die Wählerliste generiert - die privaten Credentials werden an die Wählenden verschickt, die öffentlichen Credentials auf dem Wahlserver gespeichert. So ist ein nachträgliches Hinzufügen von Wählern nicht möglich. Nach dem Start der Wahl ist eine Veränderung der Wählerliste nicht mehr möglich.

(...) Auf dem Bildschirm muss der Stimmzettel nach Absenden der Stimmeingabe unverzüglich ausgeblendet werden. Das verwendete elektronische Wahlsystem darf die Möglichkeit für einen Papierausdruck der abgegebenen Stimme nach der endgültigen Stimmabgabe nicht zulassen. Die Speicherung der Stimmabgabe in der elektronischen Wahlurne muss nach einem nicht nachvollziehbaren Zufallsprinzip erfolgen. Die Anmeldung am Wahlsystem, die Auswahl und Abgabe der Stimme sowie persönliche Informationen und IP-Adressen der Wahlberechtigten dürfen nicht protokolliert werden.

Belenios blendet den Stimmzittel nach Stimmeingabe unverzüglich aus. Eine Möglichkeit des Papierausdrucks der abgegeben Stimme nach der engültigen Stimmabgabe ist nicht möglich, da diese nach der Stimmeingabe sofort ausgeblendet wird. Die Stimmabgabe wird lokal verschlüsselt und dann erst auf dem Wahlserver gespeichert und ist dem Wählenden nicht zuordbar. Es werden keine persönlichen Informationen gespeichert, lediglich die verschlüsselte Stimme. Der Webserver und das Wahlsystem sind so konfiguriert, dass IP-Addressen nicht protokolliert werden.

§ 29 e Technische Anforderungen

(1) Elektronische Wahlen dürfen nur dann durchgeführt werden, wenn das verwendete elektronische Wahlsystem aktuellen technischen Standards, insbesondere den Sicherheitsanforderungen für Online Wahlprodukte des Bundesamtes für Sicherheit in der Informationstechnik entspricht. Das System muss die in den nachfolgenden Absätzen aufgeführten technischen Spezifikationen besitzen. Die Erfüllung der technischen Anforderungen ist durch geeignete Unterlagen nachzuweisen.

Die genutzte Wahlsoftware Belenios implementiert "State of the Art" Sicherheit für Internet-Wahlen - insbesondere wird das Wahlgeheimnis durch Verschlüsselung geschützt und durch eine Trennung von Aufgaben ("Seperation of Duties") ist die Integrität der Wählerliste und Entschlüsselung der Wahl an verschiedene unabhängige Autoritäten innerhalb der Universität aufgeteilt, die für eine Manipulation alle zusammenarbeiten müssten. Die Transparenz der Wahl wird durch "End-to-end verifiablity" sichergestellt. Jeder Wahlberechtigte kann überprüfen, dass seine Stimme gezählt wurde und nur Personen auf der Wählenden-Liste dürfen wählen. Die Überprüfbarkeit basiert darauf, dass die Liste der (verschlüsselten) Stimmen öffentlich ist und Stimmzettel mit dem Wahlschlüssel des Wählenden signiert sind. Siehe Wie funktioniert Belenios?, [belenios-2019]. Es gibt jedoch keine offizielle Zertifizierung des BSI für Belenios. Eine Diskussion der Anforderungen und wie Belenios diese löst gibt im Abschnitt BSI-CC-PP-0037.

(2) Zur Wahrung des Wahlgeheimnisses müssen elektronische Wahlurne und elektronisches Wahlverzeichnis auf verschiedener Serverhardware geführt werden. Das Wahlverzeichnis soll auf einem universitätseigenen Server gespeichert sein.

Das elektronische Wahlverzeichnis ist in Belenios nochmals aufgeteilt: Es gibt eine Liste der Wahlberechtigten je Wahl - dafür wird lediglich die Universitäts-Mailaddresse genutzt. Diese Liste wurde durch die Credential Authority (SCC) erstellt und ist nochmals auf dem Wahlserver vorhanden um eine Bestätigung der Stimmabgabe durch den Uni-Login zu ermöglichen. Technisch ist es dem Wahlserver nicht möglich eine Stimme einer E-Mail Addresse zuzuordnen. Die für die Wahl notwendigen privaten Wahlschlüssel (Credential) werden ausschließlich durch die Credential Authority (SCC) generiert und versendet. Der Wahlserver/Wahlurne hat keine Kenntnis über die privaten Wahlschlüssel. Dort sind lediglich die öffentlichen Schlüssel der Wählerliste hinterlegt, mit denen jedoch keine Rückschlüsse auf die Identität der Wählenden möglich sind.

(3) Die Wahlserver müssen vor Angriffen aus dem Netz geschützt sein, insbesondere dürfen nur autorisierte Zugriffe zugelassen werden. Autorisierte Zugriffe sind insbesondere die Überprüfung der Stimmberechtigung, die Speicherung der Stimmabgabe zugelassener Wähler, die Registrierung der Stimmabgabe und die Überprüfung auf mehrfacher Ausübung des Stimmrechtes (Wahldaten). Es ist durch geeignete technische Maßnahmen zu gewährleisten, dass im Falle des Ausfalles oder der Störung eines Servers oder eines Serverbereiches keine Stimmen unwiederbringlich verloren gehen können.

Eine Stimmabgabe ist nur möglich wenn der Wahlberechtigte sowohl seinen privaten Wahlschlüssel kennt, wie auch seinen Universitätslogin. Der administrative Zugriff auf den Wahlserver ist lediglich den autorisierten Wahladministratoren gestattet. Es gibt eine externe Sicherung der Wahldaten für den Fall einer Störung. Die Autorisierung für die Überprüfung der Stimmberechtigung (privates Credential und Uni-Login), die Speicherung der Stimmabgabe (verschlüsselt) und die Überprüfung auf mehrfache Ausübung des Stimmrechts (durch Nutzung von privaten Credential je Wählenden, mit dem die Stimmabgabe signiert wird und durch die dem Wahlserver bekannten öffentlichen Schlüssel der Wählerliste validiert werden kann) sind durch die Architektur von Belenios gelöst. Für Details siehe Wie funktioniert Belenios?, [belenios-2019].

(4) Das Übertragungsverfahren der Wahldaten ist so zu gestalten, dass sie vor Ausspäh- oder Entschlüsselungsversuchen geschützt sind. Die Übertragungswege zur Überprüfung der Stimmberechtigung des Wählers sowie zur Registrierung der Stimmabgabe im Wahlverzeichnis und die Stimmabgabe in die elektronische Wahlurne müssen so getrennt sein, dass zu keiner Zeit eine Zuordnung des Inhalts der Wahlentscheidung zum Wähler möglich ist.

Dies ist durch die Nutzung von privaten und öffentlichen Schlüsseln in Belenios gewährleistet. Die Stimmberechtigung wird komplett unabhängig von der elektronischen Wahlurne durch die von der Wahlurne/Wahlserver unabhängige Generierung der privaten Credentials durch die Credential Authority (SCC) sichergestellt. Die öffentlichen Credentials auf dem Wahlserver lassen dritten keinen Rückschluss auf die Identität des Wählers zu. Weiterhin ist die Wahlurne/Server nicht im Besitz des privaten Schlüssels der Wahl, dieser ist auf die Trustees aufgeteilt. Die Stimmabgabe in der Wahlurne erfolgt auch verschlüsselt. Da die Verschlüsselung bereits auf dem Computer des Wählenden stattfindet ist hier eine Zuordnung ebenfalls nicht möglich. Lediglich die Liste der Wahlberechtigten (E-Mail Addresse) ist für die Korrelation der Wählenden-Liste mit der Authentifizierung durch den Uni-Login während der Wahl auf dem Wahlserver notwendig. Allerdings ist dies technisch komplett Unabhängig von der Durchführung der Wahl selbst und lediglich eine zusätzliche Sicherheitsmaßnahme.

(5) Die Datenübermittlung muss verschlüsselt erfolgen, um eine unbemerkte Veränderungen der Wahldaten zu verhindern. Bei der Übertragung und Verarbeitung der Wahldaten ist zu gewährleisten, dass bei der Registrierung der Stimmabgabe im Wählerverzeichnis kein Zugriff auf den Inhalt der Stimmabgabe möglich ist.

Jede Kommunikation mit dem Wahlserver erfolgt über eine TLS-Verschlüsselte HTTP-Verbindung mit Forward-Secrecy, weiterhin ist die E-Mail-Kommunikation zumindest Uni-Intern auch auf Verbindungsebene mit TLS und Forward-Secrecy verschlüsselt. Zugriff auf die Universitäts-Email ist auch nur über verschlüsselte Verbindungen möglich. Ein passives Belauschen der Stimmenabgabe und/oder des Versands der privaten Credentials sollte damit nicht möglich sein. Weiterhin wird die Stimme bereits lokal auf dem Computer des Wählenden

verschlüsselt und somit wird nur die bereits verschlüsselte Stimme übermittelt.

(6) Die Wähler sind über geeignete Sicherungsmaßnahmen zu informieren, mit denen der für die Wahlhandlung genutzte Computer gegen Eingriffe Dritter nach dem aktuellen Stand der Technik geschützt wird; auf kostenfreie Bezugsquellen geeigneter Software ist zu hinzuweisen. Die Kenntnisnahme der Sicherheitshinweise ist vor der Stimmabgabe durch den Wähler verbindlich in elektronischer Form zu bestätigen.

Dieser Punkt ist aktuell nicht umgesetzt.

BSI-CC-PP-0037

Aus § 29 e der BUW-Wahlordnung ergibt sich, dass für Online-Wahlen genutzte Software idealerweise dem BSI-Schutzprofil [bsi-cc-pp-0037] nach Common Criteria für Online-Wahlen entsprechen soll. Common Criteria ist ein sehr formaler Ansatz und es gibt für Belenios keine offizielle Zertifizierung.

Hier wird lediglich informell beschrieben wie eine Wahl mit Belenios die in [bsi-cc-pp-0037] Kapitel 1 genannten Anforderungen löst. Das Schutzprofil umfasst sehr viele weitere Punkte auf die hier nicht eingegangen werden kann.

Es gibt eine Sicherheitsanalyse für Belenios in französischer Sprache [belenios-cnil] für die Empfehlungen der nationalen Datenschutzbehörde Frankreichs (CNIL) zu Internet-Wahlsoftware [cnil-1917529x]. Belenios mit Trennung der Zuständigkeiten (seperate Credential Authority, mehrere Trustees) erfüllt Level 2 der Anforderungen.

Ein direkter Vergleich mit den Anforderungen des BSI-CC-PP-0037 ist nicht möglich, da die Richtlinen einen anderen Fokus haben.

Im folgenden wird auf die grundlegenden Sicherheitserwartungen von [bsi-cc-pp-0037] Kapitel 1 eingegangen. Ziel ist es die wesentlichen Punkte hier zu diskutieren, jedoch keine ausführliche Analyse aller Punkte.

Allgemein

Das Schutzprofil geht über eine reine Evaluation der Wahlsoftware hinaus und beschreibt den kompletten Wahlprozess auch aus organisatorischer Sicht.

Generell wird eine Online-Wahl in 3 Phasen unterteilt:

- Wahlvorbereitung
- Wahldurchführung inkl. Stimmenzählung
- Archivierung

Das Schutzprofil definiert Anforderung an die Phase Wahldurchführung inkl. Stimmzählung. Der gesamte Ablauf der Fachschaftswahl mit Belenios ist bereits unter Ablauf dokumentiert.

Wesentliche Sicherheitsmerkmale

1.2 EVG-Übersicht

EVG = Evaluationsgegenstand, hier die Software Belenios.

1.2.1 Art des EVG

Da Belenios aus Sicht des Wählers eine Webanwendung ist, trifft hier die Unterscheidung zwischen Clientseitigem EVG und Serverseitigen EVG nicht zu sondern Belenios wird stets als Serverseitiges EVG betrachtet.

1.2.3 Sicherheitserwartungen

Die Anforderungen, die sich aus dem BSI-Sicherheitsanforderungen ergeben leiten sich aus dem allgemeinen Wahlgrundsätzen (frei, gleich, geheim, allgemein und unmittelbar) ab. Konkret werden in Kapitel 1.2.3 Sicherheitserwartungen genannt - eine Einschätzung zu Belenios jeweils unterhalb der Erwartung

Eine Zusammenführung der Identität des Wählers mit seiner abgegebenen Stimme darf nicht hergestellt werden können. (Anonymität: geheime und freie Wahl).

Dies ist bei Belenios und dem hier beschriebenen Ablauf der Wahl zum einen durch die Trennung der Verantwortlichkeiten (organisatorische Seperation of Duty) gegeben (SCC als Credential Authority, mehrere Trustees teilen den privaten Schlüssel) sowie auch durch die Verschlüsselung der Stimme vor der Stimmabgabe. Siehe auch Konformität mit der Wahlordnung des StudierendenKonvents, Wie funktioniert Belenios?.

Der EVG (Evaluationsgegenstand = Belenios hier) darf dem Wähler nicht die Möglichkeit geben, seine Wahlentscheidung gegenüber anderen zu beweisen (Quittungsfreiheit: geheime und freie Wahl).

Durch die Verschlüsselung der Stimme ist lediglich ein Nachweis möglich, dass eine Person gewählt hat. Es ist nicht möglich zu zeigen was gewählt wurde. In [belenios-2019] wird diese Problematik ausführlich diskutiert.

Eine eindeutige und zuverlässige Identifikation und Authentisierung der Wähler muß sicherstellen, daß nur registrierte Wähler eine Stimme abgeben dürfen. (Authentisierung: allgemeine und gleiche Wahl).

Die Authentisierung der Wähler erfolgt zweifach: Einmal durch die Credential-Authority, welche die initialen private Credentials aus dem Wählerverzeichnis generiert. Durch das private Credentials ist jeder Wählende gegenüber dem Wahlsystem anonymisert authentifizerbar, da sein Stimmzettel mit dem privaten Schlüssel des Credentials signiert ist und der Wahlserver den öffentlichen Schlüssel besitzt. Weiterhin ist zur Stimmabgabe ein Uni-Login notwendig, der nochmals validiert dass der Wählende auf der Wahlliste steht. Siehe auch [belenios-2019], Konformität mit der Wahlordnung des StudierendenKonvents und Wie funktioniert Belenios?

Jeder Wähler darf nur einmal eine Stimme abgeben. (One voter – one vote: gleiche Wahl).

Dies ist durch die Signatur des Stimmzettels mit dem privaten Credential gesichert. Auf dem Wahlserver gibt es ein Logfile welches zur Überprüfung genutzt wird, dass jeder Wählende nur eine Stimme abgeben darf. Siehe auch [belenios-2019].

Es darf bei der Übertragung im Netzwerk nicht möglich sein, Stimmdatensätze unbemerkt zu verändern, zu löschen oder hinzuzufügen (Integrität des Netzwerks: allgemeine und gleiche Wahl).

Dies ist zum einen durch TLS-Verschlüsselung der HTTP-Verbindung zum Wahlserver gesichert, zum anderen wird der Stimmzettel bereits lokal auf dem Computer des Wählenden verschlüsselt und mit dem privaten Credential signiert, so dass auf dem Transport eine Manipulation sowohl die Kenntnis des privaten Credentials, wie auch die Möglichkeit die Ende-zu-Ende TLS-Verschlüsselung zu brechen benötigen würde. Veränderungen auf dem Wahlserver würden während des Audits auffallen. Siehe [belenios-2019], Konformität mit der Wahlordnung des StudierendenKonvents

Es darf in der Urne nicht möglich sein, unbemerkt Stimmen zu verändern, unbemerkt Stimmen zu löschen oder unberechtigt Stimmen hinzuzufügen (Integrität der Urne: allgemeine und gleiche Wahl).

Dies ist durch die "End-to-end verifiablity" der Wahl gegeben, die von Belenios über Kryptographische Mechanismen implementiert wird. Siehe [belenios-2019], Wie funktioniert Belenios? sowie Audit der Wahldaten

Die Berechnung von Zwischenergebnissen muß ausgeschlossen werden (Zugriffskontrolle: geheime und gleiche Wahl).

Zur Berechnung von Zwischenergebnissen ist der private Schlüssel der Wahl notwendig. Dafür müssten sich alle Trustees verabreden, da der private Schlüssel zwischen diesen gesplittet ist. Im Einklang mit dem Prinzip "Organizational Seperation of Duties" werden für die Wahl unabhängige Autoritäten der Universität als Trustees genutzt, was eine Verabredung unwarscheinlich macht. Siehe [belenios-2019], Trustee Anleitung

Technische Details zum Wahlserver

Die Belenios-Wahlsoftware wird auf dem Server `m18.uni-weimar.de` unter der Domain https://wahl.m18.uni-weimar.de betrieben. Serverstandort ist das SCC-Rechenzentrum in der Steubenstraße 6a. Der administrative Zugriff ist auf Mitglieder des Referates Digitale Infrastruktur des StuKo beschränkt. Der Server läuft unter dem Betriebssystem Ubuntu 18.04 mit aktuellen Sicherheitsupdates.

Belenios wurde in der Version 1.15 als Docker-Container gebaut und auch betrieben. Der Source-Code ist unter https://gitlab.com/bauhaus/belenios zu finden - der genutzte Docker-Container wurde via Gitlab-CI aus diesen Sourcen gebaut. Die genutzte Domain `wahl.m18.uni-weimar.de` wird intern an den Docker-Container weitergeleitet siehe `/etc/nginx/sites-enabled/belenios`.

Das Wahl-Frontend wurde auf Webserver-Ebene durch Filter angepasst um z.B. Logos und CSS einzubinden - Alle Änderungen sind rein kosmetischer Natur und hier zu finden. Wichtig ist, dass sich diese Änderungen nicht im Belenios-Quellcode wiederfinden.

Der Belenios-Quellcode der auf dem Wahlsystem läuft ist unter https://wahl.m18.uni-weimar.de/belenios.tar.gz herunterladbar.

SHA-512 Prüfsumme von `belenios.tar.gz`

```
09e5167f5b9cc14aa2308d0e4b446d84a6adaee5c54d76e21b5a6f7692c8ad5e1735967e23c9dc425ef570
c65e7aaa78abf79328cd5f517138f186d0ec14b016  belenios.tar.gz
```

Die genutzt Webserver-Konfiguration zur Weiterleitung an das Belenios-Docker-Image ist folgende:

`/etc/nginx/sites-enabled/belenios`

```
server {
  listen 80;
  server_name wahl.stuko.uni-weimar.de wahl.m18.uni-weimar.de;

  location / {
      return 301 https://wahl.m18.uni-weimar.de$request_uri;
  }
}

server {
  listen 443 ssl;
  server_name wahl.stuko.uni-weimar.de;

  # ssl magic
  include snippets/wildcard-stuko;

  location / {
      return 301 https://wahl.m18.uni-weimar.de$request_uri;
  }
}

server {
  listen 443 ssl;
  server_name wahl.m18.uni-weimar.de;

  # ssl magic
  include snippets/wildcard-m18;

  location / {
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto https;
    proxy_pass http://127.0.0.1:8001;
    proxy_http_version 1.1;
  }

  location = /elections/TSyGN5mcMF34X9/ {
    return 302 https://m18.uni-weimar.de/wahlen2021/ups;
  }
```

```
  location = /static/chart.png {
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto https;
    proxy_pass http://127.0.0.1:8002;
    proxy_http_version 1.1;
  }
}
```

Referenzen

- [belenios-howitworks] How does Belenios work? Main principles of Belenios - https://www.belenios.org/howitworks.html
- [belenios-whodoeswhat] Who does what during a Belenios election? - https://gitlab.inria.fr/belenios/belenios/-/blob/master/doc/instructions-en.md
- [inria-belenios-intro] Belenios: fully transparent electronic voting - https://www.inria.fr/en/belenios-fully-transparent-electronic-voting
- [belenios-2019] Belenios: a simple private and verifiable electronic voting system - https://hal.inria.fr/hal-02066930/document
- [zk-slides] Zero-Knoweldge Proofs, with applications to Sudoku & Where's Waldo? - http://web.engr.oregonstate.edu/~rosulekm/pubs/zk-waldo-talk.pdf
- [belenios-spec] Belenios specification - https://www.belenios.org/specification.pdf
- [belenios-zk-proof] Some ZK security proofs for Belenios - https://www.belenios.org/ZK-securityproof.pdf
- [belenios-source] Belenios sources - How to get Belenios? - https://www.belenios.org/software.html
- [stuko-wahlordnung-neu] Erste Änderung Wahlordnung der Studierendenschaft - https://stuko.uni-weimar.de/wahl/stuko-wahlordnung-aenderung-2021.pdf
- [stuko-wahlordnung] MdU: Wahlordnung der Studierendenschaft 2.4.2009 - https://www.uni-weimar.de/fileadmin/user/uni/universitaetsleitung/kanzler/mdu_akad/09/08_2009.pdf
- [buw-wahlordnung-neu] MdU: Erste Änderung der Wahlordnung der Bauhaus-Universität Weimar - 30.4.2020 - https://www.uni-weimar.de/fileadmin/user/uni/universitaetsleitung/kanzler/mdu_akad/20/17_2020.pdf
- [bsi-cc-pp-0037] Common Criteria Schutzprofil für Basissatz von Sicherheitsanforderungen an Online-Wahlprodukte - https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Reporte/ReportePP/pp0037b_pdf.pdf?__blob=publicationFile&v=1
- [belenios-cnil] Analyse de sécurité de la plateforme de vote Belenios Conformité avec les recommandations 2019 de la CNIL - https://www.belenios.org/analyse-secu.pdf
- [cnil-1917529x] Délibération n° 2019-053 du 25 avril 2019 portant adoption d'une recommandation relative à la sécurité des systèmes de vote par correspondance électronique, notamment via Internet - https://www.legifrance.gouv.fr/jorf/id/JORFTEXT000038661239

- [group] `group.json` von Belenios 1.15 - https://wahl.m18.uni-weimar.de/static/groups/default.json


```
{"g":"24023526775018522092276877035323999327122876573783649165100753187876632741463
532193202856761552696787996946682987493890950838965734256019006010684771644917354 74
137283104610458681314511781646755400527402889846139864532661215055797097162016168 27
031288643245666383486363578210615491841998253431518974065818686865115135857641013 88
822153960160432288436039309893336627728484065931384060102316750957637779826651036 06
822406635076697764025346253773085133173495194248967754052573659049492477631475991 57
519877517771148149092045660020547812705472823814097251863985833411570056835369555 34
237814755824918960502966800377453084606 27","p":"20694785691422546401013643657505008
064922989295751104097100884787057374219242717401922237254497684338129066633138078 95
840496005438963628979639303877390572280360597374942767137677761889858987273586504 90
811670993105358677809800307904916540637771737641986785272734744763418356000356983 05
193144284561701911000786737307333564123971732897913240474578834468260652327974647 95
113767265869358218004631792207366886005262718636338608879688212076943236614949100 29
234443463732221458841005864210502421203654335612013204811188524087310770141516662 00
162313177169372189248078507711827842317498073276598828825169183103125680162072880 719","q":"78571733251071885079927659812671450121821421258408794611510081919805623223441"}
```