Vorgenommene Änderungen

Christoph Bieringer, Simon Schneider

27.10.2022

In diesem Dokument werden die im Rahmen dieser Arbeit am System vorgenommenen Änderungen aufgelistet und beschrieben.

Inhalt

Hinzufügen eines Setup-Skripts

Um die zukünftige Arbeit an dem System zu erleichtern, wurde ein Setup-Skript (setup_election.py) erstellt, mit dem die Datenbank initialisiert werden kann. Dieses Skript führt bei seiner Ausführungen folgende Aktionen durch:

- Die Tabellen für Wählerinformationen (persönliche IDs und Authentifizierungscodes) und Wählerschlüssel werden geleert.
- Ein neuer Administratorschlüssel wird erzeugt und in der entsprechenden Tabelle gespeichert.
- Für die in der Konfigurationsdatei festgelegte Anzahl an Wählern (siehe weiter unten) werden neue Authentifizierungsinformationen zufällig erzeugt und in der entsprechenden Tabelle abgelegt.

Hinzufügen einer Konfigurations-Datei

Im Rahmen dieser Arbeit wurden dem System mehrere Konfigurationsmöglichkeiten hinzugefügt. Die hierfür gewählten Werte werden zentral in einer Datei (BlockchainTIF19AGruppeC\dhbw-blockchain-encryption\config\config.json) gespeichert. Bei dieser Datei handelt es sich um eine JSON-Datei, die ein Objekt enthält. Dieses hat folgende Keys:

- „start“: Der zugehörige Value ist ein String, der einen ISO-8601-konformen Timestamp in UTC enthält.
- „end“: Der zugehörige Value ist ein String, der einen ISO-8601-konformen Timestamp in UTC enthält.
- „options“: Der zugehörige Value ist ein Array von Strings.
- „voterCount“: Der zugehörige Value ist ein Integer, der die Anzahl an Wählern angibt, für die (durch das Setup-Skript) Authentifizierungsinformationen erstellt werden sollen.

Einschränkung der zeitlichen Verfügbarkeit verschiedener Funktionen

Da die permanente Verfügbarkeit der Auszählfunktion, wie im Dokument „Analyse und Schwachstellen“ beschrieben, eine potentielle Sicherheitslücke darstellt, wurde beschlossen, sie nur noch nach einem bestimmten Datum (gewissermaßen dem Ende der Wahl) verfügbar zu machen.

Dieser Zeitpunkt befindet sich in dem neu angelegten Config-File (der Value zum Key „end“). Der Wahlserver liest diesen Wert beim Start einmal ein (der für dieses Feature zuständige Code befindet sich in dem neu angelegten Modul config_management.py). Wird nun eine Auszählanfrage vor dem in „end“ gespeicherten Zeitpunkt gestellt, so antwortet der Server (semantisch korrekt) mit dem Antwortcode 403 (Zugriff verweigert) sowie einem JSON-Objekt, dass den Fehlergrund angibt („Election still in progress“). Eine Anfrage nach dem Ende der Wahl wird normal beantwortet.

Damit diese Einschränkung auch tatsächlich eine Sicherheitsverbesserung bewirkt, darf natürlich nach der Freigabe der Auszahlfunktion auch nicht mehr abgestimmt werden. Auch diese Änderung wurde im Rahmen dieser Arbeit vorgenommen. Beim Zugriff auf den API-Endpoint /api/transmit nach dem Ende der Wahl wird eine Fehlermeldung (analog zur oben beschriebenen) zurückgegeben. Dass Frontend kann diese Meldung erkennen und den Benutzer über ein Pop-Up informieren.

Konfigurierbarkeit der wählbaren Alternativen

Im bisherigen System waren die Wahloptionen hardcoded. Im Rahmen dieser Arbeit wurde nun eine Möglichkeit hinzugefügt, die Wahloptionen zu konfigurieren, ohne den Wahlserver- oder Frontend-Code verändern zu müssen.

Eine Liste aller Wahloptionen befindet sich in dem neu angelegten Config-File (der Value zum Key „options“, wobei jeder String in diesem Array den Namen einer Option enthält). Der Walserver liest diesen Array beim Start ein. Über einen zusätzlichen API-Endpoint (/api/getOptions) ist die Liste der Optionen auch öffentlich verfügbar (das Frontend verwendet diesen Endpoint ebenfalls, um dem Benutzer alle Optionen anzuzeigen).

An verschiedenen Stellen musste der Code leicht verändert werden, um mit einer erst zur Laufzeit bekannten Liste an Optionen umgehen zu können. Hierfür wurde i.d.R. die Verwendung von einer fixen Anzahl Variablen mit dem Namen der entsprechenden Wahloption (z.Bsp. „value_spd“) durch die Verwendung eines Dictionaries (im Python-Code des Wahlservers) bzw. eines Objektes (im JavaScript-Code des Frontends) ersetzt.

Usability-Updates im Frontend

Im Frontend des Systems (d.h. den Websites im Ordner dhbw-blockchain-website) wurden verschieden Updates durchgeführt, die das System leichter benutzbar machen sollen. Einige der durchgeführten Updates sind u.a.:

- An verschiedenen Stellen wurden erklärende Text hinzugefügt.
- Die File-Input-Elemente auf der Verifikationsseite wurden so verändert, dass sie dem Benutzer anzeigen, ob schon eine Datei ausgewählt wurde (und wenn ja, welche). Zudem wird der Button zum Starten des Verifikationsprozess nun erst dann freigeschaltet, wenn der Benutzer beide der benötigten Dateien bereitgestellt hat.
- Die technisch nicht zu rechtfertigende Beschränkung der PIN auf eine sechsstellige Zahl wurde aufgehoben und durch eine neue Bedingung ersetzt. Ein PIN muss nun mindestens acht Zeichen lang sein, Groß- und Kleinbuchstaben sowie Sonderzeichen und Ziffern enthalten. Auf diese Weise wird verhindert, dass Benutzer einfach zu erratende PINs (etwa ihr Geburtsdatum) verwenden, und Brute-Force-Angriffe zur Deanonymisierung von Wählern werden erschwert.

Veröffentlichung der kompletten Blockchain

Damit die Manipulationssicherheit der Blockchain gewährleistet ist, muss ihr Inhalt öffentlich von Dritten überprüft werden können. Hierfür wurde im Register ein neuer API-Endpoint (/api/getFullChain) geschaffen. Analog zu /api/getTransactions enthält dieser einen JSON-Array mit einem Objekt für jeden Block der Blockchain. Diese Objekte enthalten jeweils die Blocknummer, die Transaktionen (als Array), den Hash des Vorgängerblocks, und den Hash des Blocks. Mit diesen Informationen kann eine Anwendung die Integrität der Blockchain unabhängig überprüfen, indem sie die Hashwerte für jeden Block selbst berechnet und mit den in der Blockchain gespeicherten vergleicht.

Diese Änderung ist ein wichtiger Schritt, um die globale Verifizierbarkeit und Manipulationssicherheit der Wahl sicherzustellen.

Hinzufügen einer Verifikationsmöglichkeit für die Blockchain

Im Frontend wurde eine vierte Seite erstellt. Auf dieser kann sich der Benutzer die gesamte Blockchain (von /api/getFullChain auf dem Register) anzeigen lassen. Clientseitiger Code überprüft die Blockchain vollautomatisch (mithilfe der Web Crypto API) und informiert den Benutzer, falls einer der Hash-Wert in der Blockchain nicht korrekt ist. Auf diese Weise kann sich jeder Benutzer selbst davon überzeugen, dass die Blockchain nicht manipuliert wurde und alle Stimmen korrekt

gespeichert sind, ohne dabei einer anderen Komponente (Wahlserver/Register) vertrauen zu müssen. Dies ist ein wichtiger Schritt hin zur vollständigen Verifizierbarkeit.

Der gesamte Ablauf bei der Verifikation der Blockchain wird im Folgenden in einem Sequenzdiagramm noch genauer dargestellt. Hierbei ist zu beachten, dass der Wähler für die Verifizierung die vom System bereitgestellte Website benutzen *kann*, aber nicht *muss*. Falls er dem System nicht vertraut, kann er eine eigene Anwendung/Website/… schreiben (oder die eines Dritten verwenden), um auf den entsprechenden API-Endpoint zuzugreifen und die Verifikation durchzuführen.