Universelle Verifizierung des Genfer E-Voting Systems

Studiengang: BSc in Informatik | Vertiefung: IT-Security
Betreuer: Dr. Rolf Haenni
Experte: Thomas Hofer (DGSIN)

Die fortan wachsende Digitalisierung macht auch keinen halt vor der Politik. So begann das Parlament im Jahr 2000 mit den Vorbereitungen für die elektronische Stimmabgabe kurz E-Voting. Doch ein solches System zu bauen ist ziemlich komplex. Denn es muss den aktuellen Sicherheitsanforderungen standhalten. Einer dieser Anforderungen ist die universelle Verifizierbarkeit. Das Ziel der Bachelor-Thesis war es, einen Verifier zu entwickeln, welcher diese Anforderung erfüllt.

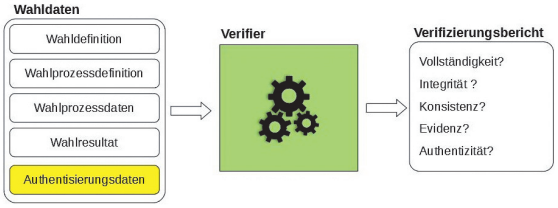
Die universelle Verifizierbarkeit

Bei der universellen Verifizierbarkeit geht es darum, die Verlierer einer Abstimmung davon zu überzeugen, dass das Resultat korrekt ist. Bei der brieflichen Stimmabgabe geschieht dies durch eine mögliche Nachzählung der Stimmen. Im E-Voting ist dies leider nicht so einfach. Deshalb gibt es eine Applikation, genannt Verifier, welche von einer unabhängigen Instanz ausgeführt werden kann. Diese überprüft nun systematisch alle Abstimmungsdaten. Somit lässt sich feststellen, ob die Daten manipuliert wurden. Aber auch Softwarefehler lassen sich damit aufdecken. Eine einzelne solche Überprüfung wird als Test bezeichnet.
Die Tests lassen sich in fünf Kategorien einordnen.
Bei der Kategorie Vollständigkeit wird überprüft, ob alle erforderlichen Daten vorhanden sind. Denn nur so kann man eine lückenlose Verifizierung gewährleisten. Jeder Test überprüft, ob ein Parameter vorhanden ist oder nicht.
Bei der Kategorie Integrität wird die Integrität der Parameter geprüft. Es wird also geprüft, ob die Parameter in sich schlüssig sind. Beispielweise wird geprüft, ob sie sich in den geforderten Wertebereichen befinden.
Bei der Kategorie Konsistenz wird geprüft, ob die Parameter zu den anderen konsistent sind, zum Beispiel zwei Vektoren, welche die gleiche Länge haben sollten. Nun wird geprüft, ob diese wirklich die gleiche Länge haben.
Bei der Kategorie Evidenz wird geprüft, ob die verschieden kryptographischen Beweise stimmen.
Bei der Kategorie Authentizität wird die Gültigkeit der Zertifikate und Signaturen überprüft.

Christian Wenger

Umsetzung

Das Ziel der Arbeit war es, die bestehende Applikation «NextGenVote Visualization» mit dem Verifier zu vervollständigen. Die Algorithmen für die Tests wurden in Python implementiert. Da bei dem System die Zertifikate und Signaturen fehlten, wurden «successful», «failed» und «skipped» als Ausgabe der Tests definiert. Der Ausgabewert «skipped» wurde verwendet, wenn keine Testdaten vorhanden waren.
Das Front-End wurde mit Vue.js umgesetzt. Dabei war ein Hauptziel, den User möglichst genau zu informieren, was das Backend im Hintergrund macht. Dazu benötigt das Frontend eine Kommunikationsschnittstelle, welche mit Sokket.io umgesetzt wurde.



Funktionsweise des Verifiers mit den fünf Test-Kategorien

Das Front-End des Verifiers, ist hauptsächlich dazu da um das Resultat anzuzeigen.