Stellungnahme zum Protokoll „Termin Polyas 16. Januar 2026“

10. März 2026

1 Allgemeines

Aufbauend zum oben stehenden, von Polyas zur Kenntnis genommenen, Protokoll zum Termin mit Polyas, folgt hier die Stellungnahme der studierenden Mitglieder. Die in diesem genannten Fakten sollen hier eingeordnet werden und die Konformität von Polyas Core 2 mit bestehenden Wahlgrundsätzen eingestuft werden.

1.1 Rahmenbedingungen

Laut Aussage von Polyas ist unser Besuch der erste seiner Art. Wir bedauern an dieser Stelle eine mangelnde Leidenschaft und ein mangelndes Interesse an Wahlen. Transparenz ist eine wichtige Grundkomponente in Wahlen, Einsichten sollten daher regelmäßig wahrgenommen werden.

Wir wollen das Gespräch mit den Entwickelnden von Polyas positiv hervorheben. Wir haben den Eindruck, dass auf unsere Fragen ehrlich geantwortet wurde. Kritische Punkte wurden offen kommuniziert.

Jedoch haben auch über zwei Stunden nicht ausgereicht, um alle Punkte anzusprechen. Das zeigt, wie komplex die Software und die Konzepte sind. Eine Einarbeitung in die Thematik von Laien scheint nahezu ausgeschlossen. Aufgrund des Zeitrahmens und des Aufbaus konnte außerdem nur wenig Code angesehen werden. Unser Besuch hat also bedauerlicherweise nicht das Vertrauen in die Software stärken können. Hierfür wäre ein leichterer Zugriff auf den Quellcode notwendig.

2 Software

2.1 Allgemeines

Wir kritisieren den Einsatz einer Software mit geheimem Quellcode für öffentliche Wahlen. Wir sind der Meinung, dass – insbesondere häufig verwendete Software – an Sicherheit gewinnt, wenn deren Quellcode öffentlich ist. Die Veröffentlichung des Quellcodes steht dabei einer Gewinnerzielung nicht unbedingt im Weg. Zahlreiche Unternehmen pflegen offene Software und erzielen dann Gewinn mit deren Hosting oder Support. Außerdem kann die Software so lizenziert werden, dass der Quellcode offen ist, der Einsatz der Software jedoch nur zahlungspflichtig möglich ist. Ein Beispiel hierfür ist die in Estland eingesetzte Wahlsoftware IVXV[1].

3 Technische Details

Den Einsatz eines selbst implementierten, mTLS-ähnlichen Protokolls sehen wir kritisch. Was den Einsatz von Kryptografie angeht, sollten so weit irgend möglich offene, weit verbreitete und getestete Algorithmen und Implementierungen verwendet werden. Das widerspricht auch nicht der proprietären Lizenz von Polyas, da die meisten dieser Softwarebibliotheken auch in proprietärer Software komplett frei verwendbar sind.

Wir betrachten das Management der Tokens im System mit Skepsis. Die komplette Anonymisierung der Stimmen – also die Trennung von Tokens und den eigentlichen Stimmen – basieren maßgeblich auf der Löschung von Informationen nach der Stimmabgabe. Das erfordert ein besonders hohes Vertrauen in die Funktionstüchtigkeit dieser Softwarekomponente. Wie bereits im Protokoll beschrieben, ist bei kleinen Wählendengruppen eine komplette Deanonymisierung möglich. Diese Deanonymisierung ist im Anhang genauer beschrieben. In der Version Core 2 wird diese Schwäche einzig dadurch verhindert, dass nur die zusammengefassten Wahlergebnisse veröffentlicht werden. Aus der Datenbank heraus könnten allerdings personenbezogene Stimmen ermittelt werden. Die Sicherheit von Wahlen darf unserer Meinung nach jedoch nicht von der Geheimhaltung von bereits vorliegenden Daten abhängen. Der Vorschlag seitens Polyas, mehrere Wahlen aufzusetzen, ist zu begrüßen. Hierbei würden die Wählenden unterschiedliche Tokens bekommen, es könnten allerdings auch Mehrkosten anfallen.

Die Vermittlung der Zugangslinks per Self-Service Portal mittels sog. SecureLink [2] ist bzgl. des Datenschutzes eine deutliche Verbesserung, da die Pseudonymisierung bereits auf Seiten der TU-Server passiert. Im Gegensatz zum Versand per E-Mail ist der Zugang dadurch auch mit zwei Faktoren gesichert. Ein Verzicht auf derartige Maßnahmen wäre besonders kritisch zu betrachten, da ohne diese Trennung unmittelbar personenbezogene

[1] `https://github.com/valimised/ivxv`

[2] `https://support.polyas.com/de/online-wahlmanager/features/authentifizierung/securelink/`

Daten an Polyas übertragen werden würden. Der Versand per Post stellt dennoch eine transparentere Anonymisierung dar. Wenn die versiegelten Briefe gemischt und dann adressiert werden, ist die korrekte Ausführung der Anonymisierung besonders gut sichtbar und auch von Laien nachprüfbar.

Der Prozess der universellen Verifikation bietet unserer Meinung nach keinerlei Vorteil. Die Fehler, die durch dieses Verfahren erkannt werden können, sind zu vernachlässigen. Der Algorithmus könnte höchstens eine zufällig aufgetretene Beschädigung von Dateien aufdecken. Die von uns benannten Angriffsszenarien kann das Verfahren nicht erkennen. Es ist zu befürchten, dass das Verfahren technisch wenig versierten Nutzenden ein falsches Gefühl der Sicherheit vermittelt. Gerade die Bezeichnung von verknüpften Prüfsummen als „Blockchain“ könnte einigen ein falsches Verständnis der Prozesse vermittelt haben. Dass keine echte Blockchain eingesetzt wird, begrüßen wir. Wir stimmen Polyas darin zu, dass diese Technologie für diese Art von Wahlen kaum Vorteile bietet und den Aufwand im Betrieb und die Komplexität der Software nur unnötig vergrößern würde.

4 Betrieb und Pflege

4.1 Sicherheitsmanagement

Wie bereits im Protokoll beschrieben, kann der Zeitaufwand bei der Zertifizierung ein zeitnahes Reagieren auf kritische Sicherheitslücken erschweren. Nach Zertifizierung werden Abhängigkeiten nur selektiv aktualisiert, wenn Polyas eine konkrete, das Produkt betreffende, Schwachstelle bekannt ist. Die Einstufung, ob konkrete Sicherheitslücken kritisch sind, ist eine zusätzliche Fehlerquelle im Betrieb der Software.

Ein allgemeines von Polyas nicht lösbares Problem ist die zunehmende Verbreitung von kritischen Sicherheitslücken. Die im Protokoll genannte Lücke in der Software Log4j war bereits vor Bekanntwerden als Zero-Day-Exploit weit verbreitet. Auch weniger schützenswerte Infrastruktur, wie zum Beispiel Minecraft-Server, waren von Angriffen mithilfe der Sicherheitslücke betroffen. Dass auch Personen, welche von einem bestimmten Wahlausgang profitieren, Schadsoftware entwickeln oder kaufen könnten, um Polyas zu manipulieren, scheint nicht weit hergeholt.

4.2 Deployment

Die Erklärung, warum Polyas nicht On-Premise angeboten wird, erschließt sich uns nicht. Dass die Kund:innen (z. B. die TU Dresden) als Hoster durch Fehler die Wahl gefährden könnten ist korrekt. Wir sind jedoch der Meinung, dass die Entscheidung, wer den Server bereitstellt, bei den Universitätsgremien liegen sollte. Im Fall der TU Dresden wäre ein On-Premise-Hosting zu begrüßen, da das ZIH über Rechenzentren mit sehr hohem Schutzlevel verfügt. Da außerdem jährlich mindestens zwei Wahlen anstehen

(Universitätswahlen und Wahlen des Promovierendenrates) lohnt sich die Pflege auf den Universitätsservern außerdem mehr als bei anderen Institutionen, die seltener wählen.

Gemäß des Schutzprofils BSI-CC-PP-0037-2008[3] müssen Wahlverzeichnisserver, Validator und Wahlurnenserver physisch getrennt sein. Diese Trennung ist in der Wahlordnung der TU Dresden zusätzlich explizit gefordert. [4] Wahlverzeichnisserver, Validator und Urnenserver würden nach Aussage von Polyas über die Anti-Affinity-Einstellung in Kubernetes auf verschiedenen physischen Servern deployt. Auf diese Dienste wird vom Browser der Wählenden nicht direkt zugegriffen, sondern über einen Ingress-Reverse-Proxy der Open Telekom Cloud, der auch die TLS-Verbindung zu den vorgenannten Servern terminiert. Da der Reverse-Proxy als eine Komponente so den gesamten Verkehr lesen und manipulieren könnte, wird das Ziel der Komponententrennung nur eingeschränkt erfüllt. Statt drei Komponenten, auf die laut Konzept das Risiko verteilt werden sollte, wird in dieser Konfiguration ein einziger „Single Point of Failure“ geschaffen. Unserer Meinung nach verstößt dies gegen die Wahlordnung der TU Dresden, da die Komponententrennung nicht zweckgemäß erfüllt ist. Hieraus ergibt sich ein relevantes Angriffsszenario. Dieses wird im Anhang näher beschrieben.

5 Zertifizierung der Software

5.1 Inhalt des Schutzprofils

Das im Protokoll und den Selbstberichten beschriebene Verfahren zur Komponententrennung soll einen Angriff erschweren. Da im von Polyas gewählten Deployment alle Server von demselben Unternehmen in demselben Kubernetes-Cluster verwaltet werden, bietet das Verfahren tatsächlich nur einen minimal erhöhten Schutz. Ein Deployment, welches den Datenzugriff stärker vor den Serververwaltern schützt, wird im Schutzprofil auch nicht vorgeschrieben. Tatsächlich verkompliziert es das Verständnis der Software und erschwert das Aufsetzen der Software, da dies auf drei verschiedenen Servern erfolgen muss. Verantwortlich für das – aus unserer Sicht – falsche Verständnis von IT-Sicherheit ist vor allem das veraltete Schutzprofil. Dieses wurde 2008 veröffentlicht und entsprach bereits zum damaligen Zeitpunkt nicht mehr den aktuellen Konzepten der Kryptografie. Das neuste Schutzprofil für Online-Wahlen, BSI-CC-PP-0121-2024[5], stellt stattdessen den individuellen kryptografischen Nachweis der Stimmenabgabe in den Mittelpunkt. Unserer Meinung nach sollte bei einer Neuauflage des Schutzprofils insbesondere die Transparenz über die Funktionsweise der Software mehr in den Mittelpunkt gestellt werden. Weiterhin sollte genauer abgegrenzt werden, für welche Wahlen keine Onlinewahl eingesetzt werden darf. Die Bezeichnung „unpolitische Wahl“ ist hierfür nicht ausreichend.

[3] `https://www.bsi.bund.de/SharedDocs/Zertifikate_CC/PP/Archiv/PP_0037.html`
[4] § 13 Abs. 11 – 13 Wahlordnung
[5] `https://www.bsi.bund.de/SharedDocs/Zertifikate_CC/PP/aktuell/PP_0121.html`

5.2 Nachteil der Zertifizierung

Auch der aktuelle Zertifizierungsprozess wirft Fragen darüber auf, ob er für Wahlsoftware geeignet ist. Die Aktualität der Software und ihrer Abhängigkeiten ist von entscheidender Bedeutung für die Sicherheit der Software. Dadurch dass der Zertifizierungsprozess zeitaufwändig ist, wird eine zeitnahe Aktualisierung der Software jedoch gehemmt. Auch wenn bekannte Sicherheitslücken beseitigt werden, vergeht bis zur Rezertifizierung der aktualisierten Software wertvolle Zeit, in welcher die Software nicht konform mit einschlägigen Wahlordnungen eingesetzt werden kann. Im konkreten Fall dauerte die Zertifizierung der Version, die die Sicherheitslücke in Log4j schloss drei Monate[6][7]. Das macht die Abhängigkeit von der Software riskant, da längere Zeit nach Bekanntwerden einer Sicherheitslücke die Software nicht in zertifiziertem Zustand eingesetzt werden kann. Dies stünde nicht in Einklang mit der Wahlordnung. Die unsichere Zertifizierungssituation schafft außerdem Verunsicherung unter Wählenden. Dass die Betriebsumgebung nicht Teil des Schutzprofils ist, kann einigen Wählenden auch ein falsches Gefühl der Sicherheit vermitteln.

6 Fazit

Wahlen an Hochschulen sind politische Wahlen und das Missbrauchsrisiko einer Wahlsoftware ist generell hoch. Unabhängig davon, dass wir den Einsatz von Onlinewahlen an Hochschulen für problematisch halten, raten wir von einer weiteren Nutzung der Software ab. Wenn Hochschulen dennoch weiterhin elektronische Wahlen einsetzen möchten, müssen folgenden Anforderungen erfüllt sein:

- Verwendung von Software mit offenem Quellcode („Public Money, Public Code“)
- On-Premises-Hosting (auf Servern der Hochschule)
- Öffentliche Zeremonie mit Aufsetzung der Server (ähnlich zum estnischen Wahlrecht)
- Ausschließen der Möglichkeit einer Verknüpfung von Stimmzetteln verschiedener Wahlen
- Versenden der Tokens per Post ohne Zuordnung von Tokens zu Namen
- Bestmögliche Einhaltung der allgemeinen Wahlgrundsätze ähnlich zu Art. 38 Abs. 1 GG

[6] `https://www.polyas.de/blog/de/online-wahlen/sicherheit/polyas-log4j-sicherheit`
[7] `https://www.bsi.bund.de/SharedDocs/Zertifikate_CC/CC/Sonstiges/0862_0862V2.html`

7 Anhang

7.1 Angriffsszenario Wählergruppen

In folgenden fiktiven Szenario wird die Anonymität der Wahl angegriffen. Betrachtet werden die Stimmzettelformulare „Senatswahl“, „Fakultätsrat SLK“ und „Fakultätsrat Informatik“. Alle Wählenden haben sich bei der Urne mit einem Token authentifiziert, welches hier schematisch dargestellt wird. Da dieser von Polyas, wie im Konzept vorgesehen, gelöscht wurde, gibt es keine dezidierte Übersicht, welcher Token zu welcher Person gehört. Eine Deanonymisierung ist dennoch möglich:

Datenbankauszug Senat

Wahl (Liste)	Token (pseudonym)
Liste A	`tf21523rf6`
Liste B	`dd8e2h832E`
Liste B	`edhz721jhf`
Liste A	`hdg329qkwd`
...	...

Die Datenbank hat insgesamt ca. 400 Einträge.

Datenbankauszug Fakultätsrat SLK

Wahl (Person)	Token (pseudonym)
Person A	`tf21523rf`
Person B	`hdg329qkw`

Die oben stehenden Einträge sind die einzigen Einträge. Daraus lässt sich schließen, dass die einzigen beiden Wählenden des Fakultätsrates SLK die Liste A gewählt haben. Sind diese Datenbankeinträge bekannt, reicht das für die Deanonymisierung des Wahlverzeichnisses aus:

Auszug Wahlverzeichnis, Seite SLK Studierende

Name	Fakultät	Sperrvermerk
Alice	SLK	Sperrung Onlinewahl
Andrew	SLK	
Bob	SLK	Sperrung Onlinewahl
Cody	SLK	
...	...	

Alle weiteren Einträge besitzen keinen Sperrvermerk für die Onlinewahl. Hierdurch kann abgelesen werden, dass die beiden einzigen Onlinewählenden, Alice und Bob der Fakultät SLK im Senat die Liste A gewählt haben.

Dieser Angriff wäre bei einer korrekt durchgeführten Briefwahl nicht möglich, da die Stimmzettel verschiedener Wahlen kein gemeinsames Element (wie ein Token) aufweisen und separat behandelt werden.

7.2 Angriffsszenario Reverse-Proxy

In diesem Beispiel wird die Freiheit der Wahl und das Wahlgeheimnis angegriffen. Möglich ist dies durch die vollständige Umgehung der Komponententrennung. Angenommen Alice möchte an einer Online-Wahl mit Polyas Core 2.5 teilnehmen. Die drei Komponenten des Wahlsystems (Wahlverzeichnisserver, Urnenserver, Validator) befinden sich auf drei verschiedenen physischen Servern. Jedoch kommuniziert Alice bei Polyas Core 2.5 nicht direkt mit diesen Komponenten, sondern sendet ihre Anfragen TLS-verschlüsselt an einen Reverse-Proxy, welcher diese Anfragen entschlüsselt und an die drei Komponenten weitergibt. Angenommen Martin würde sich Zugriff auf den Reverse-Proxy verschaffen können, dann könnte er sämtliche Kommunikation zwischen dem Reverse-Proxy und den Komponenten abhören und manipulieren. Hierbei ist auch die vollständige Umleitung der Kommunikation auf eine Attrappe möglich. Alice würde dann nicht, wie sie denkt, mit den Komponenten des tatsächlichen Wahlsystems kommunizieren, sondern mit der von Martin kontrollierten Attrappe.

Diese Attrappe würde so aussehen und sich so verhalten als wäre sie der korrekte und offizielle Dienst für die Online-Wahl. Für Alice wäre sie also nicht erkennbar. Martin könnte dann das Login-Token, welches nur TLS-verschlüsselt ist, also für den Reverse-Proxy und die dahinter liegenden Dienste im Klartext lesbar, auslesen. Da die Stimmabgabe über die Attrappe erfolgt, kann Martin hier offensichtlich das Wahlgeheimnis umgehen, da eine klare Zuordnung vom Login-Token zur abgegebenen Stimme möglich ist. Mit dem abgefangenen Token kann Martin nun zusätzlich eine Stimmabgabe beim tatsächlichen Wahlsystem durchführen und somit Alices Freiheit der Wahl verletzen.

Dieses Dokument ist ohne Einsatz von Large Language Models entstanden.