Kontextsensitive CAPTCHAS im Online-Banking

Tobias Urban[1] · René Riedel[1]
Ulrike Schmuntzsch[2] · Norbert Pohlmann[1]

[1]Institut für Internet-Sicherheit – if(is) Westfälische Hochschule
{urban | riedel | pohlmann}@internet-sicherheit.de

[2]Institut für Psychologie und Arbeitswissenschaft
Technische Universität Berlin
ulrike.schmuntzsch@mms.tu-berlin.de

Zusammenfassung

In der modernen Informationsgesellschaft nehmen Online-Transaktionen einen wichtigen Teil unseres täglichen Lebens ein. In dieser Arbeit stellen wir ein nutzerzentriertes Protokoll vor, dass es Nutzern erlaubt vertrauenswürdige und sichere Transkationen durchzuführen, selbst wenn sie ein nicht vertrauenswürdiges oder mit Schadsoftware infiziertes Gerät nutzen. Das Protokoll nutzt einen CAPTCHA-artigen Ansatz, der verhindert, dass ein Angreifer eine Transaktion verändert ohne, dass Server oder Client dies bemerken. Dazu stellen wir dem Nutzer eine Aufgabe, die kontextsensitive Informationen der Transaktion enthält. Die Aufgabe wird so gestellt, dass sie einfach von Menschen lösbar ist aber nur schwer automatisiert gelöst werden kann. Zur Evaluation des Systems haben wir eine Nutzerstudie ($n = 30$) durchgeführt und berechnet mit welcher Wahrscheinlichkeit ein Angreifer erfolgreich die richtige Antwort auf die Frage erraten kann. Wir zeigen, dass ein Großteil der Transaktionen ($> 94\%$) geschützt werden kann, während das System selbst nutzbar bleibt.

1 Einführung

Online-Banking und Online-Transaktionen sind ein bedeutender Teil der modernen Informationsgesellschaft und gewinnen stetig an Bedeutung. Allein zwischen 2007 und 2017 verdoppelte sich die Nutzung von 25 % auf 51% in Europa [1]. Dieses Wachstum, das Aufkommen von immer mehr Applikationen, die sich über Micro-Transaktionen finanzieren wird dazu führen, dass wir in Zukunft immer mehr Online-Transaktionen durchführen werden.

Betrüger haben den Bereich bereits auf unterschiedlichsten Wegen kompromittiert [2]. Laut offiziellen Angaben des Deutschen Bundeskriminalamts belief sich in Deutschland der finanzielle Schaden im Online-Banking zwischen 2014 und 2016 auf über 54 Millionen Euro [3].

Erfolgreiche Angriffe auf das Online-Banking werden durch sogenannte *Man-in-the-Browser* (MitB) Angriffe ermöglicht [4]. Bei diesem Angriffsvektor übernimmt die Angreiferin die vollständige Kontrolle über den Browser des Nutzers (z.B. durch bösartige Browser-Erweiterungen).

P. Schartner · N. Pohlmann (Hrsg.) · D•A•CH Security 2018 · syssec (2018) 346-359.

Somit erlangt sie volle Kontrolle über z.B. die übertragenden Daten oder die Darstellung der Website auf dem Endgerät des Nutzers. In Bezug auf Online-Banking könnte die Angreiferin zum Beispiel den Zahlungsempfänger, der an den Server gesendet wird, einer Transaktion manipulieren dem Nutzer aber trotzdem noch den von ihm eingegebenen Empfänger anzeigen. Typischerweise wird als Verteidigungsstrategie gegen diesen Angriffsvektor ein zweiter Kanal verwendet, über den die Daten, welche die Bank erhalten hat, verifiziert werden können (z.B. das Smart Phone des Nutzers). Dementsprechend muss die Angreiferin auch den zweiten Kanal übernehmen, um das System erfolgreich anzugreifen.

In dieser Arbeit stellen wir ein Protokoll vor, dass Transkationen absichert, selbst wenn das Gerät des Nutzers mit Schadsoftware infiziert oder nicht vertrauenswürdig ist (z.B. ein Gerät in einem Internet-Café). Dabei wird kein weiterer Kanal als vertrauenswürdige Anzeige benötigt. Das Protokoll stellt mit hoher Wahrscheinlichkeit sicher, dass eine manipulierte Transaktion von dem Server oder Nutzer erkannt werden kann. Dazu wird das Prinzip von sog. CAPTCHAs *(„Completely Automated Public Turing Test To Tell Computers and Humans Apart"*) [5] auf das gegebene Problem angewendet (siehe Kapitel 3).

Wir adaptieren dieses Prinzip verwenden aber kontextsensitive Information aus der Transaktion, um diese abzusichern. Der Nutzer muss eine automatisch generierte Frage zu der Transaktion, die er durchführen möchte, beantworten. Bei der Manipulation einer Transaktion würden sich so zwei Szenarien ergeben: (1) Wenn die Angreiferin die Darstellung auf dem Client ändert passt die Antwort auf die Frage nicht zu der manipulierten Transaktion; oder (2) Wenn die Darstellung nicht geändert wird erkennt der Nutzer, dass die Transaktionsdaten geändert wurden. Die Angreiferin wird natürlich versuchen, die Antwort auf die Frage zu erraten. Dieser und weitere Angriffe auf das System werden in Kapitel 4 beschrieben. Die Auswirkungen der Angriffe auf das Protokoll werden in Kapitel 6 analysiert. Ein wichtiger Teil der IT-Sicherheit ist die Nutzbarkeit entwickelter Lösungen und ob diese von Nutzern akzeptiert werden. Daher haben wir eine Nutzerstudie ($n = 30$) für das entwickelte Protokoll durchgeführt, um dessen Nutzbarkeit zu überprüfen (siehe Kapitel 7).

Zusammengefasst liefert diese Arbeit die folgenden Beiträge:

- Wir adaptieren den CAPTCHA Mechanismus und wenden ihn im Online-Banking an.
- Wir stellen ein Protokoll vor, dass es erlaubt sichere Transkationen durchzuführen, selbst wenn dem Endgerät nicht vertraut werden kann oder mit Schadsoftware infiziert wurde.
- Wir analysieren das vorgestellte Protokoll in einer Nutzbarkeitsstudie.

2 Grundlagen

Wenn man sichere Online-Transaktionen durchführen will lassen sich die meisten auftretenden Probleme in drei Kategorien einteilen. Dies sind einerseits technische Probleme, die von Angreiferinnen ausgenutzt werden können und andererseits Limitierungen, die sich aus der Spannung zwischen Nutzbarkeit und Sicherheit ergeben.

1. **Gestohlene Anmeldedaten:** Wenn die Angreiferin die Kontrolle über das Konto des Nutzers erlangen will muss sie die Anmeldedaten des Nutzers stehlen. Heutzutage nutzen fast alle Web-Applikationen Nutzername und Passwort für die Authentifizierung. Daher muss die Angreiferin genau diese Informationen stehlen, um Zugang zu erhalten.

2. **Manipulation von Transaktionen:** Das Hauptproblem, bei der Durchführung der Autorisierung von Transaktionen ist, dass das verwendete Endgerät mit spezialisierter Schadsoftware infiziert wird, die der Angreiferin clientseitig volle Kontrolle über die Online-Banking Umgebung gibt (z.B. Ändern von Kontoständen oder Transaktionsempfängern) [4]. Dies wird von Angreifern genutzt, um komplexe *Social-Engineering* Angriffe durchzuführen, um beispielsweise private Daten zu stehlen (z.B. Handynummern) [6]. Daten, die an den Server gesendet werden, können von der Angreiferin geändert werden, ohne dass der Nutzer die Änderung bemerkt, was dazu führt, dass weder Client noch Server prüfen können, ob die übertragenden Daten verändert wurden. Auch eine TLS gesicherte Verbindung löst dieses Problem nicht, da die Manipulation vor bzw. nach dem Verschlüsseln durchgeführt werden.
3. **Vertrauenswürdige Anzeige:** Aus dem beschriebenen Problem ergeben sich zahlreiche weitere Problemstellungen. Eine Lösung ist die Nutzung eines unabhängigen Kommunikationskanals als „vertrauenswürdige Anzeige“ (z.B. ein Smart Phone oder externe TAN-Generatoren), um zu prüfen ob die Daten, die der Server erhalten hat tatsächlich die eingegebenen Daten sind. Der Vorteil von externen Geräten ist, dass diese praktisch nicht mit Schadsoftware infiziert werden können. Da zur Autorisierung von Transkationen ein spezielles Gerät benötigt wird, werden diese meist nur zuhause genutzt. Daher bevorzugen Nutzer das Smart Phone als zweiten Kanal [7], welches allerdings einfacher anzugreifen ist und bereits häufig erfolgreich angegriffen wurde [2].

3 Konzept

Das in dieser Arbeit vorgestellte Protokoll nutz neben einer digitalen Identität keine weitere Software oder Hardware Entitäten und könnte somit auf beliebigen Geräten genutzt werden, die über einen Web-Browser verfügen. Dies umfasst auch Geräte, die mit Schadsoftware infiziert sind oder Geräten denen nicht vollständig vertraut wird. Von nun an nutzen wir Online-Banking Überweisungen als Beispiel für Online-Transaktionen und beschreiben unseren Ansatz anhand dieses Beispiels. Im Allgemeinen lässt sich unser Ansatz aber auf alle Online-Transkationen anwenden (z.B. Einkaufen in einem Online-Shop).

3.1 Nutzung digitaler Identitäten

In dem Protokoll wird eine digitale Identität zur Authentifikation (2-Faktor-Authentifikation) und Autorisierung der Transaktion (digitale Signatur) genutzt. Die Transaktion wird digital signiert, um sie kryptografisch zu sichern und zu autorisieren. Wir können nicht davon ausgehen, dass die digitale Identität über eine geeignete Anzeige verfügt, um die Transaktionsdaten zu sichern. Daher stellen wir in dieser Arbeit den Ansatz kontextsensitiver CAPTCHAs vor, die diese Aufgabe übernehmen. Da eine physische Interaktion mit der digitalen Identität nötig ist, kann eine potentielle Angreiferin nicht unbemerkt eigene Transaktionen durchführen - und die CAPTCHAs lösen - nachdem sich ein Nutzer eingeloggt hat. Beispiele für solche digitalen Identitäten sind elektronische Personalausweise oder das YubiKey Token [8].

3.2 Adaption des CAPTCHA Prinzips

Wie bereits mehrfach erwähnt, kann der Nutzer der Anzeige seines Geräts nicht vertrauen. Zur Lösung dieser Herausforderung adaptieren wird das CAPTCHA Prinzip, um eine Transkation zu autorisieren.

Dazu muss der Nutzer zwei Aufgaben erfüllen: (1) Beantworten einer Frage (dem CAPTCHA) und (2) die Transaktion digital unterschreiben. Die Frage besteht grundsätzlich aus zwei Typen von Informationen: (a) Informationen, die der Nutzer zum Beantworten benötigt; und (b) weiteren Informationen, welche die Angreiferin verwirren soll (Beispiel: *„Addieren Sie die 3te und 4te Nummer der IBAN. Die Differenz der 5ten und 7ten Stelle ist nicht von Interesse.“).* Dabei wird angenommen, dass eine Angreiferin nicht die Kontrolle über den Server hat und somit nicht die Antwort, die der Server erwartet beeinflussen kann. Somit muss sie, für einen erfolgreichen Angriff, die Antwort zu der Frage bestimmen (diese Wahrscheinlichkeit wird in Kapitel 6 beschrieben).

Bei der Generierung der Frage ist es unabdingbar, sich auf Teile der Transaktion, die von der Angreiferin höchst wahrscheinlich manipuliert werden (z.B. Zahlungsempfänger), zu beziehen. Im vorangegangen Beispiel ist die Frage an die IBAN des Empfängers gebunden. Nur wenn die neue IBAN an den gleichen Stellen die gleichen Zahlen hat ist der Ansatz nutzlos. Die Wahrscheinlichkeit dafür is 2% ($^1/_{10} * \ ^1/_{10} * 2 = 0{,}02$). Die IBAN wurde gewählt, da dies der einzige Weg für die Angreiferin ist Geld auf ein Konto zu überweisen, dass unter ihrer Kontrolle steht. Unser Ansatz schützt also *nicht* unbedingt alle Teile einer Transaktion. Nur falls eine Frage generiert werden würde, die alle Teile der Transaktion mit einbezieht, würde dies der Fall sein. Letztlich bedeutet dies, dass sich der Nutzer, falls die Frage wie oben beschrieben gestellt wird, nur sicher sein kann das Geld an den vorgesehen Empfänger übertragen wird. Würde die Frage mehrere Teile der Transaktion mit einbeziehen (z.B. Betrag oder Verwendungszweck) würde diese wahrscheinlich komplexer werden, was einen negativen Einfluss auf die Nutzbarkeit des Systems hätte. Da aber, durch den Schutz der Empfänger IBAN die Angreiferin potentiell keinen finanziellen Nutzen hat, wird ihre Motivation das System anzugreifen deutlich verringert.

Als Unterstützung für den Nutzer werden ihm vier Antwortmöglichkeiten für die gestellte Frage angeboten. Dabei sind alle Antwortmöglichkeiten Antworten auf mögliche Fragen, die sich aus den zusätzlichen Informationen in der Frage beziehen (siehe Kapitel 5). Die Antwortmöglichkeiten werden so gewählt, dass der Angreifer aus diesen keine Informationen bezüglich der Frage extrahieren kann. Beispiel: Wenn in der Frage die Nummern 3, 4 und 8 genannt werden, können daraus unterschiedliche Aufgaben generiert werden (z.B. $3 + 4$, $8 - 3$, etc.). Jede Antwortmöglichkeit ist die Lösung zu einer dieser Aufgaben. Somit kann die Angreiferin keine der möglichen Antworten ausschließen. Mögliche Antworten vorzuschlagen hat den Vorteil, dass die Lösung bei einer Manipulation durch eine Angreiferin nicht angezeigt wird und der Nutzer die Transkation abbrechen kann (siehe auch Abschnitt 6.3).

Der schematische Ablauf des Protokolls wird in Abbildung 1 dargestellt und wird im Folgenden kurz beschrieben. Nach einer erfolgreichen 2-Faktor-Authentifizierung (1) gibt der Nutzer die Transaktionsdaten ein und sendet diese an den Server (2). Der Server generiert eine Frage zu der Transaktion und sendet diese an den Client (3). Die Frage wird dem Nutzer angezeigt (4) und dieser löst die Aufgabe (5). Anschließend signiert der Nutzer die Transaktion digital (6). Abschließend prüft der Server die Antwort (7a) und die digitale Signatur (7b). Wenn beide korrekt sind wird die Transkation ausgeführt.

Durch die Nutzung einer digitalen Signatur und einer kontextsensitiven Frage wird sichergestellt, dass nur der Besitzer der digitalen Identität eine Transaktion autorisieren kann. Sollte die Transaktion manipuliert werden bemerkt dies entweder der Server oder der Nutzer.

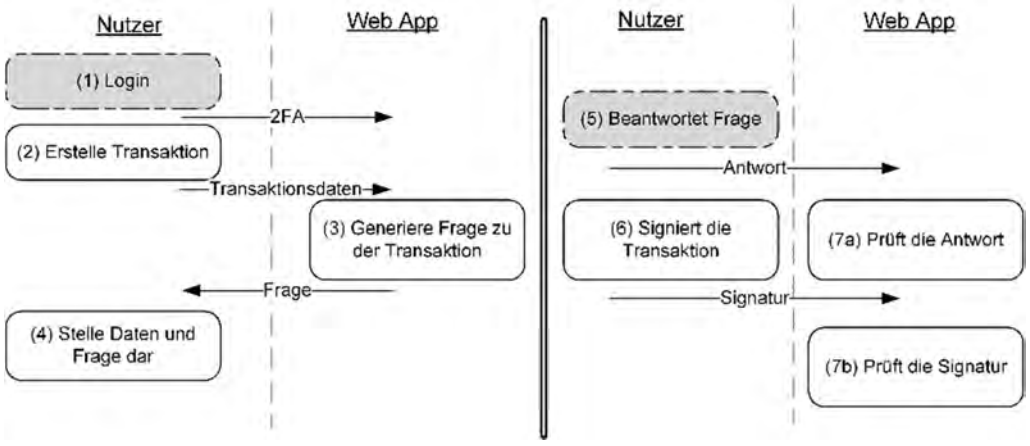

Abb. 1: Systematischer Ablauf des Protokolls

Die Angreiferin oder jemand, den sie eingestellt hat (z.B. eine Firma, die CAPTCHAs löst [9]) könnte die Transaktion per Hand lösen oder nur einzelne Transkationen live überwachen. Diese Option würde allerdings nur sehr schwer skalieren und wäre somit nicht für groß angelegten Betrug im Online-Banking geeignet.

4 Erwarteter Angriffsvektor auf das Protokoll

Vor der Sicherheitsbetrachtung stellen wir in diesem Kapitel den erwarteten Angriffsvektor auf unser Protokoll vor. Wir machen die folgenden Annahmen zu der Angreiferin:

1. **(A1)** Die Angreiferin kann die Frage *nicht* automatisch verstehen und parsen. Das automatische Parsen und Verstehen von Texten ist immer noch ein schweres Problem in der künstlichen Intelligenz (siehe auch Kapitel 8).
2. **(A2**) Die Angreiferin *kann* alle Zahlen und Wörter aus der Frage extrahieren ohne den Kontext dieser zu verstehen.
3. **(A3)** Die Fragen werden *nicht* aus einem kleinen Set aus Fragen generiert. Die Angreiferin kann sich also *nicht* auf einige wenige Fragestellungen einstellen.
4. **(A4)** Das Protokoll nutzt nur Additionen und Subtraktionen, was für Menschen die Bedingung des Systems vereinfacht.
5. **(A5)** Die Angreiferin hat *keinen* Zugriff auf den Online-Banking Server. Sollte die Angreiferin Zugriff auf den Server haben, könnte sie die Transaktion einfach dort ändern und jedes clientseitige Sicherungsverfahren wäre nutzlos.

Basierend auf diesen Annahmen gehen wir von dem folgenden Angriffsvektor aus. Die Angreiferin kann eine Transaktion nur manipulieren, wenn sie dem Server die richtige Antwort für die manipulierte Transaktion geben kann und der Nutzer die Transaktion digital signiert (durch physische Interaktion mit der digitalen Identität). Zur Bestimmung der korrekten Antwort kann die Angreiferin unterschiedliche Informationsquellen nutzen.

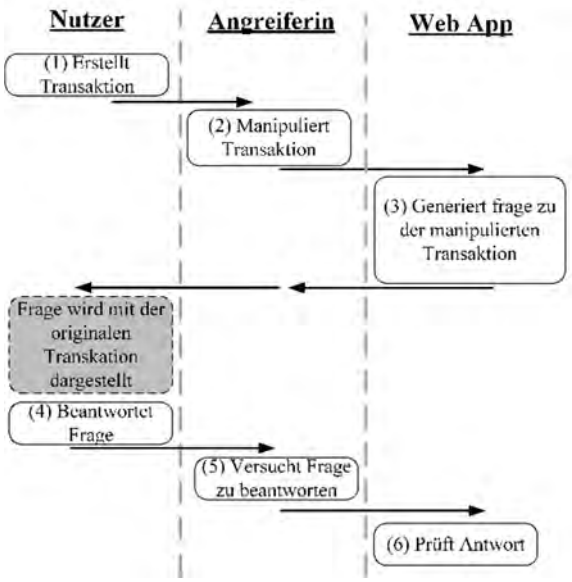

Abb. 2: Ablauf des erwarteten Angriffs

1. **(I1)** Einzelne Wörter und Zahlen aus der Frage (A1)
2. **(I2)** Alle Daten, die der Nutzer eingegeben hat
3. **(I3)** Alle Informationen über die manipulierte Transaktion (z.B. neue IBAN)
4. **(I4)** Die Antwort des Nutzers auf die Frage (falls vorhanden)

Nach dem Empfangen der Frage kann die Angreiferin die n Zahlen aus der IBAN extrahieren, die von Bedeutung sind (I1 und I3). Man bedenke, dass einige der Zahlen in der Frage zum Verwirren der Angreiferin eingesetzt werden. Aus diesen Zahlen kann die Angreiferin $3 * \binom{k}{2}$ Aufgaben generieren ($\binom{k}{2}$ Additionen und $2 * \binom{k}{2}$ Subtraktionen). Die ebenfalls übermittelten Lösungsvorschläge passen zu mindestens einer dieser Aufgaben (siehe Kapitel 3). Falls der Nutzer eine Lösung auswählt erhält die Angreiferin weitere Informationen (**I4**). Der gesamte Ablauf wird nochmals in Abbildung 2 dargestellt.

5 Generierung von Fragen

Angesichts des gerade beschriebenen Angriffsvektors ist es essentiell, dass die Angreiferin so wenig nutzbare Informationen wie möglich aus den Fragen extrahieren kann. In dieser Arbeit definieren wir eine Frage wie folgt: $Q = (F, P, S_C, T)$ mit $F = (F_1, F_2, \dots, F_k)$ den unterschiedlichen falschen Antworten; $P = (P_1, P_2, \dots, P_n)$ den unterschiedlichen Positionen, die auf die Transaktion Bezug nehmen; S_C der richtigen Antwort und T, dem Fragetext. Zur Bestimmung dieser Parameter wird Algorithmus 1 verwendet. Zuerst werden n Positionen aus der gegebenen Empfänger-IBAN (A) extrahiert. S_C wir berechnet indem ein zufälliger Arithmetischer Operator ($\otimes$) auf die ersten beiden zufällig gewählten Zahlen (an den bestimmten Positionen) angewendet wird. Dementsprechend werden die k falschen Antworten mit den restlichen bestimmten Zahlen berechnet. Zuletzt wird der Fragentext generiert. Bei dem Generieren des Textes ist es unabdingbar viele unterschiedliche Satzstrukturen und Formulierungen zu nutzen, damit der Angreifer die Struktur der Frage nicht vorhersagen kann (siehe auch Kapitel 10).

Das folgende Beispiel zeigt wie eine Frage generiert werden könnte. Gegeben sei die IBAN $A = 12345$. Die folgenden Positionen werden bestimmt: $P = (0,4,2,1,3)$. Als arithmetischer Operator wird + gewählt und somit gilt: $S_C = 1 + 4$. Die folgende gekürzte Liste enthält mögliche falsche Lösungen $F = (3,7,-2,\dots)$. Mit diesen Parametern könnte die folgende Frage generiert werden: „*Addieren Sie die 1te und 5te Stelle der IBAN. Die Subtraktion der 3ten und 2ten Stelle ist uninteressant*."

Algorithm 1: Erstellen der Fragen

Eingabe : Eine IBAN A
Ausgabe: Eine Frage für die gegebene IBAN

```
1 P|| = bestimmeZufälligePositionen(A)    /* n zufällige Ziffern aus A. */
2 ⊗ = bestimmeZufälligenOperator()       /* zufälliger arith. Operator. */
3 S_C = A[P[1]] ⊗ A[[P[2]]               /* Berechne die richtige Lösung. */
4 S|| = bestimmeLösungen(A, P)  /* Bestimme mögliche Lösungen mit A, P. */
5 F = S \ S_C                            /* Entferne die richtige Lösung. */
6 T = generiereFragentext(P, F, ⊗, A)      /* Bestimme den Fragentext. */
7 return Q(F, P, S_C, T)
```

Abb. 3: Generierung aller Parameter einer Frage

6 Sicherheitsbetrachtung

In diesem Kapitel berechnen wir die Wahrscheinlichkeit (WK) mit der ein motivierter Angreifer unser Protokoll erfolgreich angreifen kann.

Wir machen die folgenden Annahmen:

1. **S1:** Die Frage bezieht sich *nur* auf die Empfänger IBAN
2. **S2:** Die Frage enthält fünf Positionen aus der IBAN
3. **S3:** Es kommen nur Additionen und Subtraktionen in den Fragen vor (siehe A5)

Die WK, dass die manipulierte IBAN an den relevanten Stellen die gleichen Zahlen hat und der Angreifer die Lösung somit einfach übernehmen kann, ist für Additionen $2 * {}^1/_{10} * {}^1/_{10} = 0{,}02$ und $0{,}01$ für Subtraktionen, wo die Reihenfolge der Zahlen eine Rolle spielt. Wir gehen von zwei unterschiedlichen Angreifer-Typen aus und bestimmen jeweils die WK, dass diese die Fragen automatisch richtig beantworten. Zum einen gehen wir von einer *einfachen Angreiferin* aus, die keine Informationen aus der Frage extrahieren konnte und zum anderen von einer *spezialisierten Angreiferin*, die relevante Daten aus der Frage extrahieren konnte.

6.1 Einfache Angreiferin

Die einfache Angreiferin hat keinerlei Informationen aus der Frage extrahiert und kann somit die Lösung auf die Frage nur „blind" raten. Für jede k-Stellige IBAN können $3 * \binom{k}{2} = \Omega$ unterschiedliche Aufgaben generiert werden. Die WK, dass eine Lösung eindeutig, für eine gewisse Frage, ist ergibt sich wie folgt: ${}^n/_\Omega$ mit *n* der Anzahl eindeutiger Lösungen für eine IBAN. Eine Lösung ϵ ist eindeutig, wenn es genau eine Aufgabe in Ω gibt, die ϵ als Lösung hat. Beispiel mit $k = 18$ und $n = 6$ ergibt sich ${}^6/_{459} = 0{,}013$. Zum Berechnen der totalen Wahrscheinlichkeit, ob eine eindeutige Frage gestellt wird definierten wir die Funktion $\pi(j)$, welche die Anzahl an IBANs bestimmt, die genau *j* eindeutige Lösungen haben. Die WK, dass eine eindeutige Aufgabe gewählt wird ergibt sich dann über das Verhältnis der eindeutigen Aufgaben zu allen Aufgaben: ${}^j/_\Omega = \sigma(j)$.

Mit dem Verhältnis des prozentualen Aufkommens einer n-Stelligen IBAN mit *j* eindeutigen Lösungen zu allen n-stelligen IBANs ($^{\pi(j)}/_{10^n} = \omega(j)$) ergibt sich die gewichtete WK, die angibt wie wahrscheinlich es ist, dass eine eindeutige Aufgabe gewählt wird, die genau j Lösungen hat ($\sigma(j) * \omega(j)$). Insgesamt ergibt sich die totale Wahrscheinlichkeit, dass eine eindeutige Frage gewählt wird, welche die Angreiferin direkt lösen kann, durch die Summierung aller gewichteten WKs: $\sum_j \sigma(j) * \omega(j)$. Wie haben die totale WK für k = 18 berechnet. Wir haben uns für k = 18 entschieden, weil wir die Berechnung in vertretbaren Zeit und unter Verwendung einer realistischen IBAN-Länge durchführen wollten (IBANs sind zwischen 15 und 32 Zeichen lang). Die totale WK, dass eine solche Aufgabe gewählt wird ist fast 0 (0,02%). Die extrem hohe Zahl an möglichen unterschiedlichen Aufgaben ist hauptverantwortlich für diese geringe WK. Es kann davon ausgegangen werden, dass bei längeren IBANs (größeres *k*) die WK weiter fällt, da die Anzahl der berücksichtigten Stellen steigt.

6.2 Spezialisierte Angreiferin

Bei diesem Angreifer-Typ nehmen wir an, dass sie die Frage parsen konnte, die genannten Positionen und entsprechenden Zahlen extrahieren konnte, aber den Kontext, in dem diese genannt wurden nicht verstehen konnte. Die Angreiferin hat ebenfalls die Antwort, die der Nutzer an den Server gesendet hat mitgelesen. Insgesamt kennt die Angreiferin also P, die aus T extrahiert wurden und S_C für die tatsächliche Transaktion. Mit diesen Informationen hat die Angreiferin eine deutlich höhere Chance das System erfolgreich anzugreifen. Die totale WK, dass eine Frage gewählt wird, zu der eine Angreiferin eine korrekte Antwort bestimmen kann, kann wie folgt berechnet werden. Da die fünf relevanten Positionen (S2) aus der Frage extrahiert werden konnten sinkt die Anzahl an möglichen Aufgaben. Es gibt $3 * \binom{5}{2} = \Omega = 30$ unterschiedliche Aufgaben in diesem Szenario. Analog zu der Berechnung des einfachen Angreifer-Typs erhalten wir eine totale WK von 21,07%, dass die Angreiferin das System erfolgreich brechen kann. Dies ist die WK, dass sie die Frage anhand der gegebenen Informationen identifizieren kann.

Eine hoch motivierte Angreiferin wird des Weiteren versuchen mehr über die Struktur der Frage zu lernen, um so ihre Chancen eines erfolgreichen Angriffs zu erhöhen. Dies könnte durch komplexere Fragen verhindert werden, was einen negativen Einfluss auf die Nutzbarkeit des Systems hätte. Ein vielversprechenderer Ansatz wäre es Fragen in einer Weise zu stellen, dass ein Computer diese nicht verstehen kann (z.B. sog. *Winogard* Fragen [10] - siehe Kapitel 8).

6.3 Hinzufügen von Multipel-Choice Fragen

Wie bereits erwähnt muss der Nutzer die richtige Lösung zu der Frage aus vier Lösungskandidaten auswählen. In diesem Kapitel beschreiben wir den Einfluss des Ansatzes auf die Sicherheit des Systems. Der größte Vorteil des Systems ist, dass die richtige Lösung für eine manipulierte Transkation dem Nutzer nicht angezeigt wird und dieser die Transaktion abbrechen und Schaden vorbeugen kann. Natürlich könnte die richtige Antwort zufällig unter den angezeigten Lösungen vorkommen. Daher ist es wichtig zu entscheiden, wie viele Lösungskandidaten den Nutzern angezeigt werden sollen. Die Angreiferin könnte eine Frage erstellen, zu der sie die Antwort kennt und dem Nutzer Lösungskandidaten dazu zur Verfügung stellen. Allerdings wären dies keine Lösungen, die der Server erwartet und der Angriff würde fehlschlagen.

Sei *j* die Anzahl der unterschiedlichen Lösungen einer fünfstelligen Zahl (S2) und sei *k* die Anzahl der angezeigten Lösungen (mit $1 \leq k \leq 28$). Grundsätzlich muss also die Frage „Wie

viele unterschiedliche Lösungen hat eine fünfstellige Zahl“ (Im Kontext dieser Arbeit) beantwortet werden. Wir haben keine fünfstellige Zahl mit $j > 18$ gefunden. Sei $\pi_{\%}(j)$ die Funktion, die den prozentualen Anteil an fünfstelligen Zahlen ermittelt, die genau *j* Lösungen haben. Im Folgenden berechnen wir die WK das, bei einer manipulierten Transkation, die korrekte Lösung unter den *k* angezeigten Bildern ist. Über das Verhältnis der Differenz der eindeutigen Lösungen (*j*) und der angezeigten Bildern (*k*) zu den Lösungen kann die WK berechnet werden, dass die korrekte Lösung *nicht* angezeigt wird: $\delta(k) = \max\left({}^{(j-k)}/_{j}; 0\right)$

Beispiel: Ein Aufgabe hat 4 unterschiedliche Lösungen (alle möglichen Aufgaben haben einen dieser Lösungen zum Ergebnis) nun kann mit unterschiedlichen k bestimmt werden wie hoch die WK ist, dass die korrekte Lösung nicht angezeigt wird. Für eine fünfstellige Zahl mit genau *j* Lösungen und *k* angezeigten Lösungskandidaten wird über $\pi_{\%}(j) * \delta(k)$ die WK berechnet mit der die richtige Lösung *nicht* angezeigt wird. Die totale WK, dass die korrekte Lösung zu der originalen Transkation *nicht* angezeigt wird, kann über die Summierung der einzelnen WKs berechnet werden: $\varphi(k) = \delta(k) * \sum_i \pi_{\%}(i)$. Die WK, dass die Angreiferin die Lösung einfach raten kann ist: $h(k) = {}^{1}/_{k}$.

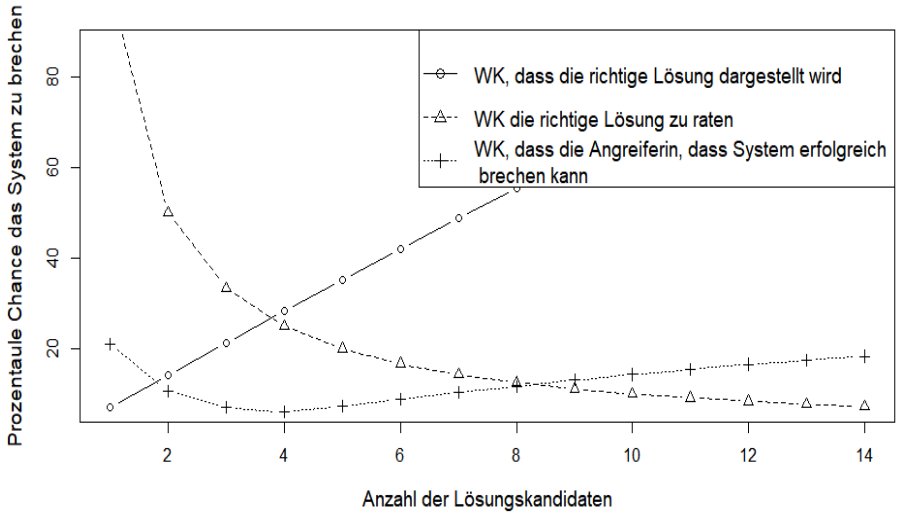

Abb. 4: WK, das System zu brechen

Die WK, dass der Nutzer die Transkation autorisiert, obwohl diese manipuliert wurde, kann über Beziehung $f(k) = \max(1 - \delta(k); h(k))$ bestimmt werden. Zur Härtung des Systems muss $f(k)$ minimiert werden ($\min \cup_k f(k)$). Abbildung 3 zeigt die Funktionsverläufe der Funktionen $h(k)$ [○], $1 - \delta(k)$ [Δ] und $f(k)$ [+]. $f(k)$ erreicht an $k = 4$ das absolute Minimum mit $f(4) = 0{,}25$ an dem Schnittpunkt von $h(k)$ und $1 - \delta(k)$. Zusammen mit der Wahrscheinlichkeit, dass der Angreifer die korrekte Lösung bestimmen kann (21,02%) und der Wahrscheinlichkeiten, dass der Nutzer die Frage beantwortet (25%) erhält man eine gesamte WK, dass die Angreiferin das System bricht, von 5,93% ($0{,}25 * 0{,}2102 = 0{,}0593$).

6.4 Limitierungen

Es ist davon auszugehen, dass die Angreiferin mehrere Bankkonten (unter unterschiedlichen Namen) kontrolliert, um es schwieriger zu machen sie zu identifizieren und den potentiellen

Schaden zu minimieren, falls ein Konto gesperrt wird. Dies hat keinen direkten Einfluss auf die Sicherheit des vorgestellten Systems, da die Angreiferin die IBAN vor dem Generieren der Frage wählen muss. Allerdings könnte sie eine IBAN wählen, die der originalen IBAN ähnlich ist, um ihre Chancen zu erhöhen. Der vorgeschlagene Ansatz schützt ebenfalls nur einige Teile der Transkation, wenn die Angreiferin andere Teile der Transkation manipuliert, werden diese nicht geschützt. Da sie so aber keinen finanziellen Vorteil daraus ziehen kann, ist dieser Angriff eher unwahrscheinlich.

7 Nutzerstudie

Zusätzlich zu der Sicherheitsbetrachtung haben wir eine Nutzerstudie zu dem vorgeschlagenen System durchgeführt. Dazu haben wir einen Prototyp eines Online-Banking Systems implementiert, der unser vorgeschlagenes Protokoll umsetzt. 30 Personen haben an unserer Studie teilgenommen (15 Frauen und 15 Männer). Die Personen wurden in drei Altersklassen sortiert, die jeweils aus 10 Personen bestanden: „jung“ (16-29 Jahre), „mittel“ (30-49) und „alt“ (50-69). Das Durchschnittsalter lag bei 38,96 Jahren. 20 der 30 Teilnehmer nutzen Online-Banking mindesten einmal pro Woche. Innerhalb der Studien wurden zwei verschiedene Szenarien simuliert (1) *nicht* manipulierte Transaktionen und (2) manipulierte Transaktionen (die korrekte Lösung wurde nicht angezeigt). Das Ziel war es herauszufinden, ob die Teilnehmer eine Manipulation erkennen würden. Nur 15 der Teilnehmer wurde vor dem Experiment explizit gesagt, dass eine Manipulation stattfinden könnte und, dass das System vor dieser Bedrohung schützen könnte.

Die Fragen wurden in zwei unterschiedlichen Formen dargestellt („schwarz“ und „bunt“ siehe Abbildung 4). Die bunte Darstellung wurde gewählt, um es der Angreiferin zu erschweren, Informationen zu extrahieren. Allerdings wurden CAPTCHAs, die auf verzerrten farbigen Text setzen, bereits gebrochen [11] und die Nutzbarkeit des System leidet ebenfalls deutlich (siehe weiter unten). Die IBAN wurde in Viererblöcke formatiert, sodass diese einfacher zu lesen ist. Jeder Teilnehmer musste 4 Transaktionen (2 pro Typ) durchführen, die pro Teilnehmer in zufälliger Reihenfolge aufgetreten sind.

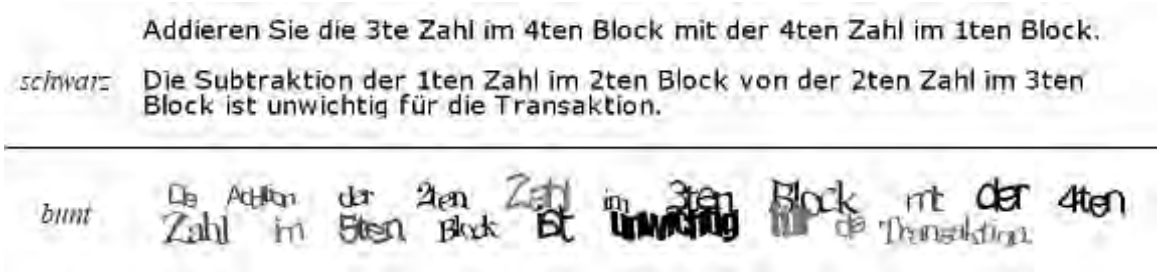

Abb. 5: Schwarze(oben) und ‚bunte‘ Darstellung der Fragen (unten)

Ein verwendetes Maß war es zu messen, wie viele Teilnehmer die Frage im ersten Versuch ohne jegliche Hilfe beantworten konnten. Abbildung 5 zeigt, dass 80% aller Teilnehmer die „schwarze“ Frage beim ersten Versuch beantworten konnten. Jüngeren Teilnehmern fiel das beantworten der Frage leichter als älteren Teilnehmern.

Die unterschiedlichen Szenarien haben deutlichen Einfluss auf die Geschwindigkeit mit der eine Transaktion abgeschlossen werden kann (siehe Tabelle 1). Die Zeiten umfassen die Dauer der gesamten Transaktion inklusive der Eingabe der Transaktionsdaten.

Tab. 1: Zeiten für die Durchführung von Transaktionen (inkl. Dateneingabe)

	Schwarz	Bunt
Keine Manipulation	30s	37s
Manipulation	48s	52s

Der empfundene Stress der Versuchspersonen wurde mittels der NASA-TLX Score [12] bestimmt. Die NASA-TLX Score ist eine weit verbreitete mehrdimensionale Skala, die es erlaubt die Arbeitslast eines Menschen zu messen, wenn dieser eine Aufgabe durchführt. Die Auswertung der NASA-TLX Score hat ergeben, dass die Teilnehmer die bunte Frage als mental schwieriger zu lösen eingestuft haben (schwarz: 4,20 gegen bunt: 5,25).

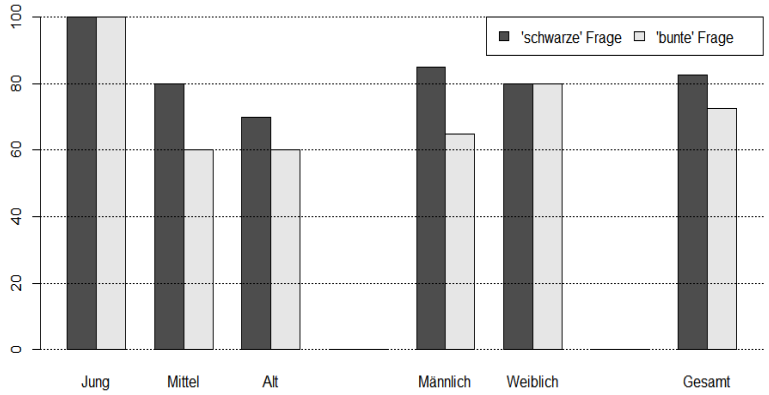

Abb. 6: Teilnehmer, die eine Transaktion im ersten Versuch erfolgreich abgeschlossen haben

73% der Teilnehmer könnten sich vorstellen, das vorgestellte Verfahren im Online-Banking zu verwenden. Allerdings haben sich fast alle Teilnehmer mehr Informationen dazu gewünscht wie das Verfahren sie vor möglichen Angriffen schützt. Die Teilnehmer haben auch angemerkt, dass die Durchführung einer Transaktion insgesamt zu lange dauert. Ebenfalls wurde die Formulierung der Fragen als sehr kompliziert angesehen. Die „bunten“ Fragen wurden als schwer leserlich bezeichnet, daher bevorzugen 90% der Teilnehmer die schwarzen Fragen. Die Nutzerstudie hat gezeigt, dass bei der Weiterentwicklung des Systems vor allem die Fragen vereinfacht werden sollten. Da die bunten Fragen letztlich keinen hohen Sicherheitsgewinn darstellen, sollte auf diese komplett verzichtet werden. Die jüngeren Teilnehmer hatten ebenfalls größere Probleme mit Subtraktionen und haben daher den Wunsch geäußert nur Additionen zu verwenden.

8 Verwandte Arbeiten

CAPTCHAs anderer Domänen: Der Ansatz CAPTCHAs im Online-Banking zu nutzen ist nicht neu. In vorherigen Ansätzen wurde die IBAN in Bildern, den CAPTCHAs, hinterlegt. Diese Bilder haben einige Transaktionsdaten und Informationen zu dem Nutzer enthalten und wurden verzerrt, damit Computer sie nicht einfach lesen können. Li et al. haben erfolgreich gezeigt wie solche Systeme angegriffen werden können [19].

Dazu haben sie Bilderkennungs-Algorithmen genutzt, um die Teile der Bilder zu identifizieren, die sich auf die Transkation bezogen und haben diese manipuliert. Unser Ansatz ist nicht von

diesem Angriffsvektor betroffen, da die Angreiferin die Frage verstehen muss, um die Transaktion zu manipulieren. Die Frage zu manipulieren ist nicht zielführend, da der Server die Antwort auf die originale Frage erwartet.

In [20] untersuchen Shahreza et al., wie Fragen in CAPTCHAs eingebunden werden könnten. Dabei setzen sie auf eine Kombination von Bildern und Texten, um die CAPTCHAs vor der Erkennung von Buchstaben und Zahlen zu schützen. Allerdings sind diese Fragen nicht Kontext bezogen und folgen einfachen Mustern.

1. **Winogard Fragen:** Levesque untersucht in [16] wie Fragen generiert werden können, die nicht von Computerprogrammen, die dem aktuellen Stand der Entwicklung entsprechen, verstanden und beantwortet werden können. Die Autoren nutzen dafür Winogard Fragen. Das automatisierte Beantworten von Fragen, durch Computerprogramme, ist ein aktives Forschungsfeld. Zur Beantwortung von allgemeinen Fragen nutzen diese Ansätze meist große Wissensdatenbanken [17] oder Ergebnisse aus Suchmaschinen [18]. Diese Ansätze sind nur wenig hilfreich bei der Beantwortung von Fragen, die in dieser Arbeit zur Absicherung von Transaktionen vorgeschlagen wurden, da es keine großen Datenquellen gibt, die bei der Beantwortung hilfreich wären.
2. **Phishing Angriffe:** Mechanismen, um Passwort Phishing zu verhindern, sind ebenfalls ein aktives Forschungsfeld. Han Yan et al. stellt ein System vor, dass es Nutzern erlaubt Passwörter auf mobilen Geräten einzugeben, ohne dass diese mitgelesen werden können [21]. Dazu wird dem Nutzer eine versteckte Nachricht angezeigt, um die Beziehung zwischen eingegebenen Zeichen und den Bewegungen, die eine potentielle Angreiferin beobachten kann, verhindert.

Limitierungen und Design Prinzipien für Systeme, die keinen zweiten Kanal zur Übermittlung eines Geheimnisses (z.B. einer TAN) nutzen werden in [22] von Qiang Yan et al. diskutiert. Qiang Yan et al beschrieben die Spannung zwischen Nutzbarkeit und Sicherheit für das in ihrer Arbeit vorgestellte Framework und folgern, dass entweder eine hohe kognitive Arbeit oder aber ein gutes Erinnerungsvermögen bei den Nutzern nötig sind. Die Arbeit befasst sich ebenfalls mit Passwörtern, die nicht mitgelesen werden können.

9 Zukünftige Arbeiten

Neben der Erhöhung der Nutzbarkeit des Ansatzes sollten ebenfalls die Fragen verbessert werden. Die Sicherheit des Systems beruht darauf, dass die Angreiferin die gestellten Fragen nicht automatisiert parsen und verstehen kann.

Dazu sollten sog. Winogard Fragen [10] genutzt werden. Winogard Fragen bestehen aus zwei Sätzen, die eine Uneindeutigkeit ausweisen, die in unterschiedliche Wegen aufgelöst werden kann. Beispiel (übernommen aus [13])*: "The trophy doesn't fit in the brown suitcase because it's too [big/small]. What is too [big/small]? Answer 0: the trophy - Answer 1: the suitcase."*. Zum Beantworten der Frage muss mit „gesundem Menschenverstand" geschlussfolgert werden und es wird ein gewisses Wissen über die Welt benötigt (Trophäen sind meist kleiner als Koffer). Daher wurden Winograd Fragen von Levesque als alternative für den Turing Test vorgeschlagen [13]. Das Beantworten dieser Fragen ist immer noch ein offenes Forschungsfeld der künstlichen Intelligenz [14, 15]. Natürlich müssen sich diese Fragen noch auf die Transkation beziehen.

Zum Beispiel: *Peter und Norbert haben an einem Wettbewerb teilgenommen. Peter, der a+b berechnen musste, fühlte sich* ***[erleichtert / deprimiert]*** *als Norbert, der c+d, berechnen musste,*

*verkündetet, dass **er** den Wettbewerb gewonnen hat.* Frage: *Was hat der Gewinner des Wettbewerbs berechnet?*

10 Fazit

Das von uns vorgestellte Protokoll ist dazu geeignet einen Großteil aller Transkationen im Online-Banking zu sichern (>94%). Die Transaktionen werden gesichert, selbst wenn das verwendete System mit Schadsoftware infiziert wurde oder generell nicht vertrauenswürdig ist (z.B. Öffentlich zugängliche Geräte). Die von uns durchgeführte Nutzerstudie hat gezeigt, dass eine Großzahl der Teilnehmer eine Transkation beim ersten Versuch erfolgreich durchführen konnten und ebenfalls dazu bereit wären, das System im Alltag zu verwenden. Allerdings sollten sich weitere Verbesserungen des Systems auf die Nutzbarkeit beziehen.

Danksagung

Diese Arbeit wurde von dem Bundesministerium für Bildung und Forschung (Förderkennzeichen: 13N13251 & 13N13252, BOB) und dem Ministeriums für Innovation, Wissenschaft und Forschung des Landes Nordrhein-Westfahlen (Förderkennzeichen: 005-1703-0021, MEwM) unterstützt. Wir möchten Herrn R. Widdermann für die Implementierung des Systems danken.

Literatur

[1] Eurostat: Individuals using the internet for internet banking, http://ec.europa.eu/ eurostat/tgm/table.do?tab=table&language=en&pcode=tin00099

[2] S. Golovanov, D. Makrushin, A. Monastyrsky: Staying safe from virtual robbers, https://securelist.com/analysis/user-advice/58328/staying-safe-from-virtual-robbers/

[3] Bundeskriminalamt: Lagebilder Cybercrime, https:// www. bka. de/ DE/ AktuelleInformationen/ StatistikenLagebilder/ Lagebilder/Cybercrime/cybercrime_node.html

[4] T. Dougan, K. Curran: Man in the Browser Attacks. International Journal of Ambient Computing and Intelligence 4 (2012) 29-39.

[5] L. von Ahn, M. Blum, N. J. Hopper, J. Langford: CAPTCHA: Using Hard AI Problems for Security. International Conference on the Theory and Applications of Cryptographic Techniques (2003)

[6] S. Abraham, I. Chengalur-Smith: An overview of social engineering malware. Trends, tactics, and implications. Technology in Society 32 (2010) 183-196.

[7] Initiative D21: Online - Online Banking 2014 Sicherheit zählt! (2014).

[8] Yubico: YubiKey. Trust the Net with YubiKey Strong Two-Factor Authentication, https://www.yubico.com/.

[9] 2Captcha: Human-powered CAPTCHA-solving service, https://2captcha.com/.

[10] T. Winograd: Understanding natural language. Cognitive Psychology 3 (1972) 1-191.

[11] E. Bursztein, M. Martin, J. Mitchell: Text-based CAPTCHA strengths and weak nesses. In: Chen, Y., Danezis, G., Shmatikov, V. (eds.) the 18th ACM conference, p. 125

[12] S. G. Hart, L. E. Staveland: Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In: Human Mental Workload, 52 (1988) 139-183.

[13] H. J. Levesque, E. Davis, L. Morgenstern: The Winograd schema challenge. In: AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning, 46 (2011) 47.

[14] S. Arpit, H. V. Nguyen, A. Somak, B. Chitta: Towards addressing the winograd schema challenge - Building and using a semantic parser and a knowledge hunting module. International Joint Conferences on Artificial Intelligence (2015).

[15] P. Schüller: Tackling Winograd Schemas by Formalizing Relevance Theory in Knowledge Graphs, http://www.aaai.org/ocs/index.php/KR/KR14/paper/view/7958 (2014).

[16] H. J. Levesque: On our best behaviour. Artificial Intelligence 212 (2014) 27-35.

[17] Q. Cai, A. Yates: Large-scale Semantic Parsing via Schema Matching and Lexicon Extension. Proceedings of the Annual Meeting of the Association for Computational Linguistics (2013).

[18] C. Kwok, O. Etzioni, D. S. Weld: Scaling question answering to the web. ACM Trans. Inf. Syst. 19 (2001) 242-262.

[19] S. Li, S. A. H. Shah, M. A. U. Khan, S. A. Khayam, A.-R. Sadeghi, R. Schmitz: Breaking e-banking CAPTCHAs. Annual Computer Security Applications Conference (2010) 171-180.

[20] M. Shirali-Shahreza, S. Shirali-Shahreza: Question-Based CAPTCHA. In: International Conference on Computational Intelligence and Multimedia Applications, 2007.

[21] Q. Yan, J. Han, Y. Li, J. Zhou, R. H. Deng: Designing leakage-resilient password entry on touchscreen mobile devices, 8th ACM SIGSAC symposium, 2013.

[22] Q. Yan, J. Han, Y. Li, R. D. Huijie: On Limitations of Designing Usable Leakage-Resilient Password Systems: Attacks, Principles and Usability. 19th Network and Distributed System Security Symposium, 2012.