Entwicklung und Evaluierung eines Grafischen Passwortsystems Basierend auf Google Maps

Developing and Evaluation of a Graphical Password System Based on Google Maps

Bachelor-Thesis von Natalie Faber

April 2012

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich 20 - Informatik
Sicherheit in der Informationstechnik

CASED

Entwicklung und Evaluierung eines Grafischen Passwortsystems Basierend auf Google Maps
Developing and Evaluation of a Graphical Password System Based on Google Maps

Vorgelegte Bachelor-Thesis von Natalie Faber

Prüfer: Prof. Dr. Michael Waidner
Betreuer: Lukas Kalabis

Tag der Einreichung:

Hiermit versichere ich, die vorliegende Bachelor-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 1. April 2012

(N. Faber)

Kurzfassung

In den letzten Jahren wurden bereits viele grafische Passwortsysteme entwickelt, dennoch werden heutzutage immernoch fast nur textuelle Passwörter verwendet. In dieser Bachelor-Thesis werden viele der bereits existierenden grafischen Passwortsysteme veranschaulicht. Diese werden dann mit einem neu entwickelten, auf Google Maps basierendem, grafischen Passwortsystem in Bezug auf ihre Benutzbarkeit und Sicherheit, sowie der Schwierigkeit sich das Passwort zu merken, verglichen. Es werden Ergebnisse einer Evaluation des neuen Systems präsentiert, welche eine erfolgreiche Anmeldequote von 97% aufweisen.

Inhaltsverzeichnis

Heutzutage wird für fast jede Anwendung eine Authentifizierung durch den Benutzer benötigt. Standard hierfür sind textuelle Passwörter mit Kombinationen aus kleinen und großen Buchstaben, Zahlen, sowie Sonderzeichen. Textuelle Passwörter haben den großen Vorteil, dass sie leicht zu programmieren und bei den meisten Benutzern bereits bekannt sind. Benutzer können direkt ein Passwort erstellen, ohne Anweisungen zu lesen oder sich einarbeiten zu müssen. Allerdings sind textuelle Passwörter schwer zu merken. Deswegen wählen viele Personen schwache Passwörter, wie ihr Geburtsdatum, den Namen eines Familienmitglieds oder eine Kombination aus einem Namen und dem aktuellem Jahr bzw. Geburtsjahr [Elf06, YBAG04].

Mit der Einführung eines Passwort-Managers müssen Benutzer sich nur noch ein textuelles Master-Passwort merken und der Manager generiert und merkt sich alle weiteren Passwörter. Passwort-Manager sind zwar benutzerfreundlicher als normale textuelle Passwörter, sollte aber ein Angreifer das Master-Passwort rausfinden, hat er direkt alle. Chiasson et al. zeigten einige Sicherheitslücken in Passwort-Managern auf [COB06].

Schon vor langer Zeit wurde gezeigt, dass sich das menschliche Gehirn viel besser Bilder als Texte merken kann [Kir94, She67]. Diese Eigenschaft soll für grafische Passwörter genutzt werden. Seit 1999 wurden viele grafische Passwörter entwickelt. Diese können die verschiedensten Arten haben, wie z. B. ein vom Benutzer selbstgezeichnetes Bild, das Setzen von Punkten auf Bildern, oder das Merken von kompletten Bildern. Ziel der Entwicklung ist es, Benutzern Passwörter zur Verfügung zu stellen, die leicht zu bedienen, zu merken und dennoch sicher sind. Obwohl inzwischen viele grafische Passwörter existieren, werden im Alltag größtenteils nur textuelle Passwörter verwendet, dabei haben grafische Passwörter einige Vorteile gegenüber textuellen Passwörtern. Diese sind nicht nur leichter zu merken, sondern auch gegen viele Angriffe sicherer.

Neben den bereits veröffentlichten grafischen Passwortsystemen [BCO11, HAIM08] soll in dieser Bachelor-Thesis auch ein neu entwickeltes, auf Google Maps basierendes, grafisches Passwortsystem vorgestellt werden. In diesem kann ein Benutzer sein Passwort durch das Setzen von Punkten auf einer Google Maps Karte erstellen. Es soll gezeigt werden, dass das grafische Passwort sowohl leicht zu merken, gut zu bedienen als auch sicher ist. Das Passwortsystem wurde einer Evaluation mit 37 Personen unterzogen, deren Ergebnisse im Kapitel 6 präsentiert werden.

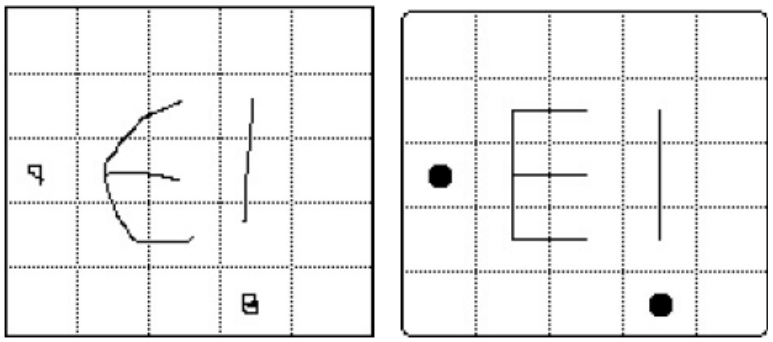
2 Arten von grafischen Passwörtern

Sehr viele Personen wählen unsichere Passwörter [YBAG04], oder vergessen diese und müssen ihr Passwort immer wieder zurücksetzen. Mittels grafischer Passwörter soll die Erinnerung an das eigene Passwort gesteigert werden. Wie durch mehrere Studien gezeigt, kann das menschliche Gehirn sich viel besser Bilder merken als Wörter [Kir94, She67]. Seit dem Ersten grafischen Passwortsystem (DAS [JMM+99]), entwickelt im Jahre 1999, wurden viele weitere grafische Passwortsysteme erfunden. In diesem Kapitel werden einige dieser Systeme vorgestellt. Die einzelnen Passwortsysteme werden in die Kategorien *Recall-based systems*, *Cued-recall systems* und *Recognition-based systems* eingeordnet [RS92, BCO11]. Die Kategorien beschreiben wie schwierig es ist, sich an ein Passwort zu erinnern.

2.1 Recall-based systems

Alle grafischen Passwörter, die in diesem Abschnitt aufgezählt werden, sind sogenannte *recall-based systems*. Wie bei textuellen Passwörtern muss der Benutzer auch hier sein erstelltes Passwort auswendig wissen und es jederzeit wieder aufrufen können.

Draw-A-Secret (DAS) (Abbildung 2.1) wurde 1999 von Jeremyn et al. entwickelt [JMM+99]. Benutzer können ein Passwort mit DAS erstellen, indem sie es in ein 2D Gitter zeichnen. Dass Passwort kann in einem Zug oder mit mehreren Strichen gezeichnet werden. Nachdem die Zeichnung beendet wurde wird das Bild als Sequenz von durchgezogen Gitterzellen kodiert. Wenn sich ein Benutzer am System anmelden möchte, muss er das Bild wieder so zeichnen, dass es die gleiche Kodierung hat.



(a) Benutzereingabe (b) Interne Repräsentation

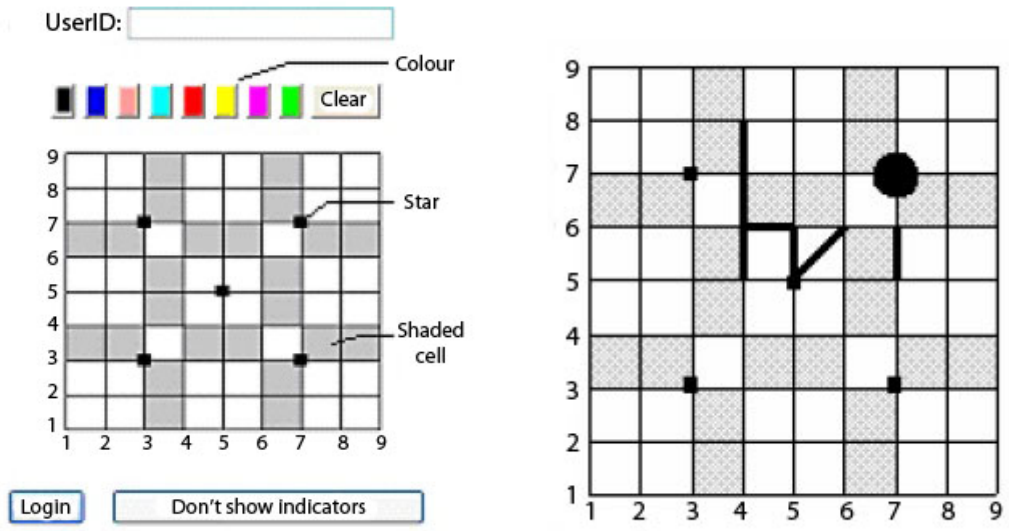
Abbildung 2.1: Draw-A-Secret [JMM+99]. In (a) ist die Zeichnung eines Benutzers zu sehen und in (b) die dazugehörige Repräsentation durch welche Gitterzellen gezeichnet wurde.

In Evaluationen hat sich ergeben, dass viele Benutzer ihre Zeichnung direkt in die Mitte des Gitters setzen und meistens auch in einem Zug oder mit wenigen Strichen erstellen [NT04]. Dies erleichtert Angreifern das Passwort zu erraten (siehe Kapitel 3). Dunphy und Yan fügten zum DAS System einen Hintergrund (**Background Draw-A-Secret**) hinzu, in der Hoffnung Benutzer zu motivieren, ihre Passwörter komplizierter zu wählen und sie sich dennoch gut merken zu können [DY07]. Mit BDAS haben die Benutzer mehr Striche und weniger Symmetrie in ihren Bildern verwendet, ohne dass die Erinnerung an das Passwort nachgelassen hat.

Mit **Passdoodle**, einem DAS ähnlichem System, können Benutzer ebenfalls eine Zeichnung als Passwort erstellen [GHS02, Var04]. Im Gegensatz zu DAS gibt es bei Passdoodle allerdings kein

Gitternetz im Hintergrund. Des Weiteren hat das System extra Eigenschaften, wie das Speichern von verschiedenen Stiftfarben, der Anzahl der gemachten Striche, sowie der Zeichengeschwindigkeit des Benutzers.

Pass-Go, entwickelt von Tao und Adams [Tao06, TA08], baut auf einem Brettspiel namens Go auf. Beim Erstellen eines Passworts wählt der Benutzer Schnittpunkte anstatt von Zellen (DAS) auf einem 9x9 großen Gitter aus. Das Gitter hat schattierte Felder und Sterne auf einigen Feldern zur leichteren Orientierung auf dem Gitter und einer besseren Erinnerung an das Passwort (siehe Abbildung 2.2 (a)). Der Benutzer kann auf diesem Gitter Punkte und/oder Linien in verschiedenen Farben setzen. Ein mögliches Passwort ist in der Abbildung 2.2 (b) zu sehen. Eine Benutzerstudie [TA08] zu Pass-Go ergab, dass 78% der Teilnehmer sich am System anmelden konnten.

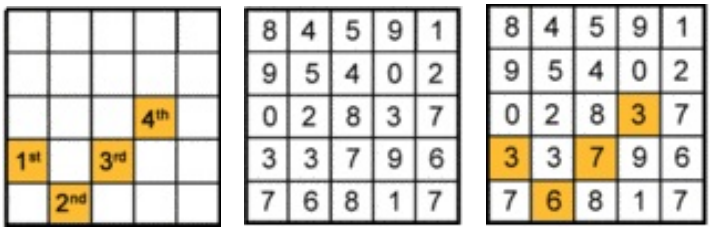


(a) Passwortfeld (b) Benutzereingabe

Abbildung 2.2: Pass-Go [Tao06]. In (a) ist das Pass-Go Feld zu sehen, in welches ein Passwort wie in (b) eingezeichnet wird.

GrIDsure ist ein weiteres *recall-based* System [BIS10, GRI, BCO11]. Der Benutzer erstellt sein Passwort, indem er sich aus einem 5x5 Gitter ein Muster wählt (siehe Abbildung 2.3 (a)). Bei der Anmeldung im System wird dem Benutzer wieder ein 5x5 Gitter angezeigt, in dem zufällige Zahlen angezeigt werden. Damit die Anmeldung erfolgreich ist, muss er sein Muster auf dieses Gitter anwenden (siehe Abbildung 2.3 (c)). Hierbei werden die Zahlen die auf den Positionen des Musters stehen in ein Passwortfeld eingegeben. In einer Benutzerstudie zu GrIDsure stellte sich heraus, dass auch bei diesem System die Benutzer ihre Passwörter schwach wählen [BIS10]. Viele haben ihr Muster als Linie und die Reihenfolge der Punkte im Muster entweder von oben nach unten oder von links nach rechts gewählt. 87% der Teilnehmer an dieser Studie konnten sich direkt mit dem ersten Versuch erfolgreich anmelden. Zwei Jahre später haben einige der Teilnehmer an einer zweiten Studie teilgenommen. Von diesen konnten sich 12% noch mit dem ersten Versuch erfolgreich anmelden.

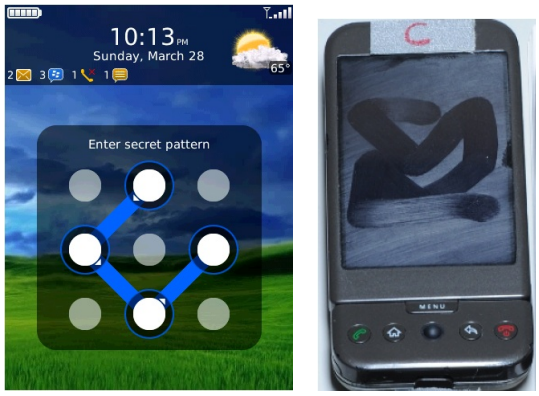
PatternLock ein unter Smartphone Benutzern weiter verbreitetes grafisches Passwort, wurde von Tafasa für das *BlackBerry* entwickelt [Taf]. Anstelle von der Eingabe des eigenen PINs, wird hier der Finger über die Zellen eines 3x3 großen Gitters gezogen. Für die Eingabe des PINs 2486, wird der Finger von der 2 zur 4 dann zur 8 und zuletzt zur 6 gezogen (siehe Abbildung 2.4 (a)). Die Zahlen auf dem Bildschirm werden allerdings nicht angezeigt, sondern nur ein Gitter aus Kreisen.



(a) Muster (b) Feld mit zufälligen Zahlen (c) Anwendung des Musters

Abbildung 2.3: GrIDsure [GRI]. In (a) ist das vom Benutzer gewählte Muster zusehen, welches auf ein Feld mit zufälligen Zahlen (b) angewendet wird (c) und somit dem Benutzer ermöglicht sich anzumelden.

Aviv et al. [AGM+10] zeigten, wie mit *smudge attacks* ein Passwort wie beim PatternLock herausgefunden werden kann. Bei *smudge attacks* kann der Angreifer, von den nach der Bildschirmentsperrung hinterlassenen Schmierflecken der Finger, das Passwortmuster ableiten (siehe Abbildung 2.4 (b)).



(a) PatternLock [Taf] (b) Smudge attack [AGM+10]

Abbildung 2.4: PatternLock. In (a) ist ein PatternLock für den PIN 2486 auf einem BlackBerry zu sehen. In (b) ist ein *smudge attack* zu sehen. Hier ist klar erkennbar, dass das Muster entweder 215368479 oder genau in umgekehrter Reihenfolge ist.

2.2 Cued-recall systems

Im Gegensatz zu *recall-based systems*, müssen bei *cued-recall systems* die Passwörter nicht direkt wieder aufrufbar sein. Der Benutzer kriegt zunächst einen Hinweis zu seinem Passwort, woraufhin er erst das Passwort wieder aufrufen muss. Dies macht es ihm leichter, sich sein Passwort zu merken. Bei solchen Systemen muss sich der Benutzer meistens Orte in einem Bild merken, aber nicht das Bild selbst.

In **PassPoints**, von Wiedenbeck et al. [WWB+05], kriegt der Benutzer vom System ein Bild vorgelegt und kann in diesem Bild fünf Punkte als sein Passwort setzen (siehe Abbildung 2.5 (a)). Bei der Anmeldung am System muss der Benutzer die Punkte in der richtigen Reihenfolge und mit einer vom System festgesetzten Klickpunkttoleranz genau anklicken. Die Anzeige des Bildes bei der Anmeldung zum System gilt als Hinweis zum Passwort, daher ist PassPoints das erste entwickelte *cued-recall* System.

Suo hat PassPoints um eine Eigenschaft erweitert [Suo06]. Das Bild wird bis auf einen kleinen Bildausschnitt verschwommen angezeigt. Während der Anmeldung muss der Benutzer Y (Klickpunkt ist im angezeigten Bildausschnitt enthalten), oder N (Klickpunkt ist im angezeigten Bildausschnitt nicht enthalten) auf der Tastatur eintippen. Dabei werden bis zu 10 Runden (bis alle fünf Klickpunkte gefunden wurden) durchgeführt (siehe Abbildung2.5 (b) und (c)). Diese Erweiterung wurde zum Schutz gegen sogenannte *Shoulder-surfing* Angriffe (siehe Kapitel 3) eingeführt. Der Angreifer muss nun mehrere Anmeldevorgänge beobachten, damit er weiß, welches der Bilder zum Passwort gehört.

(a) PassPoint [WWB+05] (b) Ereiterung von PassPoint [Suo06] (c) Ereiterung von PassPoint [Suo06]

Abbildung 2.5: PassPoint. In (a) ist ein vom System vorgegebenes Bild zusehen, sowie die vom Benutzer gewählten Klickpunkte (schwarze Vierecke im Bild). In (b) ist die Erweiterung von Suo mit einem falschen Bildausschnitt, d. h. der Benutzer hat in diesem Bildausschnitt keinen Punkt gesetzt. In (c) dagegen ist ein richtiger Bildausschnitt. Hier hatte der Benutzer ein Punkt gesetzt, wie in (a) zu sehen.

Cued Click-Points (CCP), von Chiasson et al. [COB07] entwickelt, stellt dem Benutzer fünf Bilder für das Passwort zur Verfügung. Der Benutzer wählt in jedem dieser Bilder einen Klickpunkt. Während der Anmeldung wird ein indirektes Feedback gegeben, ob das Passwort korrekt ist. Nach jedem richtig gesetzten Klickpunkt wird das nächste Bild, mit einem vom Benutzer vorhandenen Klickpunkt, angezeigt. Wenn der Benutzer einen Klickpunkt falsch setzt, wird als nächstes ein Bild angezeigt, bei dem der Benutzer keinen Klickpunkt hat. Eine Studie [COB07] ergab eine Quote von 96% erfolgreicher Anmeldungen beim ersten Versuch. Ebenfalls ergab diese Studie, dass Benutzer 50% ihrer Klickpunkte auf so genannte *Hotspots* [COB07] gesetzt haben. *Hotspots* sind Felder in Bildern, die von Benutzern sehr häufig gewählt werden, da diese beim ersten Hinschauen direkt auffallen (beispielsweise in Abbildung 2.5 (a), der Baum mitten in einer Menge von Personen). Angreifer können diese Punkte ausnutzen, indem sie eine Liste möglicher Passwörter bzw. ein sogenanntes *dictionary* erstellen. In diesem *dictionary* sind alle *Hotspots* enthalten und der Angreifer kann nun einen *dictionary* Angriff (siehe Kapitel 3) durchführen.

Persuasive Cued Click-Points (PCCP) ist eine Erweiterung von CCP, die Benutzer dazu bringen soll ihre Punkte sicherer und zufälliger zu wählen [CFBO08]. Dies wird erreicht, indem dem Benutzer in jedem der fünf Bilder ein zufälliger Bildausschnitt zur Verfügung gestellt wird (siehe Abbildung 2.6). In diesem Bildausschnitt kann der Benutzer nun seinen Punkt setzen. Er kann sich aber auch einen Neuen, vom System zufällig ausgewählten Ausschnitt anzeigen lassen, falls er auf dem aktuellen keinen Punkt setzen möchte. Während der Anmeldung zum System werden dann wieder die kompletten Bilder, ohne eine Markierung des Bildausschnitts, angezeigt und der Benutzer setzt seine Punkte wieder genauso wie bei CCP. Die Studie von Chiasson et al. [CFBO08]

ergab ein Quote von 91% erfolgreichen Anmeldungen. Die Erfolgsquote ist also ähnlich zu CCP und die Benutzer haben ihre Punkte nicht mehr so häufig auf die bereits erwähnten *Hotspots* gesetzt.

Inkblot ist ein textuelles Passwort, welches als Hinweis Bilder (*Inkblots*, siehe Abbildung 2.7) verwendet [SS04]. Zum Erstellen seines Passworts muss der Benutzer zu allen zehn zur Verfügung gestellten Bildern den ersten und letzten Buchstaben des zum Bild assoziierten Wortes eingeben. Daraus entsteht eine Passwortlänge von 20 Buchstaben. Bei der Anmeldung zum System werden dann die Bilder in einer anderen Reihenfolge angezeigt und der Benutzer gibt jeweils die zwei Buchstaben jedes Bildes ein. Die Reihenfolge ist bei jeder Systemanmeldung die gleiche und unterscheidet sich lediglich von der Reihenfolge während der Registrierung. Die Entwickler des Systems zielen darauf hinaus, dass die Bilder den Benutzer unterstützen, sich das Passwort zu merken. Nach einer Weile sollen diese aber nicht mehr benötigt, sondern das Passwort auswendig eingetippt werden.

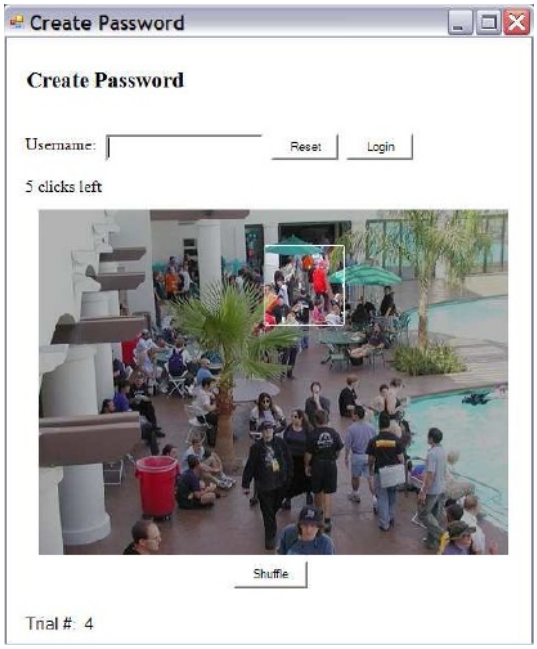


Abbildung 2.6: PCCP [CFBO08]

Abbildung 2.7: Inkblot [CFBO08]

Picture password soll das neue Anmeldesystem für Windows 8 werden [Sin]. Ein Benutzer wählt aus seinen Fotos ein Bild aus, welches er gerne für das Passwort verwenden möchte. Auf diesem Bild werden nun drei Gesten eingezeichnet, nämlich ein Punkt, ein Kreis und eine Linie (siehe Abbildung 2.8). Beim Anmelden am System muss der Benutzer seine Gesten wieder in der gleichen Reihenfolge zeichnen. Des Weiteren muss er darauf achten, dass er den Kreis und die Linie in die gleiche Richtung setzt (wie in Abbildung 2.8, eine Linie von der Nase der mittleren Person zu der Nase der linken Person und nicht umgekehrt). Diese Eigenschaft soll es einem Angreifer schwerer machen, ein Passwort zu erraten. Auch hier können Benutzer ihre Gesten auf *Hotspots* setzen. Nach [Sin] ist die Anzahl möglicher Kombinationen für solche Felder $(m * (1 + 2 * 5 + (m - 1)))^n$, wobei m die Anzahl der *Hotspots* und n die Anzahl der Gesten ist. Drei Gesten und beispielsweise zehn *Hotspots* ergeben acht Millionen mögliche Kombinationen. Wenn ein Benutzer ein Foto wählt auf dem es nur wenige *Hotspots* gibt, beispielseise drei, dann gibt es immer noch 60 Tausend mögliche Kombinationen.

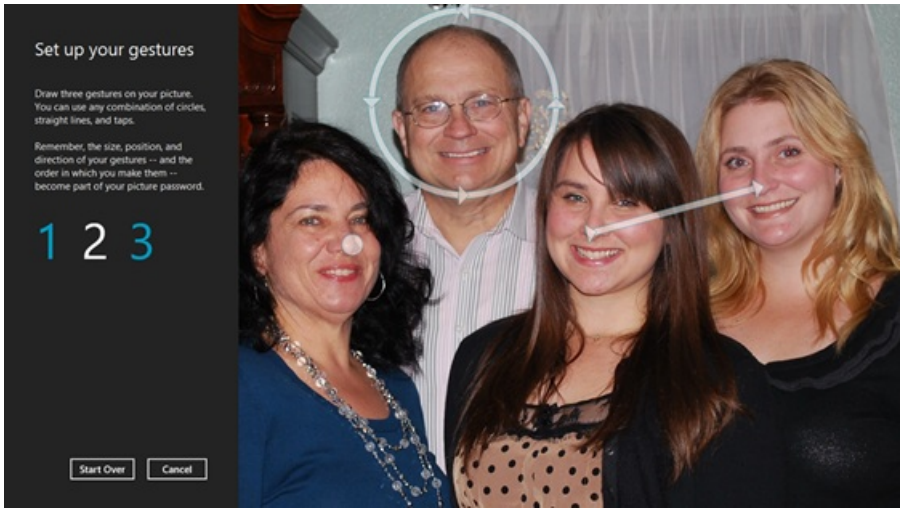


Abbildung 2.8: Picture password [Sin].

2.3 Recognition-based systems

Recognition-based systems bauen auf dem Erinnerungsvermögen des Menschen auf. Selbst kurz gezeigte Bilder kann der Mensch sehr gut wiedererkennen [SCH70]. Bei *Recognition-based systems* muss sich der Benutzer bei den meisten Systemen während der Registrierung einige Elemente oder Bilder merken und diese dann bei der Anmeldung aus einer Vielzahl von Elementen oder Bildern wiedererkennen und auswählen. Diese Systeme sind gut gegen *Phishing attacks* (siehe Kapitel 3) geeignet, da dem Benutzer zuerst die Bilder angezeigt werden müssen, bevor er sein Password eingeben kann.

Im **Passfaces** Testsystem [Cor] sollten sich Benutzer vier Gesichter aussuchen und merken. Während der Anmeldung am System werden dem Benutzer neun Bilder angezeigt, von denen eins zum Passwort gehört (siehe Abbildung 2.9). Diese Prozedur wird vier mal wiederholt, damit alle vier Gesichter vom Benutzer wieder erkannt werden können. Um sich erfolgreich anzumelden, müssen alle vier Gesichter erkannt werden. Ein Angreifer braucht also bei vier Runden von je neun Bildern zum Erraten des Passworts max. 9^4 = 6561 Kombinationen durchzugehen. Eine Benutzerstudie von Valentine [Val98] ergab eine erfolgreiche Anmeldequote von 72% bis 100% im dritten Anmeldeversuch in einem Intervall von bis zu fünf Monaten. In einer Benutzerstudie von Davis et al. [DP00] hat sich herausgestellt, dass die Teilnehmer vorhersagbare Gesichter gewählt haben (z. B. Gesichter der gleichen Nationalität). In kommerziellen Anwendungen werden dem Benutzer daher in Passfaces Gesichter während der Registrierung vom System zugeteilt. Diese muss er sich dann während eines Trainings merken.

Story [DMR04] baut auf Passfaces auf. Hier konnten sich Benutzer des Testsystems ebenfalls vier Bilder aussuchen und merken. Während der Anmeldung mussten sie dann diese Bilder aus insgesamt neun Bildern wiedererkennen und in der richtigen Reihenfolge auswählen (siehe Abbildung 2.10). Damit die Benutzer sich die Reihenfolge merken können, wurde ihnen mitgeteilt, dass sie sich eine Geschichte überlegen sollen, welche die einzelnen Bilder verbindet. Eine Sequenz von vier Bildern aus neun Bildern ergibt 9*8*7*6 = 3024 Kombinationen, also noch weniger als bei Passfaces. Eine Benutzerstudie zu Story und Passfaces von Davis et al. [DMR04] ergab, dass Teilnehmer mehr Probleme hatten, sich das Story Passwort zu merken, als das Passfaces Passwort. Die meisten Fehler sind durch das Vertauschen der Reihenfolge der Bilder aufgetreten. Dies könnte daran liegen, dass sich nur 50% der Teilnehmer eine Geschichte zu ihrem Passwort überlegt haben. Insgesamt konnten sich 85% der Teilnehmer erfolgreich anmelden.

Im **Déjà Vu** [DP00] Testsystem suchen sich Benutzer fünf Bilder aus einer Menge von zufällig generierten Kunstbildern aus. Das Passwort besteht aus diesen fünf Bildern. Während der Anmeldung werden dem Benutzer 25 Bilder angezeigt und er muss seine fünf Bilder darunter finden (siehe Abbildung 2.11). Im Gegensatz zu Passfaces gibt es in Déjà Vu nur eine Runde, in der bereits alle Bilder gefunden werden müssen. Wenn ein Angreifer das Passwort erraten will, muss er also bis zu $\binom{25}{5}$ = 53130 Kombinationen durchgehen. Die Entwickler von Déjà Vu haben sich das Ziel gesetzt, mit dem System Benutzer daran zu hindern schwache Passwörter zu wählen, da zufällig generierte Kunstbilder keinen persönlichen Zusammenhang mit dem Benutzer haben. Des Weiteren soll es dem Benutzer nicht möglich bzw. für ihn sehr schwer sein, sein Passwort zu beschreiben und weiter zu geben. Die Benutzerstudie von Dhamija und Perrig [DP00] ergab, dass die Teilnehmer Probleme damit hatten ihre Bilder zu beschreiben, und dass bei gleich gewählten Bildern die Teilnehmer verschiedene Beschreibungen angaben. In der Studie konnten sich 100% der Teilnehmer am System anmelden. Nach einer Woche wurde ein weiterer Test durgeführt. Bei diesem konnten sich noch 90% der Teilnehmer erfolgreich anmelden.

Abbildung 2.9: Passfaces [DMR04]

Abbildung 2.10: Story [DMR04]

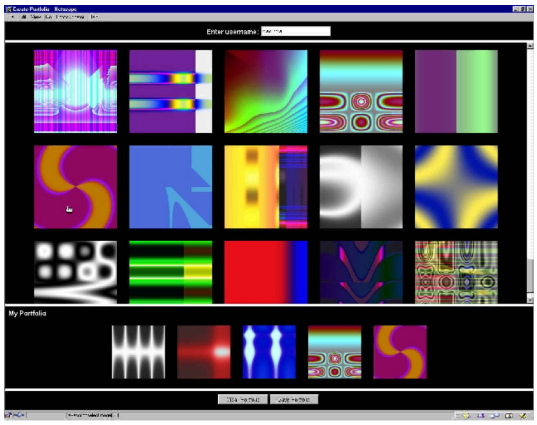


Abbildung 2.11: Déjà Vu [DP00]

Cognitive Authentication von Weinshall [Wei06] entwickelt, sollte ein sicheres grafisches Passwortsystem gegen *Shoulder-surfing* Angriffe (siehe Kapitel 3) werden. Der Unterschied von diesem System zu den bereits vorgestellten *recognition-based* Systemen ist, dass hier die Passworteingabe mittels Tastatur getätigt wird. Zuerst muss sich der Benutzer eine Untermenge von vorhandenen Bildern merken. Während der Anmeldung wird ihm dann ein Feld mit Bildern angezeigt, auf dem er einen Weg berechnen muss (siehe Abbildung 2.12). Der Benutzer startet in der linken oberen Ecke und geht ein Bild weiter runter, wenn das aktuelle Bild in seinem Passwort enthalten ist, oder nach rechts, wenn es nicht enthalten ist. Soabld der Benutzer die rechte oder untere Kante erreich, gibt er die Zahl, die neben dem letzten Bild steht, ein (siehe Abbildung 2.12, hier stehen jeweils am unteren und rechten Rand Zahlen zwischen 0 und 3). Dieses Vorgehen wird mehrmals wiederholt. Nach jeder Runde berechnet das System wie hoch die Wahrscheinlichkeit ist, dass die Eingabe nur geraten wurde. Der Login ist erfolgreich sobald eine, vom System gesetzte Wahrscheinlichkeit erreicht wurde. Sollte nach einer bestimmten Anzahl von Runden diese Wahrscheinlichkeit nicht erreicht werden, weil man beispielsweise irgendeinen Weg falsch berechnet hat, schlägt der Login fehl. Eine Studie [Wei06] ergab eine erfolgreiche Anmeldequote von 95%. In dieser Benutzerstudie haben Teilnehmer bis zu drei Minuten für die Anmeldung gebraucht. Golle und Wagner haben eine Kryptoanalyse zu *Cognitive Authentication* durchgeführt [GW07]. Im Gegensatz zur

Behauptung von Weinshall [Wei06], zeigte die Kryptoanalyse, dass das System nicht sicher gegen *Shoulder-surfing* Angriffe ist [GW07].

Abbildung 2.12: Cognitive Authentication [Wei06]

3 Sicherheit von grafischen Passwörtern

Damit grafische Passwörter verwendet werden können, sollten diese nicht nur benutzerfreundlich und leicht zu merken, sondern auch sicher sein. In diesem Kapitel werden einige Angriffe aufgezählt. Des Weiteren wird hier beschrieben welche Möglichkeiten es gibt, um textuelle und grafische Passwörter gegen diese Angriffe zu sichern.

3.1 Guessing Attacks

Guessing Attacks sind Angriffe, bei denen eine Kombination aus Benutzername und Passwort erraten wird. Hier gibt es zwei Herangehensweisen, die *Brute-Force Attack* und die *Dictionary Attack*.

3.1.1 Brute-Force Attack

Brute-Force Attack ist ein Angriff, bei dem jede mögliche Kombination von Buchstaben, Zahlen und Sonderzeichen durchprobiert wird, bis ein erfolgreiches Anmelden mit dem textuellen Passwort erfolgt ist. Solche Angriffe können aber nicht nur auf textuelle Passwörter angewendet werden, sondern auch auf grafische Passwörter. Hierzu werden alle möglichen Punkte auf Bildern, oder sonstigen Elementen die verwendet werden können, durchprobiert, bis sich der Angreifer erfolgreich anmelden kann.

Mit genug Zeit kann dieser Angriff mit Sicherheit alle Passwörter finden. Bei einem textuellen Passwort mit acht Zeichen welches kleine und große Buchstaben sowie Zahlen enthalten kann, gibt es 70^8 Möglichkeiten wie das Passwort aufgebaut sein kann. Wenn ein Angreifer 100 Millionen Passwortkombinationen pro Sekunde finden könnte, würde dies bei 70^8 möglichen Kombinationen ungefähr 67 Tage dauern, bis alle Möglichkeiten durchprobiert wurden. Wenn nur sechs Zeichen verwendet werden sind das 70^6 mögliche Kombinationen und damit nur noch ungefähr 20 Minuten bis alle Kombinationen durchgeführt wurden. Sollten sogar nur sechs Kleinbuchstaben verwendet werden, gibt es nur 30^6 Kombinationsmöglichkeiten, welche in knapp sieben Sekunden mittels eines *Brute-Force* Angriffs gefunden werden können. Im Schnitt wird ein Passwort nach der Hälfte der Zeit gefunden [BBD05], daher sollten immer gute Passwörter mit mindestens acht Zeichen und sowohl Klein-/Großbuchstaben als auch Zahlen und Sonderzeichen verwendet werden [YBAG04].

Brute-Force Angriffe können auch auf grafische Passwörter angewendet werden, insbesondere auf diejenigen mit einer geringen Kombinationszahl, wie beispielsweise Passfaces und Story. Passfaces umfasst neun Bilder und vier Runden, das macht nur 9^4 = 6561 Kombinationen. Story umfasst neun Bilder, von denen vier in der richtigen Reihenfolge angeklickt werden müssen. Das ergibt 9*8*7*6 = 3024 Kombinationen in Story, was noch weniger als in Passfaces ist. Bei solchen Systemen muss ein Versuchslimit für die Onlineanmeldung gesetzt werden, sodass Angreifer nicht die Möglichkeit haben mehrere Kombination auszuprobieren, sondern beispielsweise nur drei Versuche haben das richtige Passwort zu wählen. Wenn ein Passwort drei mal falsch eingetippt wurde, müsste das Passwort vom Benutzer zurückgesetzt werden. Dies kann z. B. über eine bei der Registrierung hinterlegte E-Mail-Adresse geschehen.

3.1.2 Dictionary Attack

Bei *Dictionary Attacks*, im Gegensatz zu *Brute-Force Attacks*, werden nicht alle Kombinationen durchprobiert, sondern nur diejenigen die am wahrscheinlichsten sind. Viele Personen wählen

schwache Passwörter, wie normale Wörter die tagtäglich benutzt werden, ihr Geburtsdatum, dem Namen eines Familienmitglieds oder eine Kombination aus einem Namen und aktuellem Jahr bzw. Geburtsjahr [Elf06, YBAG04]. Bei solchen Passwörtern ist es sehr leicht für den Angreifer, mittels einer *Dictionary Attack*, ein Passwort zu erraten. Ein Angreifer kann ein Wörterbuch der Landessprache des Benutzters verwenden, um die enthaltenen Wörter als Passwort durchzuprobieren. Es ist auch möglich eine Passwortliste bzw. ein Wörterbuch selbst zu erstellen, in das beispielweise Kombinationen von Namen und Jahreszahlen eingetragen werden.

Auch bei grafischen Passwörtern ist es möglich diesen Angriff sinnvoll zu nutzen. Es kann eine Liste mit wahrscheinlichen Passwörtern, aus gesammelten Erfahrungen oder durch Annahmen über das Benutzerverhalten, zusammengestellt werden [BCO11].

In vielen grafischen Passwortsystemen wurden nicht sichere Passwörter gewählt. Eine Studie von Davis et al. [DMR04] zeigte, dass viele Benutzer bei Passfaces Gesichter gewählt haben, die ähnlich der eigenen Herkunft waren. Bei CCP wurden von vielen Benutzern *Hotspots* [COB07] im Bild ausgewählt und in DAS wurden die Zeichnungen meistens mittig im Feld angeordnet [NT04]. Diese Eigenschaften erleichtern es einem Angreifer das Passwort zu finden. Die Wahl eines Passworts sollte nicht leichtsinnig durchgeführt werden. Es sollte darauf geachtet werden, dass das Passwort keinen Bezug zu sich selbst hat, Klickpunkte nicht auf typischen Positionen im Bild gesetzt werden, sowie Zeichnungen keine Symmetrien haben und nicht mittig auf einem Gitter gezeichnet werden.

3.2 Capture Attacks

Capture Attacks zielen darauf hinaus das Passwort direkt zu erhalten, anstatt wie bei *Guessing Attacks* das Passwort zu erraten. Diese Angriffe können auf verschiedenste Weise durchgeführt werden, z. B. über *Shoulder-surfing*, *Phishing*, oder *Malware*.

3.2.1 Shoulder-surfing

Bei einem *Shoulder-surfing* Angriff beobachtet ein Angreifer die Anmeldung eines Benutzers, entweder durch direktes Hinschauen oder über die Aufnahme mittels einer Kamera.

Shoulder-surfing Angriffe sind die große Schwäche der grafischen Passwörter. Textuelle Passwörter sind gegen diese Angriffe sicherer, da es viel schwieriger ist, Tastatureingaben zu erkennen, als gesetzte Klickpunkte auf einem Monitor [TOH06].

Manche grafische Passwörter verwenden nur einen Teil des Passworts (z. B. Cognitive Authentication [Wei06]) oder zeigen das Passwort bei der Anmeldung nicht an (z. B. die Erweiterung von PassPoints [Suo06]. Bei solchen Passwortsystemen müssen mehrere Anmeldungen beobachtet werden, damit der Angreifer das Passwort erhalten kann. Diese Eigenschaft macht das System zumindest etwas sicherer.

3.2.2 Malware

Malware ist eine bösartige Software, die auf dem Rechner der Benutzers, ohne sein Wissen, runtergeladen und installiert wird. In solch einer *Malware* können *Keystroke-loggers* (Aufnahme von Tastatureingaben), *Mouse-loggers* (Aufnahme von Mausaktivitäten) und *Screen scrapers* (Aufnahme des Bildes auf dem Monitor) enthalten sein [BCO11].

Für das Passwortsystem Inkblot ist beispielsweise nur ein *Keystroke-logger* notwendig, da hier die Passworteingabe komplett mittels Tastatur gemacht wird und die Bilder lediglich als Gedächtnis-

stütze dienen. Für textuelle Passwörter wird ebenfalls nur ein *Keystroke-logger* gebraucht. Bei den meisten grafischen Passwörtern werden ein *Mouse-logger* und ein *Screen scraper* benötigt um das Passwort zu erhalten. Damit auch der Benutzername in Erfahrung gebracht werden kann, wird hier auch noch ein *Keystroke-logger* gebraucht. Für den Erhalt von grafischen Passwörtern wird also fortgeschrittenere *Malware* benötigt als für textuelle Passwörter.

3.2.3 Phishing

Phishing Angriffe sind eine Form von *Social engineering* Angriffen [Wor07]. Bei diesen Angriffen wird der Benutzer auf einer betrügerischen Webseite aufgefordert sein Passwort einzugeben. *Social engineering* Angriffe gehen noch etwas weiter und können, nicht nur über einer Webseite, sonden auf jegliche Art von Kommunikation (z. B. über Telefon oder auch über ein persönliches Gespräch). Hierbei wird versucht den Benutzer dazu zubringen sein Passwort zu verraten.

Phishing Angriffe auf textuelle und auf *Recall-based* Systeme sind direkt ausführbar. Bei *Cued-recall* und *Recognition-based* Systemen muss der Angreifer zuerst die jeweiligen Bilder erlangen, bevor er die Benutzer zur Passworteingabe auffordern kann. Dementsprechend ist zumindest ein Teil der grafischen Passwörter vor *Phishing* Angriffen sicherer als textuelle Passwörter.

Das neu entwickelte grafische Passwortsystem bedient sich der Google Maps Karte. Hier kann sich der Benutzer drei Punkte auf der Karte auswählen. Dabei ist es egal, ob er seine Punkte auf eine Stadt, einen Stadtteil oder auf eine Staße setzt. Der Benutzer darf also soweit auf der Karte einzoomen wie er möchte. Das Passwort besteht aus den Koordinaten der gesetzten Punkte. Als Koordinaten werden die Werte der geografischen Breite (Lat) und der geografischen Länge (Lng) gespeichert. Darmstadt Stadtmitte hat beispielsweise die Koordinaten Lat: 49.87286251746749 und Lng: 8.651706576347351. Ein Beispiel zur Passwortwahl ist in der Abbildung 4.1 zu sehen. In dieser Abbildung hat der Benutzer die Punkte Staatstheater Darmstadt, die Station Willy-Brandt-Platz und das Robert-Piloty-Gebäude gewählt.

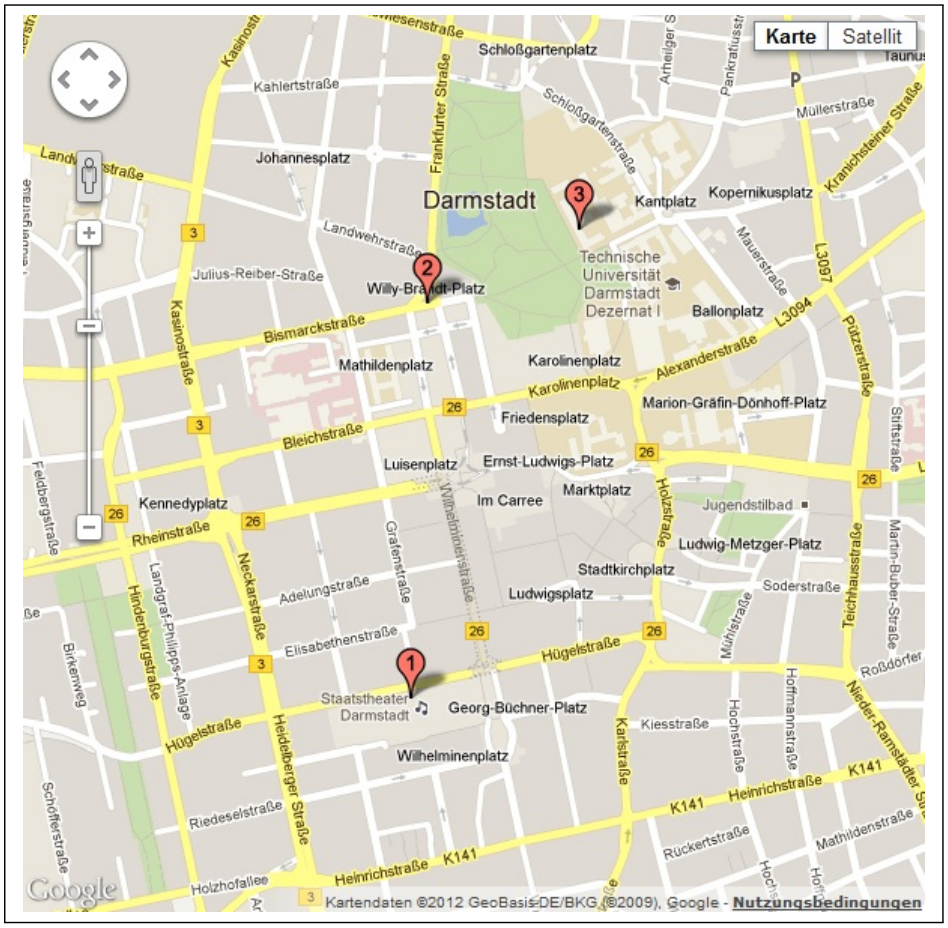


Abbildung 4.1: Ein grafisches Passwortsystem basierend auf Google Maps

Damit sich der Benutzer, nach einer abgeschlossenen Registrierung am System anmelden kann, muss er wieder das Staatstheater Darmstadt, die Station Willy-Brandt-Platz und das Robert-Piloty-Gebäude anklicken. Hierbei ist es wichtig, dass der Benutzer die Punkte in der gleichen Reihenfolge und auf der gleichen bzw. ähnlichen Zoomstufe wie während der Registrierung anklickt. Die Klickpunkte selbst müssen, genauso wie der Zoom, ebenfalls nicht exakt gesetzt werden. Der Benutzer hat einen Radius um den bei der Registrierung gesetzten Punkt. Wenn während der Anmeldung der Punkt in diesen Radius gesetzt wird, kann der Login erfolgreich abgeschlossen werden. Eine detaillierte Erläuterung der Funktionsweise der Registrierung und des Logins ist im Kapitel 5 aufgeführt.

4.1 Verwendungszweck des grafischen Passwortsystems

Mit dem auf Google Maps basierendem grafischen Passwortsystem ist die Erstellung eines neuartigen Passworts möglich. Dieses Passwort ist in die Kategorie *cued-recall systems* einzuordnen, da ein Benutzer die Karte als Hinweis zur Eingabe seines Passworts erhält. Ein Benutzer kann sich das Google Maps Passwort sehr gut merken, da er ein Passwort wählen kann, mit dem er etwas assoziiert. Eine durchgeführte Evaluation egab, dass 92% der Benutzer das Google Maps Passwort als leicht zu merken eingestuft haben (siehe Kapitel 6). Des Weiteren ist das System benutzerfreundlich. Benutzer können entweder auf der Karte nach Orten suchen, oder in einem Feld zur Ortssuche ihren gewünschten Ort eingeben, welcher dann von Google Maps für sie selbstständig gesucht wird. Ein weiterer positiver Aspekt des Systems ist seine Sicherheit. Das grafische Passwortsystem hat eine hohe Anzahl an möglichen Kombinationen, die vom Benutzer gewählt werden können. Selbst dem Benutzer vertraute Personen, müssten erst genau die drei Orte/ Stadtteile/ Straßen/ Gebäude etc. finden, die der Benutzer sich ausgesucht hat und zusätzlich noch die Zoomstufe für alle drei Punkte ungefähr treffen. Hinzu kommt noch, die Punkte in der richtigen Reihenfolge und in genau dem vom System festgesetzten Radius zu setzen. Das sind so viele Kombinationsmöglichkeiten, dass selbst Personen die den Benutzer kennen, nahezu keine Chance haben das Passwort rauszufinden. In Abschnitt 5.5 wird gezeigt, dass das System selbst gegen *Shoulder-surfing* Angriffe sicher ist, wenn der Benutzer ein schwieriges Passwort wählt.

4.2 Verwendungszweck von Google Maps

Google Maps [MAPb] wurde am 8. Februar 2005 von Google Inc. veröffentlicht und ermöglicht Orte, Gebäude, Straßen, etc. auf einer Landkarte oder auf einem Luftbild anzuzeigen. Der Kartendienst stellt dem Benutzer viele hochauflösende Bilder zur Verfügung.

Google Maps stellt eine Vielzahl von APIs (application programming interface), zur Einbettung der Google Maps Funktionalitäten auf eine eigene Homepage oder Anwendung, zur Verfügung. Hierzu gehören beispielsweise das Google Maps JavaScript-API, welches für das hier vorgestellte Passwortsystem verwendet wurde, sowie ein Google Maps-API für Flash oder ein Google Earth-API [GMA]. Im Jahr 2010 wurde die dritte Version des JavaScript APIs veröffentlicht. Diese Version ist auf eine höhere Geschwindigkeit auf mobilen Geräten als auch auf traditionellen Browser-Anwendungen für den Desktop spezialisiert. Das API bietet eine Menge Funktionalitäten mit denen Karten bearbeitet und Inhalte hinzugefügt werden können, die Kartenanwendung jedoch stabil weiterlaufen lassen. Ein weiterer Vorteil des JavaScript Maps APIs V3 ist sein kostenloser Service für Kleinnutzer. Das JavaScript Maps API V3 ist verfügbar für alle Webseiten und Anwendungen, die für deren Benutzer kostenlos sind und eine bestimmte Anzahl an Aufrufen nicht überschreiten [GMD].

Weitere ähnliche Kartendienste sind:

- MapPoint [MMP]
- Bing Maps [MAPa]
- Yahoo Maps [MAPd]
- Nokia Maps [MAPc]
- WikiMapia [WMP]
- OpenStreetMap [OSM]

Einige der Kartendienste konnten für dieses System direkt ausgeschlossen werden wie beispielsweise MapPoint, da dieser Dienst kostenpflichtig ist. Zwar wird Google Maps teilweise ebenfalls

kostenpflichtig, allerdings erst ab einer bestimmten Anzahl an Aufrufen pro Tag. Dieses Limit wird im Rahmen der Bachelor Arbeit nicht überschritten. Ebenfalls konnte Yahoo Maps ausgeschlossen werden, da das zugehörige API seit September 2011 nicht mehr aktualisiert wird [YMA]. WikiMapia ist eine Kombination aus Google Maps und Wikipedia [WMB]. Hier werden zu den einzelnen Städten, Stadtteilen, Gebäuden, usw. Informationen angezeigt, wie z. B. Fotos der Gegend, Links zu offiziellen Webseiten, oder auch Links zu Wikipedia Artikeln über die Ortschaft. Für das vorgestellte grafische Passwortsystem ist WikiMapia allerdings zu unübersichtlich, da dem Benutzer eine Vielzahl an Informationen zur Verfügung gestellt wird. Die restlichen Kartendienste sind von den Funktionalitäten her relativ ähnlich. Ein Nachteil von Bing Maps ist, dass sich der Benutzer einen neuen Entwickleraccount anlegen muss [BMD], bevor die Karte eingebunden werden kann. In Open Streetmap gibt es lediglich ein Karten Modus, Satelliten Bilder werden hier nicht zur Verfügung gestellt. In Nokia Maps gibt es zwar auch Satelliten Bilder, allerdings lässt die Qualität dieser Bilder noch zu Wünschen übrig. Daher ist aktuell für das grafische Passwortsystem Google Maps die beste Wahl.

Das Grundelement des grafischen Passwortsystems ist die Google Maps API Version 3, denn ohne die API könnte die Karte gar nicht erst angezeigt und die Punkte nicht gesetzt werden. Des Weiteren wird das Framework JQuery [jQu] verwendet, um die Werte, die Google Maps via JavaScript zurückgibt, im Hintergrund an eine PHP Seite weiterzugeben, damit diese dann in einer MySQL Datenbank, verwaltet mit der freien Software phpMyAdmin [php], gespeichert werden können. Das komplette System wurde mit den Programmiersprachen JavaScript, PHP und HTML aufgebaut, wobei HTML nur für die Gestaltung der Webseite verwendet wurde (siehe Abbildung 5.1).

Abbildung 5.1: Implementierung. Die Webseite wurde mit PHP und HTML geschrieben. Daten werden aus der Datenbank [DB] mittels PHP ausgelesen und auf die Webseite geschrieben. Die Google Maps Karte wird über JavaScript auf die Webseite eingebunden und die Informationen über gesetzte Klickpunkte werden mittels jQuery an eine PHP Seite weitergegeben, in welcher die Informationen dann in die Datenbank reingeschrieben werden.

5.1 Datenbankstruktur

Für die Registrierung zum Passwortsystem werden zwei Datenbanktabellen gebraucht. Die erste enthält die Benutzerdaten (Benutzerkennung, Benutzername und E-Mail-Adresse) und die zweite enthält die Klickpunkte. Bei den Klickpunkten werden nicht nur die Benutzerkennung und die Koordinaten, sondern auch die Reihenfolge der Klickpunkte, sowie die Zoomstufe während des Klicks gespeichert.

Für den Login existiert ebenfalls eine Tabelle für Klickpunkte, in der die eingehenden Klickpunkte für die Dauer des Logins gespeichert werden und nach dem Einloggen wieder gelöscht werden. In dieser Tabelle werden ebenfalls wieder eine Benutzerkennung, die Koordinaten, die Reihenfolge der Klickpunkte, sowie die Zoomstufe jedes Klicks gespeichert. Wofür beim Login eine extra Tabelle nötig ist, wird im Abschnitt 5.3 erläutert.

5.2 Funktionsweise der Registrierung

Bei der Registrierung werden der Benutzername, die Benutzerkennung, die geografischen Koordinaten, die Reihenfolge der Klickpunkte und die Zoomstufe während des Klicks gespeichert. Zuerst wird die Angabe eines Benutzernamens gefordert. Hierbei wird in der Datenbank geprüft, ob der Benutzername noch frei ist. Wenn dies der Fall ist, wird der Benutzername gespeichert und der Benutzer darf fortfahren, indem er Klickpunkte auf der Karte setzt. Die restlichen Daten werden dann durch ein Mausklick Ereignis, nämlich wenn ein Benutzer einen Klickpunkt setzt, entnommen (siehe Codebeispiel 5.1).

```
function addLatLng(event) {
        //Zählvariable für die Reihenfolge (order) der Klickpunkte erhöhen
        i += 1;

        //Werte setzen
        var lat = event.latLng.lat();
        var lng = event.latLng.lng();
        var zoom = map.getZoom();
        var order = i;
}
```

Codebeispiel 5.1: Mausklick Ereignis

Mittels JQuery/AJAX [AJA] wird im Hintergrund eine PHP Seite aufgerufen und die Daten des Ereignisses übergeben. Auf der PHP Seite wird überprüft, ob zu viele Klickpunkte gesetzt wurden. Wenn dies der Fall ist wird dem Benutzer eine Fehlermeldung angezeigt, dass er bereits alle Punkte gesetzt hat und sein neu gesetzter Punkt nicht gespeichert werden kann. Des Weiteren wird auf dieser PHP Seite geprüft, ob die einzelnen Punkte weit genug von einander entfernt sind. Das ist eine Eigenschaft, die den Benutzer dazu bringen soll sein Passwort sicherer zu wählen und nicht beispielsweise dreimal Darmstadt auszuwählen. Dieser Mindestabstand zwischen den einzelnen Punkten ist je nach aktueller Zoomstufe größer oder kleiner. Bei einer Anzeige der Deutschland Karte (Zoomstufe acht) ist der Mindestabstand so gewählt, dass zwei nahe aneinander liegende Städte ausgewählt werden können, aber nicht zwei mal die gleiche Stadt bzw. ganz nah an der Stadt geklickt werden darf (siehe Abbildung 5.2 (a)). Analog für höhere Zoomstufen ist der Mindestabstand so gewählt, dass nahe aneinander liegende Stadtteile (Zoomstufe neun bis zwölf, siehe Abbildung 5.2 (b)) sowie bei noch höherer Zoomstufe auch nahe aneinander liegende

Straßen (ab Zoomstufe 15) noch beide gewählt werden können. In Tabelle 5.1 ist ein Überblick über alle Mindestabstände in Koordinaten zu jeder akzeptierten Zoomstufe (acht bis 21) abgebildet.

(a) Mindestabstand auf Zoomstufe 8 (b) Mindestabstand auf Zoomstufe 11

Abbildung 5.2: Mindestabstand. In (a) ist ein Klickpunkt auf Darmstadt auf der Zoomstufe acht gesetzt. Der Kreis um den Punkt herum ist der Mindestabstand der eingehalten werden muss, um einen weiteren Klickpunkt setzen zu können. In (b) ist der Klickpunkt auf der Zoomstufe elf gesetzt worden. Auch hier zeigt der Kreis um den Punkt herum, wie groß der Mindestabstand sein muss.

Tabelle 5.1: Der Mindestabstand entspricht einem Kreis um einen gesetzten Klickpunkt. Als Beispiel werden die Koordinaten von Darmstadt genommen. Diese sind Lat: 49.87286251746749 und Lng: 8.651706576347351. Der Radius des Kreises um den Punkt mit den Koordinaten von Darmstadt ist der angegebene Mindestabstand in der Tabelle. D. h. wenn der Punkt auf der Zoomstufe acht gesetzt wurde, wird um die o.g. Koordinaten der Kreis mit dem Radius 0.03 gesetzt (in Abbildung 5.2 (a) veranschaulicht). In diesem Kreis kann kein Klickpunkt mehr gesetzt werden.

Zoom bei Registrierung	Mindestabstand
8	0.03
9	0.025
10	0.015
11	0.01
12	0.005
13	0.0025
14	0.001
15	0.00025
16	0.0002
17	0.0002
18	0.00015
19	0.00007
20	0.00003
21	0.00002

Sollte der Benutzer sein Passwort nicht wie gewünscht gesetzt haben, kann er während der Registrierung seine gewählten Punkte zurücksetzen und die Punkte neu wählen. Wenn er aus Versehen oder absichtlich die Seite neu geladen hat, hat er wieder die Möglichkeit seine Klickpunkte neu zu wählen. Dies entspricht einem Zurücksetzen der Klickpunkte. Für diesen Fall wird in der PHP Seite im Hintergrund geprüft, ob für diesen User und diese Reihenfolge bereits ein Eintrag existiert. Wenn dies der Fall ist, wird der alte Eintrag gelöscht und der neue Eintrag gespeichert. Sollte der Benutzer nun mit seiner Eingabe fertig sein, kann er diese durch einen Klick auf den bestätigen-Button absenden. Hier wird zuerst überprüft, ob der Benutzer die nötige Anzahl an Klickpunkten gesetzt hat. Hat er noch nicht die nötige Anzahl erreicht, erhält er wieder eine Fehlermeldung und muss die restlichen Klickpunkte noch setzen. Wurden die Klickpunkte alle gesetzt, so wird er auf eine Bestätigungsseite weitergeleitet. Hier werden nochmal alle gesetzten Klickpunkte und ihre Reihenfolge angezeigt, um dem Benutzer eine letzte Möglichkeit zu geben, sich diese zu merken.

5.3 Funktionsweise des Logins

Beim Login muss zuerst der Benutzername eingegeben werden. Wenn dieser in der Datenbank vorhanden ist, wird der Benutzer zum nächsten Schritt weitergeleitet. Nun kann dieser seine Klickpunkte setzen. Für eine erfolgreiche Anmeldung, müssen die Klickpunkte wieder in der richtigen Reihenfolge angeklickt werden. Des Weiteren darf man während des Logins nur um maximal zwei Zoomstufen (rein oder raus) von der Registrierung abweichen, um sich noch erfolgreich anmelden zu können. Diese Eigenschaft wurde eingeführt, da es nahezu unmöglich ist einen Punkt der beispielsweise auf eine Straße gesetzt wurde, während des Logins durch anklicken einer Stadt oder eines Stadtteils zu treffen. Es existiert eine gewisse Toleranz, wie genau ein Punkt gesetzt werden muss, abhängig davon auf welcher Zoomstufe der Klickpunkt gesetzt wurde. Ist die Toleranz jedoch so großzügig gesetzt, dass ein Benutzer sich durch Klicken auf Darmstadt anmelden kann, obwohl er bei der Registrierung beispielsweise das Robert-Piloty Gebäude gewählt hatte, kann sich genauso gut auch jeder Angreifer auf diese Weise anmelden. Daher ist die Tatsache, dass die Zoomstufe sich nur um +/- zwei unterscheiden kann, eine Sicherheitseigenschaft gegen Angreifer. Diese müssen den genauen Ort des Klickpunktes erst erfahren, bevor sie sich mit dem Passwort eines anderen Benutzers anmelden können.

Als Darstellungsbeispiel wird angenommen, dass eine Person seine Klickpunkte auf der Zoomstufe 17 gemacht hat. Diese Person möchte sich nun wieder auf der gleichen Zoomstufe anmelden, dann kann sie sich um den Wert 0.00007 „verklicken“ (siehe Abbildung 5.3 (a)). Wenn diese Person sich aber nun auf der Zoomstufe 16 oder 18 anmelden möchte, dann kann sie sich um den Wert 0.0001 verschätzen, da es schwieriger ist den Punkt wieder zu treffen wenn man auf einer Zoomstufe höher/niedriger ist. Sollte sich die Person auf Zoomstufe 15 oder 19 anmelden wollen, kann sie sich sogar um den Wert 0.00016 verschätzen (siehe Abbildung 5.3 (b)). Je weiter eine Person von ihrem tatsächlichen Punkt rein oder raus zoomt, desto schwieriger wird es diesen Punkt exakt anzuklicken.

In Tabelle 5.2 ist ein Überblick über alle Toleranzen zu jeder akzeptierten Zoomstufe abgebildet. Sowohl Zoomstufen niedriger als acht, als auch Zoomdifferenzen größer als zwei werden im Passwortsystem nicht akzeptiert und sind demnach hier auch nicht angezeigt. Auf Zoomstufen niedriger als acht können größtenteils nur noch Länder oder Kontinente ausgewählt werden. Wenn ein Passwort nur aus der Wahl von Ländern und Kontinenten besteht, kann ein Angreifer mit wenigen Tests das Passwort des Benutzers erraten. Dies soll mit dem Ausschluss von Zoomstufen niedriger als acht verhindert werden. Die Werte in der Tabelle wurden mittels mehrfachen Registrierungen und Loginversuchen erstellt und durch Testen an weiteren Personen angepasst.

(a) Toleranz beim Login auf Zoomstufe 17 (b) Toleranz beim Login auf Zoomstufe 15

Abbildung 5.3: Toleranz beim Login. In (a) ist ein Klickpunkt auf dem Robert-Piloty Gebäude auf der Zoomstufe 17 gesetzt. Der Kreis um den Punkt herum ist die Toleranz, in den ein Klickpunkt für eine erfolgreiche Anmeldung gesetzt werden muss. In (b) ist der Klickpunkt während der Registrierung ebenfalls auf der Zoomstufe 17 gesetzt worden. Allerdings will der Benutzer sich auf diesem Bild auf der Zoomstufe 15 anmelden.

Tabelle 5.2: Die Toleranz entspricht einem Kreis um einen bei der Registrierung gesetzten Klickpunkt. Als Beispiel werden hier die Koordinaten vom Robert-Piloty-Gebäude (Lat: 49.87747142768588 und Lng: 8.654526261932347) genommen. Der Radius des Kreises um den gesetzten Punkt, ist die angegebene Toleranz in der Tabelle. D. h. wenn der Punkt während der Registrierung auf der Zoomstufe 17 gesetzt wurde und der Benutzer bei der Anmeldung den Punkt wieder auf der Zoomstufe 17 setzen will, wird um die o.g. Koordinaten der Kreis mit dem Radius 0.00007 gesetzt (siehe Abbildung 5.2 (a)). In diesem Kreis kann der Klickpunkt gesetzt und die Anmeldung erfolgreich abgeschlossen werden.

Zoom bei Registrierung	Toleranz bei gleichem Zoom	Toleranz bei Zoom +/- 1	Toleranz bei Zoom +/- 2
8	0.02	0.03[1]	0.04[1]
9	0.01	0.02	0.02[2]
10	0.0075	0.0125	0.0125
11	0.005	0.001	0.003
12	0.0025	0.0005	0.0025
13	0.0012	0.0008	0.0018
14	0.0005	0.0005	0.0015
15	0.00025	0.00025	0.00075
16	0.0001	0.0001	0.00035
17	0.00007	0.0001	0.00016
18	0.00005	0.00005	0.0001
19	0.00002	0.00003	0.00008
20	0.00001	0.00001	0.00005[3]
21	0.000005	0.000005[4]	0.000015[4]

[1] Die Toleranzwerte gelten nur für Zoom +1 bzw. Zoom +2.
[2] Der Toleranzwert gilt nur für Zoom +2.
[3] Der Toleranzwert gilt nur für Zoom -2.
[4] Die Toleranzwerte gelten nur für Zoom -1 bzw. Zoom -2.

Während des Logins hat ein Benutzer die Möglichkeit sein Passwort zurücksetzen zu lassen. Hierfür wird ein acht-Zeichen langes Einmalpasswort aus einem Pool von Zahlen und Groß-/Kleinbuchstaben zufällig zusammengestellt (siehe Codebeispiel 5.2). Dieses Einmalpasswort wird in einer Datenbanktabelle gemeinsam mit der Benutzerkennung gespeichert. Der Benutzer kriegt nun eine E-Mail, auf die von ihm bei der Registrierung angegebene E-Mail-Adresse, mit dem Einmalpasswort und einem Link zur Webseite. Auf der Webseite kann er sich nun mittels seines Benutzernamens und dem generierten Einmalpasswort am System anmelden und sich ein neues grafisches Passwort erstellen.

```
function gen_pw(){
        $pool = "qwertzupasdfghkyxcvbnm";
        $pool .= "23456789";
        $pool .= "WERTZUPLKJHGFDSAYXCVBNM";

        srand ((double)microtime()*1000000);

        //passwortlänge = 8
        for($index = 0; $index < 8; $index++){
                $pass_word .= substr($pool,
                                (rand()%(strlen ($pool))), 1);
        }
        return $pass_word;
}
```

Codebeispiel 5.2: Einmalpasswortgenerierung

Im Pool aller möglichen Buchstaben und Zahlen wurden absichtlich die Buchstaben o, O und Q sowie die Zahl 0 ausgelassen, da diese eine gewisse Ähnlichkeit haben und somit in der Praxis den Benutzer verwirren könnten [Sta]. Aus dem selben Grund wurden auch die Buchstaben I, i, j und l sowie die Zahl 1 übersprungen genommen.

5.4 Geolocation

Geolocation ist ein Dienst, der den Standort von Personen bestimmt. Dieser Dienst wird von einigen Anwendungen genutzt, um einem Benutzer direkt passende Informationen zu geben, wie beispielsweise Anzeige aller Pizzerien die in der eigenen Umgebung mit einem Radius von 25 km sind. Bevor Geolocation durchgeführt wird, wird der Benutzer zuerst um seine Einverständnis gefragt. Erst wenn er zustimmt, dass der Browser seinen Standpunkt an die gerade geöffnete Anwendung weiterschicken darf, wird Geolocation ausgeführt [GEO].

Bei dem vorgestellten grafischen Passwortsystem wird Geolocation verwendet, um das Zentrum der Karte auf den aktuellen Aufenthaltsort zu setzen. Dies soll dem Benutzer die Arbeit ersparen einen bekannten Ort zu suchen, um sich orientieren zu können. Des Weiteren hat es sich in der Evaluierung (siehe Kapitel 6) herausgestellt, dass die meisten Benutzer Punkte für ihr Passwort in der Nähe ihres Wohnortes wählen.

Die Implementierung einer Geolocation ist mit der Geolocation API [W3C] recht simpel. Zum Abrufen einer Position, muss eine *successCallback*-Funktion existieren, die aufgerufen wird, wenn eine Position gefunden wurde. Es kann optional eine *errorCallback*-Funktion geschrieben werden, die aufgerufen wird, wenn eine Position nicht gefunden wurde. Ebenfalls optional kann eine Zeitbeschränkung (timeout) gesetzt werden, wie lange versucht werden darf, eine Position zu bestim-

men. Nachdem die Zeitbeschränkung abgelaufen ist, wird die *errorCallback*-Funktion aufgerufen. In dem Codebeispiel 5.3 ist die Zeitbeschränkung auf 10000 gesetzt. Dies entspricht zehn Sekunden. In der *successCallback*-Funktion werden die Koordinaten der gefundenen Position berechnet und danach die Karte auf diese Koordinaten zentriert. Die *errorCallback*-Funktion zentriert die Karte auf einen Standardort, in diesem Fall Darmstadt.

```
navigator.geolocation.getCurrentPosition(successCallback,
                                         errorCallback,
                                         {timeout:10000});

function successCallback(position){
        initialLocation = new google.maps.LatLng(
                                position.coords.latitude,
                                position.coords.longitude);
        map.setCenter(initialLocation);
}

function errorCallback(){
        //Darmstadt Koordinaten
        initialLocation = new google.maps.LatLng(49.87286251746749,
                                                 8.651706576347351);
        map.setCenter(initialLocation);
}
```

Codebeispiel 5.3: Geolocation

Der im Beispiel 5.3 aufgeführte Code zum Geolocation Dienst wurde auf verschiedenen Browsern getestet. Hierbei wurde sowohl getestet was passiert, wenn die Erlaubnis für den Geolocation Dienst erteilt wurde bzw. nicht erteilt wurde, sowie wenn die Frage nach der Erlaubnis für den Geolocation Dienst komplett ignoriert wurde. Die Browser Google Chrome und Safari haben wie oben beschrieben reagiert. Der Browser Internet Explorer (Version 9) schickt dem Benutzer erst gar nicht eine Anfrage zur Erlaubnis des Geolocation Dienstes, da dieser den Dienst nicht unterstützt. Damit das Passwortsystem in diesem Fall dennoch benutzt werden kann, wird eine Abfrage benötigt, ob ein Browser Geolocation unterstützt. Auch beim Browser Firefox gibt es Probleme. Wenn der oben aufgeführte Code in Firefox ausgeführt wird, reagiert der Browser bei einer erteilten Erlaubnis zwar richtig, wenn der Benutzer die Erlaubnis aber ignoriert oder nicht erteilt, bleibt die Karte grau. Anscheinend wartet Firefox so lange, bis die Erlaubnis erteilt wurde. Damit hier also auch bei einer ignorierten oder nicht erteilten Erlaubnis das grafische Passwortsystem noch verwendet werden kann, wird ein spezieller Timeout außerhalb des Geolocation-Aufrufs gebraucht, der die *errorCallback*-Funktion aufruft. Es wird eine spezielle Variable *geoHandlerReturned* angelegt, welche anfangs auf 0 gesetzt wird. Wenn nachdem der Timeout abgelaufen ist, in dem Fall sind es 10,5 Sekunden, die Variable immer noch auf 0 gesetzt ist, wird die *errorCallback*-Funktion aufgerufen. Wenn der Benutzer aber dem Geolocation Dienst zugestimmt hat, wird in der *successCallback*-Funktion die Variable *geoHandlerReturned* auf 1 gesetzt. Eine Lösung des Internet Explorer und Firefox Problems ist im Codebeispiel 5.4 aufgeführt.

Sollte ein Benutzer sein grafisches Passwort nicht in der Nähe seines aktuellen Standortes wählen wollen, muss er nicht vom aktuellen Standort aus die Karte zum gewünschten Ort ziehen, sondern kann dann über die Ortssuche direkt den Ort seiner Wahl auf der Google Maps Karte zentrieren lassen. Dies funktioniert mittels dem Geocoder aus der Google Maps API Version 3. Eine eingege-

bene Adresse wird vom Geocoder in ihre geografischen Koordinaten umkodiert und dann wird die Karte auf diese Koordinaten zentriert.

```
geoHandlerReturned = 0;

if(navigator.geolocation) {
        setTimeout("geoTimeoutHandlerFirefox()", 10500);
        navigator.geolocation.getCurrentPosition(successCallback,
                                                 errorCallback,
                                                 {timeout:10000});
}
else{
        errorCallback();
}

function successCallback(position){
        geoHandlerReturned = 1;
        initialLocation = new google.maps.LatLng(
                                position.coords.latitude,
                                position.coords.longitude);
        map.setCenter(initialLocation);
}

function geoTimeoutHandlerFirefox() {
        if (!geoHandlerReturned)
                errorCallback();
}
```

Codebeispiel 5.4: Erweiterung des Geolocation Dienstes für Browserunterstützung

5.5 Test des Systems

An allen Stellen im System und in der Umfrage wurden *mysql_real_escape_strings* eingefügt, damit keine *SQL Injections*[1] durch Benutzereingaben auftreten können. Ein *mysql_real_escape_string* maskiert spezielle Zeichen innerhalb eines Strings für die Verwendung in einer SQL-Abfrage [MES].

Wie in Kapitel 3 bereits erwähnt wurde, sind die meisten grafischen Passwortsysteme schwach gegen *Shoulder surfing* Angriffe. Es wurden mehrere *Shoulder-surfing* Angriffe initiiert um zu beurteilen, wie sicher dieses Passwortsystem ist. Als Vorbereitung darauf haben sich vier Benutzer im System registriert:

- Benutzer_01 hat sein Passwort auf der Standardzoomstufe (acht) gesetzt. Die Punkte, die er gewählt hat, waren die Stadt Frankfurt, die Stadt Darmstadt und die Stadt Mainz. Diese Punkte hatten eine Verbindung zum Benutzer. Frankfurt war seine alte Heimatstadt. In Darmstadt wohnt er aktuell und Mainz war einfach der dritte nötige Punkt der passend nebendran lag.

[1] SQL Injections sind Angriffe bei denen ein Benutzer des Systems Datenbankbefehle, wegen Sicherheitslücken (fehlende Maskierungen), in eine SQL-Abfrage einschleust. Diese Befehle werden ohne das Wissen des Entwicklers ausgeführt und können Passwörter auslesen oder Datenbankeinträge löschen.

- Benutzer_02 hat seine Punkte ebenfalls auf der Standardzoomstufe gesetzt. Seine Punkte hatten auch eine Verbindung zu ihm, allerdings ist diese nicht so offensichtlich wie bei Benutzer_01. Benutzer_02 wählte die Städte Stuttgart, Wuppertal und Heidelberg. In jeder der drei Städte wohnt ein Bekannter von ihm, den er sehr gerne, aber schon eine Weile nicht mehr gesehen hat.
- Benutzer_03 hat sein Passwort auf der Zoomstufe 17 gesetzt. Auch er wählte Punkte die etwas mit ihm zu tun haben. Alle seine Punkte wurden in Darmstadt gesetzt. Als ersten Punkt wählte er den Ort auf dem seine Wohnung steht. Als zweiten Punkt wählte er die Wohnung seiner Cousine, die am anderen Ende der Stadt liegt. Als dritten Punkt wählte er seine Uni (das Robert-Piloty-Gebäude).
- Benutzer_04 setzte seine Punkte genauso wie Benutzer_03 ebenfalls auf der Zoomstufe 17. Seine Punkte beschreiben seinen alten Schulweg und sind alle in Frankfurt gesetzt worden. Der erste Punkt ist auf seiner alten Wohnung. Den zweiten Punkt hat er auf eine U-Bahn Station zwischen seiner Wohnung und seiner Schule gesetzt und der dritte Punkt wurde auf die Schule selbst gesetzt.

Vier weitere Personen haben sich zur Verfügung gestellt diesen Angriff durchzuführen:

- Angreifer_01 wohnt in Frankfurt und ist ein Physik Student an der Uni Frankfurt.
- Angreifer_02, wohnhaft in Wiesbaden, hat früher in Frankfurt gewohnt und studiert jetzt Kommunikationsdesign an der Hochschule Rhein/Main.
- Angreifer_03, wohnhaft in Darmstadt, hat ebenfalls früher in Frankfurt gewohnt und studiert jetzt Informatik an der TU Darmstadt.
- Angreifer_04, wohnhaft in einem kleinen Ort nahe Frankfurt, ist genauso wie Angreifer_03 ein Informatik Student an der TU Darmstadt.

Es soll nun getestet werden, ob es für einen Angreifer schwieriger ist Punkte auf einer höheren Zoomstufe richtig zu treffen zu Vergleich zu Punkten auf der Ausgangsstufe. Des Weiteren soll getestet werden, ob bekannte Orte dem Angreifer helfen sich die Punkte eines Benutzers besser zu merken.

Jeder Benutzer hat nacheinander einmal seinen Benutzernamen und sein Passwort eingegeben, während die Angreifer hinter ihm standen und den Monitor beobachtet haben. Nach jeder Eingabe hatten die Angreifer drei Versuche sich mit dem Benutzernamen und dem Passwort des Benutzers anzumelden. Nachdem drei Versuche vorbei waren, hat der nächste Benutzer seine Login Daten eingegeben.

Das Passwort von Benutzer_01 wurde von allen vier Angreifern richtig eingegeben, wobei Angreifer_03 einen zweiten Versuch brauchte. Das Passwort von Benutzer_02 wurde von drei der vier Angreifer richtig eingegeben. Angreifer_03 konnte sich eine der drei Städte nicht merken. Benutzer_03 hat ein Passwort gewählt, das als einziges von niemandem in den drei vorgegebenen Versuchen richtig eingegeben werden konnte. Alle vier Angreifer gaben an, dass sie mit dem zweiten Punkt Probleme hatten, da sie sich die Straße nicht merken konnten und der Weg vom ersten zum zweiten Punkt zu kompliziert war. Das Passwort von Benutzer_04 wurde von drei der vier Personen richtig eingegeben, wobei Angreifer_04 drei Versuche gebraucht hat. Die drei Angreifer die das vierte Passwort richtig eingegeben haben, sagten dass es leichter war dieses Passwort wieder aufzurufen, als das dritte Passwort, da hier ein Weg vom ersten zum dritten Punkt gelaufen wurde und die einzelnen Punkte nicht so weit von einander entfernt waren. Des Weiteren waren dem ersten und zweitem Angreifer alle drei Punkte von Benutzer_04 bekannt, wodurch sie sich diese leicht merken konnten. In der Tabelle 5.3 ist nochmal zusammengefasst welcher Angreifer welches Passwort richtig eingeben konnte.

Tabelle 5.3: Shoulder-surfing Angriff. Ein Haken (✓) bedeutet das der Angreifer sich mit dem Passwort einloggen konnte. Ein Stern (*) hinter einem Haken bedeutet, dass der Angreifer mehr als einen Versuch gebraucht hat. Das X zeigt an, dass der Angreifer sich mit dem Passwort, innerhalb von drei Versuchen, nicht anmelden konnte.

	Passwort von Benutzer_01	Passwort von Benutzer_02	Passwort von Benutzer_03	Passwort von Benutzer_04
Angreifer_01	✓	✓	X	✓
Angreifer_02	✓	✓	X	✓
Angreifer_03	✓*	X	X	X
Angreifer_04	✓	✓	X	✓*

Ein sicheres Passwort sollte also auf einer höheren Zoomstufe gesetzt werden und die einzelnen Klickpunkte möglichst weit von einander entfernt sein. Am sinnvollsten wäre eine Kombination die nichts mit dem Benutzer selbst zu tun hat. Beispielsweise Wohnort, Universitätsgebäude, oder Schule sind für Personen, die den Benutzer kennen, leicht zu erraten. Es ist sinnvoll mindestens einen Punkt mit einzubeziehen, so wie Benutzer_03 es getan hat (Wohnung der Cousine), der nichts mit dem Benutzer selbst zu tun hat, der aber trotzdem für den Benutzer leicht zu merken ist.

6 Evaluation

Benutzerstudien sind eine der wichtigsten Aufgaben, die durchgeführt werden sollten, wenn ein neues System entwickelt wird. Dabei geht es darum folgende Fragen zu beantworten:

- Ist das System verständlich?
- Ist das System leicht zu benutzen?
- Kann ein Benutzer sein Passwort problemlos erstellen?
- Kann ein Benutzer sich problemlos anmelden?

Wichtig ist es auch zu sehen, wie Benutzer das System verwenden und wie sie ihre Passwörter wählen. Ist es eventuell notwendig Einschränkungen beim Setzen von Klickpunkten einzuführen, damit das Passwort sicher ist? Das sind alles Informationen die ein Entwickler, ohne Durchführung einer Benutzerstudie, nicht erhält.

6.1 Aufbau der Umfrage

Die erste Umfrage wurde in vier Bereiche aufgeteilt. Im ersten Bereich wurden fünf Fragen zur Person gestellt. Diese sollen einen Überblick verschaffen, ob Personen einiger Altersgruppen oder Arbeitsgebieten das Passwortsystem anders bewerten als der Rest. Sollte dies der Fall sein, könnte, je nach Altersgruppe und Arbeitsbereich, eine abgeänderte Form des Systems zur Verfügung gestellt werden, beispielsweise mit mehr Hinweisen zur Benutzung.

Im zweiten Teil der Umfrage geht es darum das System zu bewerten. Dies ist der wichtigste Teil der Umfrage. Hier konnte der Benutzer angeben, ob er mir dem System gut zurecht kam und sich sein Passwort gut merken konnte. Des Weiteren wurde gefragt, ob die Dauer der Registrierung und des Logins angemessen war. Wenn die Dauer der Registrierung etwas zu lang ist dies noch akzeptabel. Die Dauer eines Logins sollte allerding gering bleiben, da dieser meistens täglich verwendet wird. Ebenfalls in diesem Teil der Umfrage wurden die Teilnehmer gefragt wie sie ihr Passwort gewählt haben und warum sie dieses so gesetzt haben. Diese Frage hat den Zweck zu sehen, wie sicher Benutzer ihre Passwörter wählen und wenn schwache Passwörter gewählt werden, woran es liegt. Mit diesen Informationen kann das System eventuell um weitere Eigenschaften erweitert werden, die den Benutzer dazu bringen ihr Passwort noch sicherer zu wählen und es sich trotzdem noch genauso gut merken zu können.

Der dritte Teil der Umfrage ist dafür da, dieses grafische Passwortsystem mit anderen grafischen Passwortsystemen zu vergleichen. Hierbei wird wieder auf die Dauer der Registrierung und des Logins, wie gut das Passwort zu merken ist und wie leicht das System zu bedienen ist, eingegangen. Mit diesen Informationen kann bereits ein objektiver Vergleich unter den vielen verschiedenen grafischen Passwortsystemen gezogen werden.

Im vierten Bereich wird das grafische Passwortsystem mit textuellen Passwörtern (Kombination aus Klein-/ Großbuchstaben, Zahlen und Sonderzeichen) verglichen. Hier werden zuerst einige Fragen zu textuellen Passwörtern gestellt (siehe Abbildung 6.1). Diese sollen einen ersten Eindruck verschaffen, ob sich der Benutzer überhaupt Gedanken zum Thema Sicherheit macht.

Danach soll das System mit textuellen Passwörtern verglichen werden. Hier geht es wieder darum, die Dauer der Registrierung und des Logins, wie gut das Passwort zu merken ist und wie leicht das System zu bedienen ist, zu bewerten. Mit diesen Informationen kann wieder ein erster objektiver Vergleich zwischen dem grafischen Passwortsystem und textuellen Passwörtern gezogen werden.

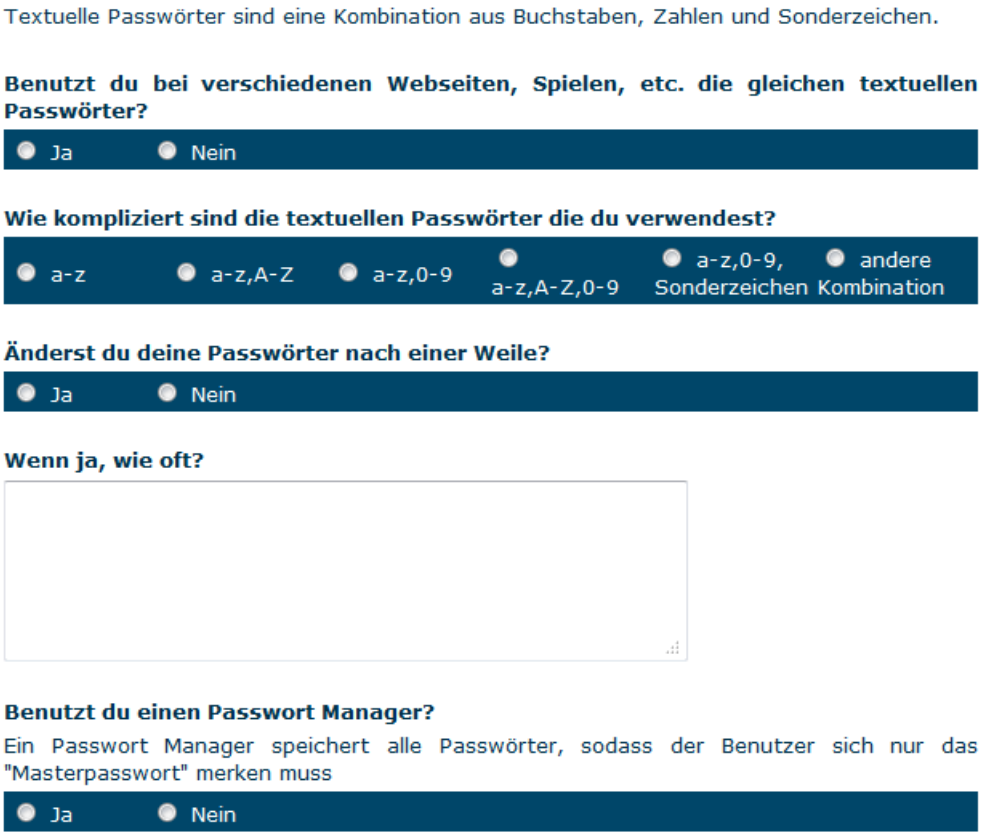


Abbildung 6.1: Umfrage 1 - Fragen zu textellen Passwörtern

Ebenfalls noch im vierten Bereich der ersten Umfrage befinden sich einige Fragen zur Sicherheit des grafischen Passwortsystems. Hierbei werden die Fragen, ob der Benutzer glaubt, dass ein Bekannter oder eine Person, die ihm beim Anmelden zusieht, sein Passwort erraten könnte, gestellt. Des Weiteren wird der Benutzer gefragt, ob er das grafische Passwortsystem oder textuelle Passwörter sicherer findet. Es ist wichtig, dass ein Passwortsystem nicht nur sicher ist, sondern dass die Benutzer auch wissen, dass es sicher ist. Wenn ein Benutzer der Meinung ist, dass das Passwortsystem nicht sicher ist, dann wird er dieses auch nicht verwenden wollen.

Die zweite Umfrage bestand nur aus einem einzigen Teil, wieder zum System selbst. Hier sollten die Teilnehmer angeben, ob sie sich noch gut an ihr Passwort erinnern konnten und ob sie es auch erwartet haben. Des Weiteren wurden sie nach Problemen während des Logins gefragt und sollten angeben wie sie mit der Bedienung des Systems zurecht kamen. Die zweite Umfrage diente der Kontrolle, ob sich Benutzer auch nach einigen Tagen ohne Benutzung des Passworts noch daran erinnern können. Ebenfalls sollte damit getestet werden, ob die Passworteingabe dem Benutzer, nach einer wiederholten Eingabe, leichter fällt.

6.2 Analyse

Insgesamt haben sich 50 Personen im System registriert. 37 haben die erste Umfrage (siehe Abschnitt 6.2.3) komplett abgeschlossen und 22 davon haben auch noch an der zweiten Umfrage (siehe Abschnitt 6.2.4) teilgenommen. Während der Registrierung und der Anmeldung wurde mit aufgezeichnet, wo und auf welcher Zoomstufe die Klickpunkte gesetzt wurden (siehe Abschnitt

6.2.1). Des Weiteren wurden fehlgeschlagene Anmeldungen sowie die Dauer der Anmeldung gespeichert (siehe Abschnitt 6.2.2).

6.2.1 Passwortwahl

53% aller Klickpunkte wurden direkt auf der Ausgangszoomstufe gesetzt. Benutzer gaben an, dass sie sich Städte besser merken konnten, als Stadtteile oder Straßen. Andere gaben an, dass sie ihre Punkte direkt gesetzt haben, ohne auf die Idee zu kommen weiter einzuzoomen. Insgesamt sagten 62% der Benutzer, dass sie Bekannte Orte gewählt haben. Viele von ihnen hatten Angst, dass sie sich andernfalls das Passwort nicht merken könnten. Sehr wenige schrieben, dass sie ihre Punkte zwar auf bekannte Orte, aber dafür diese sehr weit auseinander gesetzt haben, damit Angreifer nicht das Passwort herausfinden können. 13% gaben an, dass sie einfach große Städte ohne Verbindung zu sich selbst gewählt haben. In Tabelle 6.1 ist nochmal zu allen Zoomstufen aufgelistet, wieviele Klickpunkte auf dieser gesetzt wurden. Insgesamt wurden 150 Klickpunkte gesetzt (3 Klickpunkte pro Person).

Tabelle 6.1: Wahl der Zoomstufe bei der Registrierung. Acht ist die Ausgangszoomstufe auf der die Karte geladen wird. Zoomstufen niedriger als acht werden vom System nicht akzeptiert.

Zoomstufe	gesetzte Klickpunkte (in Personen)	gesetzte Klickpunkte (in Prozent)
8	80	53%
9	12	8%
10	4	2,7%
11	9	6%
12	9	6%
13	12	8%
14	0	0%
15	6	4%
16	10	6,7%
17	4	2,7%
18	4	2,7%
19	0	0%
20	0	0%
21	0	0%

6.2.2 Login

Es wurden 76 Anmeldeversuche auf dem System gestartet. Davon waren 80,26% mit dem ersten Versuch erfolgreich. Insgesamt waren 97,37% der Anmeldungen erfolgreich. Zwei Personen konnten sich an ihr Passwort nicht mehr erinnern und mussten es zurücksetzen lassen. 12 Teilnehmer haben einen zweiten Versuch und einer hat einen dritten Versuch gebraucht, damit die Anmeldung erfolgreich abgeschlossen wurde. In Tabelle 6.2 ist die Anmeldequote noch einmal zusammengefasst.

Die Dauer der Anmeldung lag bei den meisten Teilnehmern zwischen zehn Sekunden und einer Minute. Personen die komplizierte Passwörter gewählt haben und mehrere Versuche brauchten, haben für die Anmeldung bis zu zweieinhalb Minuten gebraucht.

Tabelle 6.2: Anmeldequote

	Versuchte Anmeldungen (in Personen)	Erfolgreiche Anmeldungen (in Personen)	Erfolgreiche Anmeldungen (in Prozent)	Fehlgeschlagene Anmeldungen (in Personen)	Fehlgeschlagene Anmeldungen (in Prozent)
1. Versuch	76	61	80,26%	15	15,79%
2. Versuch	15	12	15,79%	3	3,96%
3. Versuch	3	1	1,32%	2	2,63%
Insgesamt	76	74	97,37%	2	2,63%

6.2.3 Umfrage 1

An der ersten Umfrage haben 40 Personen teilgenommen. Von diesen haben 37 die Umfrage bis zum Ende ausgefüllt. Die Analyse der ersten Umfrage wird in die vier Bereiche, die im Abschnitt 6.1 erläutert wurden, aufgeteilt.

Umfrage 1 - Fragen zur Person

Der erste Teil der Umfrage ergab eine Teilnahme von 63% Informatiker (Studium und/oder Beruf) und 37% Nichtinformatiker. Nur einer der Nichtinformatiker beschäftigt sich in seiner Freizeit mit der Informatik. 50% der Teilnehmer waren im Alter von 23 bis 30 Jahren. 40% waren zwischen 18 und 22 Jahren und 10% waren über 30 Jahre alt.

Umfrage 1 - Fragen zum System

Im zweiten Teil der ersten Umfrage haben die Teilnehmer das System bewertet. Zwei Drittel konnten mit dem System gut umgehen und ein Drittel empfanden die Benutzung des Systems als schwierig. 92% fanden, dass das Passwort leicht zu merken ist. Bei der Sicherheit des Systems waren sich die Teilnehmer unsicher. Hier gaben 50% an, dass das Passwort sicher ist. 40% sagten es sei unsicher und 10% haben keine Angabe gemacht. Die Registrierungsdauer empfanden 85% der Teilnehmer als angemessen. Die Logindauer war noch für 75% angemessen. Die Ergebnisse, wie viele Personen was gewählt haben, sind nochmal in der Tabelle 6.3 abgebildet.

Teilnehmer wurden gefragt ob sie gerne mehr als drei Punkte im System verwendet hätten. Ein Drittel hätten gerne bis zu fünf Punkte gehabt, da dies die Sicherheit erhöht. Zwei Drittel wollten nicht noch mehr Punkte haben, da sie der Meinung waren, sie könnten sich mehr als drei Punkte nicht merken. Einige schlugen vor, dass die Anzahl der Punkte vom Benutzer selbst gewählt werden sollte (beispielsweise zwischen drei und fünf Punkten).

Am Ende des zweiten Teils wurden die Benutzer gefragt, was sie am System besonders positiv oder negativ fanden. Als gut wurde sehr häufig die Idee selbst genannt, da die meisten nie etwas von grafischen Passwörtern gehört haben und solch ein System zum ersten Mal verwendet haben. Den Benutzern hat es gefallen, dass sie ihr Passwort mal auf eine spielerische Art und Weise eingeben konnten und es trotzdem nicht so leicht erraten werden kann. Als schlecht empfanden einige, dass das Zoomen auf der Google Maps Karte zu viel Zeit in Anspruch nimmt und somit ein Login viel länger dauert als bei textuellen Passwörtern. Einige haben auch bemängelt, dass ihrer Ansicht nach das System gegen *Shoulder-surfing* Angriffe unsicher ist.

Tabelle 6.3: Benutzerstudie - Bewertung zum Google Maps basierendem Passwortsystem. In dieser Tabelle wird angezeigt wie viele Benutzer das System wie bewertet haben. Insgesamt haben 39 Personen eine Antwort abgegeben. Das Kürzel k.A. steht für „keine Angabe“.

	Trifft sehr zu	Trifft zu	Trifft etwas zu	Trifft nicht ganz zu	Trifft nicht zu	Trifft gar nicht zu	k.A.
Das Passwortsystem ist leicht zu benutzen	7	14	5	4	8	1	0
Das Passwort ist leicht zu merken	16	14	6	2	1	0	0
Das Passwort ist sicher	8	8	4	10	5	0	4
Die Registrierungsdauer ist angemessen	16	13	4	3	2	1	0
Die Logindauer ist angemessen	11	13	5	4	5	1	0

Einige der Benutzer haben noch Verbesserungsvorschläge zum System eingetragen. Hier ist öfters der Wunsch nach einer etwas höheren Toleranz zum Setzen der Klickpunkte aufgekommen. Viele hätten auch gerne eine leichtere Handhabung der Karte gehabt, sodass das Zoomen schneller geht. Ein Benutzer hat vorgeschlagen in dem Passwortsystem, ähnlich zu Picture Password (siehe Kapitel 2), Gesten einzuführen. Hier könnten dann Linien, Dreiecke und andere geometrische Figuren zwischen den Punkten gezogen werden.

Umfrage 1 - Fragen zu grafischen Passwortsystemen

Nur ein Drittel der Teilnehmer kannten bereits grafische Passwortsysteme. Die anderen gaben an, dass sie noch nie etwas von grafischen Passwörtern gehört haben, da alle Anwendungen nur textuelle Passwörter als Login fordern. Die meisten, die grafische Passwörter bereits kannten, haben als bekannte Systeme das PatternLock für Android, CAPTCHAS [CAP] oder das Picture Password für Windows 8 angegeben. CAPTCHAS sind Programme, die Webseiten gegen Bots[1] beschützen, indem verzerrte Wörter dargestellt werden, die von Menschen gelesen werden können, aber von Bots nicht. Diese dienen lediglich der Identifizierung von Bots und Menschen und sind keine grafischen Passwörter.

60% der PatternLock Nutzer finden, dass PatternLock leichter zu verwenden ist und 50% finden, dass ein PatternLock Passwort leichter zu merken ist. 80% sind der Meinung, dass die Anmeldung mit dem PatternLock schneller geht und 40% glauben, dass PatternLock sicherer ist. Die Ergebnisse der Bewertung durch die Teilnehmer ist nochmal in Tabelle 6.4 zusammengefasst.

Teilnehmer die Picture Password kennen, sagten dass die Bedienung des Systems bei beiden ungefähr gleich schwierig ist aber das Passwort vom Google Maps basierendem System viel leichter zu merken ist. Benutzer nehmen an, mit Picture Password würde die Registrungsdauer ungefähr gleichlang und der Login kürzer sein. Zum Vergleich der Sicherheit haben die Teilnehmer keine Angaben gemacht. Die Ergebnisse der Bewertung durch die Teilnehmer sind nochmal in Tabelle 6.5 zusammengefasst.

[1] Roboter (Computerprogramme), die selbstständig Aufgaben abarbeiten. Sie werden unter Anderem zum Durchsuchen von Webseiten verwendet.

Tabelle 6.4: Benutzerstudie - PatternLock gegen Google Maps basierendes Passwortsystem

	PatternLock	Google Maps basierendes Passwortsystem
Das Passwortsystem ist leichter zu benutzen	60%	40%
Das Passwort ist leichter zu merken	50%	50%
Das Passwort ist sicherer	40%	60%
Die Registrierung ist schneller	80%	20%
Der Login ist schneller	80%	20%

Tabelle 6.5: Benutzerstudie - Picture Password gegen Google Maps basierendes Passwortsystem. Das Kürzel k.A. in der Tabelle steht für „keine Angabe“.

	Picture Password	Google Maps basierendes Passwortsystem
Das Passwortsystem ist leichter zu benutzen	50%	50%
Das Passwort ist leichter zu merken	0%	100%
Das Passwort ist sicherer	k.A.	k.A.
Die Registrierung ist schneller	50%	50%
Der Login ist schneller	100%	0%

Ein Teilnehmer hat einen Vergleich mit CCP gezogen. Dieser gab an, dass das Google Maps basierende System und CCP ungefähr gleich gut zu bedienen und auch die Passwörter ungefähr gleich gut zu merken sind (mit einer leichten Tendenz, dass das Google Maps basierende System etwas simpler ist). Außerdem merkte er an, dass das Google Maps basierende System sicherer und die Registrierungs- und Logindauer kürzer seien. Ein weiterer Teilnehmer zog einen Vergleich zu Passfaces. Hier wurde ebenfalls angegeben, dass das Google Maps basierende Passwort mit einer leichten Tendenz simpler zu bedienen, leichter zu merken und sicherer ist. Passfaces dagegen soll eine kürzere Registrierungsdauer haben und auch eine ungefähr gleichlange Logindauer mit leichter Tendenz, dass Passfaces schneller ist.

Im dritten Teil der Umfrage wurde ein Training für grafische Passwörter vorgestellt. Dieses Training würde direkt im Anschluss an die Registrierung stattfinden. Ein Benutzer müsste sich, nach Erstellen seines Passworts, nochmal versuchen mit diesem einzuloggen. Dies soll ihm helfen sich die Punkte zu merken und mit den Bedingungen eines Logins schon im Voraus vertraut zu machen, da während des Logins die gesetzten Punkte, zum Schutz gegen *Shoulder-surfing* Angriffe, nicht angezeigt werden. Ein Drittel der Teilnehmer gaben an, dass sie ein Training für unnötig empfinden, da dadurch die Registrierung noch länger dauern würde. 50% der restlichen zwei Drittel gaben an, dass sie das Training für wichtig empfinden um sich ihr Passwort besser merken und das System leichter bedienen zu können. Mehrere Personen gaben an, dass sie das Passwort zumindest einmal unter Login Bedingungen noch während der Registrierung eingegeben hätten, da sie durch die fehlende Markierung der gesetzten Punkte im Login verwirrt waren. Einige Teilnehmer schlugen vor das Training dem Benutzer freiwillig anzubieten. Somit könnten Benutzer, die mit dem System gut zurecht kommen und sich ihr Passwort gut merken können, das Training überspringen. Diejenigen, die sich noch nicht sicher sind, ob sie sich das Passwort merken können, oder Probleme mit der Benutzung des Systems haben, könnten dann mit dem Training nochmal den Umgang mit dem System, durch wiederholte Eingabe des Passworts, üben. Vier Teil-

nehmer sagten, sie hätten gerne vor dem Login ausprobiert oder gesehen, wie die Genauigkeit für die Klickpunkte dann später beim Login sein wird. Ein Teilnehmer hätte gerne ein Training für das System und nicht für den Benutzer gehabt. Hierbei sollte das System eine individuelle Toleranz für jeden Benutzer erstellen, nachdem es vergleicht, wie groß der Unterschied zwischen den gesetzten Punkten bei der Registrierung und beim Training ist.

Umfrage 1 - Fragen zu textuellen Passwörtern und zur Sicherheit

Im vierten Teil der ersten Umfrage haben die Teilnehmer Fragen zu textuellen Passwörtern erhalten. Die Auswertung ergab, dass 50% der Teilnehmer für verschiedene Anwendungen die gleichen textuellen Passwörter verwenden. 8% der Benutzer wählen Passworter aus einer Kombination von Kleinbuchstaben und Zahlen. Weitere 33% wählen ihr Passwort aus einer Kombination von Klein- und Großbuchstaben, sowie Zahlen und nur 35% verwenden ein Passwort, welches auch Sonderzeichen enthält (siehe Abbildung 6.2). Knapp 28% verwenden einen Passwort-Manager. Insgesamt gaben 38% der Teilnehmer an, dass sie ihre Passwörter nach einer Weile ändern. Von diesen wechseln zwei Drittel ihr Passwort nach drei bis sechs Monaten und die restlichen ein Drittel sogar nur jährlich oder noch seltener.

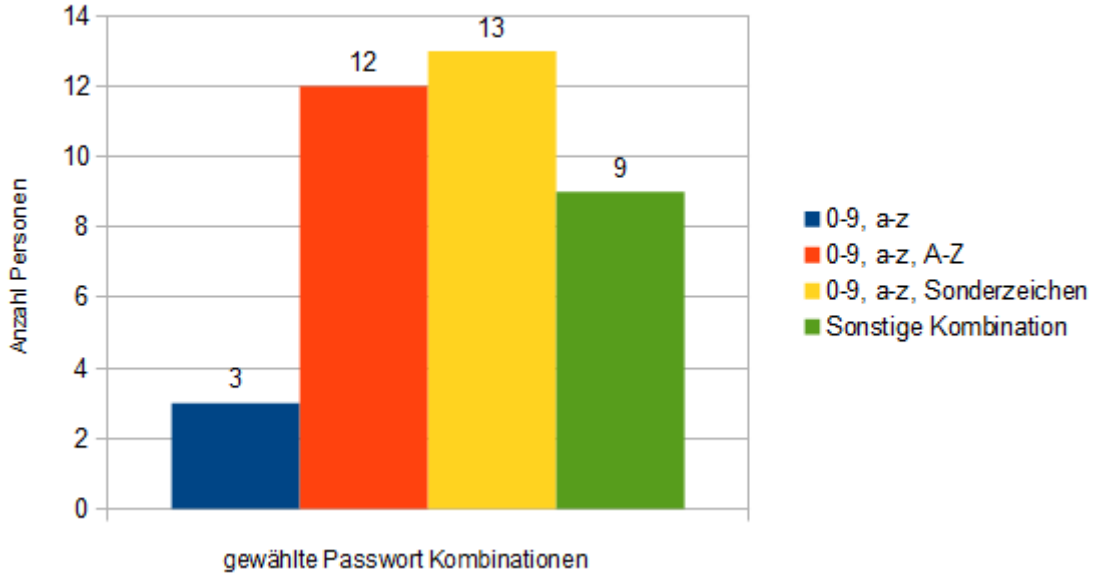


Abbildung 6.2: Verwendete Kombinationen bei textuellen Passwörtern

Benutzer sollten im Anschluss ihre textuellen Passwörter mit dem auf Google Maps basierendem Passwortsystem vergleichen. Die Auswertung zeigte, dass die Benutzung und die Sicherheit für beide Systeme ungefähr gleich eingeschätzt wurde. 70% der Teilnehmer empfanden, dass die Registrierungs- und Logindauer von textuellen Passwörten kürzer ist. Dafür konnten sich 81 % das Google Maps Passwort viel besser merken als ihr herkömmliches textuelles Passwort. In Tabelle 6.6 sind die Ergebnisse der Bewertung nochmal aufgelistet.

Als nächstes wurden den Teilnehmern Fragen zur Sicherheit des Systems gestellt. Die Frage, ob ein bekannter des Benutzers sein Passwort erraten könnte, haben 62% bejaht. Als Erläuterung wurde angegeben, das sie entweder große Städte in der Nähe des Wohnortes oder Orte die mit ihnen im Zusammenhang stehen gewählt haben. Einige der restlichen Teilnehmern sagten, dass sie ihre Punkte zufällig gewählt haben und diese nichts mit ihnen zu tun haben. Die Anderen

Tabelle 6.6: Benutzerstudie - Textuelle Passwörter gegen Google Maps basierendes Passwortsystem

	Textuelle Passwörter	Google Maps basierendes Passwortsystem
Das Passwortsystem ist leichter zu benutzen	54%	46%
Das Passwort ist leichter zu merken	19%	81%
Das Passwort ist sicherer	51%	49%
Die Registrierung ist schneller	70%	30%
Der Login ist schneller	70%	30%

gaben an, dass selbst bei Orten die mit ihnen selbst zu tun haben, es für einen Bekannten ohne Hinweis nahezu unmöglich ist von den vielen Möglichkeiten die richtigen Orte in der richtigen Reihenfolge und auch noch in der richten Klickpunkttoleranz zu treffen.

Auf die Frage, ob ein *Shoulder-surfing* Angriff auf das Google Maps basierende grafische Passwortsystem erfolgreich sein würde, haben 81% der Teilnehmer mit „Ja“ geantwortet. Ein Teil von ihnen schrieb, dass drei Punkte, insbesondere drei Städte leicht zu merken sind. Andere gaben an, dass beim Setzen einen Klickpunkts die Maus kurz angehalten wird und auch das Klicken der Maus zu hören ist, wodurch sich der Angreifer zumindest den ungefähren Bereich, in dem der Punkt liegt, merken kann. Sie nehmen an, dass auch wenn der Angreifer sich nicht mit dem ersten Versuch einloggen kann, er es doch mit mehreren Versuchen schaffen wird. Teilnehmer, die die Frage mit „nein“ beantwortet haben, schrieben dass der Angreifer zum einen das System erstmal kennen muss um überhaupt zu wissen, was er mit den Informationen anfangen kann. Des Weiteren braucht ein Angreifer ein gutes geografisches und fotografisches Gedächtnis, um die Orte wiederfinden zu können. Neben der Schwierigkeit die Orte erstmal zu finden, muss der Angreifer danach auch noch die Toleranz zu dem Klickpunkt, den er ja gar nicht genau gesehen hat, abschätzen und den Punkt in dem Toleranzradius setzen. Die Teilnehmer nehmen an, dass bei komplizierten Passwörtern und Orten die der Angreifer nicht kennt, er keine Möglichkeit haben wird diese wiederholen zu können.

Die Frage, ob textuelle Passwörter sicherer sind als das auf Google Maps basierende grafische Passwortsystem wurde von 50% der Benutzer mit „Ja“ beantwortet. Sie gaben an, dass textuelle Passwörter komplexer sind und eine längere Kombination ermöglichen. Viele schrieben auch, dass textuelle Passwörter sicherer gegen *Shoulder-surfing* Angriffe und deswegen allgemein sicherer sind. Weitere Teilnehmer nehmen an, dass ein Google Maps basierendes Passwort viel leichter zu erraten ist, wenn die Orte eine Verbindung zum Benutzer haben. Personen, die die Frage mit „Nein“ beantwortet haben, schrieben, dass je nach Wahl des textuellen Passworts dieses viel schwächer ist, da viele Benutzer öfters Passwörter der Form Name und Geburtsjahr oder Ähnliches wählen. Weitere Teilnehmer antworteten, dass es beim Google Maps basierendem grafischen Passwortsystem viel mehr Kombinationsmöglichkeiten für Passwörter gibt, im Vergleich zu textuellen Passwörtern. Des Weiteren können bei diesen auch viele verschiedene Passwörter verwendet werden, beispielsweise bei einem Passwort die drei Lieblingsrestaurants und bei einem anderen die drei Lieblingsschwimmbäder. Bei textuellen Passwörtern werden dagegen meistens die gleichen Passwörter verwendet, da sie sonst einfach nicht zu merken sind. Einer der Teilnehmer merkte noch an, dass für textuelle Passwörter nur ein *Keystroke-logger* gebraucht wird, um diese herauszufinden. Für das Google Maps basierende System wird zumindest ein *Maus-logger* und ein *Screen scraper* benötigt.

Zuletzt wurden die Teilnehmer noch gefragt, ob sie gerne das grafische Passwort irgendwo als Login verwenden würden. Hier haben wieder 50% mit „Ja“ geantwortet. Bei dieser Frage differen-

zierten sich die Erläuterungen der Teilnehmer. Ein Teil schrieb, dass sie es bei Anwendungen wie *Online Banking* und Ebay verwenden würden, da sie sich hier nicht so oft einloggen und deswegen ein Passwort brauchen, das leicht zu merken ist. Des Weiteren benutzen sie diese Anwendungen nur zu Hause, sodass keine *Shoulder-surfing* Angriffe auftreten können. Die anderen Teilnehmer gaben als Erläuterung an, dass sie es gerne für Spiele, ihr Handy oder soziale Netzwerke verwenden würden, allerdings auf keinen Fall für *Online Banking* oder ähnliches, da sie der Sicherheit von grafischen Passwörtern nicht trauen. Personen, die die Frage mit „Nein“ beantwortet haben, schrieben als Erläuterung, dass ihnen das Passwort zu umständlich und zu langsam einzugeben sei. Andere gaben an, dass sie einen Passwortmanager haben und mit diesem und auch mit textuellen Passwörtern bis jetzt noch keine schlechten Erfahrungen gemacht haben. Viele finden eine Bedienung über die Tastatur viel angenehmer und würden diese gerne beibehalten. Weitere Gründe waren, dass grafische Passwörter im Allgemeinen unsicher wirken, gerade da sie sehr unbekannt sind und die meisten Benutzer auch einfach nicht irgendetwas neues erlernen wollen.

Zwei Benutzer schrieben noch eine letzte Anmerkung zum System. Der erste Benutzer würde eine Kompatibilität des Systems zu einem Passwort-Manager sehr begrüßen, da hier dann auch ein komplizierteres grafisches Passwort gewählt werden könnte. Dieser neue Passwort-Manager könnte wiederum für alle Anwendungen verwendet werden, indem für diese beispielsweise zufällige komplexe textuelle Passwörter generiert werden, da aktuell alle Systeme nur textuelle Passwörter annehmen. Der zweite Benutzer schlug vor das System irgendwie auf eine Tastatureingabe umzuwandeln, sodass es sicherer gegen *Shoulder-surfing* Angriffe ist.

6.2.4 Umfrage 2

An der zweiten Umfrage haben noch 22 Personen teilgenommen. Teilnehmer der ersten Umfrage mussten mindestens drei Tage warten, bis sie auch die zweite Umfrage ausfüllen konnten. Dies war notwendig um zu sehen, ob sich die Teilnehmer immer noch an das Passwort erinnern können, auch wenn sie es schon einige Tage nicht mehr eingegeben haben.

Die zweite Umfrage ergab, dass vier der 22 Teilnehmer einen zweiten Versuch für die Anmeldung gebraucht haben. Zwei haben ihre Klickpunkte zu ungenau geklickt und die zwei anderen haben zum Zoomen einen Doppelklick verwendet, welches allerdings zwei Klickpunkten entspricht. 20 Personen gaben an, dass sie sich an ihr Passwort gut bis sehr gut erinnern konnten und es auch so erwartet haben. Drei Viertel gaben an, dass sie das System leicht zu bedienen finden und haben ihr Passwort auch schnell eingeben können. Die restlichen ein Viertel fanden die Systembedienung mittelmäßig bis nicht so gut, diese konnten ihr Passwort auch nicht so schnell eingeben.

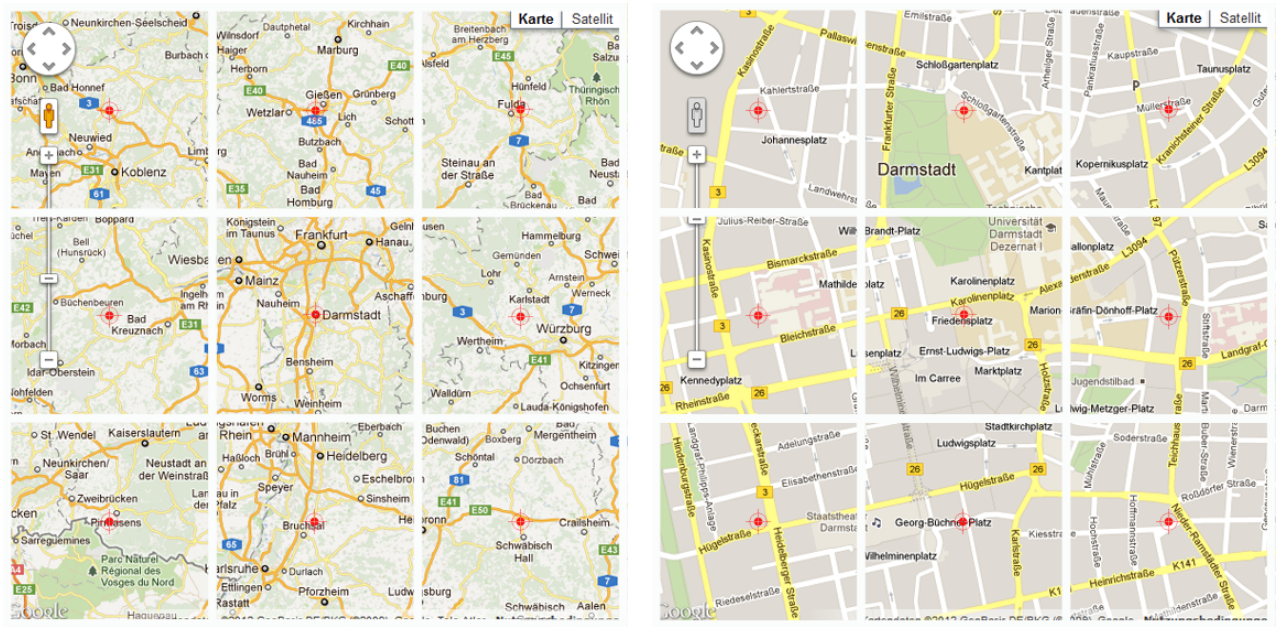
In dieser Bachelor-Thesis wurden einige der bereits entwickelten und veröffentlichten grafischen Passwörter zusammengefasst. Es wurden verschiedene Angriffe auf textuelle und grafische Passwörter vorgestellt, und erläutert wie die jeweiligen Passwörter gegen diese Angriffe geschützt werden können. Des Weiteren wurde ein neues grafisches Passwortsystem, welches auf Google Maps basiert, entwickelt und hier päsentiert. Das Passwortsystem hat eine hohe Anzahl an möglichen Kombinationen zum Setzen von Klickpunkten auf der Karte. Einige Tests ergaben, dass ein gut gewähltes Passwort sogar gegen *Shoulder-surfing* Angriffe sicher ist. Neben der Entwicklung wurde das System auch noch evaluiert. Die Ergebnisse der Evaluation zeigten, dass das Passwortsytem größtenteils leicht zu bedienen und das mit dem System erstellte Passwort leicht zu merken ist.

Für die Zukunft kann das System noch weiter ausgebaut werden. Die Evaluation ergab, dass die Registrierung und der Login noch zu lange dauern. Des Weiteren hatten manche Nutzer noch Probleme mit der Bedienung (Verwendung von einem Doppelklick zum Zoomen, wodurch zwei Klickpunkte gesetzt wurden). Diese Aspekte sollten auf jeden Fall ausgebessert werden. Das Setzen von zwei Klickpunkten kann verhindert werden, indem zum Setzen eines Klickpunktes eine Kombination aus einer Taste auf der Tastatur und dem Klick auf die Maus benötigt wird (beispielsweise Strg + linke Maustaste). Dies wäre insbesondere für Laptop Nutzer sehr von Vorteil, die lediglich ein Touchpad zur Verfügung haben und demnach beim Verschieben der Karte manchmal ausversehen einen Klickpunkt setzen.

Damit die Bedienung des Systems schneller wird, kann diese als Tastatursteuerung umgesetzt werden. Dabei könnte die Karte mittels der Pfeiltasten (↑, ↓, ←, →) nach oben, unten, links oder rechts bewegt werden. Mittels der Tasten + und − könnte dann auf der Karte ein- bzw. ausgezoomt werden. Als weitere Eigenschaft könnte hier auch die Passwortwahl über die Tastatur stattfinden. Dabei wird die Karte in ein 3x3-Gitter aufgeteilt und der Benutzer kann auf die Zellen im Gitter durch das Drücken von Zahlen auf dem Nummernblock zugreifen (siehe Abbildung 7.1). Dabei wird beispielsweise beim Drücken der Zahl 9 ein Klickpunkt in das Zentrum der Zelle rechts oben gesetzt. Diese Eigenschaft kann allerdings nur mit einer Tastatur verwendet werden und bei kleinen Laptops, die kein Nummernblock besitzen, müssten die Klickpunkte weiterhin mit der Maus gesetzt werden.

Eine Anmerkung, die in der Umfrage aufgetaucht ist und in der Zukunft umgesetzt werden sollte, ist die Erweiterung des Passworts auf fünf Klickpunkte. Dabei sollten die Benutzer die Möglichkeit haben selbst zu entscheiden ob sie drei, vier oder fünf Punkte wählen. Dies erhöht wiederum die Sicherheit, da ein Angreifer nicht weiß wie viele Punkte der jeweilige Benutzer gesetzt hat. Noch eine Anmerkung, die in der Umfrage aufgetaucht ist, ist die Erhöhung der Toleranzwerte für den Login. 20% der Benutzer konnten sich nicht erfolgreich mit dem ersten Versuch anmelden, da sie ihren Klickpunkte nicht genau genug getroffen haben. Es ist immer schwierig eine Toleranz zu finden, die für den Benutzer ausreichend ist, aber einen Angreifer daran hindert sich anmelden zu können. Hierfür sollten nochmal einige Tests durchgeführt werden, um die Toleranzwerte zu optimieren.

Wenn das System dann fertig ausgebaut wurde, kann eine Kompatibilität zu Passwort-Managern erstellt werden, wie es in der Umfrage von einer Person gewünscht wurde. Beim Passwort-Manager wird sich der Benutzer dann mittels dem Google Maps basierendem Passwort anmelden können. Der Passwort-Manager selbst wird dann zufällige komplexe und lange textuelle Passwörter generieren. Diese textuellen Passwörter werden an die Anwendungen, die textuelle Passwörter erwarten, übergeben und im Passwort-Manager gespeichert.



(a) Passwortwahl mittels Tastatur auf der Anfangszoomstufe

(b) Passwortwahl mittels Tastatur auf der Zoomstufe 15

Abbildung 7.1: Passwortwahl mittels Tastatur. Das Gitter entspricht dem Nummernblock auf einer Tastatur. Die Zelle links unten kann demnach mit einer 1 gewählt werden und die Zelle in der Mitte mit einer 5. Beim betätigen der Zahl zugehörig zur Zelle wird der Klickpunkt in die Mitte der gewählten Zelle gesetzt. Dies ist im Bild mit dem Zielsymbol verdeutlicht.

[AGM+10] AVIV, Adam J. ; GIBSON, Katherine ; MOSSOP, Evan ; BLAZE, Matt ; SMITH, Jonathan M.: Smudge Attacks on Smartphone Touch Screens / USENIX. 2010. – Forschungsbericht

[AJA] *AJAX mit der JQuery API*. http://api.jquery.com/category/ajax/, Zuletzt besucht: März 2012

[BBD05] BECHER, Alexander ; BENENSON, Zinaida ; DORNSEIF, Maximillian: Tampering with Motes: Real-World Physical Attacks on Wireless Sensor Networks / Aachener Informatik-Berichte (AIB). 2005. – Forschungsbericht

[BCO11] BIDDLE, Robert ; CHIASSON, Sonia ; OORSHOT, Paul van: Graphical Passwords: Learning from the First Twelve Years / ACM Computing Surveys. 2011. – Forschungsbericht

[BIS10] BROSTOFF, Sascha ; INGLESANT, Philip ; SASSE, M. A.: Evaluating the unability and security of a graphical one-time PIN system / 24th BCS Conference on Human Computer Interaction. 2010. – Forschungsbericht

[BMD] *Bing Maps*. http://www.microsoft.com/maps/developers/web.aspx, Zuletzt besucht: März 2012

[CAP] *CAPTCHA*. http://www.captcha.net/, Zuletzt besucht: März 2012

[CFBO08] CHIASSON, Sonia ; FORGET, Alain ; BIDDLE, Robert ; OORSCHOT, P.C. van: Influencing Users Towards Better Passwords: Persuasive Cued Click-Points / British Computer Society. 2008. – Forschungsbericht

[COB06] CHIASSON, Sonia ; OORSCHOT, P.C. van ; BIDDLE, Robert: A Usability Study and Critique of Two Password Managers / 15th USENIX Security Symposium. 2006. – Forschungsbericht

[COB07] CHIASSON, Sonia ; OORSCHOT, P. C. ; BIDDLE, Robert: Graphical Password Authentication Using Cued Click Points / ESORICS, LNCS 4734, pp.359-374. 2007. – Forschungsbericht

[Cor] CORPORATION, Passfaces: *Passfaces White Papers*. http://passfaces.com/enterprise/resources/white_papers.htm, Zuletzt besucht: März 2012

[DB] *Datenbankentwicklung*. http://www.simplethings.de/webentwicklung/datenbanken/, Zuletzt besucht: März 2012

[DMR04] DAVIS, Darren ; MONROSE, Fabian ; REITER, Michael K.: On User Choice in Graphical Password Schemes / 13th USENIX Security Symposium. 2004. – Forschungsbericht

[DP00] DHAMIJA, Rachna ; PERRIG, Adrian: Déjà Vu: A User Study Using Images for Authentication / 9th USENIX Security Symposium. 2000. – Forschungsbericht

[DY07] DUNPHY, Paul ; YAN, Jeff: Do Background Images Improve Draw a Secret Graphical Passwords / 14th ACM Conference on Computer and Communications Security (CCS). 2007. – Forschungsbericht

[Elf06] ELFTMANN, Patrick: *Secure Alternatives to Password-based Authentication Mechanisms*, RWTH Aachen, Diplomarbeit, 2006

[GEO] *Standort bezogenes Surfen*. http://www.mozilla.org/de/firefox/geolocation/, Zuletzt besucht: März 2012

[GHS02] GOLDBERG, Joseph ; HAGMAN, Jennifer ; SAZAWAL, Vibha: Doodling Our Way to Better Authentication / ACM Conference on Human Factors in Computing Systems (CHI). 2002. – Forschungsbericht

[GMA] *Google Maps-API*. http://code.google.com/intl/de/apis/maps/, Zuletzt besucht: März 2012

[GMD] *Google Maps JavaScript API Version 3*. http://code.google.com/intl/de/apis/maps/documentation/javascript/, Zuletzt besucht: März 2012

[GRI] *Wie GrIDsure funktioniert*. http://www.maiks.biz/site/index.php?id=225, Zuletzt besucht: März 2012

[GW07] GOLLE, Philippe ; WAGNER, David: Cryptanalysis of a Cognitive Authentication Scheme (Extended Abstract) / IEEE Symposium on Security and Privacy. 2007. – Forschungsbericht

[HAIM08] HAFIZ, Muhammad D. ; ABDULLAH, Abdul H. ; ITHNIN, Norafida ; MAMMI, Hazinah K.: Towards Identifying Usability and Security Features of Graphical Password in Knowledge Based Authentication Technique / Second Asia International Conference on Modelling & Simulation, pages 369 - 403, IEEE. 2008. – Forschungsbericht

[JMM+99] JERMYN, Ian ; MAYER, Alain ; MONROSE, Fabian ; REITER, Michael K. ; RUBIN, Aviel D.: The Design and Analysis of Graphical Passwords / 8th USENIX Security Symposium. 1999. – Forschungsbericht

[jQu] *JQuery*. http://jquery.com/, Zuletzt besucht: März 2012

[Kir94] KIRKPATRICK, E. A.: An experimental study ot memory. / Psychological Review 602 - 609. 1894. – Forschungsbericht

[MAPa] *Bing Maps*. http://www.bing.com/maps/, Zuletzt besucht: März 2012

[MAPb] *Google Maps*. http://maps.google.de/, Zuletzt besucht: März 2012

[MAPc] *Nokia Maps*. http://maps.nokia.com/, Zuletzt besucht: März 2012

[MAPd] *Yahoo Maps*. http://de.maps.yahoo.com/, Zuletzt besucht: März 2012

[MES] *mysql real escape string*. http://de3.php.net/manual/de/function.mysql-real-escape-string.php, Zuletzt besucht: März 2012

[MMP] *MapPoint*. http://www.microsoft.com/germany/mappoint/home.aspx, Zuletzt besucht: März 2012

[NT04] NALI, Deholo ; THORPE, Julie: Analyzing User Choice in Graphical Passwords / Carleton University. 2004. – Forschungsbericht

[OSM] *Open Streetmap*. http://www.openstreetbrowser.org/, Zuletzt besucht: März 2012

[php] *phpMyAdmin*. http://www.phpmyadmin.net/home_page/index.php, Zuletzt besucht: März 2012

[RS92] RAAIJMAKERS, Jeroen G. W. ; SHIFFRIN, Richard M.: Models for recall and recognition. / Annual Reviews. 1992. – Forschungsbericht

[SCH70] STANDING, Lionel ; CONEZIO, Jerry ; HABER, Ralph N.: Perception and memory for pictures: Single-trial learning of 2500 visual stimuli / Psychonomic Science, 19(2). 1970. – Forschungsbericht

[She67] SHEPARD, R. N.: Recognition memory for words, sentences, and pictures / Journal Of Verbal Learning And Verbal Behavior, Volume: 6, Issue: 1, Pages: 156-163. 1967. – Forschungsbericht

[Sin] SINOFSKY, Steven: *Signing in with a picture password*. http://blogs.msdn.com/b/b8/archive/2011/12/16/signing-in-with-a-picture-password.aspx, Zuletzt besucht: März 2012

[SS04] STUBBLEFIELD, Adam ; SIMON, Daniel R.: Inkblot Authentication / MSR-TR-2004-85, Microsoft Research. 2004. – Forschungsbericht

[Sta] STADTHAUS.COM: *Passwörter mit PHP generieren*. http://www.stadtaus.com/tutorials/content_cat_7_id_19.php, Zuletzt besucht: März 2012

[Suo06] SUO, Xiaoyuan: *A Design and Analysis of Graphical Password*, Georgia State University, Diplomarbeit, 2006

[TA08] TAO, Hai ; ADAMS, Carlisle: Pass-Go: Aproposal to Improve the Usability of Graphical Passwords / International Journal of Network Security. 2008. – Forschungsbericht

[Taf] TAFASA: *PatternLock*. http://tafasa.com/patternlock.html, Zuletzt besucht: März 2012

[Tao06] TAO, Hai: *Pass-Go, a New Graphical Password Scheme*, University of Ottawa, Diplomarbeit, 2006

[Tay] TAYLOR, Bret: *Mapping your way*. http://googleblog.blogspot.com/2005/02/mapping-your-way.html, Zuletzt besucht: März 2012

[TOH06] TARI, Furkan ; OZOK, A. A. ; HOLDEN, Stephen H.: A Comparison of Perceived and Real Shoulder-surfing Risks between Alphanumeric and Graphical Passwords / Symposium On Usable Privacy and Security (SOUPS). 2006. – Forschungsbericht

[Val98] VALENTINE, T.: An evaluation of the Passface personal authentication system / Goldsmiths College, University of London. 1998. – Forschungsbericht

[Var04] VARENHORST, Christopher: Passdoodles; a Lightweight Authentication Method / Research Science Institute. 2004. – Forschungsbericht

[W3C] *Geolocation API Specification*. http://dev.w3.org/geo/api/spec-source.html, Zuletzt besucht: März 2012

[Wei06] WEINSHALL, Daphna: Cognitive Authentication Schemes Safe Against Spyware (Short Paper) / IEEE Symposium on Security and Privacy. 2006. – Forschungsbericht

[WMB] *Wikimapia*. http://wikimapia.mattjonesblog.com/2006/08/09/wikimapia/, Zuletzt besucht: März 2012

[WMP] *Wikimapia*. http://wikimapia.org, Zuletzt besucht: März 2012

[Wor07] WORKMAN, Michael: Gaining Access with Social Engineering: An Empirical Study of the Threat / Information Systems Security, 16:6, 315-331. 2007. – Forschungsbericht

[WWB+05] WIEDENBECK, Susan ; WATERS, Jim ; BIRGET, Jean-Camille ; BRODSKIY, Alex ; MEMON, Nasir: Authentication Using Graphical Passwords: Basic Results / 11th International Conference on Human-Computer Interaction (HCI International). 2005. – Forschungsbericht

[YBAG04] YAN, Jeff ; BLACKWELL, Alan ; ANDERSON, Ross ; GRANT, Alasdair: Password Memorability and Security: Empirical Results / IEEE COMPUTER SOCIETY. 2004. – Forschungsbericht

[YMA] *Yahoo! Maps Web Services - AJAX API Getting Started Guide*. http://developer.yahoo.com/maps/ajax/, Zuletzt besucht: März 2012

API Application Programming Interface

SQL Structured Query Language

PHP Hypertext Preprocessor

Lat Latitude

Lng Longitude

DAS Draw-A-Secret

BDAS Background Draw-A-Secret

CCP Cued Click-Points

PCCP Persuasive Cued Click-Points