Säkerhetsanalys av ett kognitivt autentiseringsschema

FREDRIK BERGENLID
och TOMAS LYSEDAL

KTH Datavetenskap
och kommunikation

Examensarbete
Stockholm, Sverige 2010

Säkerhetsanalys av ett kognitivt autentiseringsschema

F R E D R I K B E R G E N L I D
o c h T O M A S L Y S E D A L

Examensarbete i datalogi om 15 högskolepoäng
vid Programmet för datateknik
Kungliga Tekniska Högskolan år 2010
Handledare på CSC var Mikael Goldmann
Examinator var Mads Dam

URL: www.csc.kth.se/utbildning/kandidatexjobb/datateknik/2010/bergenlid_fredrik_OCH_lysedal_tomas_K10041.pdf

Kungliga tekniska högskolan
Skolan för datavetenskap och kommunikation

KTH CSC
100 44 Stockholm

URL: www.kth.se/csc

Referat

I en rapport skriven av D. Weinshall presenteras ett autentiseringsschema i syfte att vara säkert mot avlyssnigar. Därefter har P. Golle och D. Wagner visat att användarens lösenord ändå kan hittas efter bara några få avlyssnade inloggningar. Vi har analyserat de faktorer som påverkar säkerheten i detta schema. Detta för att kunna peka på ändringar i schemat som kan bidra till en säkrare autentisering. Målet är helt enkelt att öka antalet avlyssnade inloggningar det krävs innan man kan hitta användarens lösenord. Resultatet av undersökningen är att utan att göra stora ändringar på schemat kan man inte öka säkerheten markant, dock ser vi att de grundläggande beståndsdelarna i schemat bör kunna användas för att skapa ett säkrare och fortfarande användarvänligt autentiseringsschema.

Abstract

Security Analysis of a Cognitive Authentication Scheme

In a report, D. Weinshall proposes an authentication scheme safe against an eavesdropping adversary. As a result of that P. Golle and D. Wagner presented an attack that can recover a users secret key after observing only a few number of successful logins with that scheme. We have analysed the security effect of the different factors the attack and the scheme depends on to be able to show weaknesses and propose suggestions for a better, more secure, scheme. Our results from this research conclude that without changing the scheme radically the scheme can not be sufficiently secure against eavesdroppers. However we argue that the foundation this schemes lies upon can be used to build a more secure and still user friendly authentication scheme.

Innehåll

Kapitel 1

Bakgrund

1.1 Inledning

I de flesta datorsammanhang finns behovet av att användare måste autentisera sig för att få åtkomst till olika tjänster eller till information. Detta behov har ökat kraftigt då allt fler tjänster görs tillgängliga på internet. Historisk sett har denna autentisering skett med hjälp av lösenord där användaren anger sitt användarnamn och sitt lösenord bestående av en hemlig teckenföljd. Denna metod har dock visat sig ha brister då exempelvis användare ofta tenderar att välja lösenord som är lätta komma ihåg [1]. Detta gör att lösenorden blir lättare att gissa för utomstående och exempelvis är det vanligaste lösenordet idag ”123456” [2]. Även om man inte har ett lösenord som är lätt att gissa så finns risken att ett program som kan avlyssna alla tecken som skrivs in finns installerat på datorn [3]. Detta gör att även de mest komplicerade lösenorden inte kan anses vara helt säkra.

Säkerheten hos autentiseringsmetoderna har blivit allt viktigare då många privata tjänster, som exempelvis bank och skattedeklarationer, nu även finns på internet. I dessa fall uppfyller inte vanliga lösenord säkerhetskraven så andra metoder har utvecklats. De flesta banker använder t.ex. en personlig säkerhetsdosa som generar en säker kod som enbart kan användas vid ett specifikt tillfälle [4]. En annan metod är erbjuda inloggning via så kallad e-legitimation som bygger på att användaren har en fil med en säker nyckel som endast går att få genom att legitimera sig på något annat sätt [5]. Dessa metoder kräver dock att användaren har tillgång till någon form av redskap vilket man inte alltid har så den ökade säkerheten innebär alltså minskad tillgänglighet till tjänsterna.

En annan typ av autentiseringsmetoder som försöker lösa säkerhetsbristerna hos vanliga lösenord men samtidigt inte förlita sig på något extra hjälpmedel är de kognitiva autentiseringsscheman [6]. Dessa bygger på människans naturliga förmåga att lättare minnas bilder och mönster än text och teckenföljder [6, 7, 8] vilket borde resultera i att mer komplexa lösenord används. Ett exempel på en sådant schema är att användaren presenteras en bild och ska klicka på förutvalda punkter i bilden för att autentisera sig [8]. Positionen och ordningen av de punkter användaren matat

in utgör lösenordet. En annan variant är att man autentiserar sig genom att välja ut valda bilder eller figurer ur en mängd presenterade som innehåller såväl de valda som ett antal andra bilder [8]. Dessa scheman skyddar inte mot avlyssning, eftersom man fortfarande kan se exakt vilka bilder eller punkter användaren väljer, utan riktar sig enbart mot att få till mer komplexa lösenord. Det finns dock en variant på det sistnämnda schemat som strävar efter att även vara säker mot avlyssning. Detta schema finns beskrivet i rapporten "Cognitive Authentication Schemes for Unassisted Humans, Safe Against Spyware" [6] skriven av Daphna Weinshall. Schemat bygger på att man aldrig explicit anger de valda bilderna utan resultatet av en funktion som beror på vilka bilder som är användarens valda. I rapporten ger hon inget formellt bevis för säkerheten mot avlyssning utan enbart ett generellt resonemang som bygger på att komplexiteten för problemet, att ur den informationen som ges ta fram lösenordet, är tillräckligt hög för att schemat ska vara säkert. Det har dock visat sig att det går att komma runt problemet med den höga komplexiteten. I rapporten "Cryptanalysis of a Cognitive Authentication Scheme" [9] skriven av Philippe Golle och David Wagner visas nämligen att användarens lösenord kan tas fram på ett fåtal sekunder efter bara några få avlyssnade inloggningar. Metoden som de använder i rapporten bygger på att den information som ges vid varje inloggning går att översätta till logiska samband mellan bilderna och svaret som, efter ett antal observerade inloggningar, enbart är uppfyllda av det korrekta lösenordet. Dessa samband går sedan, med hjälp av intelligenta algoritmer, att lösa på rimlig tid.

I den här rapporten undersöks hur olika faktorer i schemat beskrivet av Daphna Weinshall påverkar säkerheten mot metoden beskriven av Philippe Golle och David Wagner. Detta för att se om schemat går att göra säkrare mot den metoden samtidigt som användarvänligheten inte minskas eller säkerheten mot andra kända metoder inte blir sämre. För att göra detta kommer vi att implementera metoden beskriven i [9] och använda den mot olika varianter av schemat i [6] och analysera resultaten. Vi kommer inte att utföra någon egen användarstudie utan utgår från studien gjord i [6] och vi kommer inte heller att gå in på bildigenkänningsdelen av problemet utan anta att all information användaren matar in är tillgängligt för den som avlyssnar.

1.2 Det kognitiva schemat

I det här avsnittet beskrivs protokollet schemat bygger på samt de exakta implementationerna som presenteras i [6]. Den innehåller även en sammanfattning av användarstudien och säkerhetsanalysen som är gjord. All information i detta avsnitt kommer, om inget annat anges, från [6].

Protokollet bygger på att varje användare tilldelas två slumpmässigt valda mängder av bilder varav den första $\mathcal{B}$ innehåller N stycken bilder som är allmänt kända och kan delas mellan flera olika användare. Den andra mängden $\mathcal{F}$ är en delmängd av $\mathcal{B}$ ($\mathcal{F} \subset \mathcal{B}$) med $M < N$ stycken bilder och utgör den aktuella användarens hem-

liga lösenord. Denna mängd $\mathcal{F}$ av bilder måste användaren lära sig att urskilja från mängden $\mathcal{B}$ och för att sedan autentisera användaren presenterar systemet denne med följande procedur.

1. En mängd av n stycken slumpvalda bilder ur $\mathcal{B}$ presenteras för användaren.
2. Användaren ställs en enkel flervals fråga med P stycken möjliga svar som enbart kan besvaras av någon som vet vilka bilder i den presenterade mängden som tillhör $\mathcal{F}$.
3. Steg 1 och 2 upprepas k gånger och vid varje upprepning beräknas sannolikheten att svaren på alla hittills ställda frågor är gissade. Detta går till genom att, om användaren har gjort $e \leq k$ fel, beräkna sannolikheten att uppnå e eller färre fel på de k stycken försöken.
4. När sannolikheten för att svaren har gissats är mindre än ett förbestämt gränsvärde T autentiseras användaren. Om detta inte sker inom ett visst antal försök misslyckas inloggningen.

I rapporten beskrivs två olika implementationer av detta protokoll som skiljer sig både i valet av parametrarna och i frågan som ställs om de presenterade bilderna. I den ena implementationen ställs en något komplex fråga där svaret beror på många av bilderna och i den andra presenteras två olika enklare frågor där svaren istället beror på få bilder. I den första presenteras även fler bilder n i varje omgång än i den andra.

1.2.1 Högkomplexitetsfråga

Figur 1.1. Illustrering av högkomplexitetsvarianten

P_{19}	P_5	P_7	P_{42}	P_{26}	P_{23}	P_{13}	P_{77}	P_{27}	P_{16}	2
P_{54}	P_4	P_{67}	P_{37}	P_{53}	P_{11}	P_{68}	P_{10}	P_{14}	P_{43}	0
P_{62}	P_3	P_{76}	P_{70}	P_{52}	P_{64}	P_{74}	P_{22}	P_{12}	P_{69}	1
P_{57}	P_{73}	P_{39}	P_{65}	P_{49}	P_{18}	P_{63}	P_{75}	P_{56}	P_{78}	1
P_6	P_{34}	P_{79}	P_{17}	P_{36}	P_{38}	P_{44}	P_9	P_{28}	P_{31}	2
P_{35}	P_{48}	P_{41}	P_{71}	P_{45}	P_{32}	P_{60}	P_{21}	P_{66}	P_{24}	3
P_{59}	P_{58}	P_{15}	P_{30}	P_{29}	P_{61}	P_{46}	P_{72}	P_{80}	P_2	0
P_{50}	P_{55}	P_1	P_8	P_{33}	P_{47}	P_{20}	P_{25}	P_{40}	P_{51}	2
1	2	1	0	3	0	3	0	3	1	

I denna variant är den kända mängden 80 bilder stor $N = |\mathcal{B}| = 80$ och användarens hemliga mängd innehåller 30 bilder $M = |\mathcal{F}| = 30$ och $\mathcal{F} \subset \mathcal{B}$. I varje fråga som användaren ställs presenteras alla $n = N = 80$ bilderna från $\mathcal{B}$ i slumpmässig ordning och användaren har $P = 4$ svarsalternativ. I figuren 1.1 kan man se ett exempel på uppställningen av den här varianten där de 80 bilderna är presenterade

i en panel med $R = 8$ rader och $C = 10$ kolumner med utgångspunkter till höger respektive under panelen. Frågan som en användare ska svara på är, vilken siffra står i utgångspunkten man kommer till om man följer en stig genom matrisen givet följande regler?

1. Starta på positionen högst upp till vänster.
2. Om bilden på aktuell position finns i $\mathcal{F}$ gå ett steg nedåt, annars gå ett steg till höger.
3. Upprepa steg 2 tills du når en utgångspunkt.

De siffror stigen kan sluta på är $[0, 1, 2, 3]$ ($P = 4$) och dessa siffror är positionerade på ett sådant sätt att sannolikheten för att hamna på de olika siffrorna är så lika som möjligt.

1.2.2 Lågkomplexitetsfrågor

Figur 1.2. Illustrering av lågkomplexitetsvarianten

P_{82} (0)	P_{230} (1)	P_{12} (0)	P_{124} (0)	P_{201} (1)
P_{45} (1)	P_{22} (0)	P_{160} (1)	P_{17} (0)	P_{185} (0)
P_{164} (0)	P_{47} (1)	P_{218} (1)	P_{111} (0)	P_{198} (0)
P_{74} (1)	P_{64} (0)	P_{189} (1)	P_{178} (1)	P_{132} (1)

I rapporten föreslås två olika typer av den här varianten, de skiljer sig inte i upplägg och parameterval, utan endast i frågan som ställs. I båda dessa typer är den kända mängden 240 bilder stor $N = |\mathcal{B}| = 240$ och användarens hemliga mängd innehåller 60 bilder $M = |\mathcal{F}| = 60$ och $\mathcal{F} \subset \mathcal{B}$. I varje fråga som användaren ställs presenteras $n = 20$ slumpvalda bilder från $\mathcal{B}$ i slumpmässig ordning och användaren har $P = 2$ svarsalternativ. Till varje bild i panelen tilldelas en nolla eller en etta, antalet nollor och ettor är lika många och slumpmässigt distribuerade (se figur 1.2). Användaren ska sedan gå igenom panelen rad för rad, från vänster till höger och svara på en av förljande frågor.

Första typen: Identifiera första och sista bilden som tillhör $\mathcal{F}$ och svara om de tillhörande värdena är samma eller inte.

Andra typen: Identifiera första, andra och sista bilden som tillhör $\mathcal{F}$ och svara om majoriteten av de tillhörande värdena är ett eller noll.

1.2.3 Användarstudie

För högkomplexitetsfrågan med $N = 80$ gemensamma och $M = 30$ egna bilder utfördes en studie med nio personer som alla fick genomgå två eller tre tränings-sessioner på olika, direkt på varandra följande, dagar. Efter träningens slut kom de

tillbaka för testsessioner efter en dag, två dagar, fem dagar, en vecka och sedan en gång i veckan i 10 veckor. Några av deltagarna kom därefter även tillbaka enligt ett glesare schema i fyra månader till. Resultatet av studien visade att alla deltagare lyckades svara rätt på runt 95% av frågorna ända upp till sista testsessionen, vilken ägde rum nästan ett år efter träningen. Vidare tog varje fråga 10 till 15 sekunder att genomföra och deltagarna kände inget behov av att, exempelvis peka på skärmen för att följa stigen utan kunde räkna ut svaret enbart i huvudet. Om detta inte hade varit fallet hade det medfört en säkerhetsrisk ifall någon tittat på.

För lågkomplexitetsfrågan med $N = 240$ och $M = 60$ genomfördes en liknande studie men med enbart två deltagare. Resultatet blev ungefär lika mellan de två typerna och även likt det för högkomplexitetsfrågan med runt 95% rätt upp till ett år efter träningen. Varje fråga tog runt fem sekunder att svara på och deltagarna kunde även här räkna ut svaret direkt i huvudet.

1.2.4 Säkerhetsanalys

En vanlig metod för att knäcka lösenord kallas brute force[1] och säkerheten mot en brute force attack mäts i antalet möjliga kombinationer som finns för lösenordet. Med detta protokoll blir denna mängd $\binom{N}{M}$, så i fallet med högkomplexitetsfrågan blir antalet kombinationer $\binom{80}{30} \approx 2^{73}$ och med lågkomplexitetsfrågan $\binom{240}{60} \approx 2^{190}$. Detta går att jämföra med ett klassiskt lösenord byggt av åtta alfanumeriska tecken[2] vilket innebär $72^8 \approx 2^{49}$ olika kombinationer. En variant av brute force är den så kallade ordlisteattacken[3] men dessa bygger på att vissa kombinationer är mer sannolika än andra vilket inte är fallet med detta protokoll då bilderna som utgör lösenordet är slumpade till varje användare.

En säkerhetsrisk som finns med detta protokoll som inte finns med vanliga lösenord är att man kan göra en lyckad inloggning utan att veta lösenordet. Detta beror på att enbart P olika svarsalternativ finns i varje omgång och sannolikheten att gissa rätt blir alltså $1/P$. På grund av detta har protokollet en gräns T för hur stor sannolikheten att någon har gissat får vara innan användaren autentiseras. I rapporten föreslås gränsen vara 10^{-6} vilket för högkomplexitetsfrågan med $P = 4$ ger 11 omgångar och för lågkomplexitetsfrågorna 22 stycken med $P = 2$, detta om man antar att användaren gör rätt i 95% av omgångarna.

För avlyssning behandlas säkerheten mot två olika metoder att ta till vara på informationen som ges vid varje omgång av en inloggning. Båda bygger på att vid varje omgång reduceras antalet möjliga kombinationer lösenordet kan bestå av med, som mest, $1/P$. Kan man då hålla reda på vilka kombinationer som fortfarande är möjliga så kan man antingen köra en brute force attack på de kvarvarande eller fortsätta avlyssna fler omgångar tills enbart en möjlighet finns kvar. Den andra metoden

[1] Brute force innebär att man helt enkelt prövar alla möjliga kombinationer lösenordet kan bestå av [1]

[2] tecken innehållande bokstäver a-z både stora och små, samt siffrorna 0-9. Totalt 72 stycken

[3] Ordlisteattack (**eng** Dictionary attack) innebär att man istället för att pröva alla kombinationer endast testar de i en ordlista som innehåller de mest sannolika lösenorden [1]

benämns uppräkningsattack (**eng** enumeration attack) och har genom simuleringar beräknat komplexiteten för en sådan attack. Med parametrarna i högkomplexitetsfrågan ($N = 80$ och $M = 30$) medger den beräknade komplexiteten att en kraftfull motståndare kan utföra en sådan attack på rimlig tid men att med de för lågkomplexitetsfrågorna ($N = 240$ och $M = 60$) är den, med dagens mått mätt, tillräckligt hög för att vara säker.

Mot lågkomplexitetsfrågan av den första typen beskrivs även en annan typ av attack som tar till vara på informationen som ges vid varje omgång. Den bygger på att sannolikheten avtar drastiskt för att första respektive sista bilden ska komma på positioner längre bort från ändpunkterna på panelen. Kortfattat går metoden till så att, för varje omgång med tillhörande svar, vikta alla möjliga par av bilder med sannolikheten att just det paret är den första respektive sista bilden. Efter ett antal avlyssnade omgångar kan man då med hög sannolikhet lyckas logga in, för parametrarna i fråga kan man exempelvis ha $0.7 \pm 0.2\%$ chans att lyckas efter 3000 observerade omgångar. Samma attack kan även utföras mot den andra typen men då krävs betydligt fler observerade omgångar.

1.3 Attacken

Metoden som presenteras i [9] bygger, som tidigare nämnts, på att man, genom att översätta alla observationer till logiska samband, kan komma runt problemet med den höga komplexiteten. De logiska sambanden som ställs upp utgör nämligen ett booleskt uttryck med alla bilder som variabler och de restriktioner varje observation ger som satser. Problemet är då översatt till det oerhört kända satisfierbarhetsproblemet (SAT) som lyder: givet ett booleskt uttryck, tilldela de booleska variablerna sanningsvärden så att hela uttrycket evaluerar till sant [10]. Komplexiteten för SAT är dock inte lägre än för de attacker som tas upp i [6], det är till och med NP-fullständigt[4], men eftersom det har så många olika tillämpningar [11, 12] finns effektiva algoritmer utvecklade, även kallade heuristiker[5], som ändå kan lösa problemet tillräckligt snabbt för att schemat i [6] inte kan anses vara säkert. De logiska samband som ställs upp (dessa beskrivs i detalj i nästa kapitel) körs alltså genom en SAT-lösare för att få ut lösenordet. SAT-lösaren som används i [9] är UBCSATs [14] implementation av SAPS algoritmen [15] och testerna är körda på en dator med dubbla 3.40GHZ processorer och 1.00 GB RAM under Windows XP. Resultatet av undersökningen i presenteras i tabell 1.1 och 1.2.

I [9] skrivs det att metoden hittar flera olika lösningar vid färre än 60 omgångar med standardparametrarna i högkomplexitetsfrågan och samma sak för standardparametrarna för första typen av lågkomplexitetsfrågorna, men då under 250 omgångar. För resten av testerna presenteras ingen sådan information och det som

[4]Problem som NP-fullständiga anses vara de svåraste problemen en dator kan lösa och även om det inte är bevisat än tror de flesta att en lösning inte kan hittas i polynomisk tid [10].

[5]En heuristik är en snabb metod för att närma sig en korrekt lösning, dock innehåller inte en heuristik några bevis på hur nära den korrekta lösningen man kommer [10, 13]

Tabell 1.1. resultaten som [9] presenterat om högkomplexitetsfrågan

Parametrar för protokollet				Attackens komplexitet	
N	M	P	Panelstorlek	#Omgångar	Tid(s)
80	30	4	8 x 10	60	102
80	30	4	8 x 10	100	7
120	45	4	8 x 10	500	45
120	45	2	8 x 10	1000	≈960

Tabell 1.2. resultaten som [9] presenterat om lågkomplexitetsfrågorna

Parametrar för protokollet				Attackens komplexitet	
N	M	n	Typ	#Omgångar	Tid(s)
240	60	20	Första	250	<1
600	150	20	Första	800	<2.6
240	60	20	Andra	400	<1

visas är att även för större värden på N och M så kan metoden ta fram lösenordet på rimlig tid. I [9] skrivs även att, givet begränsningarna hos SAT-lösare, kan parametrar väljas så att metoden inte skulle fungera men att människans förmåga att minnas bilderna förmodligen skulle svika långt tidigare.

Kapitel 2

Metod

2.1 De logiska sambanden

I [9] presenteras och motiveras samtliga samband som ligger till grund för indatan till SAT-lösaren. I vår undersökningen används samma samband för att implementera attacken och för att kunna analysera vilka faktorer som är relevanta att förändra presenteras dessa samband i detta avsnitt. Alla samband som presenteras här är alltså hämtade från [9].

2.1.1 Högkomplexitetsfrågan

För att översätta de observationer man får vid varje omgång i högkomplexitetsfrågan till booleska satser som SAT-lösaren kan ta som indata, måste man först definiera upp de booleska variabler som ska användas. Låt de N booleska variablerna $A_1, A_2, ..., A_N$ representera de N bilderna i den allmänt kända mängden $\mathcal{B}$ och sanningsvärdet hos dessa variabler indikerar huruvida bilderna är med i den privata mängden $\mathcal{F}$, där $A_i = 1$ betyder att bilden A_i finns i $\mathcal{F}$ och $A_i = 0$ om bilden inte finns i F (A_i resp. $\bar{A}_i$). Problemet reduceras därmed till att finna sanningsvärden till variablerna $A_1, A_2, ..., A_N$.

$$(A_i = 1) \Rightarrow A_i \in \mathcal{F}$$

$$(A_i = 0) \Rightarrow A_i \notin \mathcal{F}$$

Vidare introduceras de booleska hjälpvariablerna $B^k_{(r,c)}$ med följande egenskaper (där R och C är antalet rader respektive kolumner i matrisen):

- För $1 \leq r \leq R$ och $1 \leq c \leq C$, låt $B^k_{(r,c)}$ representera sanningsvärdet för att stigen i omgång k gått igenom elementet på rad r och kolumn c.
- För $1 \leq r \leq R$, låt $B^k_{(r,C+1)} = 1$ om användarens stig under omgång k slutade på högra sidan av matrisen på rad r.

- För $1 \leq c \leq C$, låt $B^k_{(R+1,c)} = 1$ om användarens stig under omgång k slutade på botten av matrisen på kolumn c.

Givet en användares svar i omgång k kan följande logiska samband mellan de definierade variablerna ställas upp. Eftersom vi vet att användaren startar i positionen högst upp till vänster måste följande gälla.

$$B^k_{(1,1)} \ \forall k \tag{2.1}$$

Låt $f(k, r, c)$ vara indexet för bilden i rad r och kolumn c i omgång k, enligt reglerna i frågan ska användaren flytta ner en rad i matrisen om bilden i nuvarande position finns med i $\mathcal{F}$. Det vill säga: Om användaren befinner sig på position (r, c) i omgång k ($B^k_{(r,c)}$) och den aktuella bilden tillhör $\mathcal{F}$ ($A_{f(k,r,c)}$) kommer nästa position att vara $(r + 1, c)$ ($B^k_{r+1,c}$) vilket ger:

$$(A_{f(k,r,c)} \wedge B^k_{(r,c)}) \Rightarrow B^k_{r+1,c} \tag{2.2}$$
$$\forall k, \forall r \in \{1, 2, ..., R\} \ och \ \forall c \in \{1, 2, ..., C\}$$

På samma sätt, om den aktuella bilden inte tillhör $\mathcal{F}$ kommer nästa position att vara $(r, c + 1)$

$$(\bar{A}_{f(k,r,c)} \wedge B^k_{(r,c)}) \Rightarrow B^k_{r,c+1} \tag{2.3}$$
$$\forall k, \forall r \in \{1, 2, ..., R\} \ och \ \forall c \in \ \{1, 2, ..., C\}$$

Slutligen låter vi p^k vara svaret användaren matat in i omgång k samt v_r och w_c vara siffran som står i utgångspunkten till höger i rad r respektive underst i kolumn c. Vi vet då att stigen användaren följt inte kan ha slutat i de rader där $v_r \neq p^k$ och inte heller i de kolumner där $w_c \neq p^k$. Följande samband måste då gälla:

$$\bar{B}^k_{(r,C+1)} \forall r : v_r \neq p^k$$
$$\bar{B}^k_{(R+1,c)} \forall c : w_c \neq p^k \tag{2.4}$$

Omvänt så måste stigen ha slutat i en rad så att $v_r = p^k$ eller i en kolumn så att $w_c = p^k$. Vilket ger:

$$\left(\bigvee_{r|v_r=p^k} B^k_{(r,C+1)} \right) \vee \left(\bigvee_{c|w_c=p^k} B^k_{(R,c+1)} \right) \tag{2.5}$$

2.1.2 Lågkomplexitetsfrågorna

Precis som för högkomplexitetsfrågan definierar vi först de booleska variabler som kommer användas i konstruktionen av de logiska satserna för lågkomplexitetsfrågorna. Låt även här $A_1, A_2, ..., A_N$ representera de N bilderna i det allmänt kända mängden $\mathcal{B}$. Eftersom det, enligt beskrivningen i [6], inte är tydligt vad användaren

ska göra om de presenterade bilderna innehåller färre än två bilder från $\mathcal{F}$ (i första typen) eller färre än tre bilder från $\mathcal{F}$ (i andra typen) har [9] i sin undersökning endast genererat utdata där dessa situationer inte förekommer. Så är även fallet i vår undersökning och detta medför ingen stor förenkling då dessa situationer ändå inte är speciellt sannolika.

Låt för varje omgång $i_1, i_2, ..., i_n \in \{1, 2, ..., N\}$ representera indexen för de n visade bilderna dvs. A_{i_2} är sanningsvärdet för att bilden på index 2 i denna omgång är en av användarens hemliga, låt seden $b_1, b_2, ..., b_n$ representera bitarna associerade med dessa bilder. Slutligen låt b vara siffran som användaren svarade. För båda typerna vet vi trivialt att en av de presenterade bilderna måste finnas i $\mathcal{F}$ och följande sats måste alltså gälla:

$$(A_{i_1} \vee A_{i_2} \vee ... \vee A_{i_n}) \tag{2.6}$$

Första typen

I denna fråga ska användaren identifiera den första och sista bilden, av de presenterade, som tilhör $\mathcal{F}$ och svara om de tillhörande värdena är samma eller inte. Om b_i och b_j var värdena hos de identifierade bilderna svarar användaren med $b_i \oplus b_j$. För alla $1 \leq j < k \leq n$ får vi då att, om $b_j \oplus b_k \neq b$ så kan inte bilden på plats j och bilden på plats k vara den första respektive sista bilden av de presenterade som tillhör $\mathcal{F}$. Detta ger:

- Om $b_1 \oplus b_n \neq b$ kan inte båda bilderna med index i_1 och i_n tillhöra $\mathcal{F}$.

$$(b_1 \oplus b_n \neq b) \Rightarrow (\bar{A}_{i_1} \vee \bar{A}_{i_n}) \tag{2.7}$$

- För alla $1 \leq j < k \leq n$ sådana att $(j, k) \neq (1, n)$ och $b_j \oplus b_k \neq b$ får vi följande samband. Om båda bilderna med index i_1 eller i_n tillhör $\mathcal{F}$ måste det finnas minst en till bild som tillhör $\mathcal{F}$ i spannet $[1, j-1] \cup [k+1, n]$. Alltså:

$$(A_{i_j} \wedge A_{i_k}) \Rightarrow \Big((A_{i_1} \vee A_{i_2} \vee \ldots \vee A_{i_{j-i}}) \vee (A_{i_{k+1}} \vee A_{i_{k+2}} \vee \ldots \vee A_{i_n})\Big) \tag{2.8}$$

$$\forall j, \forall k : 1 \leq j < k \leq n, j \neq 1, k \neq n, b_j \oplus b_k \neq b$$

Andra typen

Här ska användaren identifiera första, andra och sista bilden som tillhör $\mathcal{F}$ och svara om majoriteten av de tillhörande värdena är ett eller noll. Vi låter $maj(b_j, b_k, b_l)$ betyda majoriteten av värdena b_j,b_k och b_l. Vi ser då att för alla $1 \leq j < k < l \leq n$, om majoriteten av de tre värdena (b_j, b_k, b_l) inte är lika med b så kan inte bilderna med index i_j,i_k och i_l vara den första, andra respektive sista av de presenterade som tillhör $\mathcal{F}$. Vi får följande samband:

- Om majoriteten av b_1,b_2 och $b_n \neq b$ så kan inte alla bilderna med index i_1,i_2 och i_n vara med i $\mathcal{F}$.

$$\Big(maj(b_1, b_2, b_n) \neq b\Big) \Rightarrow (\bar{A}_{i_1} \vee \bar{A}_{i_2} \vee \bar{A}_{i_n}) \tag{2.9}$$

- För alla $1 \leq j < k < l \leq n$ sådana att $(j, k, l) \neq (1, 2, n)$ och att majoriteten av $(b_j, b_k, b_l) \neq b$ ser vi att om alla bilderna på index i_j,i_k och i_l tillhör $\mathcal{F}$ måste det finnas minst en till bild som tillhör $\mathcal{F}$ i spannet $[1, k-1] \cup [j+1, k-1] \cup [l+1, n]$ vilket ger:

$$\begin{aligned}(A_{i_j} \wedge A_{i_k} \wedge A_{i_l}) \Rightarrow \Big(& (A_{i_1} \vee A_{i_2} \vee \ldots \vee A_{i_{j-1}}) \vee \\ & (A_{i_{j+1}} \vee A_{i_{j+2}} \vee \ldots \vee A_{i_{k-1}}) \vee \\ & (A_{i_{l+1}} \vee A_{i_{l+2}} \vee \ldots \vee A_{i_n})\Big)\end{aligned} \tag{2.10}$$

$$\forall j, \forall k, \forall l : 1 \leq j < k < l \leq n, j \neq 1, k \neq 2, l \neq n, maj(b_j, b_k, b_l) \neq b$$

2.2 Analys av metoden

Genom att översätta problemet till SAT har författarna i [9] lyckats minska tiden det tar att få fram användarens lösenord tillräckligt mycket för att en attack ska vara möjlig. För att göra schemat säkrare måste alltså informationen som ges i varje omgång minskas så att antalet observerade omgångar som krävs istället ökar. En viktig egenskap att notera med SAT problemet är att det inte tar hänsyn till hur sannolik ett viss sats är utan behandlar alla lika. Detta är av stor betydelse för vidare resonemang. Gemensamt för både hög- och lågkomplexitetsfrågorna är självklart att antalet bilder, både M och N, har betydelse för hur många omgångar som krävs då dessa begränsar den totala sökrymden som lösenordet kan ligga i $\binom{N}{M}$. För att ta reda på vilken effekt valet av de resterande parametrarna kan ha analyseras de logiska sambanden som ställts upp.

2.2.1 Högkomplexitetsfrågan

Av de samband som ställs upp för varje omgång i högkomplexitetsfrågan så är formel 2.1 – 2.3 reglerna för hur användaren ska röra sig genom panelen och formel 2.4 och 2.5 restriktionerna som svaret som angetts innebär. Tillsammans kan de tolkas som mängden möjliga stigar genom panelen som slutar i en utgångspunkt med den siffra som användaren svarat respektive de stigar som inte gör det. Varje unik stig motsvarar en uppsättning av bilder för vilka det, om stigen är den som användaren följt, gäller att vissa är med i användarens egna $\mathcal{F}$ och andra är det inte. Med denna tolkning inses alltså att informationen som ges vid varje omgång är en mängd möjliga stigar, alltså kombinationer av bilder som är med respektive inte är med i $\mathcal{F}$, samt en mängd kombinationer som inte är möjliga. Hur mycket information som ges beror alltså enbart på hur stora dessa mängder är och i [6] motiveras distributionen av de P svarsalternativen däremot endast genom sannolikheter, för att öka

säkerheten mot en gissningsattack. Som vi ser i figur 2.1 har dock inte sannolikheten och antalet stigar till varje utgångspunkt ett direkt förhållande, exempelvis är sannolikheten för utgången i rad 2 och utgången i kolumn 6 båda 3% medan antalet vägar skiljer sig (10 resp. 792). På grund av detta kan svarsalternativen positioneras så att alla svar har samma sannolikhet men olika antal möjliga stigar till sig.

Formel 2.5 tillsammans med reglerna ger oss alltså information om vilka stigar som användaren kan ha tagit och omvänt ger formel 2.4 tillsammans med reglerna de stigar som inte kan ha tagits. För att kunna kvantifiera hur mycket information dessa ger inför vi $\mathcal{E}_\mathcal{M}$ och $\mathcal{E}_\mathcal{I}$ som väntevärdet[1] för antalet möjliga stigar respektive det antal stigar som inte är möjliga genom panelen för varje omgång. Om vi låter $\mathcal{T}_\mathcal{S}$ beteckna det totala antalet stigar genom panelen så blir alltså $\mathcal{E}_\mathcal{I} = \mathcal{T}_\mathcal{S} - \mathcal{E}_\mathcal{M}$. Med lågt $\mathcal{E}_\mathcal{M}$ (och alltså högt $\mathcal{E}_\mathcal{I}$) ger vi mycket information om de möjliga vägarna men lite om det som inte är möjliga. Detta förhållande borde ha betydelse för hur bra metoden lyckas. Sannolikheterna och antalet stigar för varje svarsalternativ för standardutförandet visas i tabell 2.1 och för dessa värden får vi $\mathcal{E}_\mathcal{M} \approx 10931$ och $\mathcal{E}_\mathcal{I} \approx 32827$.

Tabell 2.1. Antalet stigar och sannolikheter till standardutförandets svarsalternativ

P	0	1	2	3
Sannolikhet	0.252	0.247	0.248	0.253
#Stigar	9359	11752	12164	10483

Figur 2.1. Visar antalet stigar till elementen i matrisen. I den sista kolumnen och sista raden visas sannolikheten för att en godtycklig stig slutat i respektive utgång (avrundat till hela procent). Värdena är beräknade med parametrarna föreslagen i [6]

1	1	1	1	1	1	1	1	1	1	1%
1	2	3	4	5	6	7	8	9	10	3%
1	3	6	10	15	21	28	36	45	55	7%
1	4	10	20	35	56	84	120	165	220	11%
1	5	15	35	70	126	210	330	495	715	13%
1	6	21	56	126	252	462	792	1287	2002	14%
1	7	28	84	210	462	924	1716	3003	5005	13%
1	8	36	120	330	792	1716	3432	6435	11440	11%
<1%	<1%	1%	1%	2%	3%	4%	5%	6%	7%	

En annan faktor som har betydelse är antalet i varje omgång visade bilder n, alltså storleken på panelen som användaren presenteras med. I [6] visas att en mindre panelstorlek ger mer information om användarens bilder och är sämre mot en uppräkningsattack. Detta inses också genom vår tolkning då det totala antalet stigar $\mathcal{T}_\mathcal{S}$ genom panelen blir färre om storleken på panelen minskas. I standardutförandet blir $\mathcal{T}_\mathcal{S} = 43758$. Minskas storleken på matrisen beror svaret användaren ger på

[1] väntevärde definieras som bekant $E(X) = \sum_i x_i p(x_i)$

färre bilder och svaret säger mer om dessa, dock visa färre bilder och säger alltså ingenting om de bilder som inte presenterats. En minskning av panelstorlek säger alltså mer om färre bilder, medan en ökning av storleken säger mindre om fler bilder.

Tittar man på formel 2.5 kan man se att denna ger mer detaljerad information om vilka bilder som kan ingå i användarens hemliga mängd. Det formeln säger är (indirekt) att någon av de bilder som ligger på högra sidan av panelen vid en utgångspunkt motsvarande användarens svar inte kan tillhöra $\mathcal{F}$ eller att någon av de bilder som ligger på nedre sidan av panelen vid en utgångspunkt motsvarande användarens svar måste tillhöra $\mathcal{F}$. Exempelvis om användaren svarar med en tvåa säger formel 2.5, på panelen i figur 1.1, att någon av bilderna P_{16},P_{31} eller P_{51} inte kan tillhöra $\mathcal{F}$ eller att bilden P_{55} tillhör $\mathcal{F}$. Med flera utgångar till varje svarsalternativ fås alltså flera möjliga variabler i denna formel och därmed ges mindre information om användarens bilder.

2.2.2 Lågkomplexitetsfrågorna

I formel 2.6, som gäller för både första och andra typen, sägs att någon av de n stycken presenterade bilderna måste finnas med bland användarens hemliga $\mathcal{F}$. Genom att öka n kan alltså informationen, som formel 2.6 ger, minskas. Vidare säger formel 2.7 och 2.8 (första typen) att om siffrorna som tilldelats två godtyckliga bilder inte motsvarar användarens svar och båda är med i $\mathcal{F}$ så måste minst en bild innan eller efter också vara med i $\mathcal{F}$. Mängden par av godtyckliga bilder som inte uppfyller ett visst svar ökar med storleken på n så för större n ökar antalet möjliga par så det ges information om fler bilder. För större n finns dock även fler bilder som ligger innan och efter varje par så mängden information om varje bild minskar. Samma resonemang gäller även för andra typen och den enda faktorn, utöver N och M, som har inverkan på säkerheten är alltså antalet visade bilder n.

Kapitel 3

Undersökning

3.1 Allmänt om undersökningen

Syftet med undersökningen är att se hur de olika faktorerna som togs i analysen i avsnitt 2.2 påverkar säkerheten hos schemat. I undersökningen används, precis som i [9], programmet UBCSAT (ver-1.1) med algoritmen SAPS för att lösa de booleska uttryck som ställs upp enligt sektion 2.1. Testerna är körda på en dator med en Intel Core 2 Quad processor på 2.83GHz och 4Gb RAM under operativsystemet Ubuntu 9.04 med linuxkärnan 2.6.28. Undersökningen går till så att, för varje val av parametrar och fråga utförs följande fem steg:

1. Simulera k stycken omgångar (inga felaktiga svar simuleras).
2. Generera de logiska sambanden utifrån simulationen.
3. Kör UBCSAT r gånger med tidsgräns t och spara resultatet.
4. Upprepa steg 1 till 3, x gånger.
5. Upprepa steg 1 till 4 för ökande värden på k.

Anledningen till denna, något omständiga, procedur är att i både steg 1 och 3 beror resultatet på slumpmässig information. I steg 1 simuleras en slumpgenererad matris vilket kan resultera i att vissa omgångar (med tillhörande svar) ger mer information om användarens bilder än andra och på grund av detta upprepas steg 1 till 3 x gånger. Eftersom SAPS algoritmen delvis bygger på slumpgenererade värden kan det för exakt samma indata ta olika lång tid att hitta en lösning, därför kan det, efter en tid t, vara effektivare att låta UBCSAT börja om. Vidare tillåter de logiska sambanden flera olika lösningar för låga k så, genom att köra UBCSAT r gånger kan fler av dessa hittas vilket ger information om hur många möjliga lösningar som finns.

Initialt undersöks många olika värden på k med låga värden på x och r för att se vid vilket antal omgångar som intressanta resultat framträder. För dessa värden på k genomförs sedan mer omfattande tester med högre värden på x och r. I de

flesta fall används $x = 10$ och $r = 10$ initialt och för de mer ingående testerna 40 respektive 20. Tidsgränsen t är satt så att minst en lösning hittas i steg 3 för alla relevanta värden på k och typiskt har $t = 80$ sekunder varit tillräckligt.

För varje parameterval som testas presenteras antalet omgångar som krävs för få fram användarens lösenord samt tiden det tar att göra det. Som tidigare nämnts beror resultaten av varje simulerad fil på slumpen så antalet omgångar definieras som det antal för vilket mer än 90% av filerna enbart resulterar i den korrekta lösningen. Tiden som presenteras är då ett genomsnitt av de $x \cdot r$ stycken körningarna i UBCSAT. Vidare sparas även, för varje k, hur många olika lösningar som hittas och om någon av dessa är korrekt.

3.2 Tester och resultat

I [6] föreslås att man kan variera N, M och med högkomplexitetsfrågan även P, för att öka säkerheten och det är även det som är gjort i [9]. För att få referensvärden utförs samma tester i vår undersökning men även för fler varianter för att skapa en bild över vilka faktorer som påverkar säkerheten mest. För att inte komma fram till praktiskt orimliga varianter begränsas valet av parametrarna av den användarstudie som är gjord i [6].

3.2.1 Högkomplexitetsfrågan

Som i analysen i avsnitt 2.2 betecknas totala antalet stigar genom panelen med $\mathcal{T}_{\mathcal{S}}$, väntevärdet för antalet stigar till ett visst svarsalternativ $\mathcal{EM}$ och väntevärdet för antalet stigar som inte slutar ett visst svarsalternativ $\mathcal{EI}$. Vidare så är, om inget annat anges, distributionen av de P svarsalternativen gjord så att sannolikheten och antalet vägar till varje är så lika som möjligt. Eftersom simuleringen inte innefattar felaktiga svar så krävs det 10 omgångar per inloggning för att uppnå samma säkerhetsgräns som är satt i [6].

M och N

I tabell 3.1 visas resultatet av att variera N och M, första raden motsvarar standardvalet av parametrar och den andra det val som är testat i [9]. Rad tre och fyra visar resultatet för så höga värden på M respektive N som användarstudien för lågkomplexitetsfrågan visar är rimliga, detta för att få en övre gräns för hur mycket enbart variering av N och M kan öka säkerheten. För att se hur enbart den totala sökrymden $\binom{N}{M}$ påverkar säkerheten så hålls förhållandet mellan N och M lika i samtliga tester. Detta för att inte förändra sannolikheterna för varje utgångspunkt så att samma distribution av de $P = 4$ svarsalternativen kan användas. Resterande parametrar hålls också låsta för att inte förändra antalet stigar $\mathcal{T}_{\mathcal{S}}$ eller utgångspunkter i panelen. Resultaten visar, som väntat, att för större sökrymd $\binom{N}{M}$ krävs fler antal omgångar och för $N = 240$ och $M = 90$ krävs 400 omgångar vilket motsvarar 40 observerade inloggningar.

Tabell 3.1. Resultat från variering av N och M

N	M	P	Panelstorlek	#Omgångar	Tid(s)
80	30	4	8 x 10	90	15
120	45	4	8 x 10	120	72
160	60	4	8 x 10	180	32
240	90	4	8 x 10	400	70

Panelstorlek

Tabell 3.2 visar resultatet av att variera panelstorleken (R och C) och eftersom standardparametrarna redan visar samtliga bilder, testas att minska panelstorleken till 7x9 med resterande parametrar låsta (rad ett). Genom att göra detta minskas $\mathcal{T}_\mathcal{S}$ till 11440 från 43758 och antalet utgångspunkter minskas till 16 från tidigare 18. Med detta val av parametrar testas alltså att ge mer information om färre bilder.

I rad två visas även resultatet av att öka till en 10x12 panel för vilket även N och M måste ökas. Antalet stigar $\mathcal{T}_\mathcal{S}$ och utgångspunkter genom matrisen blir då 646646 respektive 22. Med detta val av parametrar testas alltså att ge mindre information om fler bilder.

Vid minskning av panelstorleken till 7x9 för standardparametrarna krävdes istället 80 omgångar. För $N = 120$ och $M = 45$ krävdes däremot 180 jämfört med 120 omgångar vid ökning av panelstorleken till 10x12 (alla bilder visade). Resultaten antyder alltså att det har betydelse att det vid mindre panelstorlek ges information om färre bilder men att det är av större vikt att det ges mindre information om varje bild med större panel.

Tabell 3.2. Resultat från variering av panelstorlek

N	M	P	Panelstorlek	#Omgångar	Tid(s)
80	30	4	7 x 9	80	3
120	45	4	10 x 12	180	43

Distributionen av svarsalternativen

Tabell 3.3 visar resultatet av de tester som på olika sätt ämnar att undersöka hur, det förväntade värdet på antalet stigar som slutar i siffran användaren svarat respektive de som inte gör det, samt antalet utgångspunkter påverkar säkerheten av schemat. ($\mathcal{E}_\mathcal{M}$, $\mathcal{E}_\mathcal{I}$ och antalet utgångspunkter)

I rad ett och två presenteras tester med samtliga parametrar valda som standard men med en distribution av de $P = 4$ svarsalternativen som gör att det förväntade värdet förändras. För att åstadkomma förändringen har inte längre alla utgångspunkter samma sannolikhet, men eftersom metoden i sektion 2.1 inte tar hänsyn till sannolikheter visar resultaten ändå bara hur förändringarna hos $\mathcal{E}_\mathcal{I}$ och $\mathcal{E}_\mathcal{M}$ påverkar säkerheten mot denna metod. Detta innebär däremot att för dessa val på distribu-

tionen minskar säkerheten mot en gissningsattack. I rad ett (med P betecknad 4H) är det väntevärdet för antal möjliga respektive icke möjliga stigar $\mathcal{E}_{\mathcal{M}} = 17477$ och $\mathcal{E}_{\mathcal{I}} = 26281$ och i rad två (med P betecknad 4L) är $\mathcal{E}_{\mathcal{M}} = 9482$ och $\mathcal{E}_{\mathcal{I}} = 34276$. Resulatet visar att om $\mathcal{EM}$ ökas (och samtidigt $\mathcal{E}_{\mathcal{I}}$ minskas, eftersom det totala antalet stigar $\mathcal{T}_{\mathcal{S}}$ är konstant) krävs det fler omgångar och i testet i rad 1 har $\mathcal{E}_{\mathcal{M}}$ ökats (och $\mathcal{E}_{\mathcal{I}}$ minskats) med 6546 stigar och antalet omgångar som krävdes ökade med 20 stycken från 90 till 110. Omvänt ser vi att, trots att vi bara sänker $\mathcal{E}_{\mathcal{M}}$ (och ökar $\mathcal{E}_{\mathcal{I}}$) med 1449 stigar ser vi en minskning i antal omgångar som krävs med 20 stycken.

I rad tre presenteras resultatet av att sänka antalet svarsalternativ till $P = 2$ med resterande parametrar satta som standard. Distributionen av svarsalternativen ger ungefär lika sannolikhet för de båda samt förväntade värden på antal möjliga respektive icke möjliga stigar $EM = 21879$ och $EO = 21878$. Genom att halvera P dubbleras även antalet utgångspunkter till varje svarsalternativ så resultatet beror på förändring av två olika faktorer, men ger ändå värdefull information om hur dessa två faktorer påverkar säkerheten. Att sänka P är även en av de föreslagna förändringarna i [6] och ett test som utförts i [9].

Rad fyra presenterar resultatet av en något mer förändrad variant (med P betecknad 4B) av schemat som är framtagen för att testa hur mycket antalet utgångspunkter till varje svarsalternativ påverkar säkerheten. Mer exakt är detta gjort genom att införa utgångspunkter (utöver de som finns som standard) i matrisen som placeras mellan elementen i sista raden och mellan elementen i sista kolumnen. Om man befinner sig på sista raden och har den aktuella bilden svarar man med siffran för utgången under bilden, har man däremot inte bilden svarar man med siffran som står mellan aktuella bilden och den till höger. På motsvarande sätt i sista kolumnen, om man inte har den aktuella bilden svarar man med siffran till höger och annars med siffran nedanför. Genom att göra på detta viset erhålls 32 utgångspunkter men enbart 22880 stigar genom matrisen och $\mathcal{EM} = 5720$. Så även här beror resultatet på två olika faktorer och $\mathcal{E}_{\mathcal{M}}$ i frågan har alltså minskat från 10931 till 5720 medan antalet utgångspunkter ökat från 18 till 32 i jämförelse med standardfrågan.

I första och andra testet visades att $\mathcal{E}_{\mathcal{M}}$ och $\mathcal{E}_{\mathcal{I}}$ har inverkan på säkerheten och med detta i vetskap verkar antalet utgångar ha en positiv påverkan på antalet omgångar som krävs, detta eftersom det krävdes fler omgångar än test 4L och bara 10 färre än standarduppställningen. Testet med $P = 2$ antal svarsalternativ pekar i samma riktning, där både $\mathcal{E}_{\mathcal{M}}$ och antalet utgångar har ökat och antalet observerade omgångar som krävdes ökades till 170.

3.2.2 Lågkomplexitetsfrågorna

I lågkomplexitetsfrågorna krävs, för att uppnå samma säkerhetsgrad föreslagen i [6], 20 omgångar. Med lågkomplexitetsfrågan kan enbart parametrarna N, M och n varieras och i tabell 3.4 presenteras resultatet för olika val av dessa. Rad ett och två motsvarar test av parametervalen i standardutförandet av första respektive andra

Tabell 3.3. Resultat från ändring av distributionen av svarsalternativen samt antalet utgångar

N	M	P	Panelstorlek	#Omgångar	Tid(s)
80	30	4H	8 x 10	110	43
80	30	4L	8 x 10	70	24
80	30	2	8 x 10	170	58
80	30	4B	8 x 10	80	29

typen. I rad tre och fyra presenteras resultaten för båda typerna med N och M som i högkomplexitetsfrågan, detta testas för att kunna jämföra säkerheten mellan hög- och lågkomplexitetsfrågan. Rad fem motsvarar det test [9] utfört för att visa att metoden klarar att ta fram lösenordet även för betydligt större N och M än vad som anses rimligt att en människa kan minnas. Rad sex visar resultatet av att öka storleken av panelen med första typen av frågan. Anledningen till att inte $N = 600$ och $M = 150$ eller $n = 40$ testas för andra typen av frågan är att de genererade sambanden som ska testas helt enkelt blir för stora, exempelvis tar dessa 17MB för standardutförandet av andra typen vid 600 omgångar.

Precis som i högomplexitetsfrågan visar resultaten att för större sökrymd $\binom{N}{M}$ krävs flera omgångar. Dessutom visar resultaten att för lika stora N och M som i standardutförandet hos högkomplexitetsfrågan krävs 160 och 240 omgångar för första respektive andra typen, vilket innebär 8 respektive 12 inloggningar. Som jämförelse krävdes 9 inloggningar (90 omgångar) för högkomplexitetsfrågan. Vidare visas att med flera visade bilder n krävs också fler observerade omgångar, men att det inte verkar ha lika stor betydelse här som i högkomplexitetsfrågan. För $N = 600$, $M = 150$ och $n = 20$ avbröts testet efter 1000 omgångar, fortfarande med många olika unika lösningar. Resultaten visar även att andra typen kräver fler omgångar i samtliga utförda test.

Tabell 3.4. Resultat av ändringar på lågkomplexitetsfrågan

N	M	n	Typ	#Omgångar	Tid(s)
240	60	20	Första	420	<1
240	60	20	Andra	700	<1
80	30	20	Första	160	<1
80	30	20	Andra	240	<1
600	150	20	Första	>1000	N/A
240	60	40	Första	450	<1

Kapitel 4

Diskussion och slutsats

4.1 Diskussion

För högkomplexitetsfrågan visar resultatet av undersökningen att det finns många olika faktorer som påverkar säkerheten hos schemat. Högst säkerhet, mätt i antal observerade inloggningar som krävs, av de tester som utförts erhålls genom att öka M och N till 240 respektive 90 då det krävs 40 lyckade inloggningar för att i 90% av fallen ha fått ut enbart det korrekta lösenordet. Dessa val av N och M borde, enligt användarstudien för lågkomplexitetsfrågan i [6], även vara rimliga i detta fall om dock med en något förminskad användarvänlighet. Om man tillsammans med ökningen av N och M samtidigt ökat panelstorleken skulle förmodligen ännu bättre resultat uppnås men det skulle då även vara markant svårare för användaren. Syftet med hela undersökning är just att ta reda på vilka faktorer i schemat som påverkar säkerheten för att se ifall en säkrare variant går att uppnå utan att försvåra för användaren. Detta verkar inte vara fallet, men testerna av resterande faktorer visar ändå på intressanta egenskaper hos själva frågan. Vi ser i resultaten att antalet utgångspunkter, väntevärdet på antalet stigar och det totala antalet stigar samtliga påverkar säkerheten och alltså bidrar med information om bilderna var för sig. Vi ser i resultaten av testerna, med distributionen av svarsalternativen betecknade 4H respektive 4L (tabell 3.3), att det går att förändra säkerheten mycket genom att enbart förändra väntevärdet av antalet möjliga stigar. Vi ser även genom resultatet av testet betecknat 4B att en ökning av antalet utgångspunkter till knappt det dubbla nästan vägar upp en minskning av antalet totala stigar till mindre än hälften. Problemet är att dessa faktorer inte går att variera utan att påverka någon annan faktor och i exempelvis fallet med 4H har sannolikheten för ett svarsalternativ ökats till omkring 40% vilket innebär att fler omgångar per inloggning skulle krävas. Ett annat bra exempel är testet med $P = 2$ svarsalternativ med lika fördelning av sannolikhet och antal vägar till båda dessa. Då dubblerar vi både väntevärdet för antalet möjliga vägar och antalet utgångspunkter för varje svarsalternativ och får resultatet att nästan dubbelt så många omgångar krävs. Dock medför även sannolikheten att gissa rätt, att dubbelt så många omgångar per inloggning krävs,

så säkerheten blir totalt sett oförändrad.

För lågkomplexitetsfrågorna finns inget underlag för hur ökande N och M påverkar användarvänligheten och förändring av den enda övriga faktorn, som identifieras i analysen, panelstorleken n ger ingen märkvärd förhöjning av säkerheten. Det som går att se är, precis som analysen i [6] visar, att frågan av andra typen som beror på tre bilder är betydligt bättre än första som bara beror på två. Högsta säkerheten av de tester som utförts, förutom det med orimliga värden på N och M, är alltså andra typen av frågan i sitt standardutförande. Denna kräver 35 inloggningar innan enbart det korrekta lösenordet fås fram. Ett intressant resultat är även att andra typen av frågan med $N = 80$ och $M = 30$ fortfarande har högre säkerhet än högkomplexitetsfrågan med samma parametrar. Detta är något förvånande då motivationen i [6] till de högre värdena på N och M är att lågkomplexitetsfrågorna ska vara mer sårbara mot avlyssningsattacker.

Genom hela undersökningen har vi precis som i [9] enbart simulerat omgångar med korrekta svar medan det i verkligheten, enligt användarstudien, ges felaktiga svar i ungefär 5% av fallen. Detta skulle innebära ett problem för en avlyssningsattack då en mängd av de observationer som gjorts kan vara helt felaktiga. Exempelvis kan då någon av de satser som utgör det booleska uttrycket vara omöjligt att uppfylla med den faktiskt korrekta lösningen. Man kan dock i UBCSAT sätta en gräns för hur många falska klausuler som får finnas i uttrycket och denna gräns går att uppskatta relativt väl då vi kan räkna ut antalet felaktiga svar i en inloggning utifrån antalet omgångar som krävdes innan användaren autentiserades. Samma problem uppstår även av det faktum att vi i lågkomplexitetsfrågorna inte simulerar fall med färre än två respektive tre bilder, detta inträffar dock ännu mer sällan än felaktiga svar.

Ett problem som finns med protokollets uppbyggnad är att man vid varje observerad omgång minskar antalet kombinationer lösenordet kan bestå av. Om man då kan hålla reda på vilka möjligheter som finns kvar kan man efter ett färre antal observerade omgångar utföra en lyckad brute force attack. Även om en SAT-lösare inte håller koll på dessa möjligheter så kan man genom att köra samma booleska uttryck flera gånger få fram några av dem och efter tillräckligt många observerade omgångar blir då sannolikheten stor att den korrekta finns med.

4.2 Slutsats

Det vi kommit fram till i undersökningen om högkomplexitetsfrågan är att, även om många faktorer som påverkar säkerheten identifierades, så går inte säkerheten att öka nämnvärt utan att samtidigt försvåra för användaren. Det vi däremot har visat är att på grund av själva frågans struktur så ges väldigt mycket information om användarens bilder i varje observerad omgång. För lågkomplexitetsfrågorna kunde enbart en faktor utöver antalet bilder hittas och undersökningen visade att den faktorn inte heller hade så stor inverkan på säkerheten. Dock så visade undersökningen att säkerheten för andra typen av frågan ändå är relativt bra i sitt standardutfö-

rande. Detta tyder på att strukturen hos lågkomplexitetsfrågorna är bättre lämpad mot den här typen av attack. Begränsningarna av säkerheten för detta schema ligger alltså förmodligen i själva frågorna och inte i grundprotokollet och genom att ändra eller helt byta fråga borde säkerheten kunna ökas. Med vår undersökning och analyser som grund presenterar vi två förslag på idéer till frågor som kan ge ökad säkerhet med acceptabel användarvänlighet.

För samma parametrar som i lågkomplexitetsfrågorna skulle en tredje typ av fråga kunna formuleras. Undersökningen har visat att andra typen som beror på tre bilder är säkrare än första typen som enbart beror på två. Med detta som grund borde en fråga som beror på fyra eller fem bilder bli ännu säkrare. I fallet med fem skulle frågan kunna vara samma som den i andra typen, men med majoriteten av första, andra, tredje, näst sista och sista av användarens bilder i panelen.

Med liknande uppställning som i högkomplexitetsfrågan skulle man kunna instruera användaren att räkna hur många av dennes bilder som är sammankopplade från den första påträffade bilden, se figur 4.1. Svaret skulle sedan kunna vara i vilket av P stycken intervall antalet ligger i. De logiska samband som kan ställas upp från en sådan fråga blir mer lik de i lågkomplexitetsfrågorna, men beror fortfarande på fler bilder.

Figur 4.1. Illustration av en variant på en ny fråga, där användarens bilder är markerad med grått och de sammankopplade bilderna som användaren ska räkna är markerade lite mörkare.

P_{19}	P_5	P_7	P_{42}	P_{26}	P_{23}	P_{13}	P_{77}	P_{27}	P_{16}	2
P_{54}	P_4	P_{67}	P_{37}	P_{53}	P_{11}	P_{68}	P_{10}	P_{14}	P_{43}	0
P_{62}	P_3	P_{76}	P_{70}	P_{52}	P_{64}	P_{74}	P_{22}	P_{12}	P_{69}	1
P_{57}	P_{73}	P_{39}	P_{65}	P_{49}	P_{18}	P_{63}	P_{75}	P_{56}	P_{78}	1
P_6	P_{34}	P_{79}	P_{17}	P_{36}	P_{38}	P_{44}	P_9	P_{28}	P_{31}	2
P_{35}	P_{48}	P_{41}	P_{71}	P_{45}	P_{32}	P_{60}	P_{21}	P_{66}	P_{24}	3
P_{59}	P_{58}	P_{15}	P_{30}	P_{29}	P_{61}	P_{46}	P_{72}	P_{80}	P_2	0
P_{50}	P_{55}	P_1	P_8	P_{33}	P_{47}	P_{20}	P_{25}	P_{40}	P_{51}	2
1	2	1	0	3	0	3	0	3	1	

Litteraturförteckning

[1] W. Stallings and L. Brown. *Computer security: Principles and practice Pearson education*. Pearson Education, 2008.

[2] M. Jenselius. Studie: Folk använder på tok för enkla lösenord. Artikel hos IDG, 2010.

[3] C. Herley and D. Florencio. How to login from an internet caf'e without worrying about keyloggers. In *Symposium on Usable Privacy and Security CMU*. Microsoft Research, 2006.

[4] Skandinaviska Enskilda Banken AB. Information om seb's digipass, 2010. `www.seb.se/pow/wcp/templates/sebcollection.cfmc.asp?DUID=DUID_623697DCF55C4576C1256DEF00637BE3\&lang=se\&sitekey=seb.se`.

[5] Svenska E-legitimationssystemet, 2010. `http://www.e-legitimation.se`.

[6] D. Weinshall. Cognitive authentication schemes safe against spyware. Hebrew University of Jerusalem, 2006.

[7] D. Weinshall and S. Kirkpatrick. Passwords you'll never forget, but can't recall. In *CHI '04 extended abstracts on Human factors in computing systems*, pages 1399–1402. ACM, 2004.

[8] S. Chiasson, R. Biddle, and P. C. van Oorschot. A second look at the usability of click-based graphical passwords. In *SOUPS '07: Proceedings of the 3rd symposium on Usable privacy and security*, pages 1–12. ACM, 2007.

[9] P. Golle and D. Wagner. Cryptanalysis of a cognitive authentication scheme. In *IEEE Symposium on Security and Privacy*, pages 66–70. IEEE Computer Society, 2007".

[10] Compiled by Viggo Kann. *Algorithms and Complexity*. Pearson Custom Publication, 2007.

[11] H. H. Hoos and T. Stützle. SATLIB: An online resource for research on sat. In *SAT 2000*, pages 283–292. IOS Press, 2000. SATLIB is available online at www.satlib.org.

[12] The International Conferences on Theory and Applications of Satisfiability Testing (SAT). `http://www.satisfiability.org/`.

[13] N.Dershowitz Z. Hanna and A. Nadel. A clause-based heuristic for sat solvers, 2005.

[14] D. A. D. Tompkins and H. H. Hoos. UBCSAT: An implementation and experimentation environment for SLS algorithms for SAT and MAX-SAT. In *Theory and Applications of Satisfiability Testing: Revised Selected Papers of the Seventh International Conference*, volume 3542 of *Lecture Notes in Computer Science*, pages 306–320. Springer Verlag, 2005.

[15] F. Hutter D. A. D. Tompkins and H. H. Hoos. Scaling and probabilistic smoothing: Efficient dynamic local search for sat. In *In Proceedings of the Eighth International Conference on Principles and Practice of Constraint Programming (CP-02)*, volume 2470 of *Lecture Notes in Computer Science*, 2002.