G–Skriverens Kryptologi

Niels Juul Munch, Midtsjællands Gymnasium

Indledning

I den foregående artikel *G–Skriverens Historie* blev det historiske forløb om G–Skriveren beskrevet og set i sammenhæng med Sveriges situation under anden verdenskrig. I denne artikel beskrives G–Skriverens kryptologi og matematik. Hensigten med de to artikler samlet er at give lærere og SRP–skrivende elever et udgangspunkt for at arbejde med G–Skriveren som et emne i matematik–historie.

Teksten forudsætter ikke nogen kryptologisk forhåndsviden hos læseren. Tværtimod sigtes på at give netop en uindviet læser nogle indtryk af matematikkens brug i kryptologi og kryptoanalyse.

En detaljeret og fuldstændig gennemgang af G–Skriverens svagheder og af operativt realistiske angreb og deres matematik vil blive meget omfattende og har ikke været hensigten. Teksten indeholder først en beskrivelse af maskinens kryptologi med introduktion af matematiske symboler og funktioner. Derefter anvendes denne beskrivelse til at give et eksempel på matematisk kryptoanalyse.

Teleprintere, bits og karakterer

For at forstå G–Skriverens virkemåde er det nødvendigt at vide noget om teleprintere og deres teknik.

Produktet SFM T52 fra firmaet Siemens & Halske var en synkron telex–maskine med en integreret kryptering. Det synkrone betød, at to maskiner og to operatører måtte være aktive samtidig for at kommunikationen kunne finde sted. Hver operatør skrev på sin egen maskine den tekst, som efterfølgende skulle vise sig som modtaget tekst på den anden maskine. En forbindelse begyndte med en opringning fra den part, som ønskede at sende noget til den anden, men kommunikationen foregik derefter i princippet som en dialog.

En helt grundlæggende egenskab ved teleprintere er, at de sender og modtager informationer i grupper af 5 pulser. Hver puls kan være høj eller lav og er derfor matematisk set en *bit*. Før og efter hver gruppe af fem bits sendes desuden en særlig start– og stop–puls, som hjælper til at fastholde synkroniseringen mellem de to parter.

På denne måde kan teleprintere sende og modtage information i enheder af 5 bits og følgelig med $2^5 = 32$ individuelle muligheder for hver transmitteret enhed. Disse 32 muligheder fortolkes naturligt som tallene fra 0 til 31, hvor de fem bits 10011 her læses som 16 + 2 + 1, altså som decimaltallet ”19”. Den rå, transmitterede bitsekvens opfattes på denne måde som en sekvens af symboler i alfabetet {0, .. , 31}.

Disse 32 transmitteringsnære symboler må oversættes til et sæt af karakterer, som operatørerne kan indtaste og kan modtage. For at have et passende fyldigt karaktersæt benytter teleprintere sig af en dobbeltbrug, således at hvert af de 32 symboler kan anvendes både i en bogstav–mode, *Letter Mode*, og i en tal–mode, *Figure Mode*. De anvendte karakterer i hver af de to tilstande er vist i figur 1.

Eksempelvis skal de 5 bits 6 = 00110, fortolkes som et ”I”, hvis de sendes i bogstav–mode, og som et ”8”, hvis de sendes i tal–mode. Denne konstruktion kaldes Baudot–koden efter franskmanden Emile Baudot, som opfandt den omkring 1870.

Dobbeltbrugen udvider mængden af brugbare karakterer betragteligt, men betyder samtidigt, at de to maskiner altid – for hver eneste karakter – skal være enige om, hvorvidt de nu er i *letter mode* eller i *figure mode*. For selv at komme i *letter mode* kan den skrivende operatør trykke ”LS”, *LetterShift*. Dette betyder samtidig, at symbolet LS = 11111 bliver sendt til den modtagende maskine, hvor det fungerer som et kommando om også dér at gå i *letter mode*. Denne mode ved-

Decimal	Binær	Letter Mode	Figure Mode
0	00000	(ubrugt)	(ubrugt)
1	00001	E	3
2	00010	LF	LF
3	00011	A	-
4	00100	SP	SP
5	00101	S	BEL
6	00110	I	8
7	00111	U	7
8	01000	CR	CR
9	01001	D	$
10	01010	R	’
11	01011	J	4
12	01100	N	´
13	01101	F	!
14	01110	C	:
15	01111	K	(
16	10000	T	5
17	10001	Z	”
18	10010	L	)
19	10011	W	2
20	10100	H	#
21	10101	Y	6
22	10110	P	0
23	10111	Q	1
24	11000	O	9
25	11001	B	?
26	11010	G	&
27	11011	FS	FS
28	11100	M	.
29	11101	X	/
30	11110	V	;
31	11111	LS	LS

Figur 1
Baudot–koden.

varer hos modtageren indtil det alternative symbol ”FS”, *FigureShift*, er blevet sendt og modtaget.

Af øvrige specialkarakterer i tabellen kan nævnes ”LF” *LineFeed*, ”SP” *Space* og ”CR” *CarriageReturn*. Symbolet ”BEL” er vist selvforklarende.

Krypto–princip

De grundlæggende principper i G–Skriverens kryptologi kan føres tilbage til en patentansøgning, som i juli 1930 blev indgivet i Tyskland af firmaet Siemens & Halske AG. Senere blev der tildelt pa-

tenter også i eksempelvis USA, og produktionen og videreudviklingen af T52–serien fortsatte frem til 1944.

Det drejer sig principielt om en forsats til noget, som uden kryptering vil være en standard (klar–tekst) teleprinter. Umiddelbart før afsendelse af et 5–bits symbol P, laves P om til et andet 5–bits symbol C. Når C derefter er modtaget i den anden ende, er det første som sker, at modtagermaskinen laver C tilbage til P. Alt andet i de to maskiner er helt uberørt af, at der foregår en kryptering lige før afsendelsen og en dekryptering lige efter modtagelsen.

Selve krypteringen, ændringen af P til C, kan kompakt skrives som $C = E_{A,B}(P)$. Den foregår i to krypteringstrin og bruger i alt 10 bits hemmelig nøgle til at kryptere de 5 bit i P. Først anvendes 5 bits nøgle A til at omdanne P til $D = P \oplus A$ og derefter anvendes 5 andre nøglebits B til at omdanne D til $C = \pi_B(D)$. Vi vil se nærmere på hvert af disse skridt.

Binær addition og egenskaber ved D

Regneoperationen $\oplus$ er *koordinatvis binær addition*. Den binære addition $0 \oplus 0 = 0$, $1 \oplus 0 = 1$, $0 \oplus 1 = 1$, $1 \oplus 1 = 0$ udvides til addition af 5–bits størrelser fx således, at

$$01011 \oplus 11001 = 10010$$

Her kan man tænke på 01011 som klarteksten P og på 11001 som nøglen A. Højre siden 10010 er i så fald størrelsen D, der fungerer som en slags *midlertidig cifferkarakter*.

Hvis nøglen A er valgt hemmeligt og tilfældigt blandt de 32 muligheder på en måde, som er uafhængig af P, er allerede D sikker mod en hypotetisk angriber, uden at det egentlig er nødvendigt efterfølgende at anvende B. For at se sikkerheden i $D = P \oplus A$ kan man gå frem på følgende måde.

Først kan det udregnes, at sandsynlighedsfordelingen for D er flad, $Prob(D = d) = \frac{1}{32}$ for ethvert d. Alle værdier d for D er altså lige sandsynlige, og dette er tilfældet helt uanset den indledende fordeling af klarteksten P. Dernæst kan vi for hver mulighed $P = p$ og $D = d$ udnytte denne ligefordeling af D til at verificere, at

$$Prob(P = p \mid D = d) = Prob(P = p)$$

Selv en angriber, som ved alt om, hvordan P og A kombineres for at danne D, må således indse, at hans betingede sandsynlighedsfordeling for P efter en observation af $D = d$, er identisk med hans sandsynlighedsfordeling inden observationen. Det er det samme som at sige, at D ikke afgiver information om P.

Transpositionen π

Det er heldigt, at D er sikker, fordi den anden halvdel af krypteringen – π_B – er bestemt ikke sikker. Og den perfekte ligefordeling af D over de 32 muligheder for $D = d$ er ironisk nok en stor hjælp i en angribers analyse af π_B.

De 5 bits i nøglen B anvendes til at lave D om til $C = \pi_B(D)$ ved hjælp af en transposition π_B, som flytter om på bittene i D. Vi kan beskrive virkningen af π_B på følgende måde. Lad de 5 bits i B være $B = b_4 b_3 b_2 b_1 b_0$, lad $X = x_4 x_3 x_2 x_1 x_0$ være et input til π_B og lad $Y = \pi_B(X)$ være det tilsvarende output med bits $Y = y_4 y_3 y_2 y_1 y_0$. Vi starter med at sætte $Y = X$, hvilket vil sige, at hver bit x_i fra X kopieres over i den tilsvarende bit y_i i Y. Derefter sker der følgende 5 skridt i den anførte rækkefølge:

Hvis $b_0 = 1$, byt om på y_0 og y_4
Hvis $b_1 = 1$, byt om på y_4 og y_3
Hvis $b_2 = 1$, byt om på y_3 og y_2
Hvis $b_3 = 1$, byt om på y_2 og y_1
Hvis $b_4 = 1$, byt om på y_1 og y_0

Slutværdien for y–bittene danner den ønskede outputværdi Y.

Det er let at se, at outputværdien Y afslører en del om inputværdien X, eller i vores sammenhæng, at C fortæller en del om D. Svagheden er grundlæggende, at det er de samme bits som ligger i C og i D. Der vil være lige mange 0–bit og 1–bit i den ukendte D, som der er i den observerede C. Dette er information for en angriber.

Fordeling af D givet C

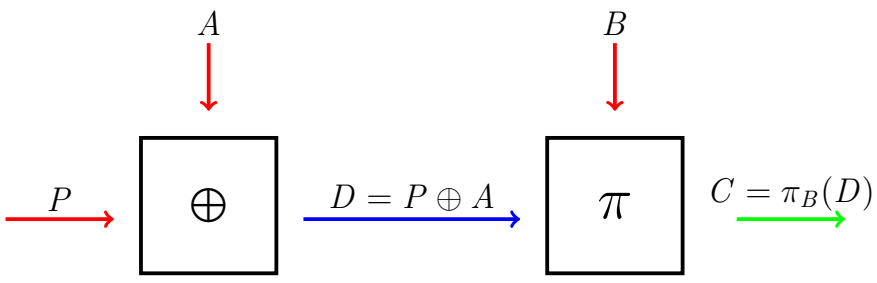
Svagheden i π_B er mest eklatant, hvis angriberen observerer eksempelvis $C = 31$. Da må også $D = 31$, fordi 31 simpelthen er den eneste mulighed for et 5–bits tal med fem 1–bits og ingen 0–bits.

Vi vil se nærmere på et andet tilfælde. Hvis angriberen observerer værdien $C = 2$, er der $\binom{5}{1} = 5$ muligheder for D med netop én 1–bit ud af 5, og der viser sig ved en undersøgelse at være følgende muligheder for parret (D, B)

D	B	Antal B
1	15, 16, 18, 20, 22, 24, 26, 28, 30	9
2	0, 1, 2, 3, 4, 5, 6, 7	8
4	8, 9, 10, 11	4
8	12, 13	2
16	14, 17, 19, 21, 23, 25, 27, 29, 31	9

Figur 3
Mulige værdier for D og B svarende til C = 2.

Disse 32 løsninger kan findes ved en simpel computerbaseret efterprøvning af de mulige værdier for D og for B. Eller ved at anvende de fem skridt i beskrivelsen af π_B som udgangspunkt for først at lave en tilsvarende beskrivelse af den inverse



Figur 2
Skitse af krypteringen $C = E_{A,B}(P)$. Alle variable er 5–bits størrelser med værdier mellem 0 og 31.

transposition π_B^{-1}, og derefter at generere de 32 værdier $\pi_B^{-1}(2)$, når B varierer.

Vi får derfor, at i fravær af anden information om D, vil en observation af C = 2 umiddelbart fortælle os, at D må være en af 1, 2, 4, 8 og 16, og at sandsynligheden for hver af de fem muligheder for D efter observationen af $C = 2$ nu er henholdsvis $\frac{9}{32}$, $\frac{8}{32}$, $\frac{4}{32}$, $\frac{2}{32}$ og $\frac{9}{32}$. Bemærk at argumentet udnytter, at både D og B er ligefordelte.

For at illustrere den potentielle nytte af at kende fordelingen af D kan vi hypotetisk forestille os, at en angriber efter en observation af $C = 2$ gennemløber mulighederne for værdierne af $D = d$, og at angriberen for hver d–værdi ved en videre undersøgelse er i stand til at finde ud af, om netop denne d var den rigtige. Hvor mange d–er skal en angriber indstille sig på at undersøge?

Uden information om D kan angriberen kun gætte tilfældigt og må regne med i middel at gætte og undersøge 16,5 gange, før den rigtige d findes. Hvis den første d er den rigtige gættes 1 gang, hvis den sidste er den rigtige gættes 32 gange, og alle værdier derimellem er lige sandsynlige. Efter observationen af $C = 2$ er der færre mulige værdier for d, og man kan endda gætte på de mest sandsynlige værdier af d først. I middel kan angriberen derfor nøjes med at gætte

$$1\cdot\frac{9}{32}+2\cdot\frac{9}{32}+3\cdot\frac{8}{32}+4\cdot\frac{4}{32}+5\cdot\frac{2}{32}=2{,}4$$

gange. Det begrænsede antal muligheder nedsætter middeltallet fra 16,5 til 3, og den skæve sandsynlighedsfordeling nedsætter middeltallet yderligere fra 3 til 2,4.

G–Skriveren som kryptosystem

En besked bestående af en strøm af klartekstsymboler $P^0, P^1, \ldots, P^i, \ldots$ krypteres i G–Skriveren til en tilsvarende strøm af ciffersymboler $C^0, C^1, \ldots, C^i, \ldots$ ved

$$C^i = E_{A^i,B^i}(P^i)$$

Problemet for konstruktørerne af maskinen var, hvor alle disse uafhængige og tilfældigt fordelte nøgler A^i og B^i skulle komme fra. Løsningen var at lave et system, en *pseudorandom generator*, til at danne de tilfældige nøgler A^i og B^i. Disse nøgler ville ikke være virkeligt tilfældige, men tanken var, at med en passende kompliceret metode til at frembringe nøglerne, ville de i praksis være lige så gode som virkeligt tilfældige værdier.

Alle maskiner på det tidspunkt benyttede sig af *hjul*, som man havde den elektromekaniske teknik til at implementere. I dette tilfælde blev hver maskine udstyret med et sæt på 10 hjul. Hvert hjul havde langs kanten en række markeringer der kunne aflæses som bits. For hver karakter, der skulle krypteres, blev der aflæst en bit fra hvert hjul, og de 10 aflæste bits blev brugt til A^i og B^i. Derefter tog hvert hjul et skridt fremad og proceduren gentog sig med næste klartekst–karakter.

Hjulene var af bakelit, og næsten alle eksemplarer af maskinen havde de samme 10 hjul med de samme bits. Kun maskiner med identiske hjul kunne kommunikere med hinanden, og det overvældende flertal af maskiner tilhørte samme serie. Hjullængderne var 47, 53, 59, 61, 64, 65, 67, 69, 71 og 73. Det betød eksempelvis, at hjulet af længde 47 efter 47 skridt ville være drejet en hel omgang. Dette hjul ville i skridt 47 og derefter afgive de samme bits som i skridtene fra karakter 0 og frem.

Lad os betegne bittene på hjul nummer 1 med hjullængde 47 i matematiske symboler med H_1^j, $0 \le j \le 46$. Tilsvarende lad os indføre h_1^i som et symbol for den bit, som til karakter nummer i udlæses fra hjul nummer 1, og lad os sige, at hjul 1 starter i positionen s_1. Da er $h_1^i = H_1^{\mathrm{mod}(i+s_1,47)}$. Tilsvarende for de øvrige hjul og hjullængder.

G–Skriverens nøgler

Nøglen til systemet bestod i to ting. Den første del var de 10 startindstillinger af de 10 hjul, som vi samlet kan beskrive med

$$S = (s_1, s_2, \ldots, s_{10})$$

Den anden del af nøglen var den måde, de 10 bits fra hjulene blev brugt på i A^i og i B^i.

Hver af de fem bits i A^i og hver af de fem bits i B^i blev dannet ved et udtag fra et bestemt af de i alt 10 hjul. Systemet er illustreret i figur 4. De hjul, som anvendes til A^i, er givet ved en delmængde $\mathcal{A}$ af {1, .., 10}, og rækkefølgen af de 5 bitspor fra $\mathcal{A}$ ved brug i A^i er angivet ved en permutation α af {0, .., 4}. Tilsvarende beskrives brugen af hjulene til B^i ved en mængde $\mathcal{B}$ og en permutation β. I det konkrete tilfælde i figur 4 bliver de fem bits i A^i bestemt ved $a_4^i = h_2^i$, $a_3^i = h_5^i$, $a_2^i = h_{10}^i$, $a_1^i = h_1^i$, $a_0^i = h_6^i$.

I ord kan vi sammenfatte, at nøglematerialet er: S, $\mathcal{A}$, $\mathcal{B}$, α, β. Værdierne $\mathcal{A}$, $\mathcal{B}$, α, β forblev uændrede under en kommunikation, når de først var sat ved kommunikationens begyndelse.

Designerne af maskinen har forladt sig på, at der er nogle store tal associeret med denne maskine. Antallet af mulige startindstillinger S for hjulene er et voldsomt stort tal. Da de 10 hjullængder er indbyrdes primiske, vil antallet af skridt før udtagspunkterne

$$U^i = (\mathrm{mod}(i+s_1,47), \mathrm{mod}(i+s_2,53), \ldots \\ \ldots, \mathrm{mod}(i+s_{10},73))$$

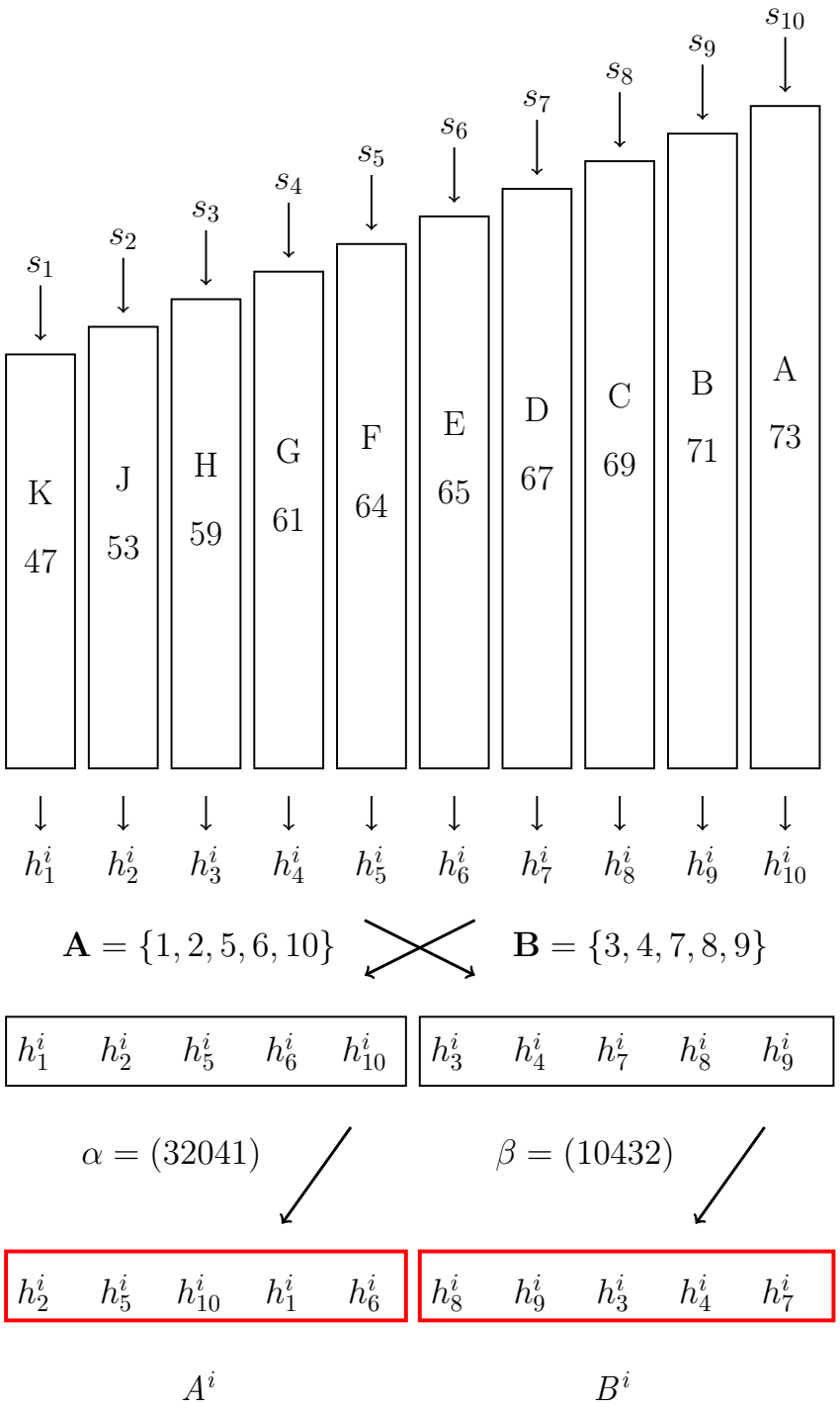
gentager sig, og nøglerne A^i og B^i dermed begynder "forfra", også være meget stort. Og endelig er der et stort antal af forskellige måder, de 10 bits kan benyttes på til at danne A^i og B^i.

I en før–computertid, hvor kryptoangreb i udgangspunktet måtte foregå ved håndkraft, har opfinderne haft god tillid til maskinens nøglestørrelse og kompleksitet.

G–Skriverens nøgleadministration

I sin krypto–administration havde man opdelt nøglen i tre dele. Der var *den indre indstilling*, der var en hemmelig nøgle, som bestod i $\mathcal{A}$, $\mathcal{B}$, α og β. Den indre indstilling bestemte hvilke udtag fra hjulene, som skulle ligge på de forskellige bitpositioner i A^i og i B^i. Hver værdi var gyldig i 3–9 dage.

Figur 4
Skitse af frembringelsen af A^i og B^i for valgte illustrerende værdier af $\mathcal{A}$ og $\mathcal{B}$, α og β. Bemærk at bittene i A^i og B^i nummereres fra 0 til 4 med 0 til højre. Den viste navngivning af de 10 hjul, med "I" udeladt, er svenskernes originale.



Der var *den daglige nøgle*, kaldet *QEK*. Den daglige nøgle var en hemmelig nøgle, som bestod i to ting. Dels en opdeling af de 10 hjul i 5 hjul til såkaldt *QEK*–brug og 5 hjul til såkaldt *QEP*–brug. Dels et valg af startindstillinger for de 5 *QEK*–hjul. For alle krypteringer som fandt sted fra den pågældende dag klokken 09:00 indtil næste morgen ville disse 5 hjul være sat til den pågældende (hemmelige) startindstilling.

Den indre indstilling og den daglige nøgle blev bestemt centralt og lange lister med værdier og gyldighedsperioder blev fordelt på papir til alle anvendere af G–Skriveren. Disse papirer var naturligvis højt klassificerede og meget blev gjort for at fordele og opbevare dem på en sikker måde.

Den sidste nøgle var *meddelelsesnøglen*, kaldet *QEP*. Meddelelsesnøglen var et valg af startindstillinger på de fem *QEP*–hjul. Disse værdier blev bestemt af opringeren og sendt med selve teksten i den kommunikation, som foregik før man skiftede til krypteret mode. De var åbne, det vil sige "ikke hemmelige", og enhver som indhentede kryptoteksten kunne aflæse de fem tal.

Selv om *QEP*–nøglen var åben, skulle tallene være tilfældige. Deres formål var at sikre, at hver krypteret session havde sin helt egen indstilling af maskinen.

Kryptologiske svagheder

Én åbenlys svaghed ved G–skriverens kryptologiske design var, at det ikke var muligt at ændre bittene H_j^i på nogen af de 10 hjul. Alle maskiner havde de samme hjul, og hvis en angriber bare én gang fik fysisk adgang til en maskine eller fandt værdierne ved en analyse, så kunne den viden genbruges ubegrænset i tid og for alle G–Skrivere.

En anden klar svaghed var, at stepningen af hjulene var regulær og forudsigelig, et skridt for hver karakter. Et bedre design ville have steppet hvert hjul et variabelt antal skridt, måske et eller to, for hver karakter og kunne have ladet hvert hjul styre af de øvrige hjul.

Men selv disse svagheder ville næppe i sig selv have givet svenskerne den praktiske succes, som de faktisk fik. De svenske angribere blev hjulpet på vej af et indgående kendskab til særheder i den (skjulte) klartekst. Og af en virkelig spektakulær fejl i tyskernes brug af maskinen.

Baggrunden for begge svagheder var, at kommunikationslinierne var voldsomt udsatte for støj.

En krypto–karakter kunne simpelthen forsvinde undervejs. Det medførte at synkroniseringen mellem afsender og modtager gik tabt, og at modtagerens klartekst var ulæselig fra synkroniseringstabet og fremad. Tilsvarende kunne en krypto–karakter blive forvansket undervejs og blive modtaget med en forkert værdi. Dette ville som oftest blot medføre, at den ene tilsvarende klartekst–karakter i den modtagne tekst blev fremvist forkert. Men hvis den pågældende krypto–karakter dækkede over en af de "usynlige" klartekst–karakter $P = FS$ eller $P = LS$, kunne den modtagende teleprinter finde på at fortolke længere tekst passager i den forkerte mode til stor gene for modtageren.

For at begrænse virkningen af den sidstnævnte type fejl havde operatørerne vænnet sig til i stedet for *SP* (mellemrum) mellem ord, systematisk at skrive de to karakterer *LS SP*. Det ekstra *LetterShift* var usynligt for læseren på modtagersiden, men sikrede, at den modtagende maskine i hvert fald fra det punkt af var i den rigtige mode. Som resultat af dette havde klarteksten en voldsom overhyppighed af kombinationen *LS SP*.

På trods af denne type forholdsregler måtte operatørerne meget ofte begynde om igen på en given transmission. Og dette er oplægget til den helt store fejl.

Maskinen var udstyret med en *nulstillingsfunktion*, som med et enkelt greb satte hvert af de 10 hjul til en given startposition. Hensigten var, at denne funktion skulle anvendes til hurtigt og sikkert at sætte *QEK*–hjulene til dagens værdier, medens værdierne for *QEP*–hjulene derefter skulle ændres tilfældigt af operatøren. Det var ikke forudset, at operatørerne, som skulle sende mange tekster og skulle gensende samme tekst mange gange før det lykkedes, ville udvikle den praksis at begynde en kommunikation ved at anvende nulstillingsfunktionen **uden** derefter at ændre på startværdierne for *QEP*–hjulene.

Resultatet var et meget stort genbrug af indstillinger af G–Skriveren. Det var ikke uset, at svenskerne en given dag kunne arbejde med en dybde på 20–40 forskellige transmissioner krypteret med samme indstilling, såkaldte *paralleller*.

Beurlings opklaring

Da Beurling fik opgaven med at finde ud af hvordan G–Skriveren fungerede eller kunne løses, havde han dette at arbejde med: En viden om, at kryptoapparater på den tid ofte benyttede sig af binær addition og af bit–permutationer. En overordnet viden om, hvordan klarteksterne nok så ud. Et stort sæt af ciffertekster, som ud fra deres tidspunkter, længder og visse karakteristiska i teksternes indledning måtte være parallelle. Blyant og papir. På dette grundlag opbyggede han på få dage en logisk model af maskinen.

Beurling har aldrig selv redegjort for sin arbejdsproces. Men det er almindeligt antaget, at det er foregået nogenlunde på følgende måde.

Først har Beurling skrevet teksterne i parallelsættet op i lange rækker med én tekst i hver række og med karakterer på samme plads i teksten under hinanden og han har ledt efter alt, som kunne virke påfaldende.

Med baggrund i tyskernes tilbøjelighed til at indsætte *LS SP* i klarteksten, var en mulig indledende ide, at de mange forekomster af netop dette bigram i klarteksten kunne have givet sig udtryk på en eller anden måde i ciffertekserne. På et tidspunkt har han observeret, at der var påfaldende "anti"–gentagelser af ciffersymboler, forstået på den måde, at cifferkarakterer på samme plads, men i forskellige tekster, havde alle bit forskellige på nær en. Dette er illustreret i figur 5.

Beurling gættede inspireret på, at disse tilfælde dækkede over forskudte forekomster af klarteksten *LS SP*, og at krypteringen foregik ved for et givet "i" at anvende ét bestemt A til binær addition og én bestemt bit–transposition π for alle tekster i hele parallelsættet. Med værdierne $LS = 31$ og $SP = 4$ er der netop én bit med samme værdi i de to klartekst–bogstaver. Hvis de to værdier 31 og 4 først binært adderes med samme værdi A og derefter udsættes for samme bit–transposition π vil resultaterne altid være to værdier, som stadig har netop en bit tilfælles.

Ved at gætte mere og mere klartekst i de forskellige tekster kunne han gradvist både bekræfte sin hypotese og bestemme flere og flere værdier af A og π. Efterhånden blev det klart, at visse elementer i A og π efterhånden gentog sig, og opklaringen af maskinens øvrige dele kunne begynde. Historien er fortalt i en længere og mere detaljeret form i Beckmans bog.

Produktionen

Den løbende løsning af G-Skriver trafikken var naturligvis langt enklere end den første opklaring. Med forhåndskendskab til maskinens opbygning og hjulenes indhold, manglede kun den indre indstilling og den daglige nøgle. Begge dele blev dagligt fundet med udnyttelse af den indgående viden om klartekstens særheder og med fordelen af hyppige parallelsæt. Ofte kunne dagsnøglen være klar allerede om morgenen.

Den indre indstilling og den daglige nøgle blev anvendt til at indstille en efterligning af G–Skriveren, som blev fremstillet på L. M. Ericsson Workshop For Precision Mechanics. Med denne kunne svenskerne udnytte deres fundne nøgler og dekryptere med samme lethed og hurtighed som tyskerne selv. Ledningerne i figur 6 bruges til at forbinde udtagene fra hjulene med enkeltbits i A^i og i B^i, det vil i vores notation sige, at ledningerne fastlægger $\mathcal{A}$, $\mathcal{B}$, α og β.

Ciffer4	...	...	7	11	22	0	17	20	11	3	5	11	6	12	...	...	
Klartekst4	...	...	LS	SP	H	E	U	T	E	LS	SP	...	...	...	...	...	
Ciffer3	...	12	25	4	17	11	29	26	21	10	30	10	20	4	11	18	28
Klartekst3	...	LS	SP	G	E	H	E	I	M	E	LS	SP	K	O	M	M	...
Ciffer2	...	...	7	11	18	...											
Klartekst2	...	...	LS	SP	...	...											
Ciffer1	...	12	25	30	13	8	18	24	31								
Klartekst1	...	LS	SP	D	A	S	LS	SP	...								

Figur 5
Beurlings arbejdsproces i miniatureformat. Ciffertekst med grønt og klartekst med rødt.

Kryptoanalyse

En kryptologisk beskyttelse med kendt algoritme kan som regel brydes af en angriber med ubegrænsede beregningsmuligheder.

Principielt kan angriberen forsøge sig frem med alle nøglemuligheder, for hver af dem generere en tilhørende klartekst og så udvælge den rigtige. Hvis han altså er i stand til at kende den rigtige klartekst, når den er der, og adskille den fra de øvrige forslag. Spørgsmålet er altså ikke, om der findes et angreb på kryptologien, men hvor *effektivt* et angreb, det er muligt at designe.

Denne simple *brute force* metode er dog normalt uigennemførlig i praksis på grund af det store antal muligheder, der skal undersøges. Søgningen bliver aldrig færdig. En angriber vil derfor lede efter kryptologiske svagheder i det håb, at et mere avanceret angreb måske kan gennemføres inden for en brugbar tidshorisont. Forskellige fundne svagheder kan give anledning til forskellige typer af kryptoangreb.

Efter Beurlings opklaring af G–Skriverens virkemåde angreb svenskerne maskinen ved brug af paralleller, som tyskerne uheldigt kom til at stille til rådighed. Desuden kan nævnes, at G–Skriveren også er svag overfor *isologer*, det vil sige to eller flere ciffertekster, som anvender forskellig nøgle, men har samme klartekst. Og at et sæt af *parallelle quasi–isologer*, hvor to tekster har samme nøgle og identiske, men forskudte klartekster, er et taknemmeligt angrebsmål.

De afsluttende afsnit af denne artikel diskuterer noget af matematikken i et angreb på G–Skriveren ved *kendt klartekst*. Det klassiske "kendt klartekst"–angreb består i at udnytte, at en modstander krypterer og sender en klartekst, som man selv kender, til at finde de anvendte nøgler. Nøglerne kan så forhåbentlig genbruges på andre ciffertekster med ukendt indhold.

Udgangspunktet for analysen er, at vi som angribere kender en klartekst og den tilsvarende kryptotekst, og at vi kender alt til maskinens funktion. Målet er at finde en metode til derudfra at identificere alle maskinens hemmelige dele, det vil sige tekstens nøgleelementer $\mathcal{A}$, $\mathcal{B}$, α, β samt alle bittene H_j^i på de 10 hjul. Startværdierne S kan vi sætte lig 0, da vi kun har én tekst og ingen givne *QEP*–værdier.

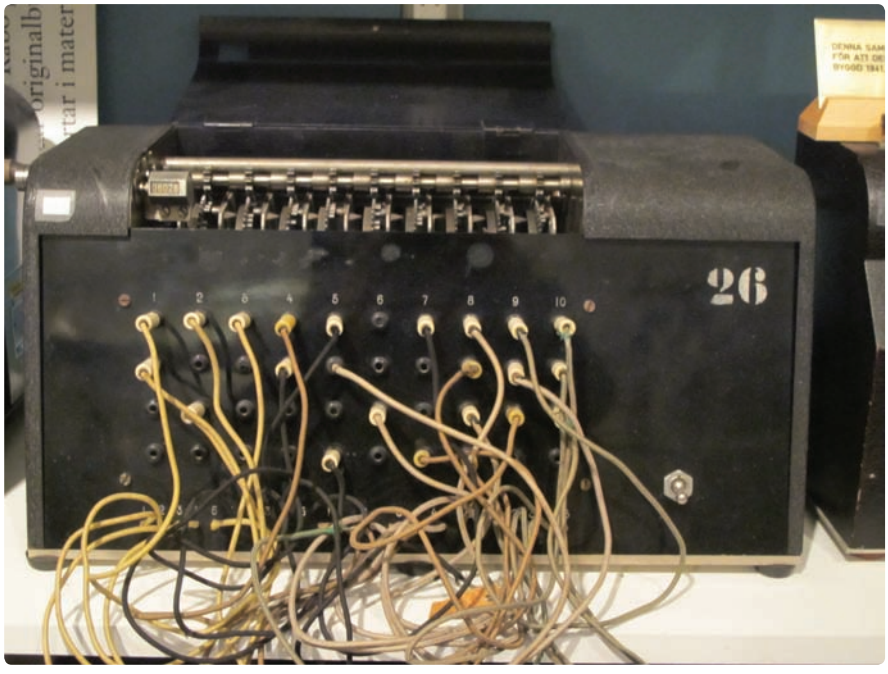
I den virkelige verden vil opdelingen af startindstillingerne S i en lukket del *QEK* og en åben del *QEP* blive kraftigt udnyttet. For ikke at komplicere fremstillingen og for at holde den rimeligt kort, ser vi dog bort fra al information, som kan udledes af nøgleadministrationen, og vi nøjes desuden med at redegøre for, hvordan et kendt klartekst–angreb kan finde $\mathcal{A}$, α og indholdet af de fem hjul hørende til $\mathcal{A}$.

Paritetsbevarelse

Angrebet starter med den observation, at hvis $X = x_4x_3x_2x_1x_0$ og $Y = y_4y_3y_2y_1y_0$ er sammenknyttede ved $Y = \pi_B(X)$, så gælder

$$y_4+y_3+y_2+y_1+y_0=x_4+x_3+x_2+x_1+x_0 \quad (1)$$

uanset værdien af B. Når x og y kun kan antage værdierne 0 og 1, tæller begge sider i ligningen (1) antallet af 1–bits og dette antal er det samme i Y som i X. Ved at reducere modulo 2, og kun se på



Figur 6
En af de originale APP–er, anvendt til dekryptering af G–Skriveren. Foto: FRA.

pariteten af antallet af 1–bits, fremkommer ligningen

$$y_4 \oplus y_3 \oplus y_2 \oplus y_1 \oplus y_0 = x_4 \oplus x_3 \oplus x_2 \oplus x_1 \oplus x_0$$

som er en binær ligning i x–erne og y–erne. I det i–te skridt af krypteringen anvendes denne ligning for $Y = C^i$ og $X = D^i = P^i \oplus A^i$. Dette fører til

$$\begin{aligned} & c_4^i \oplus c_3^i \oplus c_2^i \oplus c_1^i \oplus c_0^i \\ &= (p_4^i \oplus a_4^i) \oplus (p_3^i \oplus a_3^i) \oplus (p_2^i \oplus a_2^i) \\ &\quad \oplus (p_1^i \oplus a_1^i) \oplus (p_0^i \oplus a_0^i) \end{aligned}$$

som vi straks omorganiserer til

$$\begin{aligned} & a_4^i \oplus a_3^i \oplus a_2^i \oplus a_1^i \oplus a_0^i \\ &= (c_4^i \oplus c_3^i \oplus c_2^i \oplus c_1^i \oplus c_0^i) \\ &\quad \oplus (p_4^i \oplus p_3^i \oplus p_2^i \oplus p_1^i \oplus p_0^i) \end{aligned}$$

Pointen er at højresiden, paritetssummen

$$\begin{aligned} z^i = & (c_4^i \oplus c_3^i \oplus c_2^i \oplus c_1^i \oplus c_0^i) \\ & \oplus (p_4^i \oplus p_3^i \oplus p_2^i \oplus p_1^i \oplus p_0^i) \end{aligned}$$

af bittene i C^i og i P^i, er noget vi konkret kan beregne, når vi kender både klartekst og ciffertekst. Vi kender dermed for hvert i den binære sum z^i af alle de bits h_j^i, som indgår i nøglen A^i.

At finde $\mathcal{A}$ og første halvdel af H – binære ligninger

Hvis vi kendte $\mathcal{A}$, kunne vi omdanne de binære ligninger

$$a_4^i \oplus a_3^i \oplus a_2^i \oplus a_1^i \oplus a_0^i = z^i \qquad (2)$$

til noget umiddelbart nyttigt. Som eksempel sæt igen $\mathcal{A} = \{1,2,5,6,10\}$, hvilket betyder, at hjullængderne brugt til A^i er 47, 53, 64, 65 og 73.

Hvis eksempelvis a_4^i er den bit, som kommer fra hjul nummer 1, er $a_4^i = h_1^i = H_1^{\mathrm{mod}(i,\,47)}$ således at eksempelvis $a_4^{100} = H_1^6$, $a_4^{101} = H_1^7$, $a_4^{102} = H_1^8$ og så videre. Tilsvarende vil de øvrige hjul 2, 5, 6 og 10 fra $\mathcal{A}$ hver bidrage med en H–bit fra deres hjul til en af a_3^i, a_2^i, a_1^i og a_0^i, og vi kan bestemme det nummer, hver af disse ukendte bit har på sit hjul.

Da vi mangler permutationen α, kender vi ikke den rækkefølge, hvori disse fem hjul fra $\mathcal{A}$ bidrager til summen z^i, men det forhindrer os ikke i at danne den korrekte sum i (2) og i at danne ligningerne

$$\begin{aligned} & H_1^{\mathrm{mod}(i,\,47)} \oplus H_2^{\mathrm{mod}(i,\,53)} \oplus H_5^{\mathrm{mod}(i,\,64)} \\ & \oplus H_6^{\mathrm{mod}(i,\,65)} \oplus H_{10}^{\mathrm{mod}(i,\,73)} = z^i \end{aligned} \qquad (3)$$

Hvert i giver os én sådan ligning, og hvis vi har tilstrækkeligt mange ligninger, kan vi forvente ved lineær ligningsløsning af binære ligninger at finde alle bits fra de fem anvendte hjul.

En angrebsmetode er derfor: Prøv alle muligheder for $\mathcal{A}$ en ad gangen, og for hver mulighed opstil de binære ligninger svarende til (3) og prøv at løse dem. Dér, hvor der er en løsning, har vi fundet $\mathcal{A}$ og indholdet i alle anvendte hjul. Der er $\binom{10}{5} = 252$ muligheder for mængden $\mathcal{A}$, og det samlede antal ukendte bits i 5 hjul er maksimalt 345. Denne metode kræver således, at der er lidt flere end 345 kendte klartekstkarakterer til rådighed. Beregningsmæssigt skal vi opstille og forsøge at løse 252 binære ligningssystemer i op til 345 variable. Dette kan i dag afvikles på en tid i størrelsesordenen et enkelt sekund på en standard PC.

At finde $\mathcal{A}$ og første halvdel af H – binære polynomier

En svaghed ved ovenstående metode er, at den ikke udnytter den cykliske struktur i h–bittene. Ved at anvende lidt mere matematik og lidt flere karakterer kan vi modellere også denne side af maskinen. Belønningen bliver, at vi kan anvise konkrete formler for indholdet i de ukendte H_j^i.

At finde $\mathcal{A}$

For at finde $\mathcal{A}$ ønsker vi først at gange det binære polynomium $Q_A(x)$ defineret som produktet

$$(1+x^{47})\,(1+x^{53})\,(1+x^{64})\,(1+x^{65})\,(1+x^{73})$$

på begge sider af (3) og konstatere, at resultatet bliver 0. Det kræver en forklaring.

Lad notationen (e_i) stå for en vilkårlig sekvens af bits, som er defineret for $i \geq 0$. Vi definerer multiplikationerne $1 \cdot (e_i) = (e_i)$, $x \cdot (e_i) = (e_{i+1})$ og udvider ved sammensætning, således at eksempelvis $(1+x) \cdot (e_i) = (e_i \oplus e_{i+1})$, $x^2 \cdot (e_i) = x \cdot (x \cdot (e_i)) = (e_{i+2})$ og $x^n \cdot (e_i) = (e_{i+n})$. Dermed kan vi danne produktet $Q(x) \cdot (e_i)$ af ethvert binært polynomium $Q(x)$ med enhver ensidig bitsekvens (e_i), og resultatet vil være en ny ensidig bitsekvens. Man kan efterprøve, at konstruktionen opfylder

$$(R(x)S(x)) \cdot (e_i) = R(x) \cdot (S(x) \cdot (e_i)) \qquad (4)$$

hvor $R(x)S(x)$ er sædvanlig multiplikation af binære polynomier.

Observér nu, at $(1 + x^{47}) \cdot (H_1^{\mathrm{mod}(i,\,47)})$ er nulsekvensen (0) og tilsvarende med de øvrige faktorer i $Q_A(x)$. Ved at gange $Q_A(x)$ på begge sider af ligning (3) får man derfor, at

$$Q_A(x) \cdot (z^i) = (0)$$

Dette er vores bedre metode til at finde $\mathcal{A}$. For hver af de 252 muligheder for $\mathcal{A}$ danner vi polynomiet $Q_A(x)$ og ser efter om $Q_A(x) \cdot (z^i)$ skulle være (0). Når det er tilfældet, har vi løsningen. Da summen af de 5 største hjullængder er 345, kan graden af den rigtige $Q_A(x)$ være op til 345. Vi skal derfor have 345 kendte klartekstkarakterer, før vi kan beregne den første bit i $Q_A(x) \cdot (z^i)$ og lidt flere end 345 kendte karakterer, før vi med høj sikkerhed kan identificere den rigtige $\mathcal{A}$.

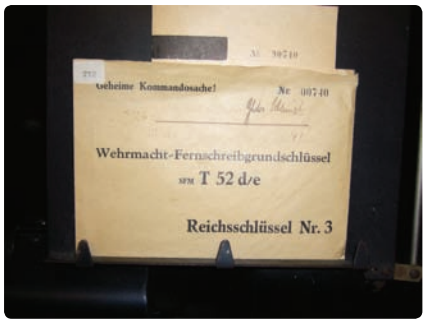
At finde første halvdel af H

Når man i kraft af $\mathcal{A}$ kender de anvendte hjul til A^i, kan man, selv uden at kende rækkefølgen α, også udlæse hjulenes bits H_j^i for de indgående j. Vi illustrerer metoden for $\mathcal{A} = \{1,2,5,6,10\}$ og for $j = 1$ hvor hjullængden er 47.

Anvend først Euklid's algoritme til at finde største fælles divisor mellem de binære polynomier $(1 + x^{47})$ og

$$(1 + x^{53})(1 + x^{64})(1 + x^{65})(1 + x^{73})$$

og til at konstruere de tilhørende binære faktorer $R(x)$ og $S(x)$. Da hjullængderne er indbyrdes primiske, vil den største fælles divisor vise sig at være $1 + x$, så



Figur 7
En af de sidst udviklede G–Skrivere, T52 d/e versionen, som faktisk var sikker. Bemærk den omhyggelige instruks om at smøre maskineriet efter 100 timers brug. Fotografier: Forsvarets Efterretningstjenestes teknisk-historiske samling.

Euklid vil give os polynomier $R(x)$ og $S(x)$, som opfylder

$$1 + x = R(x)(1 + x^{47}) + S(x)(1 + x^{53})(1 + x^{64})(1 + x^{65})(1 + x^{73})$$

I kombination med (3) giver dette en formel for $H_1^{\,i}$. Da

$$H_1^{\,\mathrm{mod(i,\,47)}} = H_2^{\,\mathrm{mod(i,\,53)}} \oplus H_5^{\,\mathrm{mod(i,\,64)}} \oplus H_6^{\,\mathrm{mod(i,\,65)}} \oplus H_{10}^{\,\mathrm{mod(i,\,73)}} \oplus z^i$$

og

$$R(x)(1 + x^{47}) + (1 + x) = S(x)(1 + x^{53})(1 + x^{64})(1 + x^{65})(1 + x^{73})$$

kan vi udregne

$$\begin{aligned}&(1 + x)\cdot(H_1^{\,\mathrm{mod(i,\,47)}})\\ &= (R(x)(1 + x^{47}) + (1 + x))\cdot(H_1^{\,\mathrm{mod(i,\,47)}})\\ &= (S(x)(1 + x^{53})(1 + x^{64})(1 + x^{65})(1 + x^{73}))\cdot\\ &\quad (H_2^{\,\mathrm{mod(i,\,53)}} \oplus H_5^{\,\mathrm{mod(i,\,64)}} \oplus\\ &\quad H_6^{\,\mathrm{mod(i,\,65)}} \oplus H_{10}^{\,\mathrm{mod(i,\,73)}} \oplus z^i)\\ &= (S(x)(1 + x^{53})(1 + x^{64})(1 + x^{65})(1 + x^{73}))\cdot(z^i)\end{aligned} \qquad (5)$$

idet alle øvrige led bliver 0. Vores metode bliver således: Identificér $S(x)$ ved Euklids algoritme, udregn udtrykket på højresiden af (5), gæt på første bit i hjul nummer 1 og anvend (5) til at finde resten. De øvrige hjul findes på tilsvarende måde.

At finde α

Med 120 muligheder for permutationen α, er det ikke noget stort beregningsarbejde at gennemløbe dem alle. Og der er en enkel måde til med stor sikkerhed at afvise de 119 forkerte forslag. Den bygger på, at vi nok har udnyttet paritetsligningen (2), men at vi endnu ikke har udnyttet den oprindelige ligning (1).

Ved brug af kryptoteksten C^i beregner vi en facitliste for, hvor mange 1–taller der skal være i hver D^i. For hver foreslået værdi for α, kan vi desuden beregne alle A^i og kan bestemme hvad værdierne $D^i = P^i \oplus A^i$ faktisk bliver, hvis denne α er den rigtige. Vi forkaster de α, som fører til en eller flere D^i–er med et forkert antal 1–taller. Med mindst 345 karakterer at checke på, bør der kun være et overlevende forslag for α.

Et hurtigt argument for den sidste påstand kan være følgende. Både for den rigtige og for de forkerte værdier af α kan vi modellere D^i som en tilfældig størrelse, ligefordelt mellem 0 og 31. Det betyder, at antallet af 1–taller i en D^i fordeles mellem mulighederne 0, 1, 2, 3, 4, 5 i forholdene 1, 5, 10, 10, 5, 1. For en forkert α bliver for hvert i sandsynligheden for ved tilfældighed at ramme det rigtige antal 1–taller i D^i givet ved

$$\left(\frac{1}{16}\right)^2 + \left(\frac{5}{16}\right)^2 + \left(\frac{10}{16}\right)^2 = \frac{126}{256}$$

når der tages højde for, at pariteterne på forhånd vides at være ens. At ramme rigtigt 345 gange i træk med denne sandsynlighed sker nok ikke, selv med 119 forsøg.

Det overlades til læseren at overveje, hvorfor den anvendte tilfældighedsmodel for mange af de forkerte α–er er ret grov og eventuelt at finde en mere præcis analyse.

Bibliografi

Beckman, B. (2002), *Codebreakers. Arne Beurling and the Swedish crypto program during world war II*, American Mathematical Society.

Lars Ulving, Frode Weierud (2000), *The Geheimschreiber Secret. Arne Beurling and the Success of Swedish Signals Intelligence. Procedings on Coding Theory and Cryptography*, Springer Verlag, men se også cryptocellar.org.

Cryptomuseum, se cryptomuseum.com/crypto/siemens/t52/