Jeg har "knækket" en kode

Af: Lars R. Knudsen, Matematisk Institut, DTU
e-mail: Lars.R.Knudsen@mat.dtu.dk

Man ser til tider i medierne, at nu er en bestemt kode blevet knækket. Jeg var i selv i vælten i januar i år, hvor det nogle steder blev påstået, at jeg havde "knækket koden til USAs ministerier, CIA og FBI" med mere end en hentydning til, at jeg har kunnet "komme ind" steder, hvor jeg ikke burde. Den korrekte version er, at jeg havde vist, at et forslag, kaldet RMAC, fra slutningen af 2002 til en ny autentificeringsstandard foreslået af amerikanske NIST, National Institute of Standards and Technology, har en alvorlig svaghed. Og selvom jeg kunne strække mig så langt som til at sige, at koden **er** knækket, så betyder det ikke, at jeg har, har haft eller nogensinde får utilsigtet adgang til nogetsomhelst. For det første var koden slet ikke taget i brug, og bliver det altså heller ikke. For det andet, så selvom min metode "knækker" koden langt hurtigere end NIST havde regnet med, så kræver angrebet stadig en del regnekraft, som ikke enhver "hacker" har. Det er rigtigt, at NIST laver standarder til brug i amerikanske regeringsinstanser, men om FBI og CIA nogensinde ville have brugt systemet, ved jeg ikke.

Men hvad betyder det at knække en kode? Som jeg skal forsøge at forklare senere, så er dette et diffust begreb.

Der er to klasser af både kryptering og autentificering. Der er de såkaldte *klassiske* (eller "secret-key") systemer og de såkaldte *moderne* (eller "public-key") systemer. I denne artikel vil jeg begrænse mig til at snakke om klassisk kryptering.

Først lidt notation. Lad k betegne en værdi af den hemmelige nøgle, lad x betegne en klartekst og lad y betegne en chiffertekst. Så skriver vi $e_k(x) = y$, dvs., at y er x krypteret med nøgleværdien k. Tilsvarende skriver vi $d_k(y) = x$ for dekrypteringen.

Vi bruger altid *Kerckhoffs' antagelse* idag. Den siger, at det antages, at angriberen ved, hvilken krypteringsmetode, der er i brug. Grunden til dette er, at erfaring og historien har vist, at sikkerhed ikke skal baseres på hemmeligheden af systemet, snarere på hemmeligheden af den brugte nøgle. Med andre ord, antagelsen er altid, at angriberen har kendskab til $e_\cdot(\cdot)$ og $d_\cdot(\cdot)$.

Normalt kalder vi afsenderen for Alice, modtageren er Bob og Eva/Oskar er angriberen. I et klassisk system udveksler Alice og Bob først en hemmelig nøgle, som kun de to kender værdien af. Vi skelner mellem forskellige former for angribere, de passive og de aktive. En passiv angriber opfanger simpelthen bare de meddelelser som afsender og modtager i kryptosystemet udveksler. En aktiv angriber kan gøre det samme som en passiv angriber, men vil ydermere forsøge at påvirke Alice og/eller Bob på forskellige måder, f.eks., kunne Eva prøve at narre Alice til at sende en bestemt krypteret meddelelse til Bob. I de senere år er det blevet ret almindeligt at antage, at Eva er "stærkest mulig", forstået på den måde, at man ser på det værst tænkelige angreb. Ideen er selvfølgelig selv i den værst tænkelige situation at vise eller blive overbevist om, at en angriber kun har ringe chance for at kunne lykkes i sine misgerninger.

En sådan angrebsmodel kaldes ofte for "black-box"-angreb. Man giver simpelthen Eva en sort boks, som indeholder den hemmelige nøgle, således at Eva ikke kan se nøglen, men ellers kan hun gøre nøjagtigt det samme som Alice og Bob. Det vil sige, at hun har adgang til funktionerne $e_\cdot(\cdot)$ og $d_\cdot(\cdot)$ (Kerckhoffs's antagelse) og funktionerne $e_k(\cdot)$ og $d_k(\cdot)$, hvor k er værdien af den hemmelige nøgle valgt af Alice og Bob. Eva's job er at finde k.

I ethvert praktisk system er der et endeligt antal værdier af den hemmelige nøgle, og derfor kan Eva i et "black-box"-angreb altid knække et krypteringssystem ved simpelthen at gætte værdien af den hemmelige nøgle, én efter én, og finde den værdi, som Alice og Bob har valgt. Dette kaldes også "udtømmende nøglesøgning". Nu er vi nået frem til det første forsøg på en definition af, hvad det vil sige at knække en kode. Vi kunne sige, det var tilfældet, hvis Eva i et "black-box"-angreb kan finde den hemmelige nøgle hurtigere end i en udtømmende nøglesøgning. Men se flg. eksempel. Antag $(e_k(\cdot), d_k(\cdot))$ er et sikkert system, og at Eva ikke kan finde den hemmelige nøgle hurtigere end en udtømmende nøglesøgning. Konstruér nu et nyt system hvor kryptering er som flg.:

$$e'_k(x \mid x_2) = e_k(x) \mid x_2,$$

dvs., klarteksten består nu af to

dele, den ene krypteres som før, medens den anden ikke krypteres overhovedet. Det er klart, at Eva ikke kan knække dette system med ovenstående definition på "knækning", men systemet er åbenlyst dårligt. En alternativ definition er som følger. Eva har adgang til funktionerne $(e.(\cdot), d.(\cdot))$, til funktionerne $(e_k(\cdot), d_k(\cdot))$ og kan evaluere alle fire på alle de argumenter, hun måtte ønske. Dernæst giver man Eva en udfordring. Man lader hende spille to spil. I det ene får hun en tilfældig valgt klartekst x og en tilfældig valgt chiffertekst y. I det andet spil får hun en tilfældig valgt klartekst x' og den tilhørende chiffertekst $y' = e_k(x')$, dvs. y' er krypteret med den pågældende kryptosystem og den hemmelige nøgle k. Dernæst vælger man en tilfældig bit. Hvis bitten er nul, lader man Eva spille første spil, ellers spiller hun andet spil. Eva skal nu forsøge at afgøre hvilket spil, hun spiller. Det er klart, at Eva altid har mindst 50% chance for at vinde. Antag, at Eva evaluerer funktionerne $(e.(\cdot), d.(\cdot))$ ialt t gange, at hun evaluerer funktionerne $(e_k(\cdot), d_k(\cdot))$ ialt q gange, og at hun har sandsynlighed $1/2 + \epsilon$ for at vinde ovennævnte spil. Da siger vi, at Eva er en "(t, q, ϵ)-distinguisher". Det er klart, at Eva vinder med sandsynlighed 1 med $q = |K|$, hvor $|K|$ er antallet af nøgler (i en udtømmende nøglesøgning). Men udover det, så er det vanskeligt, at definere, hvornår et angreb retfærdiggør at snakke om, at systemet er "knækket". Det er faktisk yderst sjældent, at vi bruger dette begreb i kryptologikredse. Vi siger hellere, at "der eksisterer et angreb".

I det følgende vil jeg beskrive den svaghed, jeg fandt i autentificeringssystemet fra NIST. Først beskriver jeg DES og trippel-DES, fordi sidstnævnte er en del af autentificeringssystemet.

DES og trippel-DES

Udenfor de militære kredse begyndte interessen for kryptering for alvor at tage fart i slutningen af 70erne. En af grundene var Lucifer, som er et krypteringssystem udviklet af IBM, som senere blev gjort til amerikansk standard, omend i en lettere modificeret udgave. Systemet var en tur indenom den amerikanske efterretningstjeneste, og da den kom ud, var antallet af mulige værdier af den hemmelige nøgle blevet drastisk reduceret. Standarden blev offentliggjort i januar 1977 under navnet DES (Data Encryption Standard) og er idag sandsynligvis det mest brugte krypteringssystem i verden (ihvertifald udenfor de militære kredse). Systemet er et klassisk krypteringssystem. Selvom DES har overlevet næsten 25 års forsøg fra alverdens kryptoanalytikere på at finde genveje til at knække systemet, så er tiden løbet ud for DES. Problemet er, at systemet "kun" har nøgler med 56 bit, hvilket betyder, at der er $2^{56} \approx 10^{17}$ mulige, forskellige værdier af den hemmelige nøgle. Og selvom om dette tal er stort (f.eks. er 2^{56} sekunder cirka 2 milliarder år) er det lille nok til, at det er muligt idag at bygge isenkram, der kan gennemløbe alle muligheder på ganske kort tid. I 1998 offentliggjorde Michael Wiener et chip design, som betyder, at med en investering på en million amerikanske dollars ville det være muligt at gå igennem alle nøgler til DES på en halv time! En excentrisk amerikansk millionær, John Gilmore, lagde året efter 250.000 dollars på bordet og fik bygget en mindre udgave af Wiener's maskine. Denne maskine kan løbe alle nøgler igennem på et par dage, dvs. i gennemsnit vil den kunne finde en ukendt DES nøgle på én dag.

Det faktum, at DES har (for) korte nøgler blev bemærket allerede kort tid efter standarden's offentliggørelse. Amerikanske forskere anbefalede allerede dengang at kryptere samme tekst tre gange i stedet for bare en. Hvis vi lader $e_k(x)$ betegne en DES kryptering af x med nøglen k og tilsvarende lade $d_k(y)$ betegne dekryptering, så anbefaledes det at kryptere x som følger: $y = e_{k_3}(d_{k_2}(e_{k_1}(x)))$. Dvs., Alice og Bob vælger tre DES nøgler på hver især 56 bit, ialt 168 bit. Dette system kaldes *trippel-DES*.[1]

RMAC

Systemet RMAC, som jeg nævnte i starten, er et såkaldt autentificeringssystem og er et klassisk system. Det vil sige, at Alice og Bob initielt udveksler en hemmelig nøgle. Her drejer det sig ikke, i udgangspunktet, om at Alice og Bob vil holde en klartekst hemmelig, men om at Bob kan checke, at den meddelelse Alice sendte også er den meddelelse Bob modtager. M.a.o., det handler om at checke om Eva har pillet med meddelelsen undervejs fra Alice til Bob. Selvom RMAC ikke er et krypteringssystem, så benytter det et sådant, NISTs RMAC var designet til at kunne bruges med trippel-DES og AES.[2] Antag her, at trippel-DES bruges. Denne krypterer blokke af 64 bit og bruger to 168 bit nøgler k og $\tilde{k}$. I RMAC deler Alice sin meddelelse m op i blokke af 64 bit, dvs. $m = m_1, m_2, \ldots, m_n$, hvor m_i er på 64 bit. Hun vælger dernæst en tilfaeldig 64 bit værdi, s (for **s**alt). Saltet udvides dernæst, så det har samme længde som den hemmelige nøgle. Dette gøres ved at tilføje 0-bit, således at $\tilde{s} = (0...0 \mid s)$ er på

[1] Grunden til at bruge en dekryptering med k_2 er, at et sådant trippel system er *kompatibelt* med (normal) DES. Med $k_1 = k_2 = k_3 = k$ reducerer trippel krypteringen til en enkelt kryptering

[2] Advanced Encryption Standard. I 1997 bestemte NIST sig for, at tiden var inde til at finde en afløser til DES. AES standarden blev offentliggjort i 2002

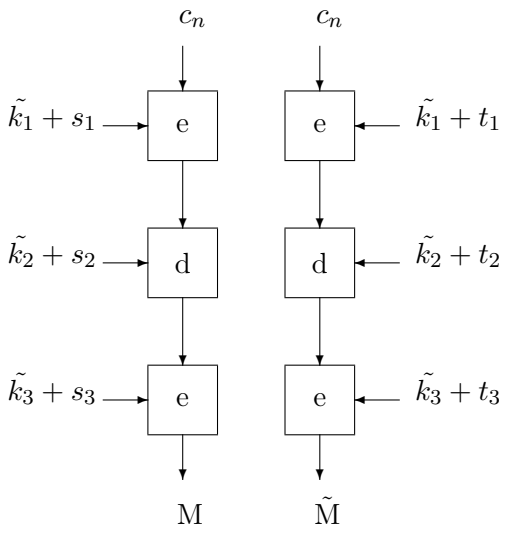
ialt 168 bit. Alice beregner dernæst en *MAC* (Message Authentication Code) som er en check kode, på følgende vis. Lad $c_0 = 0$ og beregn $c_i = e_k(m_i + c_{i-1})$ for $i = 1, \ldots, n$. Tilsidst beregnes $M = e_{\tilde{k}+\tilde{s}}(c_n)$ og Alice sender m, s og M til Bob (her og i det følgende bruges '+' om bitvis addition modulo 2). M kaldes MAC'en til m. Den "ekstra" beregning med nøglen $\tilde{k}$ og saltet er vigtig for at undgå nogle trivielle angreb. Detaljerne udelades her. Bob laver samme beregning som Alice udfra m, s og k, og får et M'. Hvis nu $M' = M$, så vil Bob have en vis sikkerhed for, hvis ellers systemet er sikkert, at m er den meddelelse, som Alice sendte ham. En af grundene til at bruge et tilfældigt salt, s, er, at hvis Alice sender den samme meddelelse m til Bob to gange, da vil saltet med stor sandsynlighed være forskelligt og dermed vil MAC'en til m med stor sandsynlighed også være forskellig. Det forhindrer visse typer angreb. Men desværre for NIST muliggør netop dette faktum også, at deres forslag med trippel-DES er svagt.

Forestil dig, at Alice sender samme meddelelse m to gange til Bob autentificeret med RMAC og trippel-DES og lad M og $\tilde{M}$ være de to MAC værdier. Lad s være saltet brugt første gang og lad t være saltet brugt anden gang, så de udvidede værdier af saltene kan skrives som flg.: $\tilde{s} = s_1, s_2, s_3$ og $\tilde{t} = t_1, t_2, t_3$, hvor $s_i$erne og $t_j$erne er på hver 56 bit. For trippel-DES adderes saltet til nøglerne, således at s_i henholdsvis t_i adderes med k_i. Definitionen af saltet på 168 bit er således, at s_1 og t_1 altid består udelukkende af 0-bit, og s_2 og t_2 altid af mindst 48 0-bit. Da s og t er tilfældigt valgt, betyder det altså at med sandsynlighed 2^{-8} eller $1/256$, så gælder, at $s_1 = t_1$ og $s_2 = t_2$. Se nu Figur 1.

Figuren illustrerer den sidste trippel-DES kryptering i udregningen af de to MACer, hvor nøglen er $\tilde{k} = (\tilde{k_1}, \tilde{k_2}, \tilde{k_3})$. Da Eva ikke kender den brugte trippel-DES nøgle, kender hun ikke (umiddelbart) værdien af c_n. Men det ses, at de to første DES krypteringer i hver trippel-DES kryptering er de samme, da de to par af DES nøgler er ens, med andre ord, $k_1 + s_1 = k_1 + t_1$ og $k_2 + s_2 = k_2 + t_2$. Givet $M, \tilde{M}, s$ og t gør Eva nu følgende for enhver mulig værdi af $\tilde{k_3}$: dekryptér M med nøglen $\tilde{k_3} + s_3$ og dekryptér $\tilde{M}$ med nøglen $\tilde{k_3} + t_3$. Hvis de to dekrypteringer er ens, er den pågældende værdi af $\tilde{k_3}$ en kandidat til den brugte, ukendte værdi. For en tilfældigt valgt værdi af $\tilde{k_3}$, vil de to resulterende dekrypteringer være ens med sandsynlighed 2^{-64}. Denne test finder altid den værdi af $\tilde{k_3}$, som Alice og Bob har valgt og brugt, og da der kun er 2^{56} mulige værdier af $\tilde{k_3}$, vil denne med god sandsynlighed også være den eneste kandidat, der bliver foreslået. Altså, Eva kan finde den tredje af de tre DES nøgler med sandsynlighed 2^{-8}, hvis Alice og Bob udveksler samme meddelelse mindst to gange, forudsat at Eva opsnapper MACen i begge tilfælde, og at hun har kapaciteten til at lave cirka 2^{56} DES krypteringer. Sandsynligheden for at angrebet vil øges, hvis Alice og Bob udveksler flere par af identiske meddelelser. Når hun først har $\tilde{k_3}$, kan hun finde $\tilde{k_2}$ på lignende måde. Tilsidst findes $\tilde{k_1}$, men detaljerne udelades her. Er RMAC med trippel-DES så knækket? Jeg har faktisk ikke anvist en måde, hvorpå Eva finder begge trippel-DES nøgler (hurtigere end forventet), kun den ene. Faktisk er mit angreb "kun" en $(2^{58}, 2^8, \epsilon)$-distinguisher, hvor ϵ er tæt på 1. Imidlertidig betyder angrebet, at en angriber efterfølgende kan "skrælle" den ekstra kryptering væk fra enhver MAC. Og da den ekstra kryptering er indført for at undgå nogle trivielle angreb (som dog ikke finder nøglen), er der her tale om en alvorlig svaghed. Tilsyneladende indså også NIST dette, og de trak kort efter forslaget RMAC tilbage. Sidenhen er et andet forslag kommet på banen, men det er en anden historie.[3]

Knæk, knak, knude.....

[3]Det skal retfærdigvis siges, at også andre forskere havde dårlig kritik af RMAC.


c_n
c_n
$\tilde{k_1} + s_1$
e
e
$\tilde{k_1} + t_1$
$\tilde{k_2} + s_2$
d
d
$\tilde{k_2} + t_2$
$\tilde{k_3} + s_3$
e
e
$\tilde{k_3} + t_3$
M
$\tilde{\text{M}}$

Figur 1: RMAC.