RSA-kryptering i Ma 5

Att skriva hemliga meddelanden som bara den tänkta mottagaren kan läsa är en konst lika gammal som skriftspråket. Genom att till exempel skifta varje bokstav fyra steg framåt kan meddelandet *kom vid åtta* skrivas som OSQ ZMH BXXE. För bokstaven Å innebär detta att vi ”börjar om från A”. Matematiskt är detta samma som att vi för varje bokstav utför beräkningen $c = m + e$ (mod 29) där $e = 4$ är krypteringsnyckeln (*e*ncryption), m är meddelandet uttryckt som ett tal. Exempelvis kan vi låta a = 0, b = 1 et cetera. Då är k = 10 och $c = 10 + 4 = 14$ motsvarar vårt kodmeddelande c. Moduloberäkningen garanterar att vi får ett resultat som motsvarar en giltig bokstav.

Dekrypteringen skulle förstås kunna utföras med beräkningen $m = c - e$ (mod 29) men vi lägger märke till att det går lika bra med beräkningen $m = c + d$ (mod 29) om vi låter $d = 25$. Det går alltså att använda *samma* beräkningsformel till både krypteringen och dekrypteringen om vi använder *olika* nycklar.

Den här typen av chiffer kallas ofta för caesarchiffer eftersom bland annat Julius Ceasar, men säkerligen andra före honom, använde sig av det.

Multiplikation istället för addition

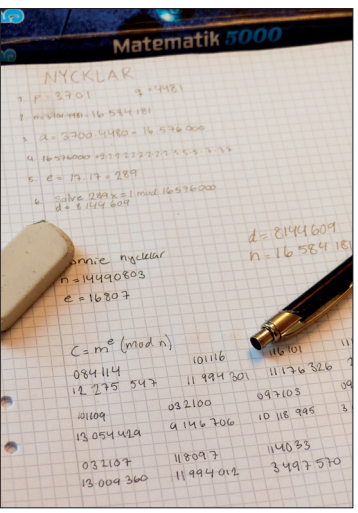
Vi kan också använda oss av multiplikation för att koda våra meddelanden. $c = m \cdot e$ (mod 29) fungerar utmärkt om $e = 4$. Det går också bra att dekryptera med multiplikation, men vilken dekrypteringsnyckel ska vi i så fall använda? Vi vet att $c = m \cdot e$ (mod 29) och vi vill hitta d så att $m = c \cdot d$ (mod 29). Sätter vi in den första i den andra så får vi $m = m \cdot e \cdot d$ (mod 29), vilket är uppfyllt om $e \cdot d \equiv 1$ (mod 29) eller om $e \cdot d - 1 = p \cdot 29$ för något heltal p.

I vårt fall, med $e = 4$ ska vi hitta ett värde på d så att $4d = 29p + 1$. Vi kan prova olika värden på p och hittar snabbt att $n = p$ ger $d = 22$.

Om vi till exempel vill koda bokstaven ”k” beräknar vi $10 \cdot 4 \equiv 11 \pmod{29}$ och skickar koden L. Vi dekrypterar detta genom att beräkna $11 \cdot 22 \equiv 10 \pmod{29}$ och får tillbaka klartexten ”k”.

Asymmetrisk kryptering

Idén att använda olika nycklar för kryptering och dekryptering kallas asymmetrisk kryptering och är en relativt modern idé från 1970-talet. Det går, förutom att använda addition och multiplikation, att kryptera meddelanden med potenser. Då har vi alltså $c = m^e \pmod{n}$ och $m = c^d \pmod{n}$ där n är ”alfabetets” längd. Med en längd på 29 kan vi dock bara kryptera en bokstav i taget och varje bokstav kommer alltså att bli utbytt mot en och samma kodbokstav varje gång. Det gör att koden blir lätt att knäcka med till exempel frekvensanalys. Om vi krypterar block om två bokstäver i taget behöver vi en längd på minst 29^2 och i praktiken använder man betydligt större blocklängder och mycket större tal n. Talet n behöver också vara stort av en annan anledning: det är produkten av två primtal, vardera med flera hundra siffror. En dator kan enkelt multiplicera ihop primtalen och därigenom beräkna n men ännu är ingen dator i världen så snabb att den på rimlig tid kan faktorisera denna produkt. Detta är kritiskt för att ingen ska kunna dekryptera meddelanden utan tillgång till den hemliga dekrypteringsnyckeln d.



Klassrumsarbete

Primtal utnyttjas i RSA-kryptering, döpt efter Rivest, Shamir och Adleman som beskrev metoden i detalj 1977. RSA-kryptering används idag oftast när nya nycklar ska distribueras till andra, effektivare krypteringssystem. Genereringen av nycklarna är lite mer komplicerad och hänger ihop med Fermats lilla sats men går utmärkt att utföra med elever. Under en vecka i maj 2021 genomförde eleverna som läste Ma 5 på Rodengymnasiet i Norrtälje ett miniprojekt kring RSA-kryptering. Alla elever skulle kryptera och dekryptera varsitt meddelande och generera sina egna nycklar. Talet n bestämdes som produkten av två primtal mellan 1000 och 5000. Eleverna genererade därefter nycklarna e och d. Varje elev kodade ett meddelande till en annan elev med dennes offentliga nycklar och den mottagande eleven dekrypterade det med sin dekrypteringsnyckel. Bokstäverna kodades med ascii-kod med blocklängden 2. Nycklar, krypterade meddelanden och de dekrypterade klartexterna publicerades allteftersom i klassens onlineplattform. Alla beräkningar utfördes med Geogebra eller Wolfram Alpha. Eleverna hade dessutom arbetat med kongruensräkning en lektion innan detta.

Vi spenderade tre lektioner på detta arbete vilket inkluderade en kort rapport från varje elev. Genom att skriva rapporten från början blev den klar när eleverna var klara med arbetet som upplevdes som både roligt och relativt verklighetsförankrat.

Jonas Hall

Länkar till bland annat elevinstruktionerna finns på Nämnaren på nätet.