SJÄLVSTÄNDIGA ARBETEN I MATEMATIK

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET

Pailliers kryptosystem - en exposé

av

Anders Mogren

2023 - No K1

Pailliers kryptosystem - en exposé

Anders Mogren

Självständigt arbete i matematik 15 högskolepoäng, grundnivå

Handledare: Jonas Bergström

2023

Pailliers kryptosystem – en exposé

Anders Mogren

August 2023

Abstract

I denna kandidatuppsats beskrivs Pailliers krypteringssystem, en asymmetrisk krypteringsmetod med applikationer inom bland annat säker datahantering och homomorf kryptering. En historisk översikt ges samt den teoretiska och matematiska bakgrunden för krypteringssystemet presenteras. Det underliggande matematiska problemet tas upp samt en ganska uttömmande beskrivning av den matematiska grupp inom vilken systemet primärt används ges.

Krypterings- och dekrypteringsprocesserna beskrivs, samt hur nycklar genereras. Algoritmerna som används för att implementera och potentiellt utmana Pailliers kryptosystem analyseras. Det beskrivs även hur systemet kan generaliseras samt ges exempel på hur det används vid tillämpningar, samt hur det kan komma att utvecklas vidare.

Resultaten betonar inte bara styrkorna i Pailliers kryptosystem utan lyfter även fram dess sårbarheter och begränsningar. En kort diskussion förs om systemets prestanda, brister och fördelar jämfört med andra krypteringar.

Nyckelord: asymmetrisk kryptering, datasäkerhet, homomorf kryptering, kryptografi, Pailler.

In this bachelor's thesis, Paillier's encryption system is described—an asymmetric encryption method with applications in secure data management and homomorphic encryption, among other areas. The paper provides a historical overview and presents the theoretical and mathematical background of the encryption system. The underlying mathematical problem is addressed, accompanied by a fairly comprehensive description of the mathematical group in which the system is primarily used.

The results emphasize the strengths of Paillier's cryptosystem and highlight its vulnerabilities and limitations. There is a brief discussion on system performance, shortcomings, and advantages compared to other encryption methods.

Keywords: asymmetric encryption, data security, homomorphic encryption, cryptography, Paillier.

1 Inledning

1.1 Bakgrund

Kryptografi är en mycket viktig del av dagens informationsinfrastruktur. Ämnet har under århundranden berört och fascinerat och är idag kanske mer aktuellt än någonsin. Det har stor betydelse för det moderna samhället och är intressant ur ett matematiskt perspektiv.

1.1.1 Varför är Paillier relevant och intressant?

Matematiskt finns flera saker att undersöka vad gäller kryptering med Paillier, såsom gruppteori och isomorfier. Pailler är ett homomorft system för kryptering vilket ger det vissa speciella egenskaper. När man använder Paillier kan man bearbeta krypterad information utan att kunna avläsa den, vilket gör krypteringssystemet lämpat för många användningsområden som exempelvis molntjänster.

1.2 Syfte

1.2.1 Mål med uppsatsen

Syftet med denna uppsats är att redogöra för kryptering med Paillier i stort och smått. Kärnan i föreliggande skrift kommer att vara att söka redogöra för och förklara de matematiska aspekterna av kryptering med Paillier. Detta är ett komplext ämne som är intressant att förstå inte minst givet hur viktigt asymmetrisk kryptering är inom ett stort antal områden.

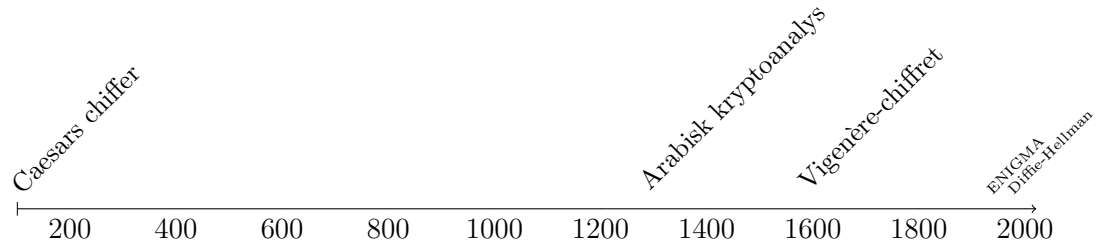
1.2.2 Beskrivning av upplägget

En teoretisk bakgrund med en översiktlig historia av kryptografi kommer att ges i början av uppsatsen. Det viktigaste här är kanske paradigmskiftet mellan äldre ”ad hoc-kryptering” till dagens betydligt mer teoriorienterade dito. Det följer en genomgång av allmänna kryptografiska principer och en matematisk bakgrund innan själva huvuddelen som behandlar Paillierkryptering. Avslutningsvis kommer en utblick att göras som innefattar dels hur Paillierkryptering kan generaliseras till andra grupper än den som ursprungligen används, dels ett exempel på en aktuell implementering. Sist finns en diskussionsdel och eventuella slutsatser som kan dras utifrån diskussionen, samt referenser och tack.

2 Teoretisk bakgrund

2.1 Kryptografins historia

2.1.1 Tidslinje



2.1.2 Klassisk kryptografi

Det hör till människans natur att vilja kunna förmedla information utan att obehöriga kan ta del av den. Därför har kryptografin en lång och rik historia. Uppfinningsrikedomen är stor, vilket visas av allt från vad vi idag kanske skulle kalla för enklare chiffer till modern asymmetrisk kryptering.

Här kommer några illustrativa exempel att nämnas. Det finns en uppsjö men i denna korta historik endast plats för ett fåtal. Dessa är valda för att ge en bild av kryptografins utveckling under historiens gång.

Ett av de tidigaste exempel man känner till är Caesars chiffer från hundratalet e.kr. [Katz s. 9]. Detta är ett substitutionskrypto, dvs det bygger på att man helt enkelt byter plats på bokstaven man vill kryptera. I fallet med Caesars chiffer låter man alla bokstäver motsvaras av bokstaven tre platser till höger, så att till exempel A blir D, B blir E, och så vidare.
Med denna metod skulle PAILLIER skrivas SDLOOHU. Det finns i detta fall i strikt bemärkelse ingen nyckel, som i de andra system som kommer att tas upp i denna historik.

Ett antal krypton liknande Caesars har förekommit genom historien. Ibland har man låtit dessa gå att variera så till vida att man haft en sorts enkel nyckel, nämligen att antalet steg man skiftat bokstäverna kunnat vara olika [Katz s. 10]. Om man tänker sig att man låter alfabetets tecken motsvaras av heltal, och får i det svenska fallet alltså $k = 0, ..., 28$, kan man beskriva detta sätt att kryptera som att man räknar modulo $k + 1$. Säg att man vill kryptera bokstaven D, som motsvaras av 3, och man godtyckligt valt $k = 28$. Då fås:

$$3 + 28 = 31 = 2 \mod 29.$$

För att dekryptera räknar man

$$2 - 28 = -26 = 3 \mod 29.$$

Man har alltså krypterat D, som motsvaras av 3 till C, som motsvaras av 2. Genom att dra bort nyckelns värde och räkna modulo har man sedan fått tillbaka den ursprungliga bokstaven.

Detta krypto är enkelt att attackera eftersom det finns så få möjliga värden på k. Det är inte alls orimligt att helt enkelt testa alla möjliga värden och på så vis knäcka kryptot. Ett sätt att göra denna typ av krypto svårare är förstås att låta k kunna anta ett mycket stort antal värden. För att detta skulle vara svårt idag, med den tillgängliga beräkningskraften i åtanke, skulle dock k behöva kunna anta cirka 2^{70} möjliga värden [Katz s. 11].

En ytterligare kategori av krypton är sådana där man ersätter ett tecken med ett annat tecken från en permutation av exempelvis alfabetet. Med det svenska alfabetet skulle man då ha 29! permutationer att använda till att välja ersättningstecknet. Dessa alfabetiska krypton finns även i en utvecklad variant där man med hjälp av en nyckel krypterar samma tecken med olika permutationer av alfabetet beroende på tecknets position i meddelandet. Principen är skisserad nedan.

P A I L L I E R
m a t e m a t i

Här krypteras det första L:et ned alfabet e, medan det andra krypteras med alfabet m. En känd variant av ett sådant polyalfabetiskt krypto är Vigenère-chiffret från 1600-talet. Detta ansågs länge omöjligt att knäcka, men i likhet med andra polyalfabetiska krypton kan det attackeras med statistiska metoder såsom frekvensen av bokstävers förekomst i språket.

Just konsten att läsa krypton utan att ha tillgång till nyckeln verkar vara ett förhållandevis nytt fenomen. Det äldsta exemplet man känner till är arabiska källor från 1300-talet [Hoffstein s. 34]. I dessa källor beskrivs hur man kan använda sig av ovan nämnda bokstavsfrekvens samt sannolikheter att olika bokstavpar förekommer i kombination för att knäcka krypton. Under europeisk medeltid förekom också kryptografiska analysmetoder, men det var inte förrän på 1800-talet mer metodiska lösningar där man tog hjälp av statistik förekom. Innan dess ägnade man sig huvudsakligen åt (kanske kvalificerade) gissningar [Hoffstein s. 35].

I takt med samhället och vetenskapens utveckling ökade även behovet av att använda krypterad kommunikation. I stora drag handlade det dock fram till moderna dagar om mer sofistikerade versioner av de typer av krypton som tagits upp ovan. Höjden av symmetrisk (dvs, med endast privat nyckel) kryptografi kan med rätta sägas vara den tyska krypteringsmaskinen ENIGMA som användes under

andra världskriget. Denna maskin hade ett system med rotorer som på ett mycket stort antal sätt kunde välja mellan alfabet. Även japanerna hade en liknande apparat där rotorernas funktion istället sköttes av en serie knappar.

2.1.3 Modern kryptografi

"We stand today on the brink of a revolution in cryptography" är de inledande orden i W. Diffie och M. Hellmans artikel "New Directions in Cryptography" från 1976 [Hoffstein s. 62]. I denna artikel introducerades konceptet kryptering med publika och privata nycklar, så kallad asymmetrisk kryptering. Detta tillsammans med utvecklingen av beräkningskraft lade grunden för dagens moderna krypteringssystem.

Den bärande idén är antagandet om One way functions, dvs funktioner som är enkla att beräkna men svåra att invertera. Med hjälp av en publik, allmänt känd nyckel kan man enkelt kryptera informationen men för att dekryptera den krävs en privat nyckel som är okänd för allmänheten. Det är viktigt att notera att det ej ännu bevisats att denna typ av funktioner existerar, utan det är just ett antagande [Hoffstein s.64].

Diffie och Hellmans upptäckt ledde bland annat till utvecklingen av RSA-kryptering, ett av de mest kända systemen inom modern kryptering. Länge var dessa algoritmer närmast statshemligheter och användandet var hårt reglerat ännu på 1990-talet. Som kuriosa sägs det att om man exempelvis tatuerat Perl-koden för RSA-algoritmen räknades man som militärt vapen med exportförbud [Hoffstein s. 62].

Det är svårt att överskatta vikten av moderna krypteringsmetoder i dagens samhälle. Exempelvis skulle förmodligen Internet och mycket annan modern kommunikation knappast vara möjligt. Kryptering används idag i allt från militära satelliter till en influencers Instagramkonto och utgör en vital och integrerad del av allas våra dagliga liv.

2.1.4 Skillnader

Det finns mycket stora kvalitativa olikheter mellan äldre tiders och dagens kryptografi. Den kanske mest väsentliga är att kryptografi tidigare hade drag av konsthantverk där man tillämpade metoder lite så att säga ad hoc, medan den vetenskap det är idag baseras på stringent teori. Även kvantitativt skiljer det sig mycket; algoritmernas komplexitet är betydligt större idag och svårigheterna att utmana dem likaså. Vidare är användningsområdena betydligt fler numera och kryptering har karaktären av var mans egendom, även om vi kanske inte alltid tänker på hur mycket vi använder det i vardagen.

2.2 Ordlista och begrepp

1. Asymmetrisk kryptering
   kryptering med två nycklar, en för kryptering och en för dekryptering.

2. Exponentiell- och polynomtid
   Med exponentiell tid menas att tidskomplexiteten ökar exponentiellt med problemets storlek. Ett litet problem kan alltså snabbt bli väldigt tidskrävande att lösa. Matematiskt uttrycks detta

   $$T(n) = O(k^n)$$

   där $k > 1$ är någon konstant och n problemets storlek (vanligen i binär form).

   I polynomiell tid begränsas problemets tidskomplexitet till en polynomfunktion av storleken på detsamma. Tidsåtgången för att lösa problemet följer då i stort storleken på problemet. Detta kan uttryckas som

   $$T(n) = O(n^k)$$

   där k är en konstant och n problemets storlek.

3. Homomorfi bevarar gruppstrukturer och tillåter manipulation av krypterade data utan att man kan läsa vad datan innehåller. I engelsk litteratur kallas detta ibland ”malleability”.

   Man skiljer mellan Fully homomorphic encryption (FHE) och Somewhat homomorphic encryption (SHE). I en svensk kontext skulle man kunna tala om fullständigt eller partiellt homomorf kryptering. Skillnaden mellan FHE och SHE är vilka operationer som kan utföras på den krypterade datamängden.

4. Längd av primtal. med längd av primtal menas i kryptografiska sammanhang talens längd i databitar. exempelvis är 11 i binär form 1011 (4 bitar) medan 3 motsvarar 11 och alltså har längd 2.

5. One-way function och Trapdoor function
   Detta kallas ibland på svenska enriktad funktion eller enkelriktad funktion. I denna uppsats kommer dock den engelska termen One-way function att användas. Detta betyder att en funktion är enkel att beräkna men svår att invertera. Notera att det är ett antagande att dessa existerar; bevis finns ej för att det är så. När det finns en privat nyckel till krypteringen talar man om en sk Trapdoor function. Den privata nyckeln ger en genväg för att beräkna inversen av funktionen.

   Detta innebär att om man har ett ingångsvärde skall det vara lätt att erhålla ett utgångsvärde men svårt att från utgångsvärdet beräkna ingångsvärdet.

Ett exempel kan vara en databas där lösenord lagrats med en One-way function. En hacker kommer över databasen men kommer få svårigheter att utifrån den räkna ut vad de faktiska lösenorden är.

6. Pseudorandom permutations
   Detta kan eventuellt översättas till pseudoslumpmässiga permutationer, ett uttryck som dock ej kommer att användas i föreliggande skrift. Vad det innebär är permutationer (av en mängd) som skapats av en pseudorandom generator. En sådan generator är en algoritm som ger till synes slumpmässiga utfall men i själva verket är en deterministisk process.

   Dessa till synes slumpmässiga utfall är ofta fullt tillräckliga inom kryptografin och används bland annat för att skapa nycklar.

7. Underliggande problem
   Moderna krypteringsmetoder baseras på att det antas finnas ett underliggande matematiskt problem. Idealiskt är detta problem lätt att lösa, dvs komma fram till ett utgångsvärde, men svåra att invertera (se One-way function ovan).

2.3 Kryptografins grundläggande principer

Till skillnad från tidigare kryptografi, som haft en air av konstnärskap och ad hoc-lösningar, är den moderna mer vetenskaplig till sin natur [Katz s. 18].

Med detta avses att den moderna kryptografin bygger på stringent teoribildning. Tre huvudprinciper [Katz s. 18] skiljer modern kryptografi från sin föregångare.

1. Formulering av exakta definitioner.
2. De antaganden på vilken säkerheten hos en kryptografisk metod vilar (till exempel svårigheten att faktorisera) måste vara precist formulerade.
3. Det skall finnas rigorösa bevis för varför en kryptografisk metod är säker med avseende på de antaganden man gjort.

2.4 Nycklar, kryptering, dekryptering, asymmetri

En modern krypteringsmetod, som RSA och Paillier, innehåller tre grundläggande algoritmer: en för att generera nycklar, en för att kryptera och en algoritm för att dekryptera ett meddelande [Katz s. 29].

I asymmetrisk kryptering skapar algoritmen för att generera nycklar en publik respektive privat nyckel utifrån vissa villkor. Vanligen, exempelvis i Paillier, beror nycklarna på primtalsfaktorerna p, q av ett tal N.

Krypteringsalgoritmen använder den publika nyckeln för att kryptera meddelandet. Denna nyckel finns allmänt tillgänglig och är inte hemlig. För att dekryptera används den privata nyckeln vilken endast mottagaren känner till.

Idén här är att meddelandet enkelt kan dekrypteras med hjälp av den privata nyckeln. Eftersom modern asymmetrisk kryptering bygger på antagandet att då det finns en funktion som är lätt att beräkna (med hjälp av den publika nyckeln) men svår att invertera, utgör den privata nyckeln en sorts genväg (Trapdoor) varmed beräkningen av den inverterade funktionen avsevärt förenklas. Detta är görbart eftersom de publika och privata nycklarna är matematiskt relaterade till varandra.

2.5 Matematisk bakgrund

2.5.1 Om grupper

Matematiska grupper besitter en rad egenskaper är användbara i modern kryptografi. Gruppteori ger den teoretiska grunden för analys och utveckling av kryptografiska algoritmer. Egenskaperna som följer utmärker en grupp.

Definition 2.1. *En grupp är en mängd G tillsammans med en operator $\star$ som uppfyller följande egenskaper:*

1. *den är sluten, dvs att när operatorn appliceras på element i G måste även resultatet ligga i G,*
2. *associativitet: för alla $a, b, c \in G$ gäller $(a \star b) \star c = a \star (b \star c)$,*
3. *det existerar en identitet $e \in G$ så att $e \star a = a = a \star e$,*
4. *det finns en unik invers sådan att för varje $a \in G$ existerar ett element $b \in G$ som uppfyller att $a \star b = e = b \star a$.*

Om det gäller att $a \star b = b \star a$ är gruppen även kommutativ. Sådana grupper kallas abelska grupper. Abelska grupper är viktiga och önskvärda inom kryptografi därför att den kommutativa egenskapen underlättar beräkning och implementation av kryptografiska algoritmer.

2.5.2 Generellt om underliggande problem

Konceptet att Public Key-kryptering bygger på ett underliggande hårt matematiskt problem medför att man vill ha vissa egenskaper för sagda problem. Det kanske viktigaste är att problemet är lätt att beräkna men svårt att invertera. Annorlunda uttryckt betyder detta att krypteringen bygger på att man gjort antagandet att det underliggande problemet inte kan lösas i polynomtid [Katz s. 24].

På grund av de egenskaper det underliggande problemet bör besitta använder man sig ofta av faktorisering av primtal i olika varianter. Detta är typiskt en One-way function. Man låter

$$pq = N$$

där p, q (stora) primtal. Uppgiften är att utifrån N hitta p (och därmed q), alltså att invertera problemet.

Ju större primtalen p, q är desto mer komplicerad blir uppgiften. Det mest primitiva sättet är förstås att genom att prova att dela med olika tal försöka hitta p. Dvs, man kontrollerar helt enkelt om p delar N för $p = 2, ..., \sqrt{N}$.

Av uppenbara skäl blir detta snabbt tidsödande. Faktum är att denna metod har exponentiell tidskomplexitet. Det finns andra snabbare, mer effektiva algoritmer som är subexponentiella, dvs har en tidskomplexitet mellan exponentiell och polynomiell. Ett exempel på en sådan algoritm är The Number Field Sieve [Hoffstein s. 162].

2.6 Användningsområden

Modern asymmetrisk kryptering har en uppsjö användningsområden. Säker kommunikation är viktigt för väldigt många tillämpningar, allt från militära till privatpersoners bankärenden. Homomorf kryptering används då man vill att data ska kunna behandlas utan att läsas. Ett exempel kan vara att man vill låta någon utomstående hantera patientjournaler utan att kunna läsa vad som står i dem.

Det är svårt att överskatta modern kryptografis betydelse för hur dagens samhälle är utformat; många saker man närmast tar för självklara idag skulle utan denna bli mycket otympliga och tidsödande.

3 Metod och implementering

3.1 Kort om Pailliers krypteringssystem

Pascal Paillier skapade detta system 1999 [Paillier s. 229]. Det är en asymmetrisk, probalistisk algoritm som används för Public Key-kryptering. Systemet baseras på svårigheterna att faktorisera stora tal i primtalskomponenter och vad som kallas The Residuosity Class Problem, dvs svårigheten att skilja residyer från godtyckliga element i gruppen vilken Pailliers krypteringssystem använder.

Systemet är mer effektivt än exempelvis bevisat säkra system som RSA och Rabin [Katz s. 385]. En av de viktigaste egenskaperna hos detta system är dess homomorficitet som medför flera fördelar, bland annat att det går att bearbeta icke dekrypterad information [Katz s. 385].

3.2 Matematiska grundvalar

Det är nu tid att undersöka vilka matematiska begrepp som ligger till grund för kryptering med Paillier. Tre viktiga områden tas upp, matematisk homomorfi, gruppen i vilken man arbetar med Paillier och det underliggande problemet som gör att metoden fungerar.

3.2.1 Homomorfi

Ur ett kryptografiskt perspektiv är fördelen med homomorfi främst att det går att utföra operationer på något som är krypterat utan att först dekryptera det [Katz s. 393].

Antag att man har två krypterade datamängder, ds_1 och ds_2 och vill att någon extern aktör ska lägga ihop dessa utan att kunna ta del av informationen de innehåller. Kanske är mängderna mycket stora och endast den utomstående aktören har tillräcklig datakraft för att utföra detta. Uppdragsgivaren skickar då ds_1 och ds_2 till den utomstående aktören, som lägger ihop dem och skickar tillbaka resultatet utan att ha kunnat ta del av informationen som är krypterad.

Det som möjliggör detta förfarande är vad som kallas homomorf kryptering. En homomorfi är en avbildning mellan objekt och bevarar deras struktur. Mer specifikt bevarar homomorfin de algebraiska operationerna och relationerna mellan element i de olika objekten.

Definition 3.1. *Låt G, H vara grupper med en operator $\star_G$ respektive $\star_H$. En funktion $f : G \to H$ är en homomorfi från G till H om $f(g \star_G h) = f(g) \star_H f(h)$ där $g \in G$ och $h \in H$.*

Homomorfin bevarar gruppstrukturen, till exempel har vi att identiteten bevaras: $f(e_G) = e_H$.

Det innebär om att om klartexterna har en gruppoperation och chiffertexterna en annan så ska krypteringsfunktionen vara en homomorfi.

Paillier är ett additivt homomorft krypteringssystem [Katz s. 393] . Att krypteringsfunktionen i Paillier är en homomorfi visas i **Sats 3.2**, punkt 3.

3.2.2 Gruppen $Z^*_{N^2}$

En speciell grupp som används inom kryptografi är enhetsgruppen $\mathbb{Z}^*_M$. Det är en matematisk struktur som innehåller restklasserna som är relativt prima modulo M. Att de är relativt prima (enhetliga) är något som används för att konstruera det underliggande problemet. Då gruppens operator är multiplikation modulo M kommer även resultatet att vara relativt prima modulo M. I gruppen finns också för varje element en multiplikativ invers, dvs att för varje a i gruppen finns ett b

sådan att $ab = 1 \mod M$. Då ordningen i vilken gruppens element multipliceras mod M, alltså

$$a \star b = b \star a \mod M$$

saknar betydelse är gruppen även abelsk. Bland annat denna kommutativa egenskap samt att abelska grupper anses förhållandevis lätta att arbeta med gör dem tillämpliga inom kryptografi.

Kryptering i Paillier sker med hjälp av gruppen $Z^*_{N^2}$. Denna grupp har flera egenskaper av betydelse för hur man krypterar och dekrypterar. Här följer en beskrivning av dessa egenskaper samt bevis för att det är som påstås.

Sats 3.2. *Låt $N = pq$ där p och q är olika udda primtal av samma längd (se Ordlista och begrepp). Då har $Z^*_{N^2}$ följande egenskaper:*

1. $\gcd(N, \phi(N)) = 1$.
2. *För ett heltal $0 \leq a \leq N$ gäller att $(1+N)^a = (1+aN) \mod N^2$. Därför är $(1+N)$ i $Z^*_{N^2}$ av ordning N. Alltså är $(1+N)^N = 1 \mod N^2$ och $(1+N)^x \neq 1 \mod N^2$ för $1 \leq x < N$.*
3. *$Z_N \times Z^*_N \cong Z^*_{N^2}$ med isomorfin $f : Z_N \times Z^*_N \to Z^*_{N^2}$ som ges av $f(a \mod N, b \mod N) = (1+N)^a \cdot b^N \mod N^2$.*

Då $f : Z_N \times Z^*_N \to Z^*_{N^2}$ betyder $x \mod N^2 = f(a \mod N, b \mod N)$ att $x \mod N^2 \in Z^*_{N^2}$ korresponderar med $(a \mod N, b \mod N) \in Z_N \times Z^*_N$.

Bevis: Med $N = pq$ och p, q udda primtal av samma längd ges bevis för egenskaperna i **Sats 3.2** enligt

1. Notera först att $\phi(N) = (p-1)(q-1)$ och antag att $p > q$.
   Då p är ett primtal och $p > p-1 > q-1$ måste

   $$\gcd(p, \phi(N)) = \gcd(p, (p-1)(q-1)) = 1.$$

   Vi har per definition att $\gcd(q, q-1) = 1$

   Om $\gcd(q, p-1) \neq 1$ måste, eftersom q är ett primtal, $\gcd(q, p-1) = q$. Men om $\gcd(q, p-1) = q$ så är $p-1$ minst lika med $2q$. Om så är fallet så är

   $$\frac{p-1}{q} \geq \frac{2q}{q} = 2.$$

   Detta motsäger att p och q har samma längd.

2. Detta kan visas med binomialsatsen. Den säger att

$$(1+N)^a = \sum_{k=0}^{a} \binom{a}{k} N^k.$$

Om man räknar $\mod N^2$ kommer alla termer med $k \geq 2$ att bli 0. För $a \in \{0, ..., N\}$ blir $1 + aN = 1 \mod N^2$ endast då $a = N$.

3. Först visas att f är väldefinierad
Låt a, b, k vara heltal. Vi har att

$$f(a+kN, b) = (1+N)^{a+kN} \cdot b^N = (1+N)^a \cdot (1+N)^{kN} \cdot b^N = (1+N)^a \cdot [(1+N)^N]^k \cdot b^N \mod N^2.$$

Notera att

$$(1+N)^N = \sum_{i=0}^{N} \binom{N}{i} 1^{N-i} N^i = \sum_{i=0}^{N} \binom{N}{i} N^i.$$

Utveckling av denna summa visar att alla termer utom den första är 0 $\mod N^2$ varför $(1+N)^N = 1 \mod N^2$.

Med andra ord:

$$f(a+kN, b) = (1+N)^a \cdot 1^k \cdot b^N = (1+N)^a \cdot b^N = f(a, b).$$

Om vi istället tittar på fallet $b + kN$ fås

$$f(a, b+kN) = (1+N)^a \cdot (b+kN)^N \mod N^2.$$

Vi har att

$$(b+kN)^N = \sum_{i=0}^{N} \binom{N}{i} b^{N-i} \cdot (kN)^N.$$

Den första termen om man utvecklar denna summa är b^N och alla andra termer innehåller N^2. Således är $(b+kN)^N = b^N \mod N^2$, vilket betyder att $(a, b+kN) = (1+N)^a \cdot b^N \mod N^2$. Alltså är funktionen oberoende av representanter för a, b i sina restklasser $\mod N^2$ och funktionen är därmed väldefinierad.

Notera att $\gcd((1+N), N^2) = 1$ samt $\gcd(b, N^2) = 1$. Alltså ligger $(1+N)^a \cdot b^N$ i $Z^*_{N^2}$ då det saknar gemensam faktor med N^2.

För att visa att f är en bijektion visas först att

$$|Z^*_{N^2}| = \phi(N^2) = pq(p-1)(q-1) = |Z_N| \cdot |Z^*_N| = |Z_N \times Z^*_N|.$$

Så ordningen av $Z_{N^2}^*$ är lika med ordningen av $Z_N \times Z_N^*$. Därför räcker det att visa att f är injektiv för att visa att den är bijektiv.

Vi låter $a_1, a_2 \in Z_N$ och $b_1, b_2 \in Z_N^*$ samt $f(a_1, b_1) = f(a_2, b_2)$. Alltså är

$$\frac{(1+N)^{a_1} \cdot {b_1}^N}{(1+N)^{a_2} \cdot {b_2}^N} = (1+N)^{a_1 - a_2} \cdot \left(\frac{b_1}{b_2}\right)^N = 1 \mod N^2.$$

Om båda sidor upphöjs med $\phi(N)$ fås

$$(1+N)^{(a_1-a_2)\cdot\phi(N)} \cdot \left(\frac{b_1}{b_2}\right)^{N\cdot\phi(N)} = 1^{\phi(N)} \mod N^2 = 1 \mod N^2.$$

Eftersom $\phi(N^2) = N \cdot \phi(N)$ och $\frac{b_1^{\phi(N^2)}}{b_2^{\phi(N^2)}} = 1 \mod N^2$ enligt Eulers sats följer att

$$(1+N)^{(a_1-a_2)\cdot\phi(N)} = 1 \mod N^2.$$

Vi har visat att ordningen av $(1+N) \mod N^2$ är N. N delar alltså $(a_1 - a_2) \cdot \phi(N)$. Dock är $\gcd(N, \phi(N)) = 1$, så N kan dela $(a_1 - a_2) \cdot \phi(N)$ endast om $a_1 = a_2 \mod N$.

Insätt $a_1 = a_2 \mod N$ i

$$(1+N)^{a_1 - a_2} \cdot \left(\frac{b_1}{b_2}\right)^N = 1 \mod N^2.$$

Då fås $\frac{b_1^N}{b_2^N} = 1 \mod N^2 \Leftrightarrow b_1^N = b_2^N \mod N^2$. Detta betyder att $b_1^N = b_2^N \mod N$.

Alltså är $b_1 = b_2 \mod N$. Det betyder att f är injektiv och tillsammans med att vi tidigare visat att $|Z_{N^2}^*| = |Z_N \times Z_N^*|$ att f är en bijektion.

Det återstår att visa att f är en isomorfi, dvs att då f ges av

$$f(a, b) = (1+N)^a \cdot b^N \mod N^2$$

så gäller att

$$f(a_1, b_1) \cdot f(a_2, b_2) = f(a_1 + a_2, b_1 \cdot b_2).$$

Vi har att

$$f(a_1, b_1) = (1+N)^{a_1} \cdot b_1^N \mod N^2,$$
$$f(a_2, b_2) = (1+N)^{a_2} \cdot b_2^N \mod N^2.$$

Vidare är

$$f(a_1 + a_2, b_1 \cdot b_2) = (1+N)^{a_1+a_2} \cdot (b_1 \cdot b_2)^N \mod N^2.$$

Till sist

$$f(a_1, b_1) \cdot f(a_2, b_2) = (1+N)^{a_1} \cdot b_1^N \cdot (1+N)^{a_2} \cdot b_2^N = (1+N)^{a_1+a_2} \cdot (b_1 \cdot b_2)^N \mod N^2.$$

□

Då man krypterar med Paillier använder man isomorfin mellan $Z_N \times Z_N^* \to Z_{N^2}^*$, dvs att $f(a \mod N, b \mod N) = (1+N)^a \times b^N \mod N^2$. Då denna funktion krypterar meddelandet kommer dess invers att dekryptera detsamma. Vi ska nu visa att så är fallet då vi har den publika nyckeln $(\lambda(N) = \phi(N), \mu = \phi(N)^{-1})$, där alltså μ är den modulära inversen till $\lambda(N)$.

Sats 3.3. *Vi påstår att givet följande förutsättningar:*

1. $0 \leq m < N$, *där* m *är meddelandet som ska krypteras,*
2. $0 < r < N$ *och* $\gcd(r, N) = 1$.
3. $g = 1 + N$
4. *krypteringsfunktionen är* $c = g^m \cdot r^N \mod N^2 = (1+N)^m \cdot r^N \mod N^2$ *där* $0 \leq c < N^2$

så är inversen till krypteringsfunktionen $\frac{(g^m \cdot r^N)^{\phi(N)} - 1}{N} \cdot \phi(N)^{-1} \mod N$.

Bevis: Låt

$$d = c^{\phi(N)} = (g^m \cdot r^N)^{\phi(N)} = g^{m\cdot\phi(N)} \cdot r^{N\cdot\phi(N)} \mod N^2,$$

där $0 \leq d < N^2$. Notera att $\phi(N^2) = N \cdot \phi(N)$. Enligt Euler är $a^{\phi(m)} = 1 \mod m$, så

$$r^{N\cdot\phi(N)} = r^{\phi(N^2)} = 1 \mod N^2.$$

Således fås att

$$(g^m \cdot r^N)^{\phi(N)} \mod N^2 = g^{m\cdot\phi(N)} \mod N^2.$$

Om detta utvecklas med hjälp av binomialsatsen erhålls

$$g^{m\cdot\phi(N)} \mod N^2 = (1+N)^{m\cdot\phi(N)} \mod N^2 = \sum_{k=0}^{m\cdot\phi(N)} \binom{m\cdot\phi(N)}{k} N^k \mod N^2$$

vilket ger

$$g^{m\cdot\phi(N)} = 1 + m\cdot\phi(N)\cdot N + \sum_{k=2}^{m\cdot\phi(N)} \binom{m\cdot\phi(N)}{k} N^k \mod N^2.$$

För alla $k > 1$ blir termerna $0 \mod N^2$ varför

$$d = 1 + m\cdot\phi(N)\cdot N \mod N^2.$$

Om vi sedan låter $e = \frac{d-1}{N}$ fås att

$$e = m\cdot\phi(N) \mod N.$$

Slutligen beräknas

$$e\cdot\mu = e\cdot\phi(N)^{-1} \mod N = m \mod N = m.$$

Så

$$m = \frac{c^{\phi(N)} - 1}{N} \cdot \phi(N)^{-1} \mod N.$$

Alltså har m fåtts tillbaka från c. □

Det kan som parentes nämnas att på liknande sätt som vi fått tillbaka m från c går det även att få tillbaka r från c.

Vi har $c = g^m \cdot r^N \mod N^2$ och låter $0 \leq c < N^2$. Sätt $f(m,r)^t \mod N = c^t \mod N$. Välj $t = N^{-1} \mod \phi(N)$ så att $0 \leq t < \phi(N)^{-1}$.

Notera: $N|N^2$ varför vi kan räkna termer $\mod N^2$ med $\mod N$. Då fås:

$$c^t \mod N = (g^m \cdot r^N)^t \mod N = (1+N)^{t \cdot m} \cdot r^{t \cdot N} \mod N.$$

Genast ser vi att $r^{t \cdot N} \mod N = r \mod N$ så vi har kvar

$$c^t \mod N = (1+N)^{t \cdot m} \mod N = r.$$

De egenskaper som visats i detta stycke tillsammans med gruppens komplexitet i form av stor mängd möjliga värden gör att det är mycket svårt och beräkningstungt att faktorisera N.

Det är dock värt att notera att i fallet Paillier (liksom liknande krypteringssystem med undantag för Rabin) vet man inte om säkerheten hos krypteringen är ekvivalent med svårigheten att faktorisera. Det kan vara så men det är ej visat [Katz s. 385].

3.2.3 Det underliggande problemet i Paillier

Eftersom $Z_N \times Z_N^* \to Z_{N^2}^*$ med isomorfi $f(a \mod N, b \mod N) = (1+N)^a \cdot b^N \mod N^2$ korresponderar ett godtyckligt element $y \in Z_{N^2}^*$ med ett godtyckligt element $(a \mod N, b \mod N) \in Z_N \times Z_N^*$.

Definition 3.4. *$y \in Z_{N^2}^*$ är en N:e residy modulo N^2 om y är en N:e-potens, dvs det finns ett heltal $x \mod N^2 \in Z_{N^2}^*$ så att*

$$y = x^N \mod N^2.$$

Notera att + är operationen i Z_N, som är en additiv grupp.

Sats 3.5. *Varje element $y \mod N^2 \in Z_{N^2}^*$ som korresponderar med $(0 \mod N, b \mod N) \in Z_N \times Z_N^*$ är en N:e residy.*

Bevis: Notera att då $\gcd(N, \phi(N)) = 1 \mod \phi(N)$ finns modulär invers $d = N^{-1} \mod \phi(N)$. Vi har då att

$$(a \mod N, b^d \mod N)^N = (N \cdot a \mod N, b^{dN} \mod N) =$$

$$= (0 \mod N, b^{N^{-1} \mod \phi(N) \cdot N} \mod N) = (0 \mod N, b \mod N).$$

□

Så ett godtyckligt element y från $Z_{N^2}^*$ som även är en N:e residy korresponderar med ett element $(0, r)$ i $Z_N \times Z_N^*$. Låt ett element som inte är en residy, dvs ett godtyckligt element w från $Z_{N^2}^*$, korrespondera med $(r', r) \in Z_N \times Z_N^*$.

Vad som kallas Nth Residuosity Class Problem är att skilja dessa element från varandra, alltså att skilja mellan $(0, r)$ och (r', r).

Detta leder till följande abstrakta sätt att kryptera meddelandet $0 \leq m < N$ med en publik nyckel N. Välj en godtycklig N:e residy $(0, r)$ och sätt kryptotexten c till

$$c = (m, 1) \cdot (0, r) = (m, r)$$

där $\cdot$ representerar gruppoperationen i $Z_N \times Z_N^*$.

Då ett residy $(0, r)$ inte kan skiljas från ett godtyckligt element (r', r) blir kryptotexten

$$c' = (m, 1) \cdot (r', r) = (m + r' \mod N, r)$$

omöjlig att skilja från den ovanstående för någon som ej vet faktorerna av N och försvårar därför olika former av avlyssning [Katz s. 389 ff]. I grund och botten är det underliggande problemet i Paillier att från den krypterade texten c erhålla ursprungsmeddelandet m. Detta kallas för Residuosity Class Problem, som även är vad krypteringsalgoritmen baseras på.

Dock är Residuosity Class Problem i sig byggt på att man ej känner till faktoriseringen av N i p och q. Alltså har man, precis som i RSA och vissa andra krypteringar, om man kan faktorisera N knäckt hela krypteringen, eftersom man då kan avläsa m ur c. Detta kan man göra eftersom om man har faktorerna p och q, varav N består, har man implicit även den privata nyckeln. Alltså kan man helt enkelt beräkna inversen till krypteringsfunktionen (se **Sats 3.3, 3.5**).

Det här kan man uttrycka som Class[N] $\Leftarrow$ Fact[N]. Man tror att om man kan lösa Residuosity Class Problem kan man även faktorisera N, men detta har ej visats [Paillier s. 233].

3.3 Hur fungerar Paillier?

3.3.1 Nyckelgenering

Det första steget i krypteringsprocessen är skapandet av en nyckel. Denna består av två delar. En publik (eller offentlig) nyckel och en privat nyckel. Den publika nyckeln är allmänt känd och kan användas av alla medan endast den som ska dekryptera meddelandet (dvs mottagaren) känner till den privata.

Det finns flera sätt att generera nycklar i Paillier. Först kommer en grundläggande version att presenteras. I denna version har man förenklat processen med avseende på hur man beräknar den privata nyckeln. Detta förenklar även förståelsen av hur nyckelgenerering i Paillier går till.

Det finns dessutom flera varianter som används vid tillämpningar där man lagt till vissa delar. En av dessa varianter presenteras för att ge ett exempel. I den senare varianten har steg lagts till för hur den privata nyckeln skapas. Anledningen till dessa tillägg är att man kan önska ökad säkerhet och ytterligare egenskaper såsom förbättrad prestanda i krypteringsprocessen.

Här följer först en beskrivning av hur den förstnämnda processen för att generera en nyckel kan gå till.

1. Välj två primtal p, q av ungefär samma storlek, där $p \neq q$.
2. Beräkna $N = pq$ samt använd Eulers ϕ-funktion för att beräkna $\phi(N) = (p-1)(q-1)$.
3. Beräkna $\mu = \phi(N)^{-1} \mod N$ där $0 \leq \mu < N$. Eftersom $p \neq q$ och både p och q är relativt prima med N finns denna modulära invers.
4. Låt $g = N + 1$. Detta är något av ett redundant steg, men kan ibland behövas för kompabilitet vid generaliseringar av Pailliers krypteringsprocess [här behövs en referens].
5. Den publika nyckeln är (N, g) och den privata nyckeln är $(\phi(N), \mu)$.

Ett exempel på hur en process för att skapa nycklar då Paillier används vid tillämpningar ges nedan.

1. Välj två stora (av minst 1024 bitars längd) primtal p och q. Om de har samma längd och är prima är $\gcd(pq, (p-1)(q-1)) = 1$.
2. Låt $N = pq$.
3. Med hjälp av Carmichaels λ-funktion beräknas $\lambda = \lambda(N)$:

$$\lambda = \operatorname{lcm}(p-1, q-1).$$

4. Välj ett godtyckligt $[g] \in Z^*_{N^2}$ där $0 \leq g < N^2$.
5. Låt $u = g^{\lambda} \mod N^2$ där $0 \leq u < N^2$. Gör följande

   Definition 3.6. $L(u) = \frac{u-1}{N}$

   Eftersom g är relativt prima med N och med valet av λ säkerställs att $L(u)$ är ett heltal.
6. Beräkna den multiplikativa inversen $\mu = (L(u))^{-1} \mod N$ där $0 \leq \mu < N$.

7. Den publika nyckeln är nu (N, g) och den privata (λ, μ).

Som synes har man här skapat den privata nyckeln på ett annat sätt än i det första exemplet. Istället för Eulers ϕ-funktion används Carmichaels λ-funktion. Notera även att $\phi(N), \phi(N)^{-1}$ i den första versionen av nyckelgenereringen motsvaras av λ, μ i den andra. Då $\lambda = \phi(N)$ och $g = 1 + N$ fås att $u = g^{\phi(N)} \mod N^2 = (1+N)^{\phi(N)} \mod N^2$ där $0 \leq u < N^2$.

Utveckling med binomialsatsen visar att $u = 1 + N \cdot \phi(N) \mod N^2$. Som tidigare har vi att

$$\mu = L(u)^{-1} \mod N = \frac{1}{L(g^{\phi(N)})} \mod N = \frac{1}{(\frac{u-1}{N})} \mod N,$$

alltså att $\mu = \phi(N)^{-1} \mod N$.

Detta beror på att Carmichaels λ-funktion ger exponenten av den multiplikativa gruppen. Med andra ord är λ den minsta exponent man kan upphöja ett element av den multiplikativa gruppen med för få identiteten. Mer precist

Definition 3.7. *Carmichaelfunktionen $\lambda(n)$ är det minsta positiva heltal m sådant att $a^m = 1 \mod n$ för alla a relativt prima n.*

För att snabba upp nyckelgenerering kan man även i det andra fallet låta $g = N+1$, och $\lambda = \phi(N), \mu = \phi(N)^{-1}$ som i den första versionen av nyckelgenerering. Detta används ofta då beräkningen av μ kan bli tidsödande med stora primtal p, q.

3.3.2 Kryptering

Med förutsättningarna

1. en publik nyckel (N, g) där $N = pq$ och $p, q, p \neq q$ primtal av samma längd samt $g = N + 1$,

2. en privat nyckel $(\phi(N), \phi(N)^{-1})$,

3. ett meddelande $0 \leq m < N$ och $\gcd(m, N) = 1$

beskrivs nu hur den krypterade texten c skapas.

1. Välj godtyckligt $0 < r < N$ med $\gcd(r, N) = 1$, och

2. beräkna $c = f(m, r)$ där $0 \leq c < N^2$.

3.3.3 Dekryptering

Givet ett krypterat meddelande $0 \leq c < N^2$, låt

$$m = \frac{c^{\phi(N)} - 1}{N} \cdot \phi(N)^{-1} \mod N.$$

3.3.4 Överblick

Då kryptering i Paillier är en ganska komplicerad process ges här en kort överblick över hur systemet konstrueras.

Låt GenereraMod vara en algoritm i polynomtid som ger N, p, q från input 1^n.

1. Nycklar genereras. N är publik nyckel och $(N, \phi(N)$ privat nyckel.

2. För att kryptera $m \in Z_N$ väljs godtyckligt r i Z_N^* som ger krypteringstext c där
$$c := (1+N)^m \cdot r^N \mod N^2.$$

3. För att dekryptera används den privata nyckeln och
$$m := \frac{c^{\phi(N)} - 1}{N} \cdot \phi(N)^{-1} \mod N.$$

3.3.5 Exempel

För att illustrera hur Paillier kan fungera ges här ett pedagogiskt exempel med mycket små tal. Låt säga att GenereraMod ger $N = 15, p = 5, q = 3$. Vi har då att $\phi(N) = 8$. Vidare blir $\phi(N)^{-1} = 2$.

Antag att vi vill kryptera meddelandet $m = 3$ och att vi väljer ett godtyckligt $r \in Z_N^*$, till exempel $r = 4$. Den publika nyckeln är $N = 15$ och den privata dito $(15, 8)$

Då fås:

$$c = (1+15)^8 \mod 225 = 154.$$

Alltså är 3 krypterat till 154. Dekrypterar vi får vi

$$m = \frac{154^8 \mod 225 - 1}{15} \cdot 2 \mod 15 = 18 \mod 15 = 3,$$

så det ursprungliga $m = 3$ är tillbaka.

3.3.6 Algoritmer och komplexitet

Ett antal algoritmer är nödvändiga för att genomföra kryptering med Paillier.

Vanligtvis genereras de två primtalen p, q som tillsammans utgör N i den publika nyckeln slumpmässigt. Detta sker med en primtalsprobabilitetsalgoritm, och testas med exempelvis Miller-Rabin. Primtalskandidaterna man fått fram testas för att se om de verkligen är primtal. Detta är snabba algoritmer som körs i polynomtid.

Vid exponentiering används olika algoritmer för Fast Powering. Ett exempel på detta är så kallad Square-and-multiply. I denna effektiva algoritm utnyttar man att $a^{2^n} = (a^{2^{n-1}})^2$ för att minska antalet multiplikationer. En variant av detta som kallas Sliding Windows används även.

Tidskomplexiteten för dessa algoritmer är polynomiell, vilket är avsevärt snabbare än naiv exponentiering vilken snabbt blir ohanterbart långsam.

För att hitta den modulära inversen vid dekryptering används Euklides utökade algoritm. Detta är ett standardförfarande inom kryptografi och denna algoritms tidskomplexitet är i någon mening linjär; dess exekveringstid ökar proportionellt med antalet bitar som utgör problemets storlek.

3.3.7 Algoritmer för att utmana Paillier

Paillier anses mycket svårt att utmana när nyckeln $N = pq$, där p, q är primtal av samma längd, är tillräckligt stor. Eftersom systemets säkerhet bygger på svårigheten att faktorisera ett stort tal i sina primtalsfaktorer är de flesta algoritmer som används för att försöka knäcka det inriktade på faktorisering. Beskrivning av några metoder som används följer.

Brute force-attacker: man provar alla möjliga faktoriseringar av N. Detta är förstås i princip omöjligt om N är tillräckligt stort.

Ett alternativ till Brute force är kvadratmetoden som fungerar särskilt väl när primtalsfaktorerna är nära varandra. Det är en förhållandevis långsam metod men tillämpas när mer generella metoder är ineffektiva.

Quadratic Sieve är en algoritm som är effektiv för att faktorisera stora heltal. Den bygger på linjär algebra och talteori och drar nytta av kvadratkroppar. Idén är att man med hjälp av kvadratiska former hittar vektorer med vars hjälp man kan faktorisera talet.

Den mest effektiva algoritmen som används inom kryptografi för att faktorisera stora heltal är General Number Field Sieve. Den fungerar genom att man undersöker relationer mellan sammansatta heltal och genom att analysera dessa kan härleda primtalsfaktorer [Hoffstein s. 162]. Number Field Sieve har subexponentiell tidskomplexitet vilket innebär att dess tidskomplexitet ligger mellan polynomiell tid och exponentiell tid.

4 Utblick

4.1 Generalisering av Paillier

Damgård och Jurik beskrev i sin artikel 2001 hur Paillier kan generaliseras till beräkningar mod $s + 1$, med $s \geq 1$ där $s = 1$ naturligtvis representerar det ursprungliga kryptosystemet. De skriver att denna generalisering gör Paillier lika säkert och medför flera fördelar.

En av de största fördelarna som tas upp är minskad expansion factor. Detta begrepp beskriver hur längden (i bitar) av den krypterade texten skiljer sig från klartext. Vanligtvis i Paillier är krypterad text betydligt längre än okrypterad. Damgård och Jurik har genom laborerande med ändrad modulus minskat expansion factor [Damgård och Jurik s. 15] från två till nära ett. Detta innebär betydligt snabbare databehandling.

4.2 Paillier och mikrorymdskepp

Med den tekniska utvecklingen har det blivit enklare att bygga mindre och lättare rymdfarkoster. Dessa blir mer och mer populära vid olika uppdrag i rymden då de har en rad fördelar såsom större flexibilitet, lägre kostnad och är mer pålitliga [Yongxia Shi m. fl. s. 2].

Med fler farkoster ökar behovet av säker kommunikation. Forskare från Hong Kong och Peking har undersökt hur Pailliers kryptosystem kan användas i dessa sammanhang. Särskilt har man intresserat sig för de homomorfa egenskaperna som bidrar till säker kommunikation även om det finns någon som avlyssnar densamma. Tack vare de homomorfa egenskaperna kan man även låta rymdfarkosterna göra beräkningar på krypterad indata till styrsystemet [Yongxia Shi m. fl. s. 1].

Forskarna kommer fram till att Pailliers kryptosystem fungerar för att hantera de känsliga data som hanteras av rymdfarkosternas styrsystem samt skriver att det kan vara intressant med ytterligare försök med ett fullt homomorft system [Yongxia Shi m. fl. s. 13].

5 Slutord

Pailliers system för kryptering är homomorft och baserat på att man använder gruppen $Z^*_{N^2}$. Homomorfin innebär att operationer kan utföras på redan krypterad text utan att man behöver ta del av informationen. Gruppen $Z^*_{N^2}$ är lämplig för kryptering framförallt för att den har modulära inverser och är abelsk.

Som visats har gruppen $Z^*_{N^2}$ flera egenskaper som gör den lämplig att användas vid kryptering med Pailler, bland annat att det finns en isomorfi till $Z_N \times Z^*_N$ [Katz

s. 385].

De homomorfa egenskaperna i Pailliers krypteringssytem möjliggörs av att gruppen $Z^*_{N^2}$ är abelsk och därmed besitter multiplikativa egenskaper. Dessa homomorfa egenskaper är en av de största styrkorna med Pailliers kryptosystem, och möjliggör bearbetning av krypterad data utan att dekryptera densamma. Som exempel kan ges att en utomstående aktör ombeds behandla patientjournaler. Denna aktör kan då bland annat sätta ihop två journaler utan att behöva eller kanske framför allt kunna läsa dem

Egenskaperna hos Pailliers kryptosystem gör att det har en rad användningsområden, inte minst som nämnts vid behandling av skyddad data (bland annat Blockchains och molnberäkningar).

Isomorfin $f(a, b) = [(1 + N)^a \cdot b^N] \mod N^2$, som beskrivits är i någon mening kryptosystemets kärna. Den funktion som beskrivs av isomorfin är den sk One-way function som antas finnas och som är lätt att beräkna men svår att invertera.

I uppsatsen har även framlagts bevis för att $Z^*_{N^2}$ besitter de egenskaper som krävs för att den ska vara lämplig att användas vid kryptering med Paillier. Beviset för att isomorfin föreligger och är korrekt beskriven åskådliggör även hur krypteringsprocessen fungerar i Paillier.

Beräkningen av, och beviset för inversen av den del varmed den krypterade texten fås från det ursprungliga meddelandet påvisar även på ett i författarens tycke elegant sätt hur konceptet med en One-way function kan se ut. Det blir tydligt hur enkelt det är att så att säga räkna baklänges för att dekryptera något.

I uppsatsen tas även ett försök att generalisera Pailliers kryptosystem upp. Forskarna Damgård och Jurik visar hur man kan generalisera systemet från gruppen $Z^*_{N^2}$ till $Z^*_{N^{s+}}$, där $s = 1, 2, ...$ Det kanske mest intressanta resultatet tycks vara att denna generalisering under vissa omständigheter gör att man kan optimera krypteringsprocessen.

Ett mer specifikt användningsområde tas upp i avsnittet om rymdfarkoster. Här illustreras hur Pailliers homomorfa egenskaper kommer väl till pass vid styrning av små och lätta rymdfarkoster som samarbetar.

Det ges i början av denna skrift en översiktlig bild av hur kryptografin utvecklats under historien. Att ta med sig från detta är främst de stora kvantitativa och kvalitativa skillnaderna mellan historisk kryptografi i form av chiffer och liknande och modern asymmetrisk kryptering.

6 Referenser (och tack)

Vid arbetet med uppsatsen har både böcker i ämnet samt vetenskapliga artiklar konsulterats. Vidare har ämneskunskaper som författaren tillägnat sig under studier vid Stockholms universitets matematiska institution varit till stor nytta.

Mestadels har Hoffsteins ”An Introduction to Mathematical Cryptography” och Katz ”Introduction to Modern Cryptography” använts vid arbetet med uppsatsens huvudsakliga innehåll. Flera artiklar har använts, inte minst för avsnittet ”Utblick”.

För en djupare förståelse av hur man i praktiken använder Pailliers kryptosystem har även videoföreläsningar, främst från Eindhovens universitet, varit till stor nytta.

Författaren vill rikta ett särskilt tack till Jonas Bergström vid Stockholms universitets matematiska institution som handlett vid skrivandet av uppsatsen och på ett mycket pedagogiskt vis bidragit till ökad förståelse för ämnet. Jonas tålamod, kunnande och kommentarer har varit till ovärderlig hjälp under processens gång.

6.1 Referenser

Diffie, W och Hellman, M. E. (1976). *New Directions in Cryptography.* IEEE Transactions on Information Theory.

Damgård, I., Jurik, M. och Buus Nielsen, J. (2001). *A Generalization of Paillier's Public-Key System with Applications to Electronic Voting.* Aarhus University.

Hoffstein, J., Silverman, J. H. och Pipher, J. (2008). *An Introduction to Mathematical Cryptography.* Springer.

Katz, J. och Lindell, Y. (2007). *Introduction to Modern Cryptography.* CRC Press.

Paillier, P. (1999). *Public-Key Cryptosystems Based on Composite Degree Residuosity Classes.* GEMPLUS.

Yongxia, S., Ehsan, N. och Quinglei, H. (2023). *Secure motion control of microspacecraft using semi-homomorphic encryption.* Sands.