Kryptografi: en blandning av datavetenskap, matematik och tillämpningar

Björn von Sydow

17 november 2010

Kryptografins historia

Fyra faser

Kryptografins historia

Fyra faser

- Antiken – ca 1920
  Papper och penna.

Kryptografins historia

Fyra faser

- Antiken – ca 1920
  Papper och penna.
- 1920 – ca 1960
  Elektromekaniska maskiner.

Kryptografins historia

Fyra faser

- Antiken – ca 1920
  Papper och penna.
- 1920 – ca 1960
  Elektromekaniska maskiner.
- 1960 –
  Datoriserade system.

Kryptografins historia

Fyra faser

- Antiken – ca 1920
  Papper och penna.
- 1920 – ca 1960
  Elektromekaniska maskiner.
- 1960 –
  Datoriserade system.
- 1990 –
  Krypto överallt.

– ca 1600. Substitutionschiffer

Exempel

```
abcdefghijklmnopqrstuvxyz
PAQCRIXOEVHYJUZGSTKLNFMDB
```

Ordet `kryptografi` krypteras till `HTDGLZXTPIE`.

Sändare och mottagare måste ha kommit överens om ovanstående nyckel.

– ca 1600. Substitutionschiffer

Exempel

```
abcdefghijklmnopqrstuvxyz
PAQCRIXOEVHYJUZGSTKLNFMDB
```

Ordet `kryptografi` krypteras till `HTDGLZXTPIE`.

Sändare och mottagare måste ha kommit överens om ovanstående nyckel.

Ansågs oknäckbart under medeltiden.

Lösningsmetoder hittades inom den arabiska kulturen: använd frekvensanalys.

Lätt att knäcka om texten ej för kort.

– ca 1600. Substitutionschiffer

Exempel

```
abcdefghijklmnopqrstuvxyz
PAQCRIXOEVHYJUZGSTKLNFMDB
```

Ordet `kryptografi` krypteras till `HTDGLZXTPIE`.

Sändare och mottagare måste ha kommit överens om ovanstående nyckel.

Ansågs oknäckbart under medeltiden.

Lösningsmetoder hittades inom den arabiska kulturen: använd frekvensanalys.

Lätt att knäcka om texten ej för kort.

Alla bokstäver förekommer inte lika ofta; i engelsk text står ’e’ för ca 13% av alla bokstäver, ’t’ för ca 10%,. . . ’z’ för < 0.1 %.

Liknande statistik kan göras för par av angränsande bokstäver.

1600 – ca 1850. Polyalfabetiska chiffer

Exampel

Nyckelordet är ’secret’.

```
 plaintextwithoutspaces
+secretsecretsecretsecr
------------------------
 HPCYRMWBVNMMYSWKWISGGJ
```

1600 – ca 1850. Polyalfabetiska chiffer

Exampel

Nyckelordet är ’secret’.

```
 plaintextwithoutspaces
+secretsecretsecretsecr
------------------------
 HPCYRMWBVNMMYSWKWISGGJ
```

Mest kända exemplet är Vigenères chiffer.

Oknäckbart i 300 år.

Knäcktes av Charles Babbage ca 1850.

1600 – ca 1850. Polyalfabetiska chiffer

Exampel

Nyckelordet är ’secret’.

```
 plaintextwithoutspaces
+secretsecretsecretsecr
------------------------
 HPCYRMWBVNMMYSWKWISGGJ
```

Mest kända exemplet är Vigenères chiffer.

Oknäckbart i 300 år.

Knäcktes av Charles Babbage ca 1850.

Knäcks i två steg:

Först bestäms nyckelordets längd.

Sedan används frekvensanalys.

Att knäcka Vigenère

Exempel

Betrakta följande chiffertext:

```
DEFVFLJCZBKFDKFDMNBFLVQLKVXCAVSKLFBVNSFBLNSBVFRTDESD
SDFLVQLKVXCFKFBVBVJJSFDKBLSNFMBSLKBNASVDXZFDZBDFSZLQ
SDVGMALBAMVLAVMALVMALVAMDLVFLVQLKVXCCVSDGSDFBVADSBFB
```

Att knäcka Vigenère

Exempel

Betrakta följande chiffertext:

DEFVFLJCZBKFDKFDMNBFLVQLKVXCAVSKLFBVNSFBLNSBVFRTDESD
SDFLVQLKVXCFKFBVBVJJSFDKBLSNFMBSLKBNASVDXZFDZBDFSZLQ
SDVGMALBAMVLAVMALVMALVAMDLVFLVQLKVXCCVSDGSDFBVADSBFB

Steg 1: Nyckelordets längd

Följden FLVQLKVXC förekommer tre gånger i chiffertexten.

Trolig orsak: samma ord förekommer tre gånger i klartexten i samma position modulo nyckelordets längd.

$\gcd(55 - 20, 132 - 55) = 7$; trolig nyckelordslängd är 7.

Att knäcka Vigenère

Exempel

Betrakta följande chiffertext:

DEFVFLJCZBKFDKFDMNB**FLVQLKVXC**AVSKLFBVNSFBLNSBVFRTDESD
SD**FLVQLKVXC**FKFBVBVJJSFDKBLSNFMBSLKBNASVDXZFDZBDFSZLQ
SDVGMALBAMVLAVMALVMALVAMDLV**FLVQLKVXC**CVSDGSDFBVADSBFB

Steg 2: Fullständig lösning

När man vet nyckelordets längd p kan man använda frekvensanalys på p deltexter:

- bokstäverna i position k, där $k \equiv 0 (\mathrm{mod} p)$ (alla dessa har skiftats lika mycket).
- bokstäverna i position k, där $k \equiv 1 (\mathrm{mod} p)$; återigen samma skift (men ett annat än för de föregående) osv.

1920–1945. Enigma

Tysk chiffermaskin.
Tre seriekopplade rotorer, som ger en ny substitution för varje skriven bokstav.

Flyttbara kablar byter plats på sex bokstavspar.

Ny nyckel varje dag.

1920–1945. Enigma

Tysk chiffermaskin.
Tre seriekopplade rotorer, som ger en ny substitution för varje skriven bokstav.

Flyttbara kablar byter plats på sex bokstavspar.

Ny nyckel varje dag.

Knäcktes på 30-talet av polska kryptografer; förbättrad version av Alan Turing m fl under andra världskriget.
Allierad tillgång till tysk kommunikation hade stor betydelse för krigets utgång.

Blockchiffer

01110100011010001111000010110	10100101011000110000011010111111

Meddelanden är följder av bitar (ASCII-text, bilder, video, . . .)

Kryptering blockvis; ett block består av ett fixt antal bitar (i dag typiskt 128). Nyckel också bitföljd, av samma storlek.

Blockchiffer

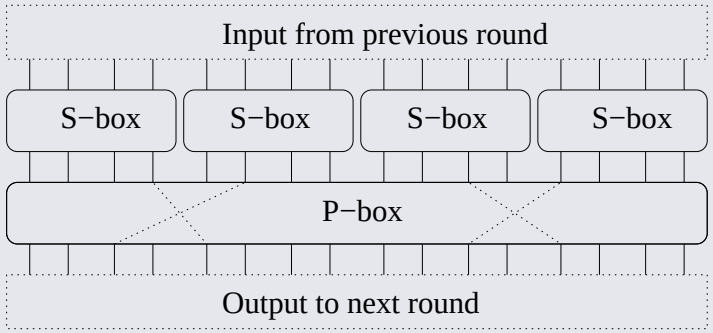
01110100011010001111000010110	10100101011000110000011010111111

Meddelanden är följder av bitar (ASCII-text, bilder, video, . . .)

Kryptering blockvis; ett block består av ett fixt antal bitar (i dag typiskt 128). Nyckel också bitföljd, av samma storlek.

Ett varv består av S-boxar (= funktioner $\{0,1\}^k \to \{0,1\}^k$) och P-boxar (= permutationer av bitar).

Itereras 10-15 varv; olika delnycklar för varje varv.



Blockchiffer de senaste 35 åren

Två chiffer har haft särskild betydelse:

Blockchiffer de senaste 35 åren

Två chiffer har haft särskild betydelse:

- DES (Data Encryption Standard)
  Standard från mitten på sjuttiotalet; dominerande i 25 år.

  64 bitars block, 56 bitars nyckel – för litet. Uttömmande sökning möjlig. På väg ut.

Blockchiffer de senaste 35 åren

Två chiffer har haft särskild betydelse:

- DES (Data Encryption Standard)
  Standard från mitten på sjuttiotalet; dominerande i 25 år.

  64 bitars block, 56 bitars nyckel – för litet. Uttömmande sökning möjlig. På väg ut.
- AES (Advanced Encryption Standard)
  Ny standard 2000. 128 bitars block och nyckel.

  Ersätter DES i nya applikationer; god säkerhet.

Blockchiffer de senaste 35 åren

Två chiffer har haft särskild betydelse:

- DES (Data Encryption Standard)
  Standard från mitten på sjuttiotalet; dominerande i 25 år.

  64 bitars block, 56 bitars nyckel – för litet. Uttömmande sökning möjlig. På väg ut.
- AES (Advanced Encryption Standard)
  Ny standard 2000. 128 bitars block och nyckel.

  Ersätter DES i nya applikationer; god säkerhet.

Fundamental svårighet

Hur förser man varje par av användare med en hemlig nyckel?

Kryptografi för 35 år sedan

Ca 1975 rådde enighet om följande fakta:

Kryptografi för 35 år sedan

Ca 1975 rådde enighet om följande fakta:

- Världsomfattande publika nätverk skulle snart erbjuda snabb och billig kommunikation.

Kryptografi för 35 år sedan

Ca 1975 rådde enighet om följande fakta:

- Världsomfattande publika nätverk skulle snart erbjuda snabb och billig kommunikation.
- För att utnyttja detta för affärs-, myndighets- och privatbruk skulle bra kryptografiska verktyg behövas.

Kryptografi för 35 år sedan

Ca 1975 rådde enighet om följande fakta:

- Världsomfattande publika nätverk skulle snart erbjuda snabb och billig kommunikation.
- För att utnyttja detta för affärs-, myndighets- och privatbruk skulle bra kryptografiska verktyg behövas.
- Existerande verktyg hade stora problem med att distribuera nycklar.

Kryptografi för 35 år sedan

Ca 1975 rådde enighet om följande fakta:

- Världsomfattande publika nätverk skulle snart erbjuda snabb och billig kommunikation.
- För att utnyttja detta för affärs-, myndighets- och privatbruk skulle bra kryptografiska verktyg behövas.
- Existerande verktyg hade stora problem med att distribuera nycklar.

Helt nya idéer behövdes.

Modern kryptografi

“We stand today on the brink of a revolution in cryptography.”

Inledningsfrasen till

New directions in cryptography
Whitfield Diffie and Martin Hellman
IEEE Transactions of Information Theory, Nov 1976.

Diffie-Hellmans nyckelalgoritm

Scenario

Alice surfar in på webbshopen MegaStore; för överföring av känsliga data krävs kryptering – men Alice och MegaStore har ingen gemensam nyckel.

Diffie-Hellmans nyckelalgoritm

Scenario

Alice surfar in på webbshopen MegaStore; för överföring av känsliga data krävs kryptering – men Alice och MegaStore har ingen gemensam nyckel.

Lösning

Diffie-Hellmans nyckelalgoritm

Scenario

Alice surfar in på webbshopen MegaStore; för överföring av känsliga data krävs kryptering – men Alice och MegaStore har ingen gemensam nyckel.

Lösning

1. MegaStore skickar ett stort primtal p och en primitiv rot g till p till Alice.

Diffie-Hellmans nyckelalgoritm

Scenario

Alice surfar in på webbshopen MegaStore; för överföring av känsliga data krävs kryptering – men Alice och MegaStore har ingen gemensam nyckel.

Lösning

1. MegaStore skickar ett stort primtal p och en primitiv rot g till p till Alice.
2. Alice väljer ett slumpmässigt heltal x med $0 < x < p - 1$ och skickar $X = g^x \bmod p$ till MegaStore.

Diffie-Hellmans nyckelalgoritm

Scenario

Alice surfar in på webbshopen MegaStore; för överföring av känsliga data krävs kryptering – men Alice och MegaStore har ingen gemensam nyckel.

Lösning

1. MegaStore skickar ett stort primtal p och en primitiv rot g till p till Alice.
2. Alice väljer ett slumpmässigt heltal x med $0 < x < p - 1$ och skickar $X = g^x \bmod p$ till MegaStore.
3. MegaStore väljer ett slumpmässigt heltal y med $0 < y < p - 1$ och skickar $Y = g^y \bmod p$ till Alice.

Diffie-Hellmans nyckelalgoritm

Scenario

Alice surfar in på webbshopen MegaStore; för överföring av känsliga data krävs kryptering – men Alice och MegaStore har ingen gemensam nyckel.

Lösning

1. MegaStore skickar ett stort primtal p och en primitiv rot g till p till Alice.
2. Alice väljer ett slumpmässigt heltal x med $0 < x < p - 1$ och skickar $X = g^x \bmod p$ till MegaStore.
3. MegaStore väljer ett slumpmässigt heltal y med $0 < y < p - 1$ och skickar $Y = g^y \bmod p$ till Alice.
4. Alice och MegaStore kan bägge beräkna nyckeln

$$K = X^y \bmod p = Y^x \bmod p = g^{xy} \bmod p.$$

Diffie-Hellman – kommentarer

Definition

g är en primitiv rot till p om g^k mod p för $k = 0, 1, \ldots p-2$ alla är olika och alltså ger alla heltalsvärden i intervallet $[1, p-1]$.

Då g är en primitiv rot kan K anta vilket värde som helst i intervallet $[1, p-1]$.

Diffie-Hellman – kommentarer

Definition

g är en primitiv rot till p om $g^k \bmod p$ för $k = 0, 1, \ldots p-2$ alla är olika och alltså ger alla heltalsvärden i intervallet $[1, p-1]$.

Då g är en primitiv rot kan K anta vilket värde som helst i intervallet $[1, p-1]$.

Exempel

Nedanstående tabell visar att 2 är en primitiv rot till 13, medan 3 inte är det.

i	0	1	2	3	4	5	6	7	8	9	10	11
$2^i \bmod 13$	1	2	4	8	3	6	12	11	9	5	10	7
$3^i \bmod 13$	1	3	9	1	3	9	1	3	9	1	3	9

Om stora primtal

Vad är stort?

Med ett “stort” primtal menas i dag typiskt ett tal som i binär form har 1024 bitar, dvs ett tal i storleksordningen 10^{308}.

Om stora primtal

Vad är stort?

Med ett “stort” primtal menas i dag typiskt ett tal som i binär form har 1024 bitar, dvs ett tal i storleksordningen 10^{308}.

Hur hittar man sådana?

Det är lätt att generera stora primtal (vi återkommer till detta).

Om stora primtal

Vad är stort?

Med ett “stort” primtal menas i dag typiskt ett tal som i binär form har 1024 bitar, dvs ett tal i storleksordningen 10^{308}.

Hur hittar man sådana?

Det är lätt att generera stora primtal (vi återkommer till detta).

Tar de inte slut?

Sannolikheten att ett slumpmässigt genererat 1024-bits tal är primtal är ca 1/700. Så det finns ca 10^{305} 1024-bits primtal.

Motståndaren då?

Den som tjuvlyssnar och hör p, g, $X = g^x \bmod p$ och $Y = g^y \bmod p$, kan inte han beräkna K?

Motståndaren då?

Den som tjuvlyssnar och hör p, g, $X = g^x \bmod p$ och $Y = g^y \bmod p$, kan inte han beräkna K?

Den enda metod man känner är att beräkna den diskreta logaritmen av X (dvs x) eller Y (dvs y).

Motståndaren då?

Den som tjuvlyssnar och hör p, g, $X = g^x \bmod p$ och $Y = g^y \bmod p$, kan inte han beräkna K?

Den enda metod man känner är att beräkna den diskreta logaritmen av X (dvs x) eller Y (dvs y).

Att beräkna diskreta logaritmer modulo p där $p \approx 10^{300}$ är ogörligt med i dag kända algoritmer – åtminstone om $p - 1$ har någon stor primfaktor.

Några andra svåra problem

Modern kryptografi bygger i mycket på den förmodade svårigheten hos kända matematiska problem.

Vi ger ytterligare ett par exempel.

Några andra svåra problem

Modern kryptografi bygger i mycket på den förmodade svårigheten hos kända matematiska problem.

Vi ger ytterligare ett par exempel.

Faktorisering

Givet $N = p \cdot q$, där p och q är stora primtal, finn p och q.

Några andra svåra problem

Modern kryptografi bygger i mycket på den förmodade svårigheten hos kända matematiska problem.

Vi ger ytterligare ett par exempel.

Faktorisering

Givet $N = p \cdot q$, där p och q är stora primtal, finn p och q.

Kvadratrötter modulo sammansatta tal

Givet $N = p \cdot q$, där p och q är stora primtal samt $y = x^2 \bmod N$, finn x.

Pohlig-Hellmans chiffer

Ett matematiskt baserat blockchiffer

Låt p vara ett (stort) primtal.
Ett meddelande m är ett heltal $m < p$. Den hemliga nyckeln k är också ett heltal $k < p - 1$. Kryptering ges av

$$c = m^k \bmod p$$

Pohlig-Hellmans chiffer

Ett matematiskt baserat blockchiffer

Låt p vara ett (stort) primtal.
Ett meddelande m är ett heltal $m < p$. Den hemliga nyckeln k är också ett heltal $k < p - 1$. Kryptering ges av

$$c = m^k \bmod p$$

Säkerhet

Om motståndaren fått tag i ett par (m, c) (klartext och motsvarande chiffertext), så kan hon ändå inte räkna ut k
– det är det diskreta logaritmproblemet.

Pohlig-Hellmans chiffer

Ett matematiskt baserat blockchiffer

Låt p vara ett (stort) primtal.
Ett meddelande m är ett heltal $m < p$. Den hemliga nyckeln k är också ett heltal $k < p - 1$. Kryptering ges av

$$c = m^k \bmod p$$

Säkerhet

Om motståndaren fått tag i ett par (m, c) (klartext och motsvarande chiffertext), så kan hon ändå inte räkna ut k
– det är det diskreta logaritmproblemet.

Men . . .

Hur dekrypterar den riktiga mottagaren meddelandet?

Kryptering med öppna nycklar

Diffie och Hellman föreslog också ett helt nytt sätt att kryptera:

Kryptering med öppna nycklar

Diffie och Hellman föreslog också ett helt nytt sätt att kryptera:

- Varje användare har en publik nyckel, som används för kryptering. Denna kan publiceras i kataloger, på webbsidor, osv.

  Fördel: Vem som helst som känner Alices publika nyckel kan skicka ett krypterat meddelande till Alice.

Kryptering med öppna nycklar

Diffie och Hellman föreslog också ett helt nytt sätt att kryptera:

- Varje användare har en publik nyckel, som används för kryptering. Denna kan publiceras i kataloger, på webbsidor, osv.

  Fördel: Vem som helst som känner Alices publika nyckel kan skicka ett krypterat meddelande till Alice.
- Varje användare har också en privat nyckel som används för dekryptering.

Kryptering med öppna nycklar

Diffie och Hellman föreslog också ett helt nytt sätt att kryptera:

- Varje användare har en publik nyckel, som används för kryptering. Denna kan publiceras i kataloger, på webbsidor, osv.

  Fördel: Vem som helst som känner Alices publika nyckel kan skicka ett krypterat meddelande till Alice.
- Varje användare har också en privat nyckel som används för dekryptering.

Tyvärr kunde de inte föreslå något fungerande system. Det första kryptosystemet med öppna nycklar blev RSA 1978.

RSA (Rivest, Shamir, Adleson)

RSA – nycklar

Alice väljer två primtal p och q samt beräknar $N = pq$. Vidare bestämmer hon ett heltal e så att $\gcd(e, (p-1)(q-1)) = 1$. Därmed kan hon beräkna ett tal d så att
$de \equiv 1 (\mathrm{mod}\, (p-1)(q-1))$.

Alices öppna nyckel är (N, e); hennes privata är d.

RSA (Rivest, Shamir, Adleson)

RSA – nycklar

Alice väljer två primtal p och q samt beräknar $N = pq$. Vidare bestämmer hon ett heltal e så att $\gcd(e, (p-1)(q-1)) = 1$. Därmed kan hon beräkna ett tal d så att
$de \equiv 1 (\mathrm{mod}\, (p-1)(q-1))$.

Alices öppna nyckel är (N, e); hennes privata är d.

RSA – kryptering

De meddelanden m som kan krypteras med RSA är naturliga tal $m < N$.

För att kryptera m beräknar man $c = m^e \bmod N$.

RSA (Rivest, Shamir, Adleson)

RSA – nycklar

Alice väljer två primtal p och q samt beräknar $N = pq$. Vidare bestämmer hon ett heltal e så att $\gcd(e, (p-1)(q-1)) = 1$. Därmed kan hon beräkna ett tal d så att
$de \equiv 1 (\mathrm{mod}\, (p-1)(q-1))$.

Alices öppna nyckel är (N, e); hennes privata är d.

RSA – kryptering

De meddelanden m som kan krypteras med RSA är naturliga tal $m < N$.

För att kryptera m beräknar man $c = m^e \bmod N$.

RSA – dekryptering

För att dekryptera c beräknar Alice $m = c^d \bmod N$.

Exponentiering

Hur beräknar man c^d mod N när alla tre ingående talen är av storleksordningen 10^{300}?

Exponentiering

Hur beräknar man c^d mod N när alla tre ingående talen är av storleksordningen 10^{300}?

Exempel

Vi beräknar 21^{29} mod 37.

n	21^n mod 37
1	21
2	$21^2 = 34$
4	$34^2 = 9$
8	$9^2 = 7$
16	$7^2 = 12$

Sedan utnyttjar man att $29 = 16 + 8 + 4 + 1$ och får resultatet $12 \cdot 7 \cdot 9 \cdot 21 = 3$.

Elektroniska signaturer

Ytterligare en ny idé i Diffie-Hellmans uppsats var användning av öppen-nyckelsystem för signering.

Elektroniska signaturer

Ytterligare en ny idé i Diffie-Hellmans uppsats var användning av öppen-nyckelsystem för signering.

RSA – signering

För att signera meddelandet m beräknar Alice $s = m^d \bmod N$, dvs hon "dekrypterar" meddelandet m och skickar både m och s till Bob.

Elektroniska signaturer

Ytterligare en ny idé i Diffie-Hellmans uppsats var användning av öppen-nyckelsystem för signering.

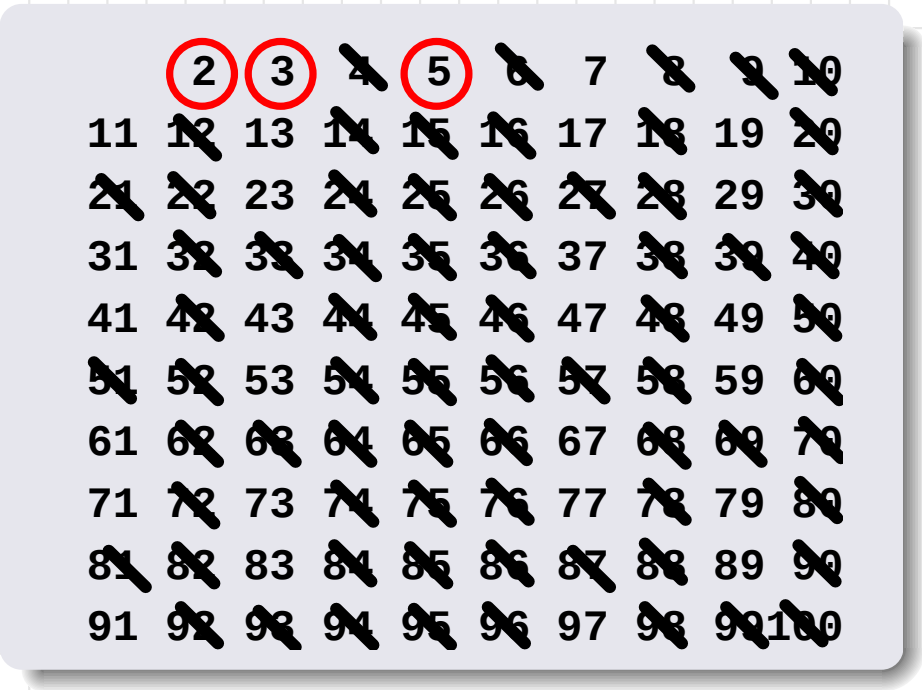
RSA – signering

För att signera meddelandet m beräknar Alice $s = m^d \bmod N$, dvs hon "dekrypterar" meddelandet m och skickar både m och s till Bob.

RSA – verifiering

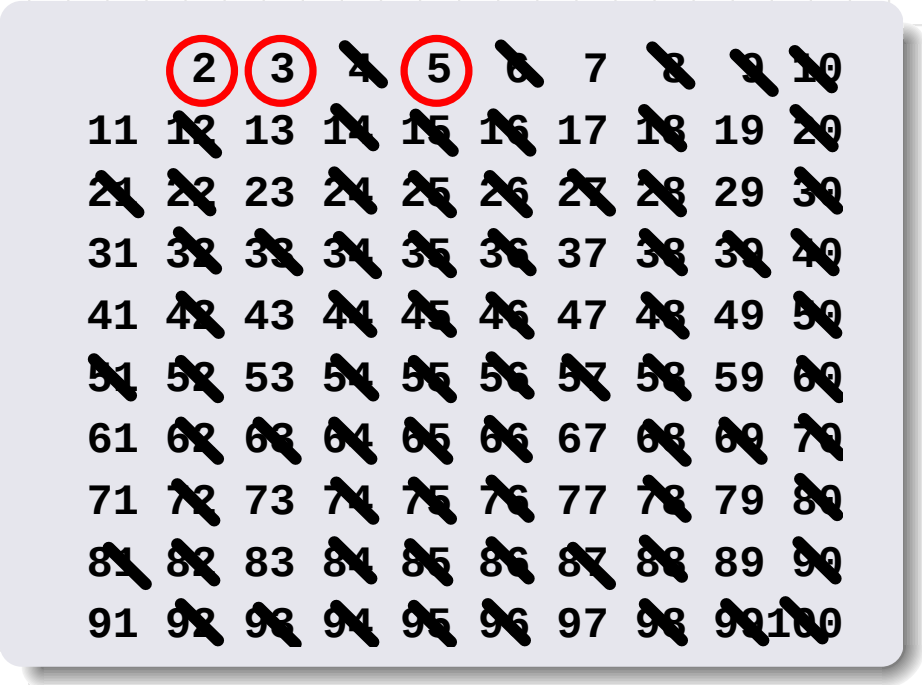
För att verifiera att s är Alices signatur på m beräknar Bob $s^e \bmod N$ och kontrollerar att han får m.

Hur hittar man stora primtal?



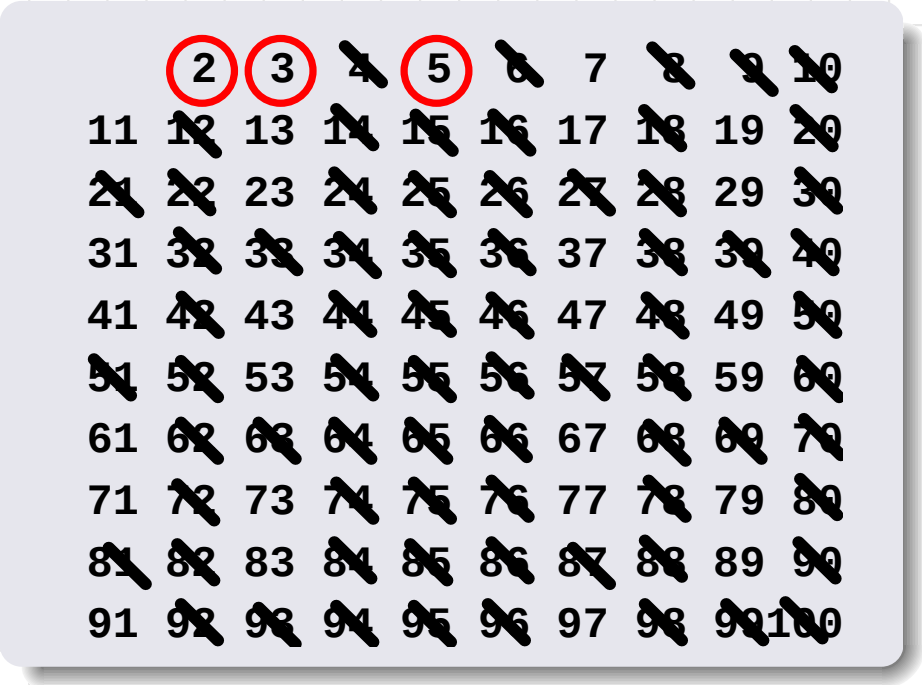
Hur hittar man stora primtal?

- 200 f.Kr. Eratosthenes. Sålla fram primtal.



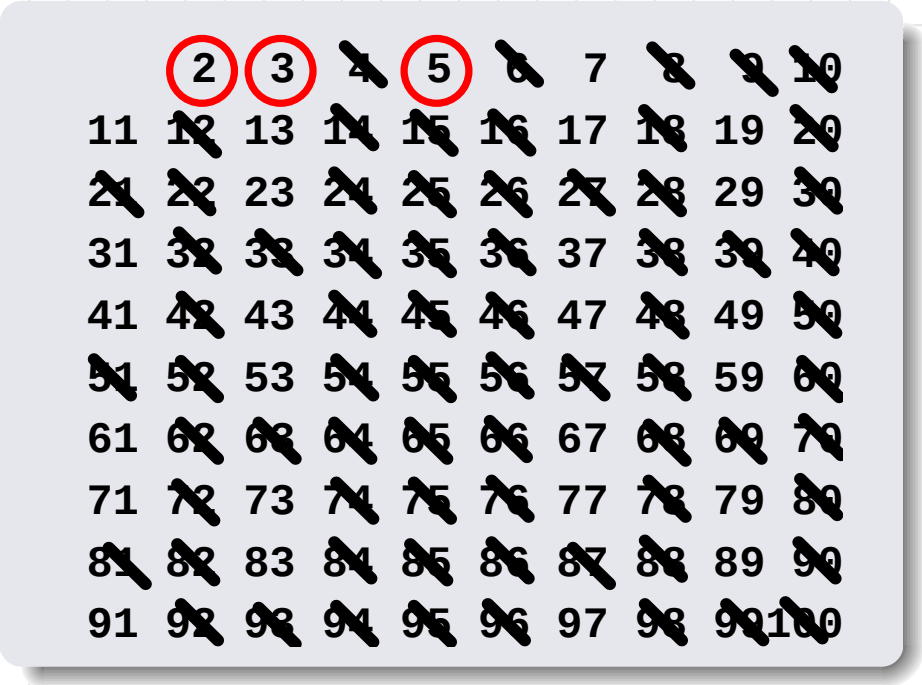
Hur hittar man stora primtal?

- 200 f.Kr. Eratosthenes.
  Sålla fram primtal.
- 1200 e.Kr. Fibonacci.
  För att hitta primtal nära N,
  sålla fram till $\sqrt{N}$.



Hur hittar man stora primtal?

- 200 f.Kr. Eratosthenes.
  Sålla fram primtal.
- 1200 e.Kr. Fibonacci.
  För att hitta primtal nära N,
  sålla fram till $\sqrt{N}$.
- Ca 1975 (Miller, m.fl.)
  Probabilistiska test.



Probabilistiska test

Fermat's test

För att testa om n är primtal, upprepa följande t gånger:

1. Generera slumpmässigt $a < n$ och beräkna $z = a^{n-1} \bmod n$.
2. Om $z \neq 1$, avbryt: n är ej primtal.

Om vi inte avbrutit efter t försök, tror vi att n är primtal.

Probabilistiska test

Fermat's test

För att testa om n är primtal, upprepa följande t gånger:

1. Generera slumpmässigt $a < n$ och beräkna $z = a^{n-1} \bmod n$.
2. Om $z \neq 1$, avbryt: n är ej primtal.

Om vi inte avbrutit efter t försök, tror vi att n är primtal.

Egenskaper

Probabilistiska test

Fermat's test

För att testa om n är primtal, upprepa följande t gånger:

1. Generera slumpmässigt $a < n$ och beräkna $z = a^{n-1} \bmod n$.
2. Om $z \neq 1$, avbryt: n är ej primtal.

Om vi inte avbrutit efter t försök, tror vi att n är primtal.

Egenskaper

- Man vill välja t så att sannolikheten för att påstå att ett sammansatt tal n är primtal blir "liten".

Probabilistiska test

Fermat's test

För att testa om n är primtal, upprepa följande t gånger:

1. Generera slumpmässigt $a < n$ och beräkna $z = a^{n-1} \bmod n$.
2. Om $z \neq 1$, avbryt: n är ej primtal.

Om vi inte avbrutit efter t försök, tror vi att n är primtal.

Egenskaper

- Man vill välja t så att sannolikheten för att påstå att ett sammansatt tal n är primtal blir "liten".
- Det går tyvärr inte; om n är ett Carmichaeltal ger Fermat's test fel resultat.

Probabilistiska test

Fermat's test

För att testa om n är primtal, upprepa följande t gånger:

1. Generera slumpmässigt $a < n$ och beräkna $z = a^{n-1} \bmod n$.
2. Om $z \neq 1$, avbryt: n är ej primtal.

Om vi inte avbrutit efter t försök, tror vi att n är primtal.

Egenskaper

- Man vill välja t så att sannolikheten för att påstå att ett sammansatt tal n är primtal blir "liten".
- Det går tyvärr inte; om n är ett Carmichaeltal ger Fermat's test fel resultat.
- Små förbättringar ger Miller-Rabins algoritm, där sannolikheten för fel slutsats kan göras godtyckligt liten.

Vad händer i framtiden?

Kapplöpningen

Snabbare datorer och bättre algoritmer för faktorisering, diskret log osv framtvingar längre och längre nycklar och därmed alltmer krävande beräkningar.
Kan man göra något annat?

Vad händer i framtiden?

Kapplöpningen

Snabbare datorer och bättre algoritmer för faktorisering, diskret log osv framtvingar längre och längre nycklar och därmed alltmer krävande beräkningar.
Kan man göra något annat?

Generaliserad diskret log

Låt G vara en mängd, $\otimes$ någon operator på G och g^n betyda

$$g^n = \underbrace{g \otimes g \otimes g \otimes \ldots \otimes g}_{(n \text{ ”faktorer”})}$$

Diskret log-problemet är då att, givet g och $y = g^n$, finna n.

Kanske man kan hitta ett bättre (effektivare, säkrare) val av $\otimes$ än multiplikation modulo p?

Framtiden är redan här!

I många nya standarder (Bluetooth, SSL,. . .) är G en elliptisk kurva över en ändlig kropp.

Mer sofistikerad matematik, men samma idé, enkla beräkningar och mycket effektivare.

Framtiden är redan här!

I många nya standarder (Bluetooth, SSL,. . .) är G en elliptisk kurva över en ändlig kropp.

Mer sofistikerad matematik, men samma idé, enkla beräkningar och mycket effektivare.

Från Bluetooth simple pairing

G består av de par (x, y) av heltal modulo p som uppfyller $y^2 = x^3 - 3x + b \bmod p$, där

$p =$
627710173538668076383578942320766641608390870039032496127 9

$b =$ 64210519e59c80e70fa7e9ab72243049feb8deecc146b9b1.

Operationen $\otimes$ ges av explicita formler med modulär aritmetik, motiverade av geometriska överväganden.

Inlogging med `ssh`

Förberedelser: Alice genererar ett nyckelpar för RSA och lagrar den publika nyckeln på fastställd plats på den dator H hon vill kunna logga in på.

Inlogging med `ssh`

Förberedelser: Alice genererar ett nyckelpar för RSA och lagrar den publika nyckeln på fastställd plats på den dator H hon vill kunna logga in på.

Inloggningsprotokoll

Inlogging med `ssh`

Förberedelser: Alice genererar ett nyckelpar för RSA och lagrar den publika nyckeln på fastställd plats på den dator H hon vill kunna logga in på.

Inloggningsprotokoll

1. Alice kontaktar H och anger sin identitet och att hon vill logga in.

Inlogging med `ssh`

Förberedelser: Alice genererar ett nyckelpar för RSA och lagrar den publika nyckeln på fastställd plats på den dator H hon vill kunna logga in på.

Inloggningsprotokoll

1. Alice kontaktar H och anger sin identitet och att hon vill logga in.
2. H skickar en slumpmässig utmaning u till Alice.

Inlogging med `ssh`

Förberedelser: Alice genererar ett nyckelpar för RSA och lagrar den publika nyckeln på fastställd plats på den dator H hon vill kunna logga in på.

Inloggningsprotokoll

1. Alice kontaktar H och anger sin identitet och att hon vill logga in.
2. H skickar en slumpmässig utmaning u till Alice.
3. Alice signerar u och skickar signaturen till H.

Inlogging med `ssh`

Förberedelser: Alice genererar ett nyckelpar för RSA och lagrar den publika nyckeln på fastställd plats på den dator H hon vill kunna logga in på.

Inloggningsprotokoll

1. Alice kontaktar H och anger sin identitet och att hon vill logga in.
2. H skickar en slumpmässig utmaning u till Alice.
3. Alice signerar u och skickar signaturen till H.
4. H verifierar signaturen och accepterar login.

e-legitimation

Exempel på tjänster som tillhandahålls över nätet:

e-legitimation

Exempel på tjänster som tillhandahålls över nätet:

- Anmälan om uttag av föräldraledighet.

e-legitimation

Exempel på tjänster som tillhandahålls över nätet:

- Anmälan om uttag av föräldraledighet.
- Självdeklaration och information om personligt skattekonto.

e-legitimation

Exempel på tjänster som tillhandahålls över nätet:

- Anmälan om uttag av föräldraledighet.
- Självdeklaration och information om personligt skattekonto.
- Personlig pensionsprognos.

e-legitimation

Exempel på tjänster som tillhandahålls över nätet:

- Anmälan om uttag av föräldraledighet.
- Självdeklaration och information om personligt skattekonto.
- Personlig pensionsprognos.

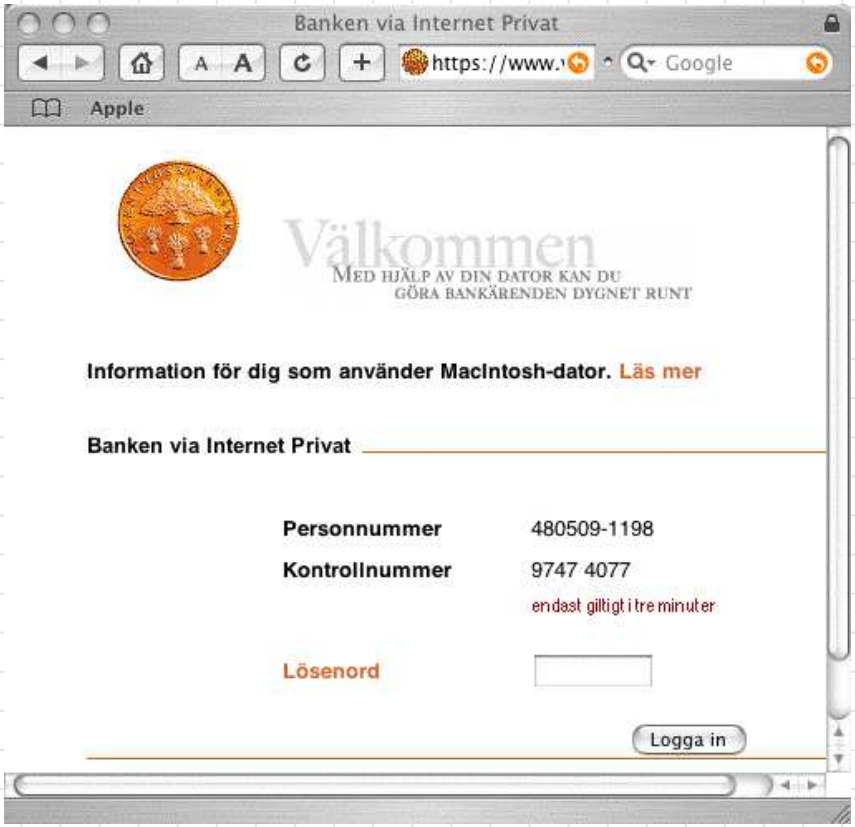
Dessa kräver identifiering av användaren.

Lösning: e-legitimation, en digital signatur.

Teknologi: RSA-signatur med 1024 bitars nyckel skyddad av 12 teckens lösenord.

Internetbank

Typiskt scenario för bankservice över Internet:
Kundens webbläsare kommunicerar med banken;
dessutom har kunden en säkerhetsdosa.



Banksäkerhet 1

Kommunikationen mellan webbläsare och bank sker över en SSL-förbindelse; alla data är krypterade. Data krypteras med blockchiffer; nycklar överenskoms med öppen-nyckelmetoder.

Banksäkerhet 1

Kommunikationen mellan webbläsare och bank sker över en SSL-förbindelse; alla data är krypterade. Data krypteras med blockchiffer; nycklar överenskoms med öppen-nyckelmetoder.

En SSL-förbindelse betraktas som säker mot tjuvlyssning.

Det återstår att skydda bank och kund mot varandra.

Banksäkerhet 2

Säkerhetsdosan har följande funktionalitet

Banksäkerhet 2

Säkerhetsdosan har följande funktionalitet

1. För att aktivera den måste användaren ge sin PIN-kod.

Banksäkerhet 2

Säkerhetsdosan har följande funktionalitet

1. För att aktivera den måste användaren ge sin PIN-kod.
2. I aktiv mod kan den beräkna ett åttasiffrigt svar på en åttasiffrig utmaning.

Banksäkerhet 2

Säkerhetsdosan har följande funktionalitet

1. För att aktivera den måste användaren ge sin PIN-kod.
2. I aktiv mod kan den beräkna ett åttasiffrigt svar på en åttasiffrig utmaning.

Svaret beräknas genom att kryptera utmaningen med ett blockchiffer (3DES) med användande av dosans hemliga kod.

Banken känner till den hemliga koden och kan kontrollera svaret.

För andra är det ogörligt att beräkna svaret.

Banksäkerhet 3

Inloggningsprotokoll

Banksäkerhet 3

Inloggningsprotokoll

1. Kund → Bank: Namn (el personnummer, el kontonummer).

Banksäkerhet 3

Inloggningsprotokoll

1. Kund $\rightarrow$ Bank: Namn (el personnummer, el kontonummer).
2. Bank $\rightarrow$ Kund: Slumpmässig utmaning.

Banksäkerhet 3

Inloggningsprotokoll

1. Kund $\longrightarrow$ Bank: Namn (el personnummer, el kontonummer).
2. Bank $\longrightarrow$ Kund: Slumpmässig utmaning.
3. Kund $\longrightarrow$ Bank: Svar beräknat av dosan.

Banksäkerhet 3

Inloggningsprotokoll

1. Kund $\longrightarrow$ Bank: Namn (el personnummer, el kontonummer).
2. Bank $\longrightarrow$ Kund: Slumpmässig utmaning.
3. Kund $\longrightarrow$ Bank: Svar beräknat av dosan.
4. Banken kontrollerar svaret och accepterar login.

Banksäkerhet 3

Inloggningsprotokoll

1. Kund $\rightarrow$ Bank: Namn (el personnummer, el kontonummer).
2. Bank $\rightarrow$ Kund: Slumpmässig utmaning.
3. Kund $\rightarrow$ Bank: Svar beräknat av dosan.
4. Banken kontrollerar svaret och accepterar login.

Betal-TV

Typiskt scenario för betal-TV:

Betal-TV

Typiskt scenario för betal-TV:

- Systemet använder en vanlig TV ansluten till en box som innehåller ett smart card.

Betal-TV

Typiskt scenario för betal-TV:

- Systemet använder en vanlig TV ansluten till en box som innehåller ett smart card.
- Den signal som sänds är krypterad (permuterade, inverterade linjer). Dekrypteras av boxen. Men, om man redan har boxen, varför betala prenumerationsavgiften?

Betal-TV

Typiskt scenario för betal-TV:

- Systemet använder en vanlig TV ansluten till en box som innehåller ett smart card.
- Den signal som sänds är krypterad (permuterade, inverterade linjer). Dekrypteras av boxen. Men, om man redan har boxen, varför betala prenumerationsavgiften?

- Med några sekunders mellanrum instruerar sändningen boxen att kontrollera kortets giltighet och stänga av sig om kortet är ogiltigt.

Hur kontrollerar man kortet?

Krav

Hur kontrollerar man kortet?

Krav

- Kortet ska kunna demonstrera sin giltighet på ett sätt som gör det svårt att tillverka piratkort.

Hur kontrollerar man kortet?

Krav

- Kortet ska kunna demonstrera sin giltighet på ett sätt som gör det svårt att tillverka piratkort.
- Leverantören måste kunna svartlista/stänga av kort som stulits etc.

Hur kontrollerar man kortet?

Krav

- Kortet ska kunna demonstrera sin giltighet på ett sätt som gör det svårt att tillverka piratkort.
- Leverantören måste kunna svartlista/stänga av kort som stulits etc.

Vag idé

Leverantören bestämmer två stora primtal p och q och beräknar $N = pq$. Varje kort har en privat nyckel x och en publik nyckel $X = x^2 \bmod N$.

Kortet identifierar sig genom att visa att det känner till x – utan att avslöja x.

Försök 1

Varje gång kortet ska identifiera sig genererar det ett slumpmässigt r och skickar $Z = rx \bmod N$ till boxen.

För att boxen ska kunna kontrollera kortet skickas också $R = r^2 \bmod N$.

Försök 1

Varje gång kortet ska identifiera sig genererar det ett slumpmässigt r och skickar $Z = rx \bmod N$ till boxen.

För att boxen ska kunna kontrollera kortet skickas också $R = r^2 \bmod N$.

Boxen kontrollerar att $Z^2 \equiv R \cdot X \pmod{N}$. Om detta gäller tror boxen att kortet känner till x.

Boxen (eller tjuvlyssnaren) kan inte beräkna vare sig x eller r.

Försök 1

Varje gång kortet ska identifiera sig genererar det ett slumpmässigt r och skickar $Z = rx \bmod N$ till boxen.

För att boxen ska kunna kontrollera kortet skickas också $R = r^2 \bmod N$.

Boxen kontrollerar att $Z^2 \equiv R \cdot X \pmod{N}$. Om detta gäller tror boxen att kortet känner till x.

Boxen (eller tjuvlyssnaren) kan inte beräkna vare sig x eller r.

Problem

Kortet kan fuska: Generera Z på måfå och beräkna $R = Z^2/X \pmod{N}$.

Fiat-Shamirs metod

När kortet ska identifiera sig sker följande:

Fiat-Shamirs metod

När kortet ska identifiera sig sker följande:

- Kortet genererar r slumpmässigt och skickar $R = r^2$ mod N til boxen.

Fiat-Shamirs metod

När kortet ska identifiera sig sker följande:

- Kortet genererar r slumpmässigt och skickar $R = r^2$ mod N til boxen.
- Boxen genererar en slumpmässig bit b och skickar till kortet.

Fiat-Shamirs metod

När kortet ska identifiera sig sker följande:

- Kortet genererar r slumpmässigt och skickar $R = r^2$ mod N til boxen.
- Boxen genererar en slumpmässig bit b och skickar till kortet.
- Om $b = 0$ så skickar kortet $Z = r$, annars skickas $Z = rx$.

Fiat-Shamirs metod

När kortet ska identifiera sig sker följande:

- Kortet genererar r slumpmässigt och skickar $R = r^2$ mod N til boxen.
- Boxen genererar en slumpmässig bit b och skickar till kortet.
- Om $b = 0$ så skickar kortet $Z = r$, annars skickas $Z = rx$.
- Om $b = 0$ så kontrollerar boxen att $Z^2 = R$, annars att $Z^2 = RX$.

Fiat-Shamirs metod

När kortet ska identifiera sig sker följande:

- Kortet genererar r slumpmässigt och skickar $R = r^2$ mod N til boxen.
- Boxen genererar en slumpmässig bit b och skickar till kortet.
- Om $b = 0$ så skickar kortet $Z = r$, annars skickas $Z = rx$.
- Om $b = 0$ så kontrollerar boxen att $Z^2 = R$, annars att $Z^2 = RX$.

Ett falskt kort har sannolikhet 0.5 att lyckas lura boxen (genom att gissa b).

Om det tvingas identifiera sig med några sekunders mellanrum så kommer ett falskt kort snart att avslöjas.

Slutord

Vi har sett på något av kryptografins grunder och tillämpningar, men det finns mycket mer:

Slutord

Vi har sett på något av kryptografins grunder och tillämpningar, men det finns mycket mer:

- Tillämpningar: Mobiltelefoni, Bluetooth, trådlösa nätverk, . . .

Slutord

Vi har sett på något av kryptografins grunder och tillämpningar, men det finns mycket mer:

- Tillämpningar: Mobiltelefoni, Bluetooth, trådlösa nätverk, . . .
- Matematik med kryptoanknytning: Primtalstest, algoritmer för faktorisering och diskreta logaritmer, . . .

Slutord

Vi har sett på något av kryptografins grunder och tillämpningar, men det finns mycket mer:

- Tillämpningar: Mobiltelefoni, Bluetooth, trådlösa nätverk, . . .
- Matematik med kryptoanknytning: Primtalstest, algoritmer för faktorisering och diskreta logaritmer, . . .
- Kryptografiska primitiver och begrepp: hashfunktioner, strömchiffer, perfekt säkerhet, . . .

Slutord

Vi har sett på något av kryptografins grunder och tillämpningar, men det finns mycket mer:

- Tillämpningar: Mobiltelefoni, Bluetooth, trådlösa nätverk, . . .
- Matematik med kryptoanknytning: Primtalstest, algoritmer för faktorisering och diskreta logaritmer, . . .
- Kryptografiska primitiver och begrepp: hashfunktioner, strömchiffer, perfekt säkerhet, . . .

En översikt av detta ges i kursen Cryptography.