Heltalsaritmetik del 2: Eulers Φ-funktion och RSA-krypto

Johan Jonasson

October 2020

Definition

Eulers Φ-funktion ges av

$$\Phi(n) = |\{1 \leq b \leq n-1 : sgd(b,n) = 1\}|.$$

Med andra ord är $\Phi(n)$ antalet heltal b mellan 1 och $n-1$ sådana att $[b]^{-1}$ existerar i $\mathbb{Z}_n$.

Omedelbar observation: p primtal $\Rightarrow \Phi(p) = p-1$.

Observation inför nästa bevis: Om $sgd(m,n) = 1$ gäller att $sgd(a,mn) = 1 \Leftrightarrow sgd(a,n) = sgd(a,m) = 1$, ty:

$\Rightarrow$ är trivial. $\Leftarrow$: Antag att $d|a$ och $d > 1$. Eftersom $sgd(a,m) = sgd(a,n) = 1$ gäller att $sgd(d,m) = sgd(d,n) = 1$. Antag nu att $d|mn$. Då har vi att $d|mn$ och $sgd(d,m) = 1$. Då säger Sats 5 att $d|n$, en motsägelse.

Theorem

Sats 10: *Om* $sgd(m, n) = 1$ *gäller att* $\Phi(mn) = \Phi(m)\Phi(n)$.

Bevis: Låt $a_1, a_2, \ldots, a_{\Phi(m)}$ vara de heltal mellan 1 och $m - 1$ som är relativt prima med m.

Då är

$$a_k + im,\ k = 1, \ldots, \Phi(m),\ i = 0, \ldots, n - 1$$

de tal mellan 1 och $mn - 1$ som är relativt prima med m. Vi vill veta hur många av dessa som är relativt prima med mn. Enligt observationen ovan är dessa precis de som är relativt prima med n.

Det gäller att då $i > j$ är

$$a_k + im \not\equiv a_k + jm \ mod \ n$$

ty om $n|(a_k + im) - (a_k + jm)$ gäller att $n|m(i-j)$, så enligt Sats 5 gäller $n|i-j$, en motsägelse ty $i - j \leq n - 1$.

Slutsats: elementen $a_k, a_k + m, a_k + 2m, \ldots, a_k + (n-1)m$ har alla olika rester modulo n, dvs de representerar varsitt av elemenen i $\mathbb{Z}_n$. Exakt $\Phi(n)$ av dessa är relativt prima med n. Detta gäller för alla k, så totalt blir det $\Phi(n)\Phi(m)$ element som är relativt prima också med n. □

Följd: om p och q är två olika primtal gäller

$$\Phi(pq) = (p-1)(q-1).$$

Theorem

Eulers sats: *Om* $sgd(a, n) = 1$ *gäller i* $\mathbb{Z}_n$ *att*

$$[a]^{\Phi(n)} = [1]$$

Bevis: Låt $x_1, x_2, \ldots, x_{\Phi(n)}$ vara de tal mellan 1 och $n-1$ som är relativt prima med n. Eftersom $sgd(a, n) = 1$ gäller enligt Sats 5 att $ax_1, ax_2, \ldots, ax_{\Phi(n)}$ alla är relativt prima med n, ty om $d|n$ gäller $sgd(d, a) = 1$ och om $d|ax_k$ säger Sats 5 att $d|x_k$, varför d måste vara 1.

Det gäller också att $ax_i \not\equiv ax_j$ mod n, ty om $n|ax_i - ax_j$ medför detta enligt Sats 5 att $n|x_i - x_j$, vilket är en motsägelse eftersom $x_i - x_j < n$.

Alltså gäller

$$\{[x_1], [x_2], \ldots, [x_{\Phi(n)}]\} = \{[ax_1], [ax_2], \ldots, [ax_{\Phi(n)}]\}$$

sedda som mängder av element i $\mathbb{Z}_n$ (båda sidor utgör nämligen

Vid räkning i $\mathbb{Z}_n$ gäller därför att

$$[ax_1][ax_2]\cdots[ax_{\Phi(n)}] = [x_1][x_2]\cdots[x_{\Phi(n)}]$$

vilket ger

$$[a]^{\Phi(n)}[x_1x_2\cdots x_{\Phi(n)}] = [x_1x_2\cdots x_{\Phi(n)}].$$

Eftersom x_i:na alla är relativt prima med n så är även $x_1x_2\cdots x_{\Phi(n)}$ relativt prima med n och har därför en invers som element i $\mathbb{Z}_n$. Multiplicera bägge sidor med denna invers och få

$$[a]^{\Phi(n)} = [1].$$

Ett specialfall av Eulers sats är Fermats lilla sats som säger att om p är ett primtal så gäller $a^{p-1} \equiv 1 \bmod p$.

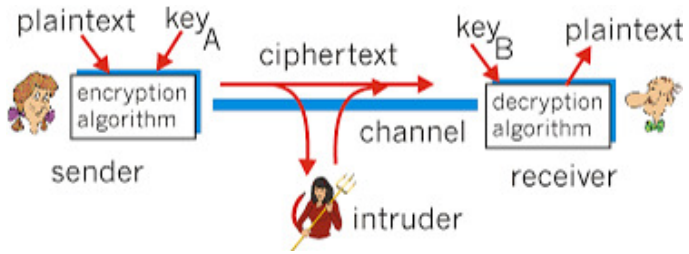
Exempel: Vad är $[15]^{100} + [64]^{200} + [33]^{800}$ i $\mathbb{Z}_{101}$?

Eftersom 101 är ett primtal får vi

$$[15]^{100} + [64]^{200} + [33]^{800} = [1] + [1]^2 + [1]^8 = [3].$$

RSA-krypto

RSA: Ron Rivest, Adi Shamir och Len Adleman (1977).



Alice vill sända ett meddelande till Bob på ett sådant sätt att

- ingen annan än Bob kan läsa,
- Bob ska veta att det var Alice som skickade,
- Eve ska inte kunna manipulera.

Det är också viktigt att Alice och Bob inte behöver träffas för att utbyta nycklar. RSA löser alltihop!

Alla användare skaffar sig två mycket stora primtal (ca 300 siffror), p och q, och ett tal a sådant att
$sgd(a, \Phi(pq)) = sgd(a, (p-1)(q-1)) = 1$ och publicerar pq och a, den s.k. *öppna nyckeln*.

Viktig observation: primtalsfaktorisering är mycket svårt, så ingen kan beräkna p och q genom att veta pq och därmed heller inte $\Phi(pq)$. Däremot är det lätt att kontrollera om ett tal är ett primtal och att kontrollera om $sgd(a, \Phi(pq)) = 1$ (Euklides algoritm).

Användaren beräknar inversen b till a modulo $\Phi(pq)$. Detta går snabbt med EUA, men går inte att göra om man inte känner till $\Phi(pq)$. Talet b hålls hemligt.

Alice vill skicka x till Bob. Här är x ett heltal som är mindre än alla p, q, a och b i systemet. Alice skickar

$$y = x^{a_B} \ mod \ p_B q_B.$$

Bob beräknar

$$z = y^{b_B} \ mod \ p_B q_B.$$

Det gäller att $sgd(x, p_B q_B) = 1$ (ty $x < p_B, q_B$), så enligt Eulers sats gäller $x^{\Phi(p_B q_B)} \equiv 1 \bmod p_B q_B$. Eftersom b_B är invers till a_B modulo $\Phi(p_B q_B)$ kan vi skriva

$$a_B b_B = 1 + t\Phi(p_B q_B)$$

för ett heltal t.

Räkning i $\mathbb{Z}_{p_B q_B}$ ger nu

$$z = (x^{a_B})^{b_B} = x^{1+t\Phi(p_B q_B)} = x.$$

Bara Bob känner till $\Phi(p_B q_B)$ så det är bara han som kan göra den här beräkningen, så Eve kan inte tjuvläsa.

Hur säkerställer vi att det verkligen var Alice som skickade meddelandet och att det inte manipulerats?
Använd en *hashfunktion* h som uppfyller

- h är svår att invertera, dvs att veta $h(x)$ avslöjar inte x,
- olika meddelanden, x_1 och x_2 har (åtminstone nästan alltid) $h(x_1) \neq h(x_2)$.

Skriv $u = h(x)$. Som signatur till sitt meddelande skickar Alice

$$w = u^{b_A} \ mod \ p_A q_A.$$

Bob beräknar w^{a_A} modulo $p_A q_A$. Detta ger vid räkning i $\mathbb{Z}_{p_A q_A}$:

$$w^{a_A} = u^{a_A b_A} = u^{1+s\Phi(p_A q_A)} = u.$$

Till sist beräknar Bob $h(x)$ för det x han avläst. Detta ska då överensstämma med u.

Ingen annan än Alice hade kunnat beräkna w.

Man kan också dubbelkryptera:

- Alice beräknar $y = x^{b_A}$ modulo $p_A q_A$.
- Alice beräknar sedan $w = y^{a_B}$ modulo $p_B q_B$ och skickar till Bob.
- Bob beräknar w^{b_B} modulo $p_B q_B$ och får y.
- Bob beräknar till sist y^{a_A} modulo $p_A q_A$ och får x.

Tyngre beräkningar men ännu lite säkrare.