**DDU onsdag 4-jan-2023 Cybersikkerhed**

**Poul Erik Thomsen (PET)**

**Off-topic:** (sådan da) <https://danskdatalogidyst.dk/dyst2>

I skal være født efter 30.6.2023

Dette er en konkurrence hvor man skal lave effektive og korrekte programmer.  
Dyst2 er startet. Man har 7 dage til at løse opgaver. Slutter 16. jan 2023 23:59  
Sidste år var der 5 ud af ca 35 fra HCØ i Lyngby

**Mere indenfor skiven:**

Cybermesterskaberne

Det er en lang process med flere konkurencer og hvis man går hele vejen kommer man på cyberlandsholdet.

<https://www.cybermesterskaberne.dk/kvalifikation/>

Kvalifikationen starter den 25. februar. Der plejer at være en mængde online træningssessioner.  
Man skal være mellem 16 og 25 for at kunne komme på landsholdet.

Kopi fra Discord:

(Skrevet af Jens Myrup, der blandt meget andet er landstræner for cyberlandsholdet)  
Jeg tillader mig at tagge @everyone - for her er datoerne for DDC 2023! Qualifiers starter 25. februar og slutter 19. marts. Regionals bliver 15. april, og Nationals bliver 6. maj. Til Nationals udtager vi 25 til bootcamp, som finder sted den 9.-11. juni. HVIS man kommer videre til Landsholdet, så er første meetup den 2. juli, og tre træningsweekender: 18.-20. august, 8.-10. september, 6.-8. oktober. Og ECSC er den 24.-28. oktober i Norge!

Hvis jeg kommer til at bruge udtrykket CTF så er det meget kendt af nogle af jer, men nok ikke alle.  
Det står for ”Capture The Flag” og er opgaver hvor man på lovlig vis kan træne sine cyberskills evner. Der findes mange konkurrencer i det. Personligt havde jeg for lidt over et år siden aldrig hørt om det, men jeg synes det er vildt sjovt at løse disse opgaver, selvom jeg ikke er særlig god til det.

**Nu til sagen:**

Der vil være opgaver i dag. Jeg har lavet 7 af forskellig sværhedsgrad og forskellige emner.  
Disse opgaver skulle ligge på skemabrikken men 5 af dem ligger inde på Aalborg Universitets opgaveplatform kaldet Haaukins. Når vi kommer til opgaverne giver jeg en kort introduktion til Haaukins. Mange af jer har brugt Haaukins mange gange, så I kan bare bruge det som i plejer.  
Der ligger forøvrigt også lidt yderligere opgaver derinde.  
Der er 2 links. En til dem der gerne vil bruge VPN og en til dem der bare vil igang. Opgaverne i de to links er de samme.  
<https://hcoddu1.haaukins.com/scoreboard>  
<https://hcoddu1vpn.haaukins.com/>

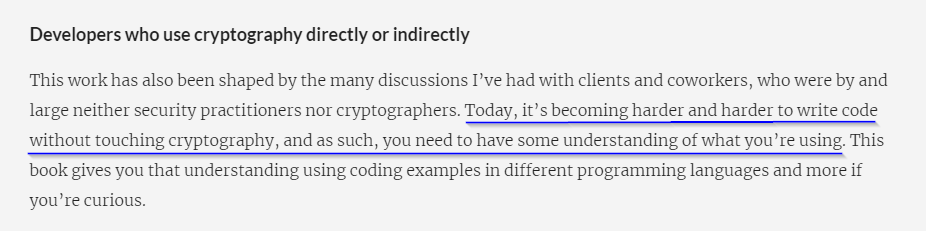
Bemærk at disse to allerede slutter i morgen, så I må gerne notere jeres løsninger ned, da jeg fredag morgen laver nogle nye som indeholder de samme opgaver plus nogle flere.   
Dette skyldes at det vist nok er første gang at der er nogen udefra som har fået lagt opgaver ind på Haaukins, så I er lidt forsøgskaniner.  
I kan sagtens løse opgaverne uden Haaukins, men derinde kan I få checket jeres flag og få point for det. I kan også få checket flagene hos mig.

Forøvrigt er det sandsynligvis på Haaukins at cybermesterskaberne bliver afholdt. Det plejer det i hvert fald at være.

**I dag:**

Password – cracking – evt finde dem ved undersøgelser på nettet eller social engineering

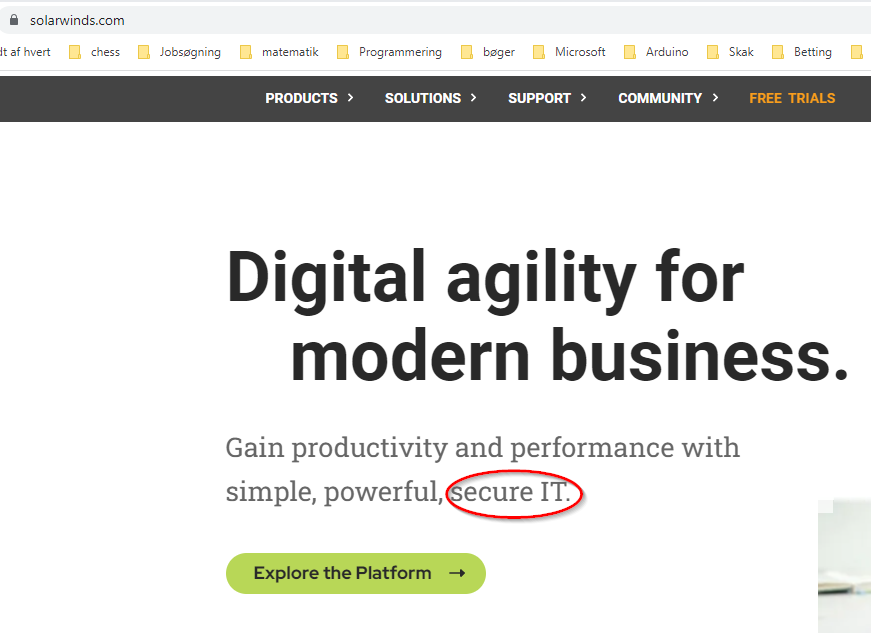
Kryptering

****

**Password:**

Dårlige passwords er bestemt noget af der kan udnyttes i cybercrime.

Solarwinds: Password: Solarwinds123 - Det lyder simpelthen utroligt.



I kan checke jeres egen mailkonti her og endda også jeres password. Siten skulle være god nok, altså ikke en hacker-site.

<https://haveibeenpwned.com/>

Den nok mest berømte passwordliste hedder **rockyou** og stammer fra et såkaldt data-breach i 2009 hvor der var mange 14 millioner password gemt i plaintekst format, altså helt læsbart.  
Deres password policy har ikke været særlig stran da der foreksempel var ”123456” ”qwerty” og mange af samme skuffe. Listen er dog utroligt effektiv som udgangspunkt for nutidige password-cracks.  
Det lyder næsten for uhyggeligt til at være sandt at passwordene var gemt i almindelig klartekst.

Hvis I i jeres program gemmer password så sørg for at de ikke bliver gemt i plaintext format. I skal encrypte dem med en af de kendte routiner (MD5, SHA256 eller andet)

Og så skal i selvfølgelig skifte jeres eget password, hvis der har været en data-breach.  
Meget få har forskellige password til hver eneste konto. I hvert fald ikke mig.  
Personligt har jeg flere password alt efter hvor vigtig kontoen er og så bruger jeg MITID hvor jeg kan.

Jeg har (som del af mit tidligere job) lavet mange passwordcrackinger og det er vildt hvor mange password der hedder noget i retning af Pernille1 Anders01 Aberdeen1 Letmein1 Platform1  
Med en smule kendskab til personen har man faktisk en god chance for at gætte passwordet.

Hvis man har adgang til en fil der indeholder encryptede passwords eller krypterede dokumenter er der mange tools der kan hjælpe med at cracke disse password.  
Det mest effektive er at bruge ordlister som udgangspunkt og så modificere alle ordene i listen. Så er der en god chance for at finde nogle af passwordene.  
Der er mange af den slags tool. Jeg har på windows aftest hashcat som I kan finde her:

<https://hashcat.net/hashcat/>

Hashcat er vist nok som standard installeret på Kali Linux, så dem der har sådan en er klar med det samme. En hashcat begynder video findes her:

<https://www.youtube.com/watch?v=rRUiRz_znLQ>

Denne video viser avanceret brug af hashcat. Temmelig komprimeret men god.

<https://www.youtube.com/watch?v=m0AGSO1LDJs&t=247s>

Der findes mange andre såsom:

John The Ripper Cain&Abel (denne brugte jeg i mit tidligere job)

Der er 3 opgaver som kan løses ved informationssøgning på nettet blandet med  
Derudover er der en opgave hvor der er en fil med 26 password som er MD5 encryptede.  
Der er et hint til et af passwordene i opgaveteksten.

Hvem kan finde flest. Hvis I kan finde mere end 20 er det flot.  
Jeg håber ikke at der er nogen der kan finde alle 26, da et af passwordene er i samme stil som et af mine egne password.

I kan bruge:

google-søgning  
online tools  
password cracking tool – ordliste, hybrid,bruteforce  
Rainbow tables hvis I kan finde ud af at downloade dem.

Lige lidt afklaring:

Hashing: matematisk process som ikke kan laves baglæns. Det er hele ideen.  
Encoding/Decoding: En omformning af tekst så den er lettere at transmitere over nettet.  
Men meningen er at man kan decode den og komme tilbage til den oprindelige tekst.  
Der findes en del. Den mest kendte er base64, men der er også base32.  
  
Base32 består af store bogstaver og tal og slutter nogle gange på et eller flere =  
Base64 består af store og små bogstaver og tal og slutter nogle gange på =  
Base58 benyttes af Bitcoin, men den har jeg aldrig brugt.  
Der findes mange online tools der kan encode og decode base64,32 og de andre typer.

Opgaver: Pension1 findes på Haaukins  
 Pension2 findes på Haaukins  
 Bad password findes på Haaukins

Password Cracking

**Kryptering gennem tiderne – meget kort gennemgang**

Histiaeus ca 500 før Kristus – klippede en slave og skrev besked på hovedbunden.

Cæsar – Måske den mest kendte kryptering overhovedet

Viginiere – i 1500 tallet , den ubrydelige kode

Enigma

DES Symmetrisk – Begge parter skal kende samme key

Diffie-Hellmann (1975) Første asymmetriske nøgledistribution

RSA (1977) Public-Private key Rivest-Shamir-Adleman - Meget brugt idag.

Utroligt at James Ellis-Clifford Cocks-Malcolm Williamson have opdaget det samme nogle år tidligere, men fordi de arbejde for GCHQ (den engelske efterretningstjeneste-dem der brød Enigma) blev deres opfindelse hemmelig indtil midt i 1990’erne. Så de kunne bare stiltiende se på at de andre blev berømte. 18.dec.1997 holdt Williamson et foredrag om hvad de lavede mere end 20 år tidligere og de fik den anerkendelse de fortjente, MEN desværre var James Ellis afgået ved døden ca 1 måned tidligere.

Elliptic curve (1985) - Nogle siger den er at foretrække fordi den er så svær rent matematisk at ikke ret mange prøver at implementere den selv. Kræver mindre nøgler end RSA

AES 1998 (afløseren for DES) DES have for korte keys så den var blevet usikker.

Kryptering er blevet til ren matematik, hvor det i gamle dage også havde noget med sprogkendskab at gøre.

**Post-quantum:**

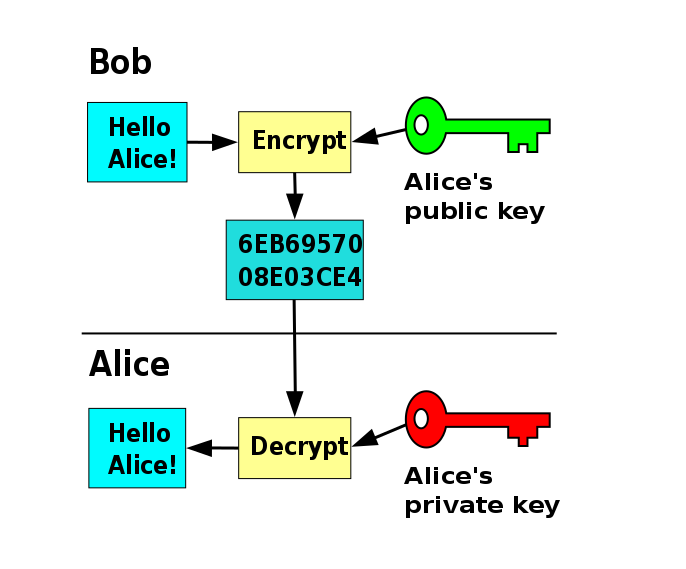
Ikke RSA sandsynligvis heller ikke elliptic curves.  
McEliece måske  
AES er så vidt jeg ved også brugbar

**Symmetrisk og assymmetrisk kryptering**

En blandt mange websider der forklarer forskellen

https://www.websiterating.com/da/vpn/glossary/what-is-asymmetric-symmetric-encryption/

Symmetrisk: Sender og modtager bruger samme nøgle  
  
Assymetrisk: Private og Publig key

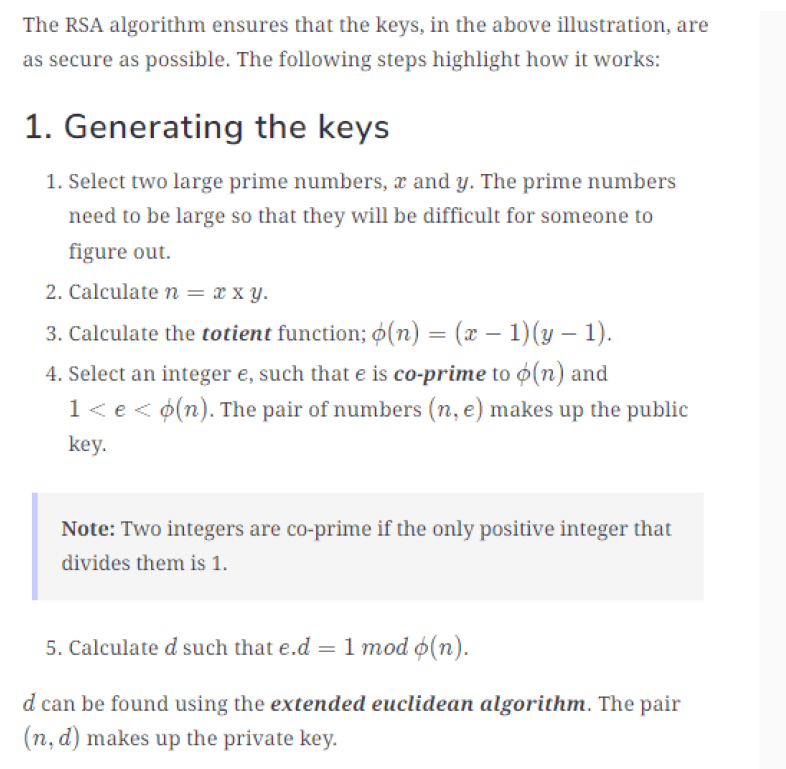


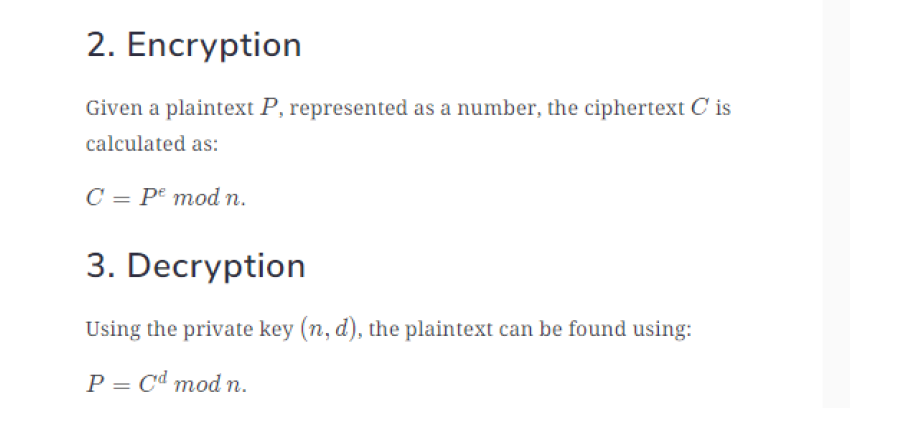
For det meste bruger man en hybrid kryptering. Private/public key til at sende nøglen til den symmetriske kryptering. Symmetrisk er meget hurtigere, så derfor bruges denne til store datamængder.

Public/private key:

RSA baseret på meget store primtal og modulus regning (regning med rester)  
  
Baserer sig på at finde to store primtal (p og q) og gange dem sammen og få et tal n. Sikkerheden består i at det er meget tidskrævende ud fra n at finde p og q. Hvis man kan det er sikkerheden brudt.

Algoritme:





P skal være mindre end n – ellers virker metoden ikke.

Altså der er en privat nøgle der består af **n og d**En offentlig nøgle der består af **n og e**Den offentlige nøgle står I certifikatet på en https side, men ikke synligt.

Hvis man kan finde p og q er det bare simple udreninger at bestemme d og så kan man dekryptere beskeden.

Husk at RSA kun krypterer tal. En tekst skal konverteres til et tal først.   
Dette gøres ikke altid på samme måde men typisk ved at man bruger asciiværdierne i hex og så laver man hex om til almindelige tal, men nogle opgaver afviger fra dette uden at man kender den algoritme som bruges.

Forøvrigt synes jeg det er helt vildt at man bruger Euclids Extended Algoritm som er mere end 2000 år gammel.

Svagheder I RSA:

Den private nøgle skal holdes hemmelig. Hvis en attacker får fat i den så er sikkerheden brudt. Husk at d og e er de samme indtil man skifter sit certifikat ud og det sker normalt med års mellemrum.  
  
Hvis p og q er for små kan man med moderne metoder faktorisere n

Husk at det samme p altid vil give det samme C, så hvis man tager et bogstav ad gangen kan krypteringen brydes ret let.

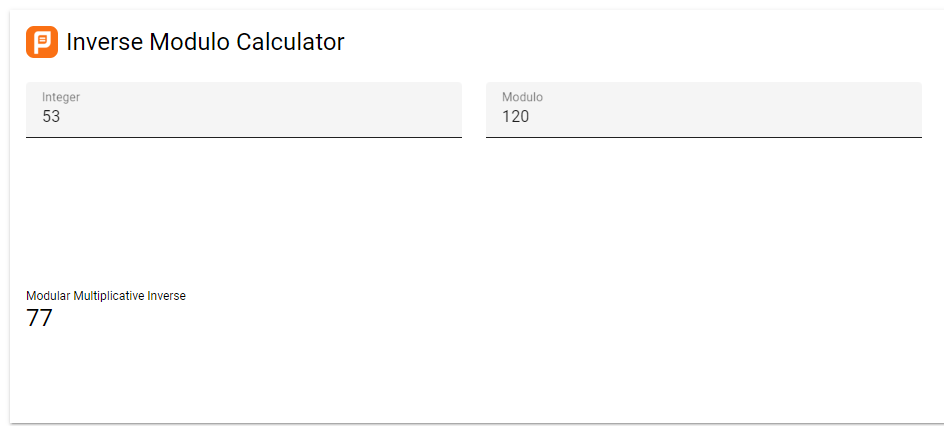
Der er flere svagheder, men det gemmes til næste gang.  
  
Husk at hvis der er en opgave i RSA, så er det sandsynligvis fordi der er en svaghed et eller andet sted.

Jeg har lavet 3 egne opgaver

RSA beginner findes på Haaukins  
RSA is key large enough findes på Haaukins  
RSA is implementation OK

RSA har den fantastiske fordel at det virker præcist lige så godt med meget små primtal, men kan brydes meget let. Altså er den god til undervisningsbrug.

Simpelt eksempel:  
p=11  
q=13  
N=11\*13=143  
   
Vi vælger e=53  
Nu skal d findes så Dette kaldes den inverse.  
Dette kan enten gøres ved euclids udvidede algoritme, men der eksisterer I de fleste programmeringssprog library funktioner til det. Der er også online beregnere. En af dem er denne:  
<https://planetcalc.com/3311/>

Svaret er at d=77  


Så nu er den offentlige nøgle 143 og 53  
Den hemmelige nøgle er 143 og 77

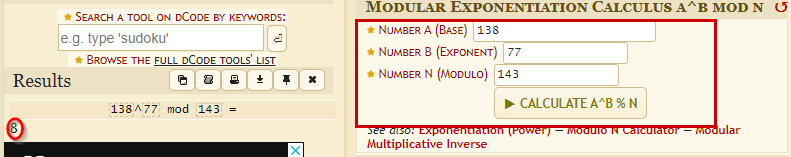
Kryptering:

Vi vil gerne sende tallet M=8 og derfor skal udregnes:  
 I python er der en function pow(8,53,143)

Altså sendes 138 til modtageren

Dekryptering:

Utroligt men sandt, så simpelt er det. At bevise at det altid gælder er noget sværere og er ikke relevant her.  
Man kan sagtens finde online værktøjer der kan foretage denne udregning for eksempel:  
<https://www.dcode.fr/modular-exponentiation>



Når man bruger store tal bliver udregningerne gigantiske, så det er nødvendigt at bruge smarte algoritmer. Hvis man bruger library funktioner er disse algoritmer allerede benyttet, men man kan faktisk også lave dem selv.  
De smarte algoritmer vil vi komme tilbage til, men ikke I dag.

Da man tit vil sende tekst skal man finde en måde at lave tekst om til tal på.  
Det gøres foreksempel ved at tage ascii værdien af en bogstav I hex og lave en streng af det og så til sidst lave det om til et tal.  
For eksempel: “ABC”  
 A er i ascii 65 I hex: 41  
 B er i ascii 66 I hex: 42  
 C er I ascii 67 I hex: 43  
Total streng i hex: “414243” det bliver i integer: 4276803

Der er funktioner til dette og mange websites.

Man skal lige huske at M=4276803 ikke må være større end n, så hvis ikke n er stort skal man begrænse længden af den streng man vil sammensætte.

Slut for i dag