NTNU | Norwegian University of Science and Technology

Introduksjon til kryptografi

Tjerand Silde @ TTM4175 – 12/11/25

Tjerand Silde

Førsteamanuensis i kryptologi ved IIK siden 2022, PhD fra NTNU

Forskningsgruppeleder NTNU Applied Cryptology Lab (NaCl)

Forsker på kvantesikker kryptografi og personvern-bevarende teknologier

Kontaktperson for masterprofil innen Kryptografisk Ingeniørvitenskap

Foreleser: TTM4205 Sikre Kryptografiske Implementasjoner (høst)
TTM4135 Anvendt kryptografi og nettverksikkerhet (vår)

Oversikt

Hva er kryptografi?

Symmetrisk kryptografi

- Hemmelig-nøkkel kryptering
- Caesar chifferet
- Hva betyr sikkerhet?
- AES og ChaCha
- Hashfunksjoner
- Meldingsauthentisering

Asymmetrisk kryptografi

- Nøkkelutveksling
- Offentlig-nøkkel kryptering
- Hva betyr sikkerhet?
- Diffie-Hellman og RSA
- Kvantesikker kryptografi

Applikasjoner

Lab-oppgaver

OBS!

Dette er en ny modul i dette kurset nå i høst.

Vi er interessert i tilbakemeldinger angående:

- Forberedelsesmateriellet
- Innholdet i forelesningen
- Oppsettet for labbene

Hva er kryptografi?

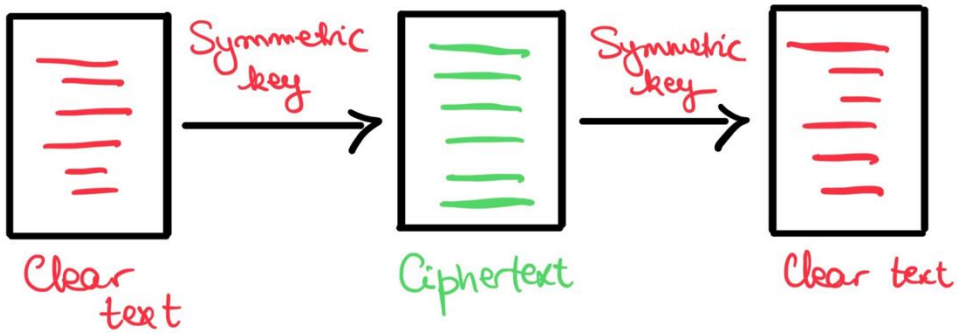
«Kryptografi er prosessen med å kode informasjon slik at vi kan oppnå egenskaper som konfidensialitet eller autentisitet.»

For dette benyttes vanligvis kryptering eller signaturer. Det er hovedfokuset i denne modulen. Men kryptografi omhandler også:

- Hvordan vi kan bevise at noe er sant uten å lekke info
- Hvordan vi kan sørge for at brukere oppfører seg ærlig
- Hvordan vi kan analysere krypterte data uten nøkler
- Hvordan vi kan implementere ting sikkert i kode

Symmetrisk kryptografi

- Ønsker å kryptere hemmelige meldinger
- Sender og mottaker må ha samme nøkkel
- Benytter nøkkelen til å forvrenge meldingene
- Benytter også nøkkelen til å sjekke korrekthet

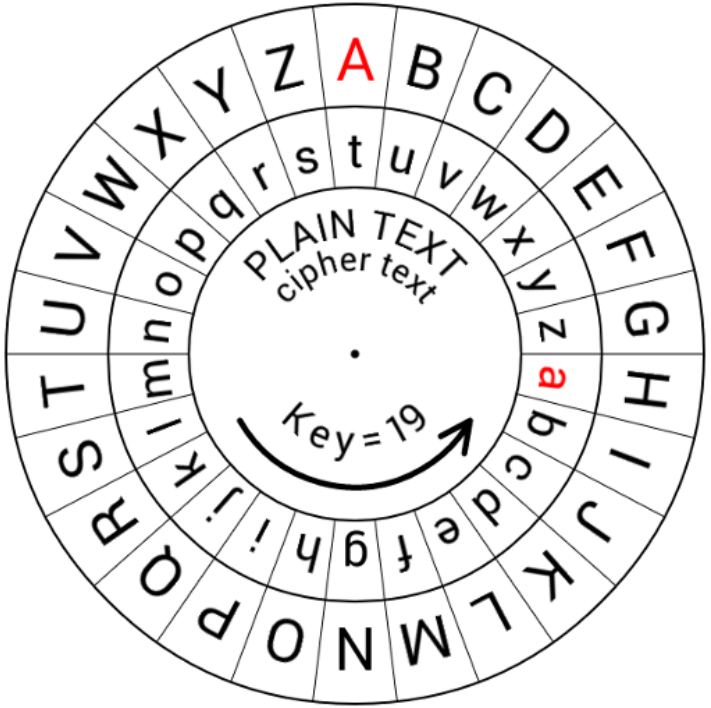


Caesar chifferet

En av de eldste måtene å kryptere en melding på er å rotere alle bokstavene.

Da flyttes bokstavene (f.eks.) 19 hakk fremover: A → t
for å kryptere meldinger.

Dette er ikke særlig sikkert...

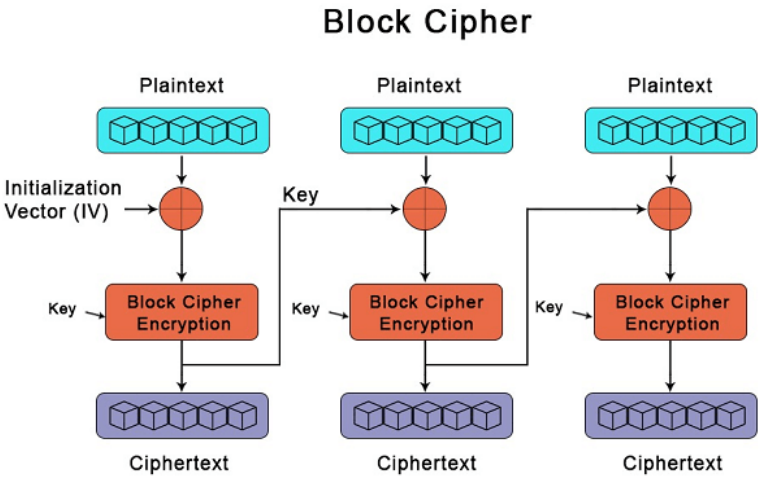


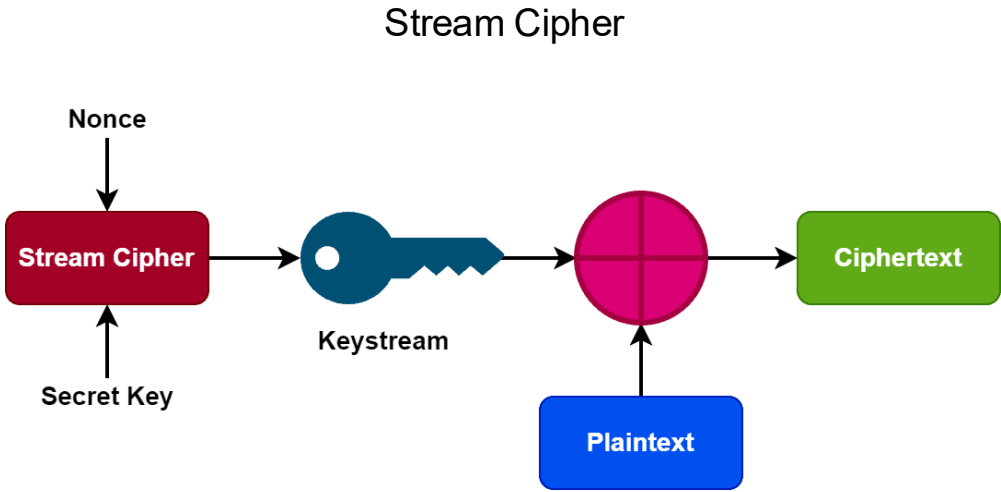
HVA BETYR SIKKERHET?

Hva betyr sikkerhet?

- Det hjelper ikke å vite hvordan krypteringsalgoritmen fungerer
- Det må finnes mange mulige krypteringsnøkler som kan benyttes
- Krypteringsnøklene må være tilfeldige / vanskelige å gjette
- Chiffertekstene må ikke lekke informasjon om meldingene
- Det hjelper ikke å få tilgang til mange meldinger og chiffertekster
- Det hjelper ikke å ha et «(de-)krypteringsorakel» tilgjengelig

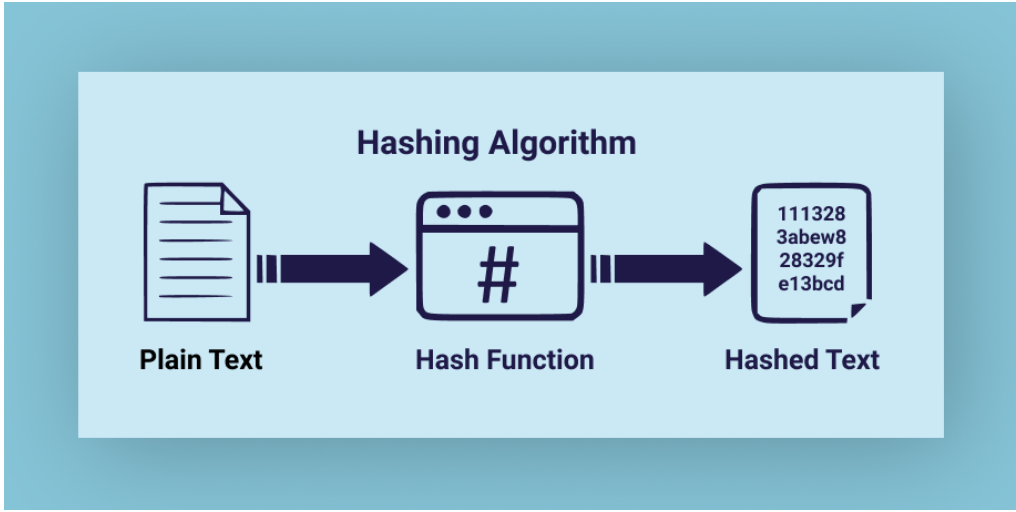
AES og ChaCha





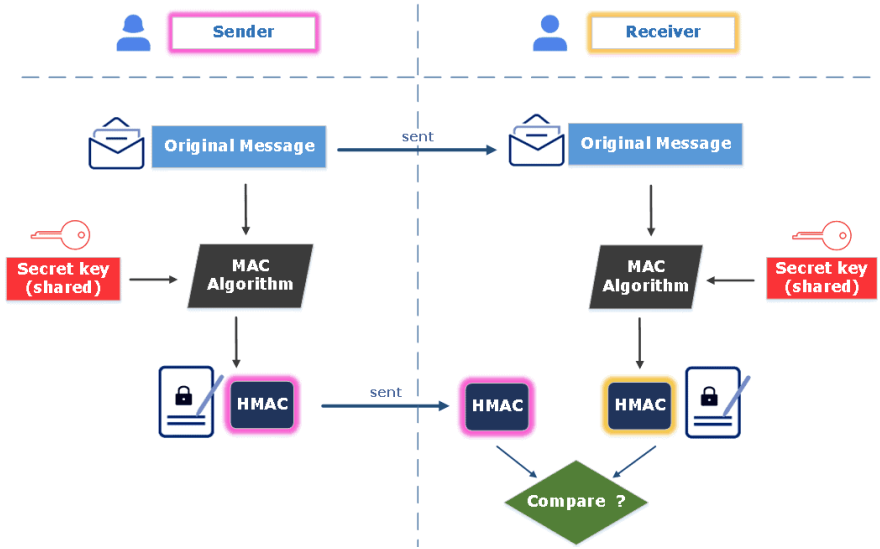
Hashfunksjoner

- Tar arbitrært input, og produserer output med fastbestemt lengde
- Vanskelig å finne input, gitt bare output av H
- Vanskelig å finne kollisjoner H(x) = H(y)



Meldingsauthentisering

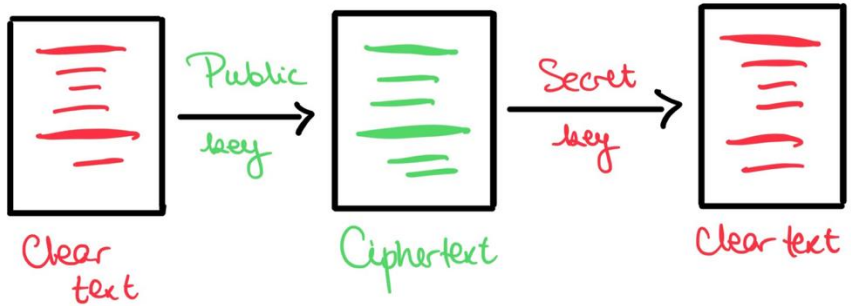
- Vi ønsker integritet av meldinger som sendes
- Da kan vi produsere en meldingsautentiseringskode
- Samme nøkkel benyttes til å lage og sjekke koden

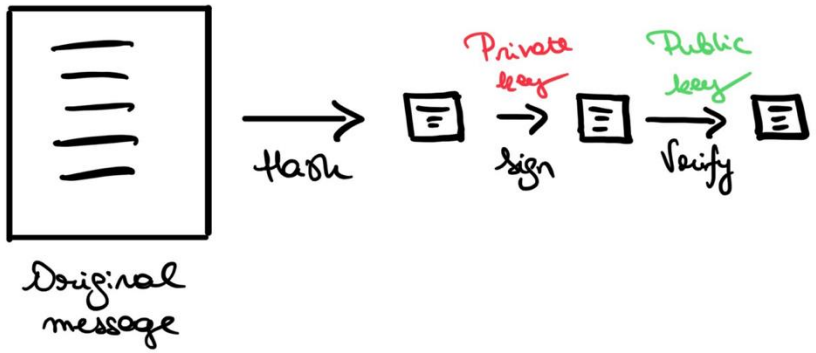


PAUSE

Asymmetrisk kryptografi

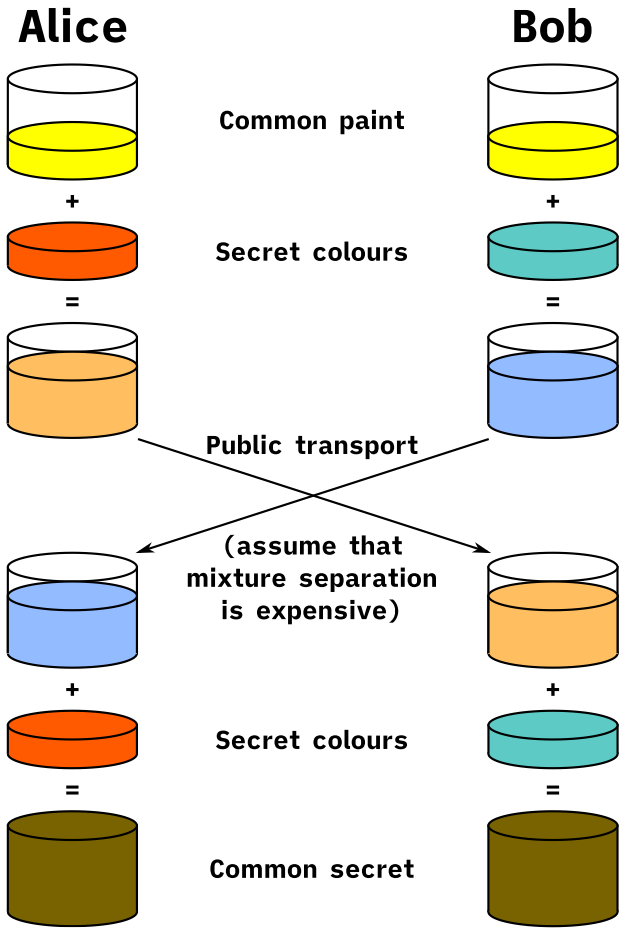
- Ønsker å kryptere eller signere meldinger
- Sender og mottaker har forskjellige nøkler
- En benyttes til å kryptere, en annen til å dekryptere
- En benyttes til å signere, en annen til å verifisere





Nøkkelutveksling

- To parter ønsker å bli enige om en nøkkel
- Nøkkelen benyttes så i AES eller ChaCha
- Utvekslingen er sikker selv om alle kan se på



Offentlig-nøkkel kryptering

- Offentlige nøkler kan publiseres online eller i database
- Tillater hvem som helst å kryptere en melding
- Tillater hvem som helst å verifisere en signatur
- Kun den som har tilgang til den private nøkkelen kan dekryptere eller signere meldinger

HVA BETYR SIKKERHET?

Hva betyr sikkerhet?

- Offentlig algoritme, mange nøkler, chiffertekst er tilfeldig
- Det hjelper ikke å få tilgang til mange meldinger og chiffertekster eller signaturer for disse meldingene
- Det hjelper ikke å ha et «dekrypteringsorakel» tilgjengelig

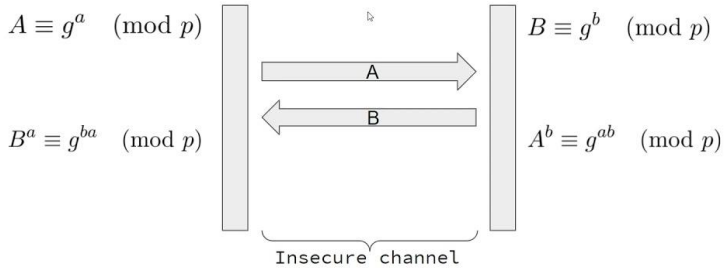
Diffie-Hellman og RSA

CYBER CAMPUS NRW

cry.college

Diffie-Hellman Protocol

Standardize $\mathbb{F}_p$ and multiplicative generator $g \in \mathbb{F}_p$.



Key material $= g^{ab} \pmod p$

RSA Algorithm

Key Generation

Select p,q	p and q both prime; $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n),e) = 1;\ 1 < e < \phi(n)$
Calculate d	$de \bmod \phi(n) = 1$
Public key	$KU = \{e,n\}$
Private key	$KR = \{d,n\}$

Encryption

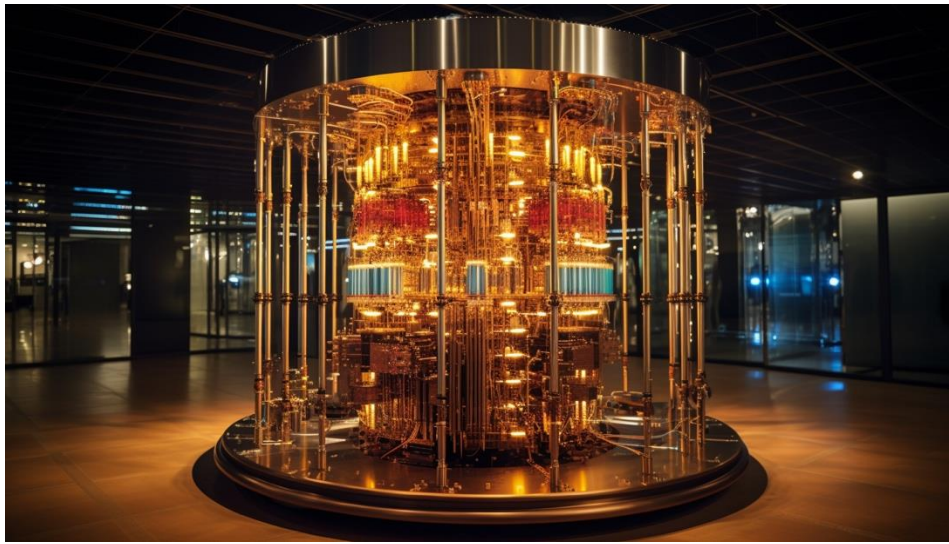
Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod n$

Decryption

Plaintext:	C
Ciphertext:	$M = C^d \pmod n$

Kvantesikker kryptografi

- Kvantedatamaskiner kan knekke DH og RSA
- Lage nye algoritmer for å beskytte oss i fremtiden
- Symmetrisk kryptografi vil fremdeles være sikker
- Pågående standardisering

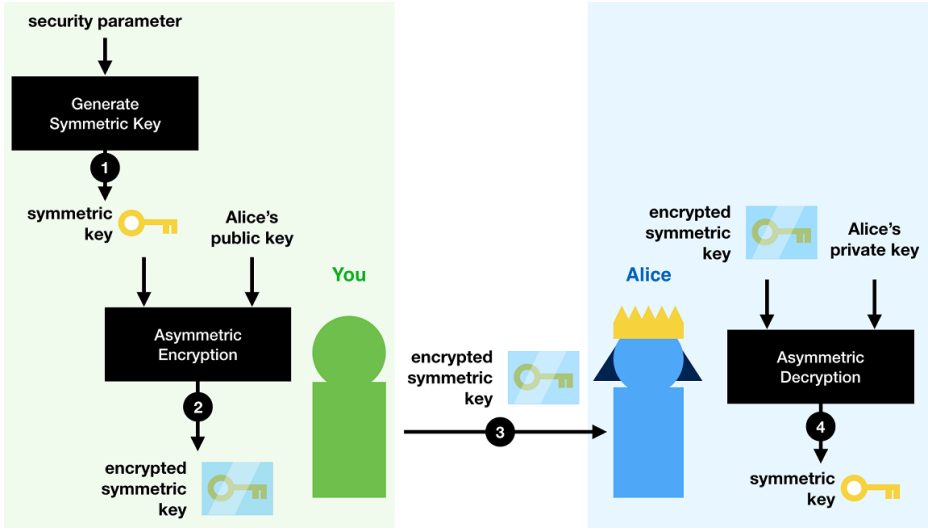


Applikasjoner

- Sikre meldingstjenester: Signal, WhatsApp, iMessage,...
- Sikre tilkoblinger: TLS, SSH, IPsec,...
- Digital autentisering: FIDO, Buypass ID, Bank ID,...
- Betalingstjenester: Vipps, VISA/Mastercard, Bitcoin, Apple/Google Pay, PayPal,...

Kryptering i TLS

1. Lag en symmetrisk nøkkel
2. Krypter under offentlige nøkkelen til mottaker
3. Krypter meldinger under symmetrisk nøkkel
4. Send kryptert nøkkel og meldinger til mottaker



Teamwork

For this week's lab, you will be using **Jupyter Notebook** to complete both the tasks and your report. The notebooks are available under the following links:

- Lab 1
- Lab 2

Be sure to **read through all the instructions carefully** so you do not miss any tasks or important information. Questions that require reflection are highlighted with a text box like the one below, where you will enter your answers:

Question 4.1.1.1: Explain what the function does at each line

```
Fill in your answer here.
```

Report

Once you have completed the lab, export the notebook (with all your answers and changes), and make it an attachment to Report 3 to be uploaded to Blackboard for evaluation. You also have to include a reflection note about the lab in the report.

Lab 1: Implementere algoritmer

```python
alphabet = "abcdefghijklmnopqrstuvwxyz"

def encode(character, key):
    """
    TODO:
    Implement Caesar cipher encoding for a single lowercase letter.
    """

    return  # Replace this with your code

def decode(character, key):
    """
    TODO:
    Implement Caesar cipher decoding for a single lowercase letter.
    """

    return  # Replace this with your code

# Test examples
print(encode("a", 17))  # Expected output: 'r'
print(decode("r", 17))  # Expected output: 'a'
```

Lab 2: Sette alt sammen på micro:bit

```
def on_receive(received_bytes):
    try:
        decrypted_bytes = chacha20_encrypt(
            received_bytes, GLOBAL_KEY)
        data = decrypted_bytes.decode('utf-8')
        return data
    except TypeError:
        return "TYPE ERROR"
    except:
        return "WEIRD ERROR"
```

Question 2.2: Explain shortly what happens in the `on_receive` function.

```
# Fill in your answer here.
```

Additionally, `def send_mode()` has been altered to send the messages written as constants instead of "A" or "B" as in part 1. Here, a new function `on_send()` is used instead of `radio.send()`.

Using the `on_receive()` function, let us try to fill out `on_send()`!

Task: Follow the steps as written in the TODO and copy your solution into the code block below

```
def on_send(msg):
    """
    TODO:
    Implement a send function that encodes the inputted
    message to bytes, encrypts the bytes using chacha20_encrypt()
    and then sends the encrypted bytes over the radio.
    """
    pass # Remove this line and implement your solution here
```

SPØRSMÅL?