Kryptologi, Quantum, og Zero-knowledge Proofs

Spionfilm eller Matematik?

Simon Skjernaa Erfurth

Institut for Matematik og Datalog
Syddansk Universitet

22. November, 2022
UNF Odense

Simon Skjernaa Erfurth

- **2016–2019:** Bachelor i Matematik og Datalogi (IMADA, SDU)
- **2019–2021:** Kandidat i Matematik (IMADA, SDU)
  Speciale: *Oblivious Transfer in Quantum Cryptography*
- **2022–2024:** Ph.D. i Datalogi (DDC/IMADA, SDU)
  Projekt: *Trust and News Authenticity*
  „Develop a digital signature to be attached to journalistic content and a recognizable label for users to see if the content is authentic and verified.“

Dagens plan

1. Kryptologi
   - Hvornår er noget sikkert
   - Symmetrisk kryptering
   - Public-key kryptering
2. Zero-knowledge proofs
   - Introduktion og et eksempel
   - Sikkerhed
   - Et graf eksempel
3. Multi-party computations og quantum
   - Oblivious transfer
   - Quantum
   - Quantum multi-party computations

Hvad er kryptologi?

For jer?

- Hemmeligheder?
- Blockchain?

For grækerne

- Kryptós: ”hidden, secret”
- -logia: ”study”

For mig

Brugen af Matematiske og Datalogiske redskaber til at holde information hemmelig overfor fjender, og samtidig tillade at man opnår meningsfulde resultater med informationen.

Eksempler på kryptologi

- Udveksle hemmelige beskeder med ens venner
- Udveksle hemmelige beskeder med folk man ikke kender
- Digitale underskrifter: Check en beskeds autenticitet og integritet.
- Identificering: på nettet og i virkeligheden
- Dele en hemmelighed i flere bider
- Zero-knowledge proofs: Bevis at man har et bevis, uden at fjenden lære beviset.
- Multi-party computations (Vi har to hemmeligheder som vi gerne vil beregne en funktion af, men vi vil ikke dele hemmelighederne med hinanden)

Hvornår er noget sikkert

Definition 1

En protokol er sikker hvis fjenden ikke lære noget.

Problem!

Vi kan ikke undgå at fjenden lære noget.

Definition 2

En protokol er sikker hvis fjenden ikke lære mere end højest nødvendigt.

Meeen....

- Hvad „højest nødvendigt“ er variere
- Hvor stærkt „ikke lære“ er variere

Symmetrisk kryptering

Generelt

Alice og Bob har samme nøgle K, som tillader dem at kryptere og dekryptere beskeder. Hvis man ikke har nøglen „lære man ikke mere end højest nødvendigt“.

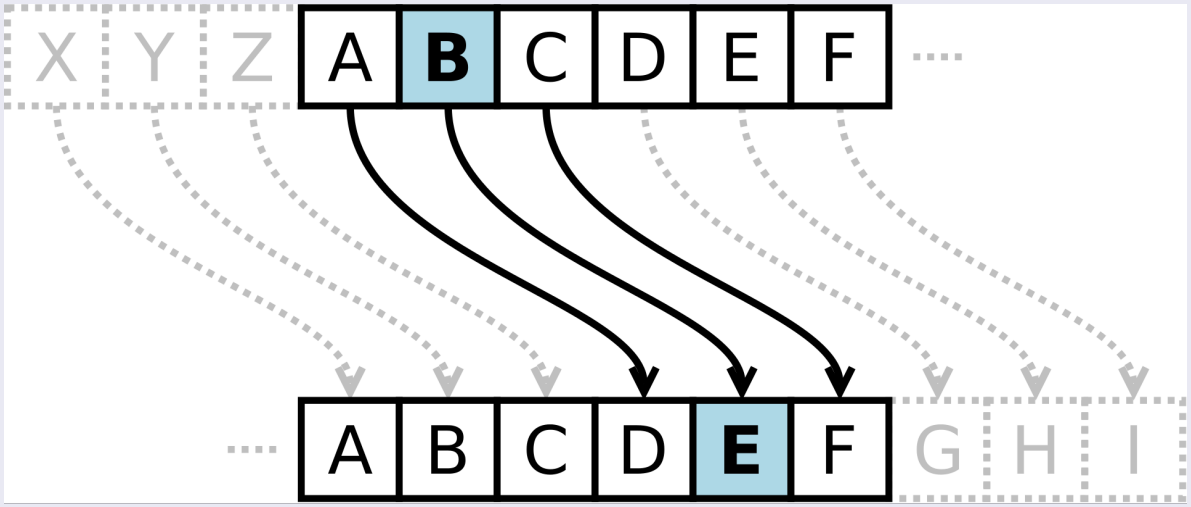
Kerckhoffs princip (1883)

Sikkerheden af en krypteringsløsning må ikke afhænge af algoritmens, men kun af nøglens hemmelighed. [For det lykkedes altid fjenden at lære algoritmen at kende.]

Historiske chifre

Cæsarchiffer

En nøgle er et tal $K \in \{1, 2, \ldots, 29\}$.



Problem!

Der er kun 29 mulige nøgler...

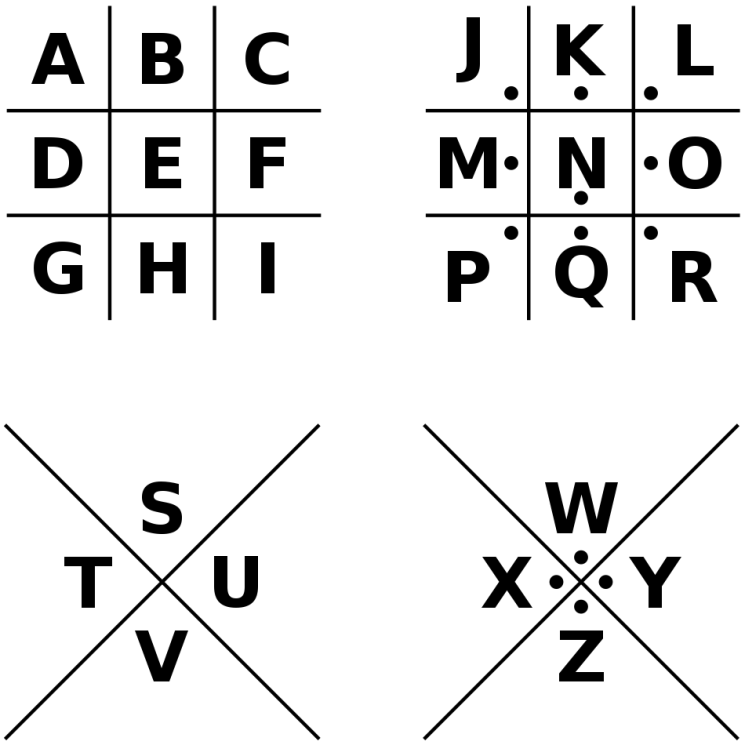
Historiske chifre

Substitutionschiffer

En nøgle er en funktion fra de 29 bogstaver til 29 symboler.

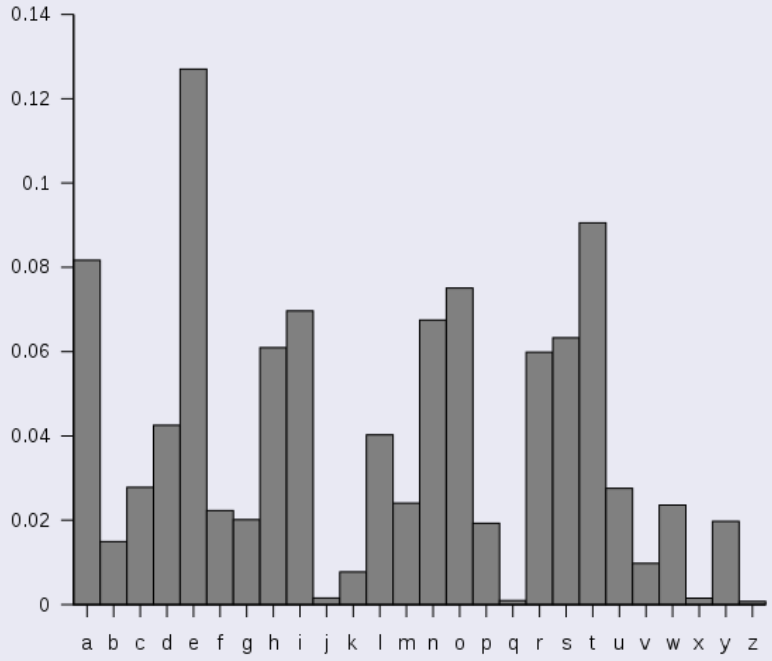
Problem!

Der er for meget system bevaret!

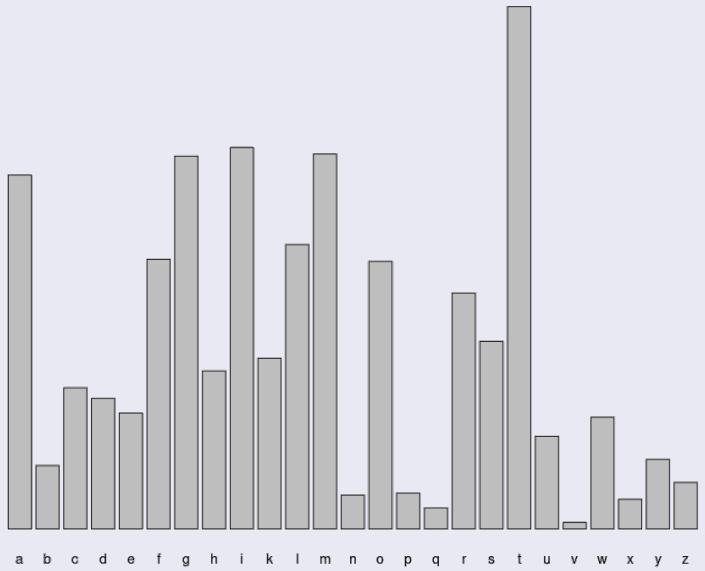


Brydning af substitutionschiffer

Bogstavs frekvens for engelsk



Frekvens for en (lang) besked



Opgaver med dette (og mere)

SCAN ME

www.serfurth.dk/FD2022

Informationteoretisk kryptering

Perfekt sikkerhed

- Nøglen har samme længde som beskeden.
- $M = \text{UNF}, K = \text{ABC} \implies C = VPI$

Hvordan er det perfekt?

- Alle (3 bogstavs) beskeder kan give cifferteksten VPI!
- Så uden nøglen lære fjenden kun længden på beskeden.
- F.eks. $M = \text{SDU}, K = \text{ILQ} \implies C = \text{VPI}$

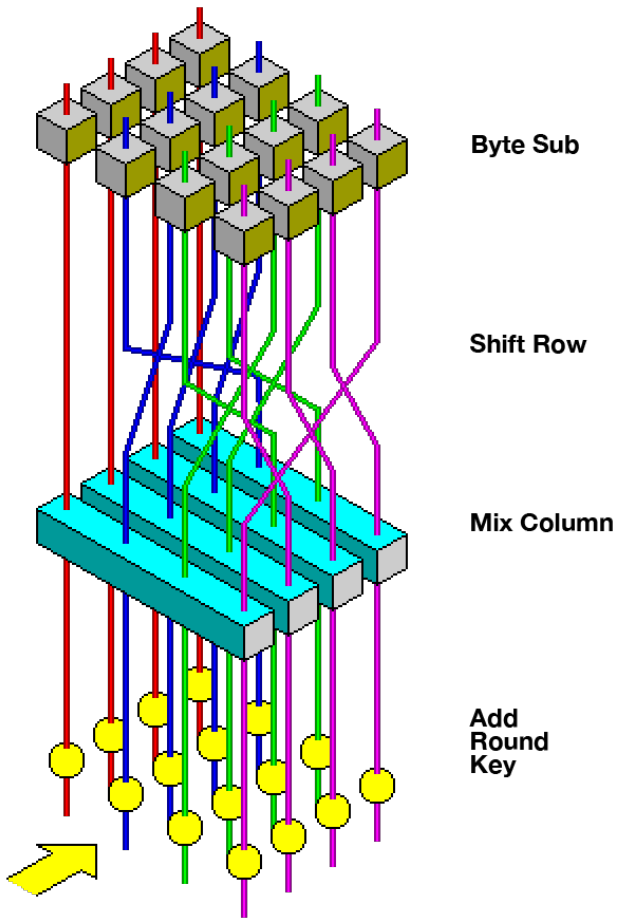
Problem?

Nøglen har sammen længde som beskeden: Hvordan udveksler vi nøglen?

Moderne chifre

Advanced Encryption Standard (AES)

- Pros: Moderne, effektiv, og kompakt symmetrisk kryptering
- Cons: Kræver sit eget foredrag



Public-key (asymmetrisk) kryptering

Problem?

Hvad hvis vi ikke kan dele en nøgle på forhånd?

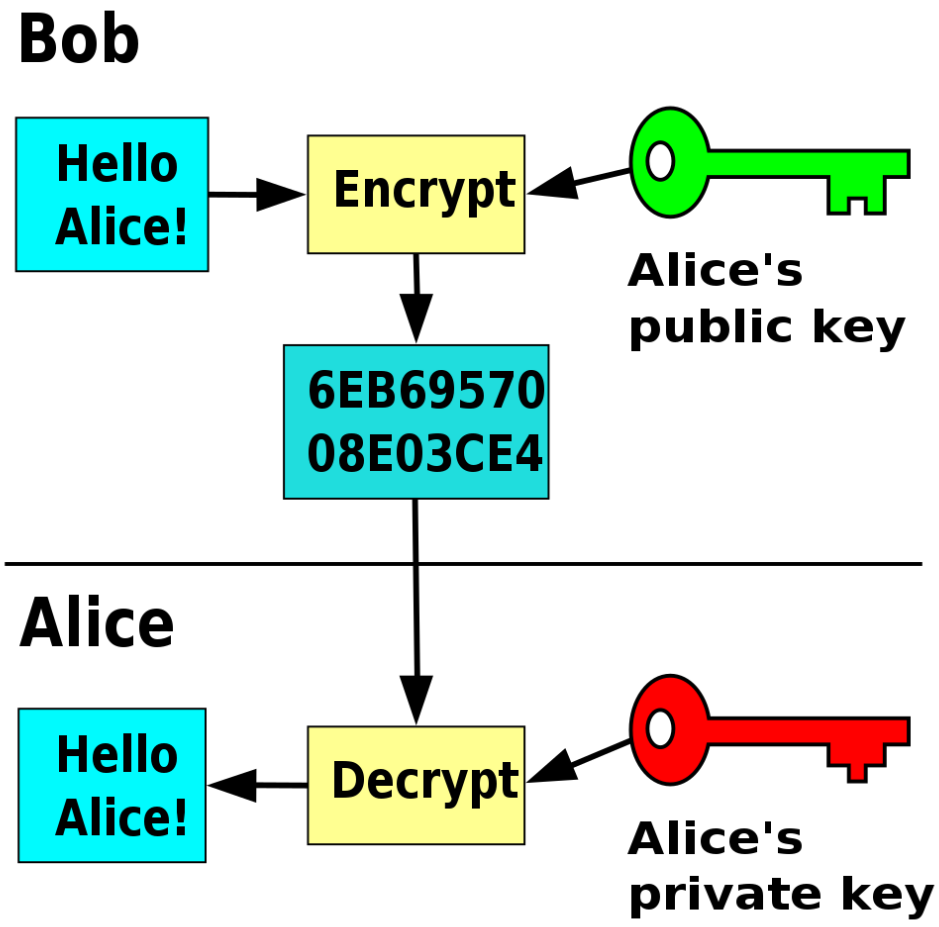
Asymmetrisk kryptering

Alice har en hemmelig nøgle sk og en offentlig nøgle pk. Alle der kender pk kan kryptere beskeder til Alice, men kun dem der kender sk kan dekryptere beskederne.

Asymmetrisk kryptering i praksis

- Alice og Bob har hver deres par $(\mathrm{pk}_A, \mathrm{sk}_A)$ og $(\mathrm{pk}_B, \mathrm{sk}_B)$.
- Bruger disse til at udveksle en nøgle til symmetrisk kryptering.

Public-key kryptering



Public-key kryptering

RSA (Rivest–Shamir–Adleman, 1977)

- Fandt ud af at det er muligt at vælge tre tal N, e, d sådan at hvis

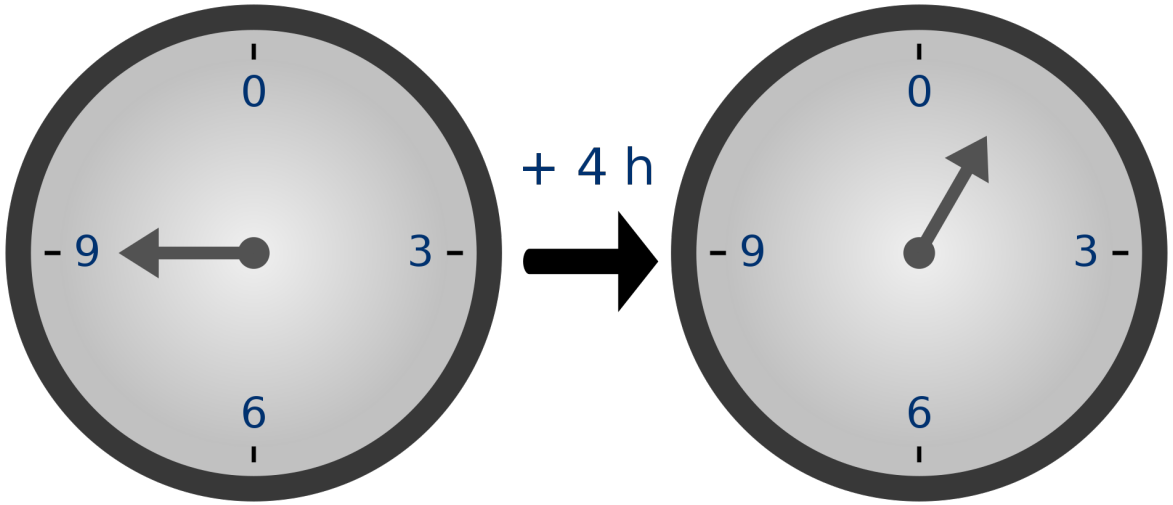
$$c \equiv m^e \pmod{N} \tag{1}$$

så er

$$c^d \equiv m \pmod{N}. \tag{2}$$

- $\mathrm{sk} = (d, N)$ og $\mathrm{pk} = (e, N)$.
- Hvis $N = pq$ for to store primtal p og q, så er det *svært* at finde m (og d) hvis man ikke kender p og q.

Modulær aritmetik



Public-key kryptering

OBS!

- Hvis man finder p og q er RSA ikke sikkert.
  - Men det tror vi er svært (factoring-assumption)
- Vi tror også det er svært at finde m hvis man kun kender c og $\text{pk} = (d, N)$.
  - RSA-assumption

$$\text{RSA-assumption} \Rightarrow \text{Factoring-assumption}$$

Public-key kryptering

RSA Eksempel

1. Vælg to (store) primtal: $p = 7; q = 19$
2. Beregn $N = pq = 7 \cdot 19 = 133$
3. Find e (som opfylder nogen krav), f.eks. $e = 29$
4. Find d (ved hjælp af gruppeteori), f.eks. $d = 41$
5. Nu er $\text{pk} = (29, 133)$ og $\text{sk} = (41, 133)$
6. Krypter 69:

$$c = 69^{29} (\approx 2.121 \cdot 10^{53}) \equiv 27 \pmod{133} \tag{3}$$

7. og dekrypter

$$27^{41} \equiv 69 \pmod{133} \tag{4}$$

Hvad er et Zero-knowledge proof?

Proof [broke]

Et matematisk overbevisende argument for at noget er sandt.

Zero-knowledge [woke]

Fjenden bliver ikke klogere.

Zero-knowledge proof [enlightened]

Et (interaktivt) matematisk overbevisende argument for at man ved noget, som ikke gør fjenden klogere.

Eksemple: Farver

Farver

- Alice er farveblind. Bob skal overbevise Alice om at han ikke er farveblind.
- IDE: Blå og Rød bold, en i hver hånd. Alice kan enten bytte rundt på dem, eller hun kan lade være. Bob skal afgøre om hun har byttet rundt på dem.
- Hvorfor virker det? Hvis Bob ikke er farveblind svare han altid rigtigt, men hvis Bob er farveblind bliver han nød til at gætte, så 50% chance for at svare forkert.
- Gentag mange gange

Eksemple: Farver

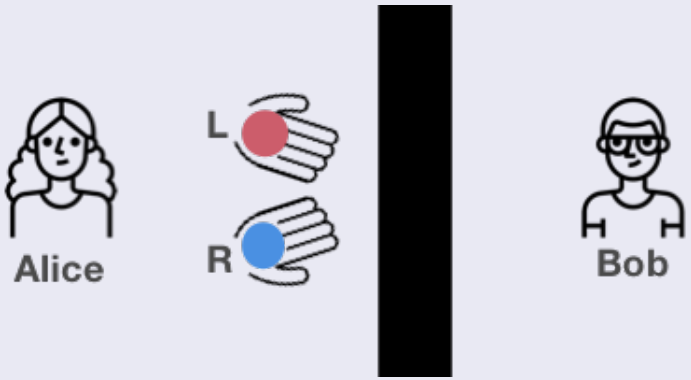
Trin 0

Alice: „Her er to bolde i forskellige farver“

Eksemple: Farver

Trin 1

Alice kan nu vælge at bytte rundt på dem i hemmelighed



Eksemple: Farver

Trin 1.5

Bob skal afgøre om Alice har byttet rundt på dem.

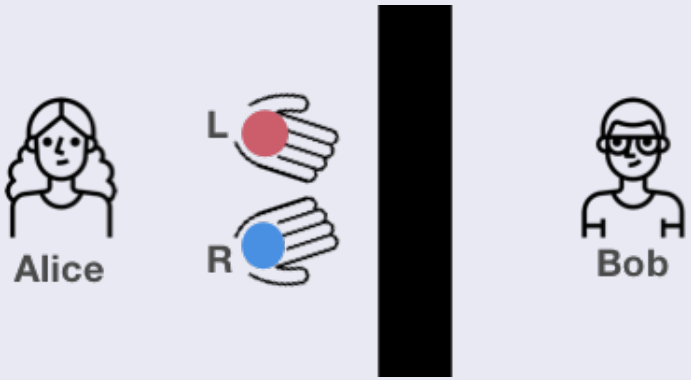
Hvis Bob gætter

Hvis Bob gætter er der sandsynlighed $\frac{1}{2}$ for at han tager fejl.

Eksemple: Farver

Trin 2

Alice kan nu vælge at bytte rundt på dem i hemmelighed



Eksemple: Farver

Trin 2.5

Bob skal afgøre om Alice har byttet rundt på dem.

Hvis Bob gætter

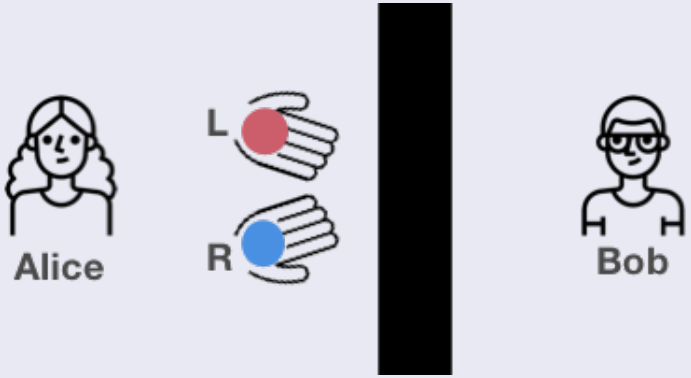
Hvis Bob gætter er der sandsynlighed $\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$ for at han har taget fejl, enten her eller tidligere.

Eksemple: Farver

Eksemple: Farver

Trin *n*

Alice kan nu vælge at bytte rundt på dem i hemmelighed



Eksemple: Farver

Trin $n.5$

Bob skal afgøre om Alice har byttet rundt på dem.

Hvis Bob gætter

Hvis Bob gætter er der sandsynlighed $\frac{1}{2^n}$ for at han har taget fejl, enten her eller tidligere.

Eksemple: Farver

Kan Bob snyde?

- Efter $n = 20$ runder er der under $\frac{1}{1000000}$ sandsynlighed for det lykkedes.
- Efter $n = 33$ runder er der under $\frac{1}{|\text{jordens befolkning}|}$ sandsynlighed for det lykkedes.

Er det zero-knowledge?

- Med andre ord: Lære Alice noget?
- ✓ Bob er ikke farveblind
- × Hvilken farve boldene har

Hvad vil det sige at være sikker?

Overfor den der beviser noget (Prover)

- Den der beviser noget kan ikke snyde.
- *Hvis Proveren ikke kender hemmeligheden er der meget, meget lille sandsynlighed for at den overbeviser Verifieren, uanset hvad den gør.*

Overfor den der bliver overbevist (Verifier)

- Verifieren lære ikke noget om hemmeligheden.
- *Man kan ved at simulere Verifieren få et transkript som ligner et rigtigt transkript, selvom man ikke kender hemmeligheden. Skal også gælde hvis Verifieren ikke er ærlig.*

Hvad vil det sige at være sikker?

Korrekthed

Hvis begge parter er ærlige så bliver Verifieren overbevist.

Er farve eksemplet sikkert?

Overfor Bob (Prover)

Ja - argumenteret undervejs.

Korrekthed

Ja (Proveren kan altid svare rigtigt).

Overfor Alice (Verifier)

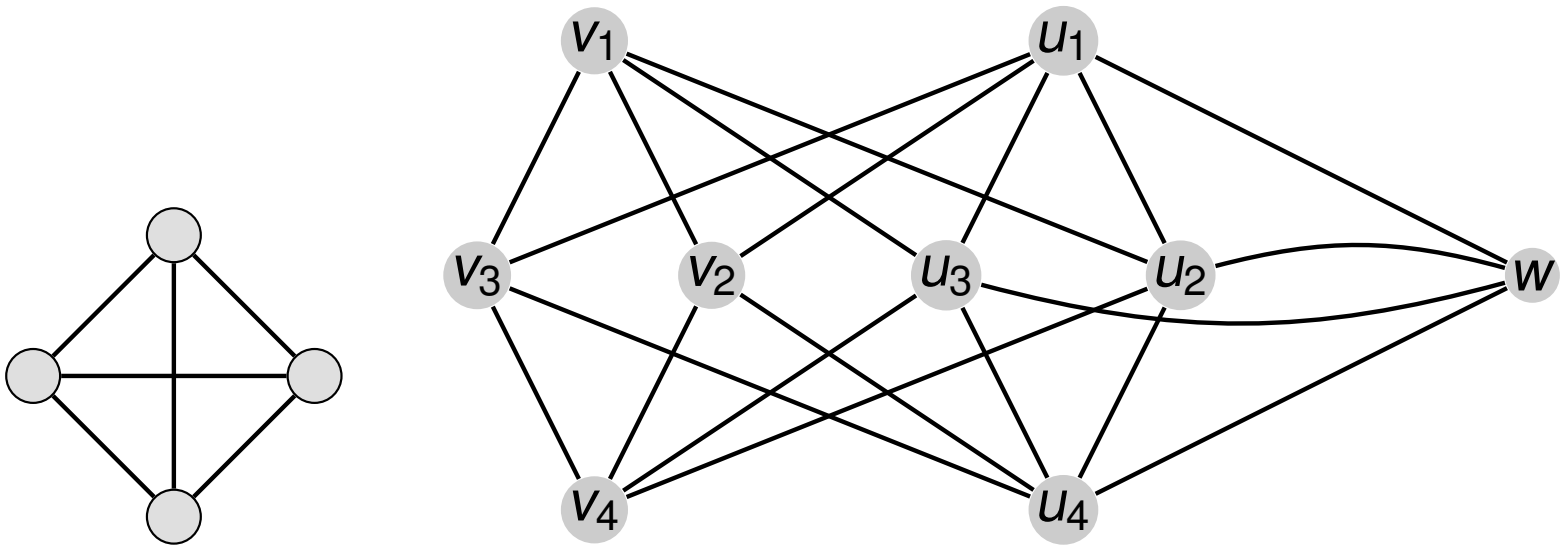
Ja.

- Hvis vi er ikke farveblinde, så svare vi bare.
- Hvis vi er farveblinde: gæt på om den har byttet rundt eller ej, hvis vi tager fejl så spol tilbage til før runden og prøve igen. (Forvent to forsøg per runde)

Graf isomorfier

Grafer

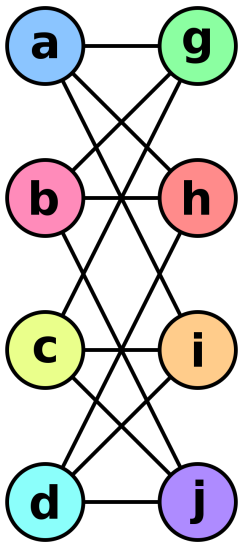
I datalogi er en graf $G = (V, E)$ en samling V af knuder og en samling E af kanter mellem knuder i V.

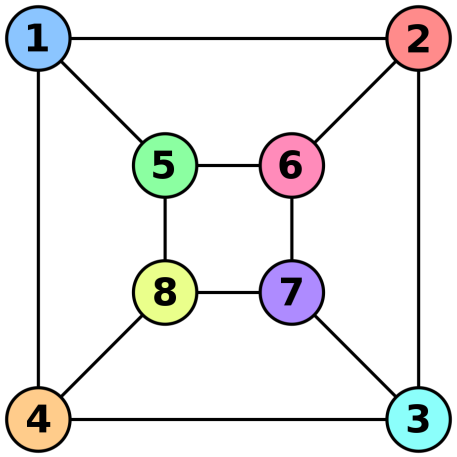


Graf isomorfier

Graf isomorfi

To grafer $G_1 = (V_1, E_1)$ og $G_2 = (V_2, E_2)$ er isomorfe hvis der er en bijektion $\pi\colon V_1 \to V_2$ sådan at der er en kant mellem $u, v \in V_1$ hvis og kun hvis der er en kant mellem $\pi(u)$ og $\pi(v)$ i V_2.
Vi skriver $\pi(G_1) = G_2$.





$$\begin{aligned} f(a) &= 1 \\ f(b) &= 6 \\ &\vdots \\ f(j) &= 7. \end{aligned}$$

Graf isomorfier

Graf isomorfi i zero-knowledge

- Begge parter kender $G_1 = (V, E_1)$ og $G_2 = (V, E_2)$.
- Proveren kender graf isomorfi π:

$$\pi(G_1) = G_2.$$

- Proveren skal overbevise Verifieren om at den kender sådan π uden at afsløre π.*

Graf isomorfier

Ide

- Vælg en tilfældig isomorfi π', send $G = \pi'(G_1)$.
- Lad Verifier vælge hvilken af G_1 og G_2 han gerne vil se er isomorf til G.

Ide: Sikkerhed

- Prover kan ikke snyde da han skal kunne vise begge er isomorfe til G (og hvis han kan det kan han finde en isomorfi mellem de to grafer).
- Verifier lærer ikke noget, da han kun ser den ene grafs isomorfi til G, så for at blive klogere skal han finde en graf isomorfi uanset hvad.

Graf isomorfier

Diagram

Prover		**Verifier**
Vælg tilfældig $\pi' : V \to V$		
$G \leftarrow \pi'(G_1)$	$\xrightarrow{\quad G \quad}$	
		$b \leftarrow_{\$} \{1, 2\}$
	$\xleftarrow{\quad b \quad}$	
Hvis $b = 1$ send $\pi = \pi'$,		
ellers $\pi = \pi' \circ \pi^{-1}$	$\xrightarrow{\quad \pi \quad}$	Check at
		$G = \pi(G_b)$

Graf isomorfier

Sikkerhed: Korrekthed

✓ Verifier acceptere altid.

Graf isomorfier

Sikkerhed: Prover

Hvis prover finder G sådan at han kender isomorfier π_1 og π_2 sådan at

$$\pi_1(G_1) = G$$
$$\pi_2(G_2) = G$$

så kender han en isomorfi mellem G_1 og G_2 nemlig

$$\pi = \pi_2^{-1} \circ \pi_1.$$

Har G ikke denne egenskab så kan Prover kun svare rigtigt med sandsynlighed $\frac{1}{2}$.

Graf isomorfier

Sikkerhed: Verifier

Ide: Simuler Verifier.

1. Gæt på om Verifier gætter 1 eller 2. Lav G sådan at man kan svare på det spørgsmål.
2. Interagere med Verifier som om man er Prover.
3. Hvis vi gættede rigtigt: ✓
4. Hvis vi gættede forkert: Spol tilbage til trin 1, og prøv igen.

Forvent to forsøg per runde.

Graf isomorfier

Konklusion

Der findes en zero-knowledge protokol til at bevise at to grafer er isomorfe.

Zero-knowledge proofs

Konklusion

- Der findes zero-knowledge proofs for næsten alt (man kan have beviser for).
- Bruges mange steder:
  - ~~Blockchain~~
  - Finans
  - Online afstemninger
  - Som et alternativ til passwords

Multi-party computations

Problemet

- Alice og Bob vil gerne beregne en fælles funktion på delt indput (Tænk $f(x_A, x_B) = (y_A, y_B)$)
- Alice og Bob vil ikke dele deres indput med hinanden.

Løsningen

- Brug en trusted party?
- Multi-party computations!

Multi-party computations

Eksempel

- Alices indput: en graf isomorfi π mellem to grafer G_1, G_2
- Bobs indput: $\times$
- Alices output: $\times$
- Bobs output: Sandt hviss π er en graf isomorfi mellem G_1 og G_2

Flere eksempler

- Gennemsnits indkomst
- Den rigtige pris på roer[1]

[1] https://doi.org/10.1007/978-3-642-03549-4_20

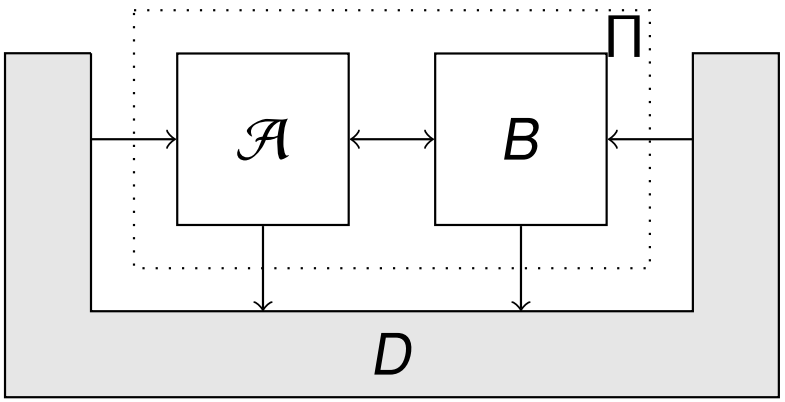
Sikkerhed

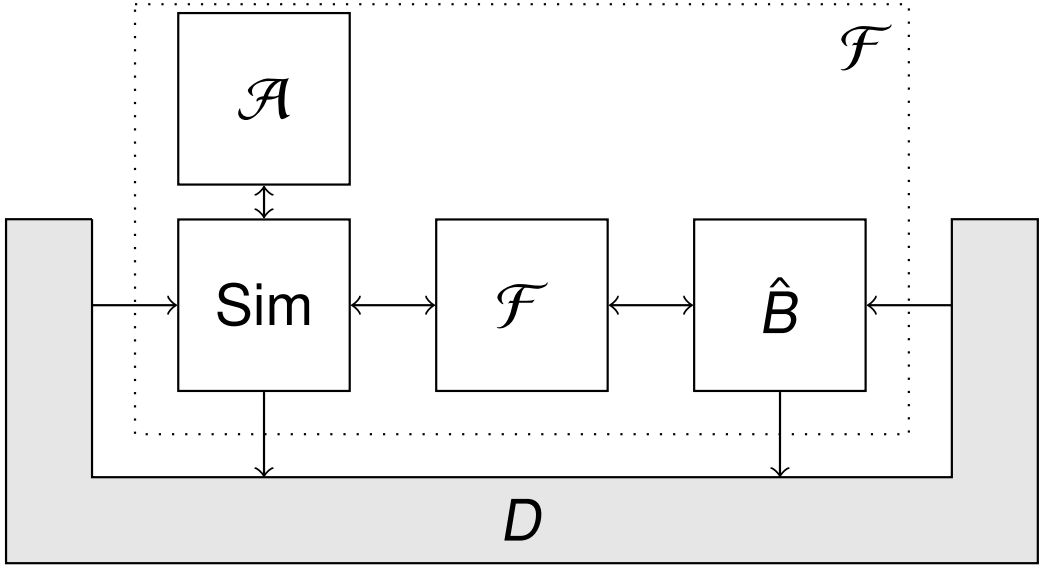
Ide

Sammenlign med en ideel verden, hvor vi har en trusted party!

Sikkerhed

Der findes en simulator sådan så det ikke er muligt at se forskel på den ideele verden og den virkelige verden.





Introduktion

Oblivious transfer (OT)

- Sender har m_0, m_1, modtager har $c \in \{0, 1\}$.
- Sender har ikke noget output, modtager skal have m_c som output.
- Sikkerhed: Sender må ikke lære c og modtager må ikke lære m_{1-c}.
- Et simpelt eksempel, **men** fra OT kan man få alle andre multi-party funktioner!

Introduktion

$\mathcal{F}_{\mathrm{OT}}$

Sender $\leftarrow m_0, m_1$	$\mathcal{F}_{\mathrm{OT}}$	**Modtager** $\leftarrow c$
	$\xrightarrow{m_0, m_1}$	
	$\xleftarrow{c}$	
	$\xrightarrow{m_c}$	
Output: $\perp$		Output: m_c

Eller som en funktion $\mathcal{S} \times \mathcal{M} \to \mathcal{S}' \times \mathcal{M}'$:

$$((m_0, m_1), c) \longmapsto (\perp, m_c). \tag{5}$$

Ide

RSA...(16)

- Fact: Krypterede RSA beskeder er umulige at kende fra tilfældige tal.
- Kan vi bruge det til at konstruere OT?

Ide

OT fra RSA

- Sender danner et RSA par $(\mathrm{pk} = (e, N), \mathrm{sk} = (d, N))$, og sender pk til Modtager.
- Modtager laver x_c en rigtig RSA kryptering af et (kendt) element og vælger x_{1-c} et tilfældigt element. Sender x_0, x_1 til Sender.
- Sender laver (perfekt) kryptering af m_0 og m_1 med $x_0^d \mod N$ og $x_1^d \mod N$, og sender dem til Modtager.
- Modtager kan nu finde m_c men ikke m_{1-c}.

OT fra RSA

Sender $\leftarrow m_0, m_1$		**Modtager** $\leftarrow c$
Dan $(\mathrm{pk}, \mathrm{sk})$	$\xrightarrow{\mathrm{pk}}$	$r_0, r_1 \leftarrow_\$ \mathbb{Z}_N$
		$x_c \leftarrow r_c^e \mod N$
	$\xleftarrow{x_0, x_1}$	$x_{1-c} \leftarrow r_{1-c}$
$d_0 \leftarrow m_0 \oplus (x_0^d \mod N)$		
$d_1 \leftarrow m_1 \oplus (x_1^d \mod N)$	$\xrightarrow{d_0, d_1}$	$m_c \leftarrow d_c \oplus r_c$
Output $\perp$		Output m_c

Sikkerhed for OT

Ide

- Sender lære ikke noget da x_0 og x_1 begge er tilfældige elementer fra $\mathbb{Z}_N$ (fra Senders perspektiv).
- Modtager lære ikke noget om m_{1-c}: hvis den kunne finde m_{1-c} kan den finde $d_1^d \mod N$ uden at kende d, dvs. den kan bryde RSA.

Sikkerhed for OT

Formelt mod Sender

- Simulator opføre sig først som Modtager med $c = 0$.
- Så spoler den Sender tilbage og opføre sig som Modtager med $c = 1$.
- Giv de opnåede m'_0, m'_1 til $\mathcal{F}_{\mathrm{OT}}$.
- Modtager i den ideele verden får præcis det af $\mathcal{F}_{\mathrm{OT}}$ som Modtager i den virkelige verden kommer frem til.

Quantum kryptografi

Problem!

Meget kryptologi (f.eks. RSA) er ikke sikkert overfor fjender med kvantecomputere!

Post-quantum kryptografi

Studiet af hvilke klassiske kryptografiske redskaber der er sikre mod kvantecomputere.

Quantum kryptografi

Klassisk kryptografi, men med færre eller svagere antagelser (f.eks. ingen RSA-antagelse).

Quantum Mechanics

Postulates of quantum mechanics

1. States are unit vectors in complex Hilbert spaces, $|\phi\rangle \in \mathcal{H}$.
2. Transformations are unitaries on $\mathcal{H}$.
3. Measuring is unreliable, and might change a state.
4. Composite systems are $|\phi\rangle \in \mathcal{H} \otimes \mathcal{H}'$.

No-cloning theorem

There is no state $|s\rangle \in \mathcal{H}$ and no unitary U acting on $\mathcal{H} \otimes \mathcal{H}$ such that for every state $|\phi\rangle \in \mathcal{H}$

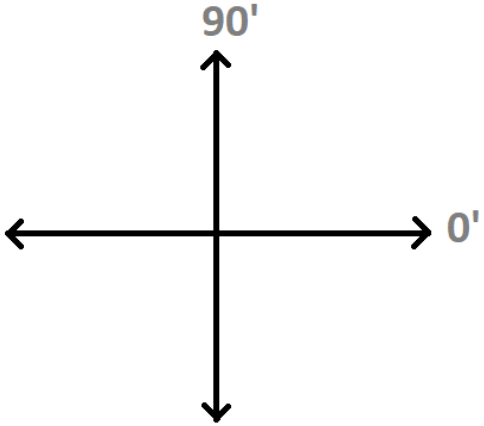
$$U(|\phi\rangle \otimes |s\rangle) = |\phi\rangle \otimes |\phi\rangle \,. \tag{6}$$

Introduktion

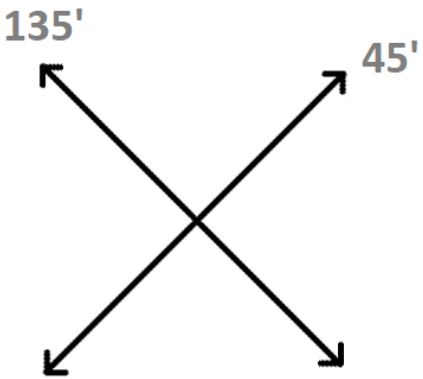
Mål

- General multi-party computation, men uden RSA-antagelse.
- Hvordan? Quantum Oblivious Transfer!

Målinger og BB84 states



Rectilinear basis



Diagonal basis

OT fra BBCS92

Sender S $\leftarrow (m_0, m_1)$		**Modtager R** $\leftarrow c$
$x^S \leftarrow_R \{0,1\}^n$		
$\theta^S \leftarrow_R \{+,\times\}^n$	$\xrightarrow{\quad |x^S\rangle_{\theta^S} \quad}$	$\theta^R \leftarrow_R \{+,\times\}^n$
		Measure $\|x^S\rangle_{\theta^S}$ wrt. θ^R, obtaining x^R
	$\xrightarrow{\quad \theta^S \quad}$	$I_c \leftarrow \{i \in [n] \colon \theta_i^S = \theta_i^R\}$
$f \leftarrow_R F$, for $i \in \{0,1\}$:	$\xleftarrow{\quad I_0, I_1 \quad}$	$I_{1-c} \leftarrow [n] \setminus I_c$
$s_i := m_i \oplus f(x_{I_i}^S)$	$\xrightarrow{\quad s_0, s_1, f \quad}$	$m_c := s_c \oplus f(x_{I_c}^R)$
Output: $\perp$		Output: m_c

OT fra BBCS92

Problem?

Hvad hvis Modtageren kan gemme sine qubits, og først måle på dem når den har modtaget θ^S?

Solution!

Commitments and cut-and-choose.

Spørgsmål?

Kilder

Billeder

Hvis ikke andet er nævnt: egen konstruktion eller wikipedia.

- Farve zero-knowledge eksempel: `https://blog.goodaudience.com/understanding-zero-knowledge-proofs-through-simple-exa`