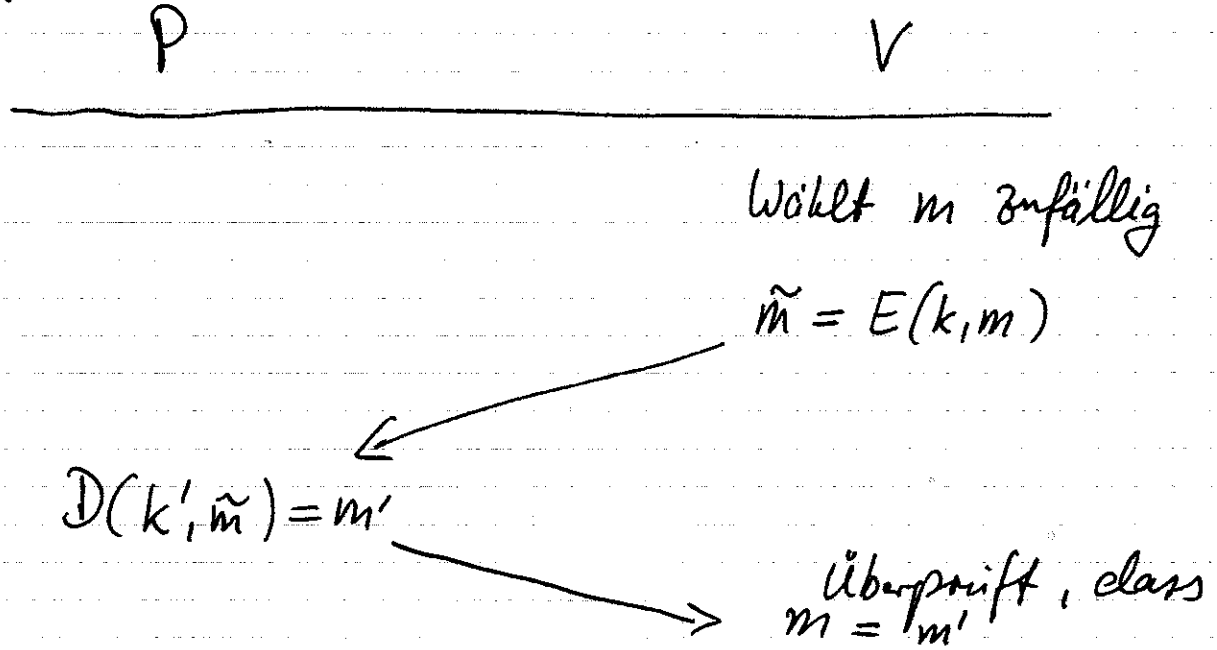
Aufgabe: Was kann man gegen dieses „Zero knowledge-Protokoll" (aus Sicht des Provers) sagen?



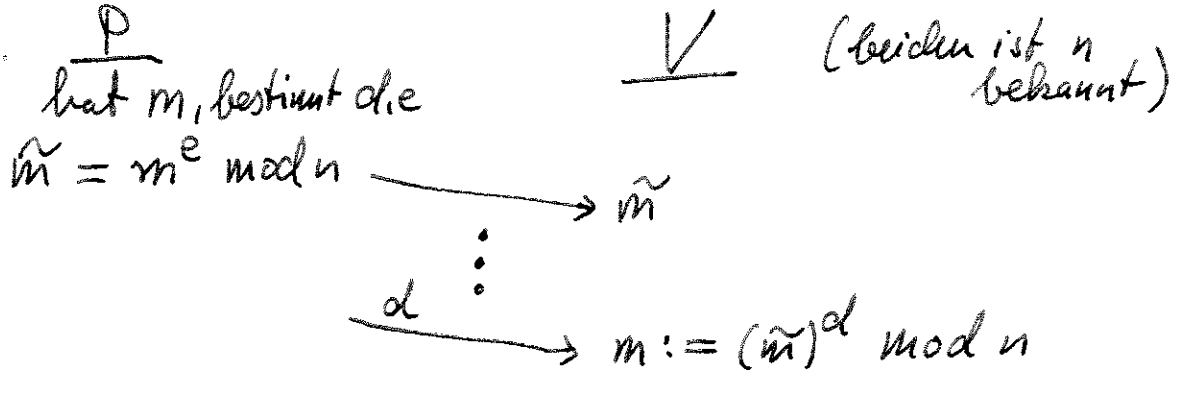
Hierbei ist k der öffentliche; k' der geheime Schlüssel von P in einem Public Key System, und $E(\,)$, $D(\,)$ sind die encryption- und decryption-Funktion.

Antwort: Es könnte sein, dass sich V (bzw. $\tilde{V}$) nicht an das Protokoll hält. Dann könnte $\tilde{m}$ nicht wie beschrieben zustande gekommen sein, sondern als verblendetes Dokument, das P durch Verwenden des geheimen Schlüssels k' mittels $D(\,)$ unterschreibt.

Aufgabe: Beim „Verschlüsseln mit Commitment" (siehe letzte Vorlesung) sendet P an V eine verschlüsselte Nachricht $\tilde{m}$ (die eigentliche Nachricht ist m). Damit ist P auf m festgelegt (an m gebunden). Zu einem späteren Zeitpunkt im Protokoll sendet P and V den Schlüssel, der V ermöglicht, m zu berechnen (oder P sendet m selbst). Es darf aber keine Möglichkeit geben, dass P statt dessen V von einer anderen Nachricht $m' \neq m$ überzeugt.

Frage: Kann man mit der Pohlig-Hellman-Verschlüsselung ein Verschlüsseln mit Commitment realisieren? (Pohlig-Hellman-Verschlüsselung: n große Primzahl. P wählt zufälliges $e < n$, berechnet $d = e^{-1} \bmod (n-1)$. Verschlüsseln: $m^e \bmod n = \tilde{m}$. Entschlüsseln: $(\tilde{m})^d \bmod n = m$.)

Also:



Antwort: Nein, auf keinen Fall.

P könnte d', e' mit $d' \cdot e' \equiv 1 \pmod{(n-1)}$ wählen und schickt (statt d) d' an V.

Damit entschlüsselt V:

$$(\hat{m})^{d'}_{\text{mod}} = m'$$

Auch P kennt m', denn $m' = (\tilde{m})^{e'} \bmod n$.

Ein Bit-Commitment Scheme haben wir bereits kennen gelernt: P will geheimes Bit $b \in \{0,1\}$ verschlüsseln.

P wählt Primzahlen p, q

berechnet $n = p \cdot q$ (p, q zunächst geheim)

bestimmt a so dass

$$\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = 1 \quad \text{(falls } b=1\text{)}$$

$$\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = -1 \quad \text{(falls } b=0\text{)}$$

In beiden Fällen ist $\left(\frac{a}{n}\right) = \left(\frac{a}{p}\right) \cdot \left(\frac{a}{q}\right) = 1$.

P übermittelt an V: (n, a)

V kann b nicht ermitteln.

Später übermittelt P an V: (p, q)

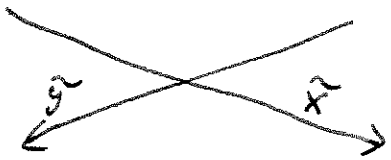
Dann kann sich V von b überzeugen.

P ist an b gebunden.

„Zufallszahl übers Telefon" (bei sich misstrauenden Teilnehmern):

Grundsätzlich könnte man das Diffie-Hellman-Protokoll nehmen, um eine beiden bekannte Zufallszahl zu erzeugen:

$$\tilde{x} = a^x \bmod n \qquad \tilde{y} = a^y \bmod n$$



$$z = (\tilde{y})^x \bmod n \qquad z = (\tilde{x})^y \bmod n$$

Falls aber Teilnehmer A zuerst sendet, so kann B seine Kommikation so wählen, dass möglicherweise eine ihm genehme Zahl z sich ergibt.

Lösung: Der zuerst Sendende commitment-verschlüsselt seine 1. Kommunikation. Dann sendet der 2. Teilnehmer seine Nachricht. Danach sendet der 1. Teilnehmer seine Entschlüsselung und beide berechnen denselben Zufallswert z.

Ein Zero knowledge Protokoll für 3-Färbbarkeit (welches ein NP-vollständiges Problem ist):

$$3\text{-COLOR} = \{ G \mid \overset{\text{Graph}}{G=(V,E)},\ V=\{1,\dots,n\},$$

ist ein 3-färbbarer Graph, d.h.

$$\exists\, c: V \to \{1,2,3\}\ \forall \{x,y\} \in E: c(x) \neq c(y) \}$$

Der Prover wählt zunächst eine zufällige Färbung $c: V \to \{1,2,3\}$. Dann baut er einen Zufallsgraphen $G=(V,E)$ so auf, dass die Färbung c zulässig ist.

Öffentlicher Schlüssel: G,

Geheimer Schlüssel: c

Protokoll:

P	V
Wählt Zufallspermutation $\pi \in S_3$. Sei $d = c \circ \pi$ die „Neufärbung" nach Permutation mit π	

Die Farben der Knoten
1, 2, ..., n werden
einzeln (mit separaten
Schlüsseln) commitment-verschlüsselt.
$k_1, \dots, k_n$

$\widetilde{d(1)}, \widetilde{d(2)}, \dots, \widetilde{d(n)}$ →

Wählt zufällig
$\{i,j\} \in E$

← i, j

Sendet Schlüssel
k_i, k_j um $d(i), d(j)$
zu entschlüsseln

k_i, k_j →

Überprüft, dass
$d(i) \neq d(j)$

Der falsche Prover $\tilde{P}$ kennt keine Färbung von G. Allerdings könnte er (im worst case) eine solche Färbung kennen, die nur bei einer Kante nicht zulässig ist. Daher wird $\tilde{P}$ bei einem solchen Protokoll mit W'keit $\frac{1}{m}$, $m = |E|$, entdeckt bzw. mit W'keit $(1 - \frac{1}{m})$ nicht entdeckt. Wiederholt man das Protokoll mt-mal, so bleibt $\tilde{P}$ nicht entdeckt mit W'keit $(1 - \frac{1}{m})^{mt} \leq e^{-t}$.

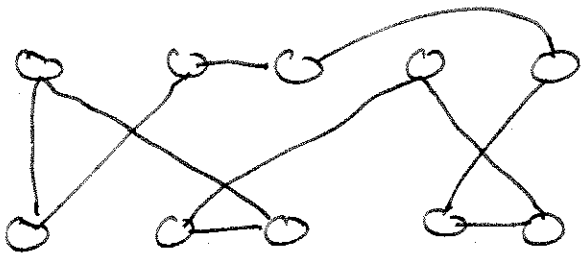
Grundsätzlich kann jedes NP-vollständige Problem im Rahmen eines entsprechenden Zero Knowledge-Protokolls eingesetzt werden (so wie hier: 3-Färbbarkeit). Dies erfordert eine sicheres Commitment-Verschlüsseln. Was das Zero Knowledge-Konzept betrifft, so führen solche Protokolle auf „Computational zero knowledge".

Ein weiteres Beispiel für ein NP-vollständiges Problem: HAMILTONKREIS =

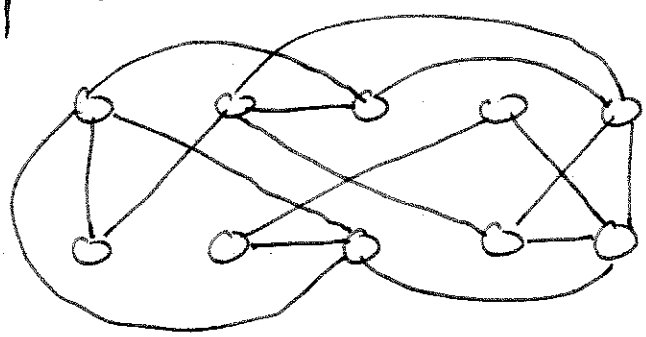
$\{ G \mid G=(V,E)$ ist ein Graph, der einen Hamiltonkreis enthält, also ein Pfad in G, der zum Startknoten zurückkehrt und der jeden Knoten des Graphen genau einmal besucht $\}$

Der Prover bereitet einen großen Zufallsgraphen mit einem ihm bekannten Hamiltonkreis vor:

- Zuerst nur den Hamiltonkreis

K: 

- Dann mit weiteren Zufallskanten auffüllen:

G: 

Öffentlich: der Graph G
Geheim: der darin enthaltene Hamiltonkreis K.

Protokoll, mit dem P gegenüber V beweist, dass er einen Hamiltonkreis in G kennt:

P	V

Wählt eine Zufalls-Permutation $\sigma \in S_n$ und berechnet $H = \sigma(G)$

H →

Wählt Zufalls-Bit $b \in \{0,1\}$

← b

Falls $b=0$, so ist $y = \sigma$

Falls $b=1$, so ist $y = \sigma(k)$

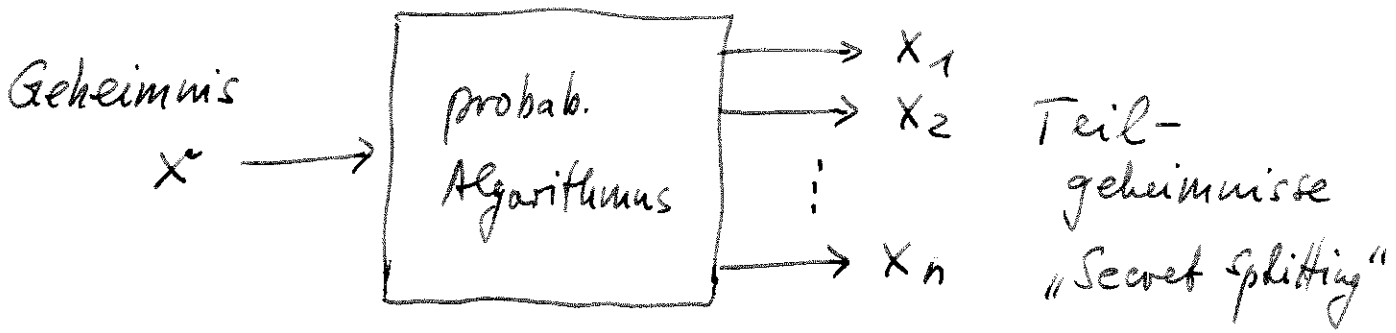
y →

Falls $b=0$, so wird überprüft: $H \stackrel{?}{=} \sigma(G)$

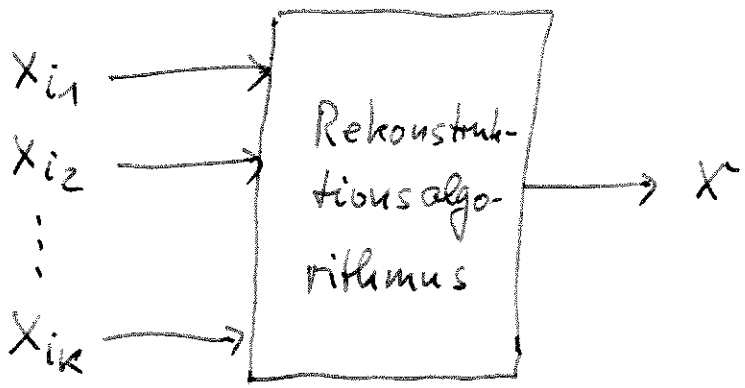
Falls $b=1$, so wird überprüft, ob y ein Hamilton-Kreis in H ist

Aufteilen von Geheimnissen

(ein weiteres Basisprotokoll, das in komplexeren Protokollen als Baustein benötigt wird).



Ein (k,n)-Schwellenwertsystem liegt vor, wenn es mit $< k$ vielen Teilgeheimnissen unmöglich ist, x zu rekonstruieren. Aber sobald k Teilgeheimnisse vorliegen, kann man x berechnen:



Zunächst einfacherer Spezialfall: $k = n = 2$

Sei $x \in \{0,1\}^m$ das Geheimnis.

Aufteilen des Geheimnisses:

x_1 ist Zufallsbitstring der Länge m

$$x_2 = x_1 \oplus x$$

↑ bitweise.

Wenn beide Teilgeheimnisse x_1, x_2 vorliegen, kann man x ermitteln: $x = x_1 \oplus x_2$

(Ähnliches passiert bei Visueller Kryptographie)

Bild x_1 besteht aus zufälligen „Pixeln"

 $\hat{=} 0$ oder  $\hat{=} 1$

Bild x_2 besteht aus denselben „Pixel" wie bei x_1, wenn an der Stelle eine Graufärbung erfolgen soll; x_2 enthält das komplementäre „Pixel", wenn Schwarzfärbung erfolgen soll.

Spezialfall: $k = n > 2$:

$x \in \{0,1\}^m$ sei das Geheimnis

$x_1, x_2, \ldots, x_{n-1}$ seien unabhängige, zufällige Bitstrings der Länge m.

$$x_n = x_1 \oplus x_2 \oplus \cdots \oplus x_{n-1} \oplus x$$

Jeder Bitstring (auch x_n) ist für sich betrachtet rein zufällig, aber mit allen zusammen (nicht jedoch mit $< n$ vielen) kann x rekonstruiert werden: $x = x_1 \oplus \cdots \oplus x_n$.

Der allgemeine Fall $k < n$ kann mit Polynomen realisiert werden.

Beachte: ein Polynom mit k Koeffizienten $a_0, a_1, \ldots, a_{k-1}$ also

$$p(x) = a_0 + a_1 x + \cdots + a_{k-1} \cdot x^{k-1}$$

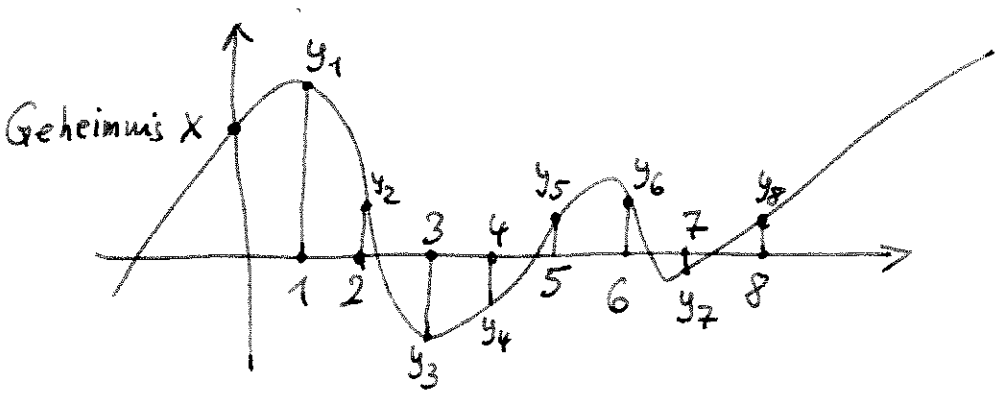
kann mittels k verschiedener Stützstellen

$(x_1, y_1), \ldots, (x_k, y_k)$ wobei $y_i = p(x_i)$

(z.B. Interpolation nach Lagrange)

eindeutig rekonstruiert werden. Zum

Beispiel $x_1 = 1, x_2 = 2, \ldots, x_k = k$.



Polynom vom Grad 5, also 6 Koeffizienten

Beispiel mit $n = 8$ und $k = 6$.

Das Geheimnis x könnte (z.B.) der Wert des Polynoms an der Stelle 0 sein (der y-Achsenabschnitt).

Wenn nicht genügend ~~Koeffizienten~~ Stützstellen vorliegen, kann x nicht ermittelt werden.

Das Polynom braucht nicht über dem Körper $\mathbb{R}$ sein (wie in der Zeichnung), sondern kann über einem endlichen Körper, z.B. $\{0, 1, \ldots, p-1\}$, p Primzahl, definiert sein.