Institut für Informatik
Prof. Dr. J. Rothe

Wintersemester 2023/24
Übungsblatt 9

Lösungsvorschläge

Kryptokomplexität 1

Bearbeitungszeit: 5. Dezember bis 13-15. Dezember
Verantwortlich: Roman Zorn

Aufgabe 1 : Fälschung von RSA Signaturen
Erich fängt die folgenden signierten Nachrichten $\langle m_i, \text{sig}_A(m_i)\rangle$ von Alice ab:

$$\langle 576, 102\rangle, \langle 215, 1595\rangle, \langle 338, 764\rangle, \langle 59, 1130\rangle, \langle 512, 339\rangle, \langle 26, 988\rangle, \langle 532, 1392\rangle$$

Der öffentliche Schlüssel ist $(n, e) = (1961, 7)$.

(a) ▶ Führen Sie den *chosen-plaintext* Angriff aus der Vorlesung für alle $r \in \{1, 2, 3\}$ durch, wobei Sie $e_i = 1$ für alle i wählen sollen.

Lösungsvorschlag: ▶ Benutze die Formeln aus der Vorlesung:

$$m = r^e \prod_{i=1}^{k} m_i^{e_i} \mod n$$

$$\text{sig}_A(m) = r \prod_{i=1}^{k} (\text{sig}_A(m_i))^{e_i} \mod n$$

$r = 1$: Wir berechnen $m_1 = 1^7 \cdot 576 \cdot 215 \cdot 338 \cdot 59 \cdot 512 \cdot 26 \cdot 532 \equiv 1333 \mod 1961$
und $\text{sig}_A(m_1) = 1 \cdot 102 \cdot 1595 \cdot 764 \cdot 1130 \cdot 339 \cdot 988 \cdot 1392 \equiv 889 \mod 1961$

$r = 2$: Wir berechnen $m_2 = 2^7 \cdot 576 \cdot 215 \cdot 338 \cdot 59 \cdot 512 \cdot 26 \cdot 532 \equiv 17 \mod 1961$
und $\text{sig}_A(m_2) = 2 \cdot 102 \cdot 1595 \cdot 764 \cdot 1130 \cdot 339 \cdot 988 \cdot 1392 \equiv 1778 \mod 1961$

$r = 3$: Wir berechnen $m_2 = 3^7 \cdot 576 \cdot 215 \cdot 338 \cdot 59 \cdot 512 \cdot 26 \cdot 532 \equiv 1225 \mod 1961$
und $\text{sig}_A(m_3) = 3 \cdot 102 \cdot 1595 \cdot 764 \cdot 1130 \cdot 339 \cdot 988 \cdot 1392 \equiv 706 \mod 1961$

(b) Nehmen Sie an, alle Nachrichten seien mit Blocklänge 2 über dem kanonischen Alphabet $\Sigma = \{A, B, \ldots, Z\}$ kodiert. Machen die Nachrichten von Alice und/oder die gefälschte Nachricht von Erich aus Aufgabenteil (a) Sinn?

Lösungsvorschlag: ▶ Die Nachrichten von Alice ergeben dekodiert:

$22, 4, 8, 7, 13, 0, 2, 7, 19, 18, 1, 0, 20, 12$, was dem deutschen Wort "Weihnachtsbaum" entspricht.

Die Nachricht von Erich ergibt keinen Sinn, da 1333 und 1225 nicht über dem

kanonischen Alphabet mit Blocklänge 2 dekodiert werden können:

Aufgabe 2 : Permutationschiffre

Geben sei die folgende DTM $M = (\Sigma, \Gamma, S, \delta, [\star,\star,\star], \square, F)$ mit $\Sigma = \{0,1\}$, $\Gamma = \{0,1,\widehat{0},\widehat{1},\square\}$, $S = \{z_N, z_L, z_E, [\star,\star,\star]\} \cup \{[X,\star,\star],[X,Y,\star],[X,Y,Z],[X,Y,Z]' \mid X,Y,Z \in \{0,1\}\}$, $F = \{z_E\}$ sowie δ durch die folgende Tabelle für $X,Y,Z \in \{0,1\}$:

δ	0	1	$\widehat{0}$	$\widehat{1}$	$\square$
$[\star,\star,\star]$	$([0,\star,\star],0,R)$	$([1,\star,\star],1,R)$			$(z_L,\square,L)$
$[X,\star,\star]$	$([X,0,\star],0,R)$	$([X,1,\star],1,R)$			
$[X,Y,\star]$	$([X,Y,0],\widehat{Y},L)$	$([X,Y,1],\widehat{Y},L)$			
$[X,Y,Z]$	$([X,Y,Z]',X,L)$	$([X,Y,Z]',X,L)$			
$[X,Y,Z]'$	(z_N,Z,R)	(z_N,Z,R)			
z_N	$(z_N,0,R)$	$(z_N,1,R)$	$([\star,\star,\star],0,R)$	$([\star,\star,\star],1,R)$	
z_L	$(z_L,0,L)$	$(z_L,1,L)$			$(z_E,\square,R)$

M verschlüsselt die Eingabe mit der Permutationschiffre und einem fixierten Schlüssel π unter der Voraussetzung, dass die Länge der Eingabe ein Vielfaches von 3 ist.

(a) ▶ Geben Sie die vollständige Konfigurationenfolge bei Eingabe von `011101` an. Bereiten Sie sich darauf vor die Funktionsweise von M anhand dieser Eingabe zu erläutern.

(b) ▶ Geben Sie die Permutation π, die M verwendet, an.

Lösungsvorschlag:

(a) (Wir geben die Konfigurationenfolge bei Eingabe `011101` an:

$[\star,\star,\star]011101 \vdash_M 0[0,\star,\star]11101 \vdash_M 01[0,1,\star]1101 \vdash_M 0[0,1,1]1\widehat{1}101 \vdash_M$
$[0,1,1]'00\widehat{1}101 \vdash_M 1z_N0\widehat{1}101 \vdash_M 10z_N\widehat{1}101 \vdash_M 101[\star,\star,\star]101 \vdash_M 1011[1,\star,\star]01 \vdash_M$
$10110[1,0,\star]1 \vdash_M 1011[1,0,1]0\widehat{0} \vdash_M 101[1,0,1]'1\widehat{1}0 \vdash_M 1011z_N\widehat{1}0 \vdash_M 10111z_N\widehat{0} \vdash_M$
$101110[\star,\star,\star]\square \vdash_M 10111z_L0 \vdash_M^* z_L\square101110 \vdash_M z_E101110.$

M merkt sich also immer 3 aufeinander folgende Symbole der Eingabe mit Hilfe der Zustände und beginnt dann diese Symbole von rechts nach links entsprechend π zu vertauschen. Das Ende der 3 Symbole wird mit einem $\widehat{}$ markiert. Nach dem Vertauschen der drei Symbole wandert der Kopf von M wieder nach rechts und wiederholt den Prozess für die nächsten 3 Symbole. Ist der rechte Rand erreicht, so wandert der Kopf von M zum Anfang der Eingabe und M hält an.

(b) Die von M verwendete Permutation lässt sich durch

$$\pi = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}$$

angeben. Dabei wird π immer auf 3 aufeinander folgende Symbole der Eingabe angewendet. Es gilt zum Beispiel $\pi(1) = 3$, damit ist gemeint, dass wir auf die Stelle des 1. Symbols das 3. Symbol abbilden. Formal gilt also für drei aufeinander folgende Symbole $x_1x_2x_3$, dass nach Anwendung von π gilt

$$x_{\pi(1)}x_{\pi(2)}x_{\pi(3)} = x_3x_1x_2.$$

Aufgabe 3 : Komplexitätsmaße für det. Maschinen

Betrachten Sie die folgende DTM $M' = (\Sigma, \Gamma, Z, \delta, z_0, \square, F)$ mit $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \square\}$, $Z = \{z_0, z_C, z_1, z_2, z_3, z_A, z_R\}$, $F = F_A \cup F_R$, $F_A = \{z_A\}$, $F_R = \{z_R\}$ und δ:

δ	0	1	$\square$
z_0	$(z_C, 0, R)$	$(z_1, 1, R)$	$(z_R, \square, N)$
z_C	$(z_R, 0, N)$	$(z_R, 1, N)$	$(z_A, \square, N)$
z_1	$(z_1, 0, R)$	$(z_1, 1, R)$	$(z_2, \square, L)$
z_2	$(z_3, 0, L)$	$(z_R, 1, N)$	
z_3	$(z_A, 0, N)$	$(z_R, 1, N)$	

Dabei bezeichnet F_A die Menge der akzeptierenden Endzustände und F_R die Menge der ablehnenden Endzustände.

(a) ▶ Bestimmen Sie $Time_{M'}(101)$ und $Space_{M'}(101)$ und geben Sie die entsprechende Konfigurationenfolge an.

(b) ▶ Bestimmen Sie $time_{M'}(3)$ und $space_{M'}(3)$. Begründen Sie Ihre Antworten.

(c) ▶ Welche Sprache akzeptiert M'?

Lösungsvorschlag:

(a) Gebe die Konfigurationenfolge bei Eingabe `101` an:

$$z_0101 \vdash_{M'} 1z_101 \vdash_{M'} 10z_11 \vdash_{M'} 101z_1\square \vdash_{M'} 10z_21\square \vdash_{M'} 10z_R1\square.$$

Folglich gilt $Time_{M'}(101) = 5$ sowie $Space_{M'}(101) = 4$.

(b) Für alle Eingaben x mit $|x| = 3$ gilt:

(i) Falls x mit 0 beginnt wird die Eingabe nach einem Schritt abgelehnt.

(ii) Falls x mit 1 endet wird die Eingabe nach fünf Schritten abgelehnt.

(iii) Falls x von der Form $1z0$ für $z \in \Sigma$ ist, dann gilt (hier für 100):

$$z_0100 \vdash_{M'} 1z_100 \vdash_{M'} 10z_10 \vdash_{M'} 100z_1\square \vdash_{M'}$$
$$10z_20\square \vdash_{M'} 1z_300\square \vdash_{M'} 1z_A00\square.$$

Somit ist $Time_{M'}(100) = 6$ das Maximum und insgesamt gilt $time_{M'}(3) = 6$. Da höchstens ein zusätzliches $\square$ bei Eingaben, die mit 1 beginnen, gelesen wird gilt $space_{M'}(3) = 4$.

(c) Die von M' akzeptierte Sprache lässt sich formalisieren als

$$L(M') = \{\ \text{bin}(n) \mid n \in \mathbb{N} \text{ mit } n \equiv 0 \bmod 4\},$$

also akzeptiert M' alle natürlichen, binär kodierten Zahlen die durch 4 teilbar sind.

Aufgabe 4 : Kodierungen für Graphen

Sei $G = (V, E)$ mit $V = \{v_1, \ldots, v_n\}$ und $E = \{e_1, \ldots, e_m\}$ ein ungerichteter Graph. Um G als Eingabe einer TM darstellen zu können, muss G als Eingabestring kodiert werden. Betrachten Sie folgende Kodierungen:

(i) *Kodierung als Knoten-/Kantenliste*: $\text{vel}_1(G) = V[\text{bin}(1)] \ldots V[\text{bin}(n)]$ und $\text{vel}_2(G) = (V[\text{bin}(i_1)]V[\text{bin}(j_1)]) \ldots (V[\text{bin}(i_m)]V[\text{bin}(j_m)])$, wobei $\{v_{i_k}, v_{j_k}\} = e_k$. Dann wird G als Zeichenkette $\text{vel(G)} = \text{vel}_1(G)\text{vel}_2(G)$ über dem Alphabet {0,1,V,(,),[,]} kodiert.

(ii) *Kodierung als Adjazenzmatrix*: Stelle G als Zeichenkette $\text{adj}(G)$ dar, indem die Zeilen der Adjazenzmatrix durch das Zeichen / voneinander getrennt werden.

Bearbeiten Sie mit diesen Kodierungen die folgenden Aufgaben:

(a) Gegeben Sei der Graph $G' = (V, E)$ mit $V = \{v_1, v_2, v_3, v_4\}$ und

$$E = \{\{v_1, v_2\}, \{v_1, v_4\}, \{v_2, v_4\}, \{v_3, v_4\}\}.$$

▶ Stellen Sie den Graph G' in beiden Kodierungen dar.

(b) Sei $\star$ eine Grapheigenschaft und seien

$$\Pi = \{\text{vel}(G) \mid G \text{ erfüllt Eigenschaft } \star\},$$
$$\Gamma = \{\text{adj}(G) \mid G \text{ erfüllt Eigenschaft } \star\}$$

zwei Probleme über Graphen, die sich nur in der Darstellung der Eingabe unterscheiden.

▶ Beweisen Sie, dass die folgende Aussage

$$\Pi \in \text{P} \iff \Gamma \in \text{P}$$

gilt. Sie müssen keine Turingmaschinenprogramme o.Ä. angeben.

Lösungsvorschlag:

(a) Kodieren von G' mittels *Kodierung als Knoten-/Kantenliste* ergibt

$$\text{vel}(G) = V[1]V[10]V[11]V[100](V[1]V[10])(V[1]V[100])(V[10]V[100])(V[11]V[100]).$$

Kodieren von G' mittels *Kodierung als Adjazenzmatrix* ergibt

$$\text{adj}(G) = 0101/1001/0001/1110.$$

(b) Wir zeigen im ersten Schritt, dass man beide Kodierungen innerhalb polynomieller Zeit bzgl. der Instanzgröße ineinander überführen kann.

(i) Knoten-/Kantenliste zu Adjazenzmatrix:
Laufe die Zeichenkette $I = vel(G)$ ab, dann ist bekannt wie viele Knoten $n \leq |I|$ der Graph hat und man kann bereits die leere Adjazenzmatrix in Zeit $\mathcal{O}(|I| + |I|^2)$ aufschreiben. Nun müssen für jede der $m \leq n^2$ Kanten zwei Einträge in der Adjazenzmatrix verändert werden, das geht in Zeit $\mathcal{O}(mn^2) \subseteq \mathcal{O}(n^4) \subseteq \mathcal{O}(|I|^4)$.

(ii) Adjazenzmatrix zu Knoten-/Kantenliste:
Laufe die Zeichenkette $I = adj(G)$ bis zum ersten / ab. Dann ist die Anzahl der Knoten n im Graph bekannt. Das geht in Zeit $\mathcal{O}(|I|)$. Schreibe nun die Knoten binär kodiert auf. Für jeden Knoten benötigen wir $\mathcal{O}(\log(n))$ Zeit, also insgesamt $\mathcal{O}(n \log(n))$ Zeit. Da $n^2 \leq |I|$ gilt, folgt $\mathcal{O}(n \log(n)) \subseteq \mathcal{O}(|I|)$. Füge für jede 1 in der oberen Dreiecksmatrix (die Adjazenzmatrix ist symmetrisch, da der Graph ungerichtet ist) eine entsprechende Kante zur Kantenliste hinzu. Es gibt $m \leq n^2$ Kanten und für das Aufschreiben jeder Kante brauchen wir Zeit $\mathcal{O}(\log(n))$. Damit können wir alle Kanten in $\mathcal{O}(m \log(n)) \subseteq \mathcal{O}(n^2 \log(n)) \subseteq \mathcal{O}(|I|^2)$ aufschreiben.

Wir wissen nun, dass wir beide Kodierungen in polynomieller Zeit ineinander überführen können. Angenommen es gilt nun $\Pi \in \text{P}$. Das bedeutet, dass wir eine Instanz I von Π in der Zeit $f(|I|)$ für ein Polynom f lösen können. Mit der vorherigen Überlegung wissen wir, dass wir eine Instanz I' von Γ in Zeit $g(|I'|)$ für ein Polynom g in eine Instanz I'' für Π überführen können. Dann gilt für die überführte Instanz $|I''| \leq g(|I'|)$.
Insgesamt brauchen wir also eine Zeit von $g(|I'|)$ um die Instanz von Γ in eine Instanz von Π zu überführen und anschließend eine Zeit von $f(|I''|)$ um die überführte Instanz zu lösen. Da $|I''| \leq g(|I'|)$ gilt, folgt

$$f(|I''|) \leq f(g(|I'|)).$$

Da Polynome unter Komposition abgeschlossen sind, ist die insgesamt benötigte Zeit

$$g(|I'|) + f(|I''|) \leq g(|I'|) + f(g(|I'|))$$

auch wieder polynomiell in $|I'|$ und wir können die Instanz von Γ auch in P lösen. Analog argumentiert man für die umgekehrte Richtung.