Informatik.Softwaresysteme

Vorlesung - Übung - Praktikum

IT-Sicherheit und Datenschutz

Zusammenfassung der Vorlseung

Sommersemester: 2019

Student:
Steffen Holtkamp
Matrikelnummer: 2016xxxxx

Westfälische Hochschule - Bocholt
Prof. G. Kroesen
Münsterstraße 265
46397 Bocholt

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

1 Einleitung

1.1 Beschreibung

IT-Sicherheit und Datenschutz

2 Vorlesung 1

Montag: 08.April 2019

2.1 Gruppen

Damit wir eine Gruppe bilden können, müssen folgende drei Regeln erfüllt werden.

$$a * (b * c) = (a * b) * c \tag{1}$$
$$id * a = a \tag{2}$$
$$a^{-1} * a = id \tag{3}$$

Dabei sei $\mathbb{M}$ eine Menge, in der IT endlich und * eine Operation zwischen zwei Elementen der Menge $\mathbb{M}$. Man beachte, dass eine Gruppe nur das Kommutativ Gesetz erfüllt.

Des weiteren gilt in den betrachteten Gruppen:

$$a * id = id * a = a \tag{4}$$

da,

$$a * id = a * (a^{-1} * a) \tag{5}$$
$$= (a * a^{-1}) * a \tag{6}$$
$$\tag{7}$$

und

$$\begin{aligned} a * id &= id * a = a && (8)\\ a &= id * a | erweitert mit a^{-1} && (9)\\ a * a^{-1} &= id * a^{-1} | mit (c * a^{-1} = id && (10)\\ &= (c * a^{-1}) * (a * a^{-1}) && (11)\\ &= ((c * a^{-1}) * a) * a^{-1} && (12)\\ &= (c * (a^{-1} * a)) * a^{-1} && (13)\\ &= (c * (id)) * a^{-1} | id ist ein neutrales Element && (14)\\ &= c * a^{-1} | mit c * a^{-1} = id && (15)\\ &= id && (16)\\ & && (17) \end{aligned}$$

Sei # die Menge der Elemente in $\mathbb{M}$. # wird auch als Leiterchen bezeichnet.

Sei $a^m = a^n$, dann ist die kleinst mögliche differenz zwischen den beiden k und es gilt.

$$\begin{aligned} a^m &= a^n && (18)\\ a^{m+k} &= a^n && (19)\\ a^m * a^k &= a^n && (20)\\ => a^k &= id && (21) \end{aligned}$$

Sei $a * b = a * c$ ist $b = c$

Es gibt die Menge $\mathbb{M} = \{id, b_1, b_2, ..., b_{\#-1}\}$ und die Menge $\mathbb{M}_0 = \{id, a, a^2, a^3, ..., a^{k-1}\}$ in der es keine Dubletten gibt, das die erste Dublette $id = a^k$ wäre.

Wenn man nun neue Mengen bildet aus $\mathbb{M}$ und $\mathbb{M}_0$ gibt es # viele Mengen. Enthält eine Menge nur ein Element, dass auch in einer anderen Menge vorkommt, so sind beide Mengen identisch.

Darauß ergibt sich, dass

$$\begin{aligned} n * k &= \# && (22)\\ a^{\#} = a^{kn} = (a^k)^n = id^n &= id && (23) \end{aligned}$$

2.2 Diffi-Hellmann

	Alice	Bob
	M, *, #	M, *,#
	$g \in \mathbb{M}$	$g \in \mathbb{M}$
private Key	1 <x <#	1 <y <#
public Key	g^x	g^y
Schlüsselvereinbarung	g^y	g^x
	$(g^y)^x$	$(g^x)^y$

Tabelle 1: Diffi-Hellmann - Schlüsselvereinbarung

Dabei muss # muss genau bekannt sein. g ist ein öffentliches Generatorpolynom. Und die Gruppe hat meist zwischen 2^{200} bis 2^{500} Werte.

Wenn man einen Text verschlüsselt, wird jedes Zeichen des Textes verschlüsselt und übertragen in der Form $c_i g^{x_i y_i}$ und dann mit Hilfe des inversen entschlüsselt: $c_i g^{x_i y_i} * g^{-x_i y_i} = c_i$

Das inverse ist einfach $g^{\#-x}$.

Einschub: Es git eine Angriffsform die die Laufzeit untersucht, um anhand der Laufzeit und geschätzten Anzahl an Operationen Rückschlüsse auf den Verschlüsselungsalgorithmus zu machen.

2.3 Gruppendarstellung und Untergruppen

Man kann eine Menge $\mathbb{M} = \{id, a, b\}$ als Tabelle aufschreiben.

*	id	a	b
id	id	a	b
a	a	b	id
b	b	id	a

Tabelle 2: abstracte Menge mit 3 Elementen

Mann kann auch einfach die + Operation mod 3 in einer solchen Tabelle aufschreiben. Das wichtige ist, dass in einer Spalte und einer Zeile, jedes Element der Menge $\mathbb{M}$ nur einmal vorkommen darf.

+ mod 3	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

Tabelle 3: Menge 3 mit Zahlen

Wenn man beide Tabellen vergleicht stellt man fest, dass sie isomorph sind. Das bedeutet, dass man einfach id = 0, a = 1 und b = 2 definieren kann und feststellt, dass die abstrakte Tabelle gleich der 3-Addition ist.

Des weiteren gilt, dass alle Gruppen die eine Größe # haben, die eine Primzahl sind haben genau eine Gruppe und besitzen keine Untergruppen. Daher sind alle Gruppen, die die selbe Größe # haben und diese Größe eine Primzahl ist isomorph.

Untergruppen entstehen wenn # keine Primzahl ist und so # einen Teiler hat. Jeder Teiler der durch Primfaktorzerlegung entsteht ist eine eigene Gruppe.

XOR	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

Tabelle 4: XOR Menge 4

Diese Menge hat die Untergruppe:

XOR	0	1
0	0	1
1	1	0

Tabelle 5: Untergruppe Menge 2

Untergruppen spielen eine wichtige Rolle bei der Wahl des Generatorpolynoms.

2.4 Generatorpolynom

Das Generatorpolynom g^x muss so gewählt werden, dass es eine möglichst große Teilmenge von $\mathbb{M}$ abdeckt. Wenn es nur einen kleinen Bereich abdeckt ist es leicht zu knacken.

Beispiel: Sei die Primzahl 11 -> Rechnung mit mod 11

Vergleich von Generatorpolynomen:

	1	2	3	4	5	6	7	8	9	10
g=2	2	4	8	5	10	9	7	3	6	1
g=4	4	5	9	3	1					
g=10	10	1								

Tabelle 6: Generatorpolynom

Man erkennt wenn man mit $g = 2$ rechnet, dass man alle Elemente des Körpers durchgeht, bevor man auf das neutrale Element, die 1 stößt. Wenn man $g = 4$ nimmt, dann werden nur noch 5 Elemente der Menge genutzt. Wenn man $g = 10$ nutzt werden sogar nur 2 Elemente genutzt. Je weniger unterschiedliche Elemente genommen werden, desto einfacher ist es herauszubekommen welchen Wert x hat, sprich was der private Key ist.