

Humboldt-Universität zu Berlin
Institut für Informatik

WS 2005/06

Proseminar: Kryptografische Algorithmen und Protokolle

Dozenten: Prof. Johannes Köbler, Olaf Beyersdorf

Multiparty Computations

Von Saskia Becker

1. Was bedeutet Multiparty – Computations?

Multiparty – Computations (Mehr – Parteien -Berechnungen) bezeichnet ein Gebiet der Kryptografie, das sich mit Protokollen beschäftigt, an deren Durchführung zwei oder mehr Teilnehmer beteiligt sind, die gemeinsam etwas berechnen möchten.

Das klassische Beispiel hierfür, klingt zunächst etwas unrealistisch:

Eine Gruppe von Personen möchte herausfinden, wer aus der Gruppe am meisten verdient. Dabei möchte keiner sein eigenes Einkommen verraten und es kann vorkommen, dass sich einige Personen aus der Gruppe misstrauen. Trotzdem soll am Schluss jeder erfahren, wer das höchste Gehalt hat.

Obwohl dieses Problem auf den ersten Blick unlösbar zu sein scheint, gibt es Methoden in der Multiparty – Computation die dafür eine Lösung bestimmen können. Wie das geht, möchte ich im folgenden anhand einiger Beispielm Modelle aus der Multiparty – Computation vorführen.

Dabei werde ich zunächst das zugrundeliegende Modell der Multiparty – Computations erläutern, um so ein besseres Grundverständnis der späteren Verfahren zu ermöglichen.

Anschließend stelle ich drei der wichtigsten Verfahren vor und veranschauliche diese jeweils mit Hilfe eines konkreten Beispiel-Protokolls. Zuletzt werde ich noch kurz auf Vor- und Nachteile von Multiparty – Computations im Vergleich zu anderen kryptografische Verfahren eingehen.

2. Zugrunde liegendes Modell

Unter einer Multiparty – Computation versteht man die Durchführung eines Protokolls mit:

- n Eingaben: die Geheimnisse der n Teilnehmer
- n Ausgaben: Ergebnisse der n Teilnehmer

Ziel: Am Ende des Protokolls soll jeder der Teilnehmer (TN) das korrekte Ergebnis erhalten, ohne die Eingabe (manchmal auch die Ausgabe) der anderen zu erfahren.

In der Theorie stellt man sich die Umsetzung dieser Forderungen wie folgt vor:

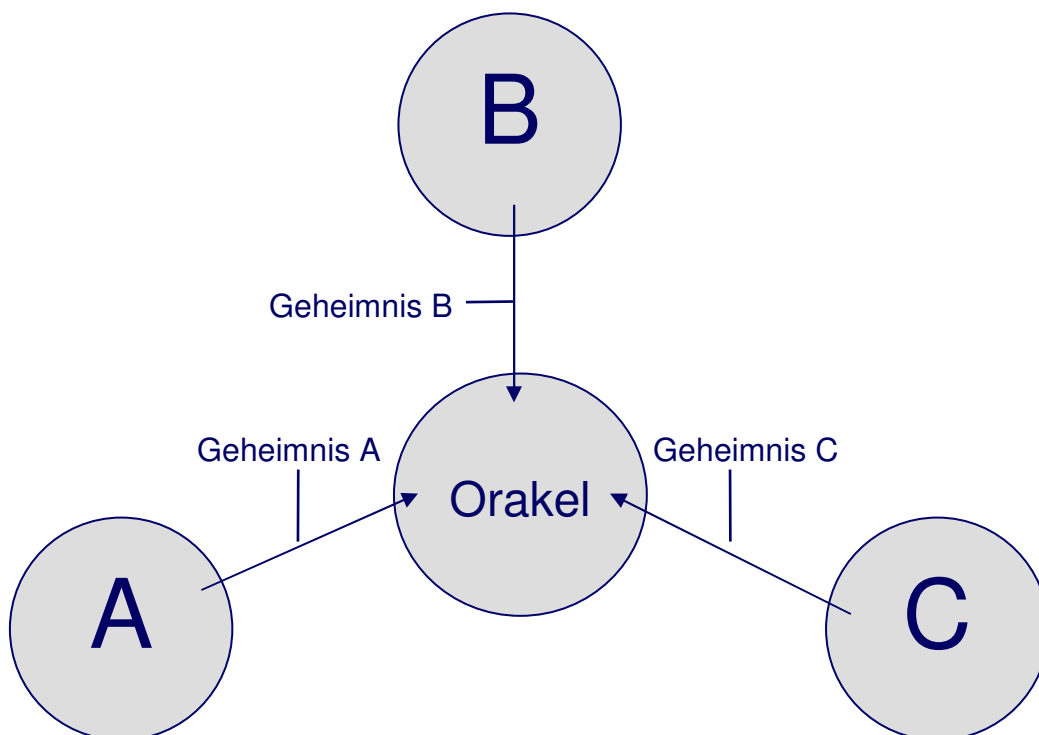


Abb.1: Idealisiertes Modell – Eingabe (Jeder TN sendet sein Geheimnis an ein Orakel.)

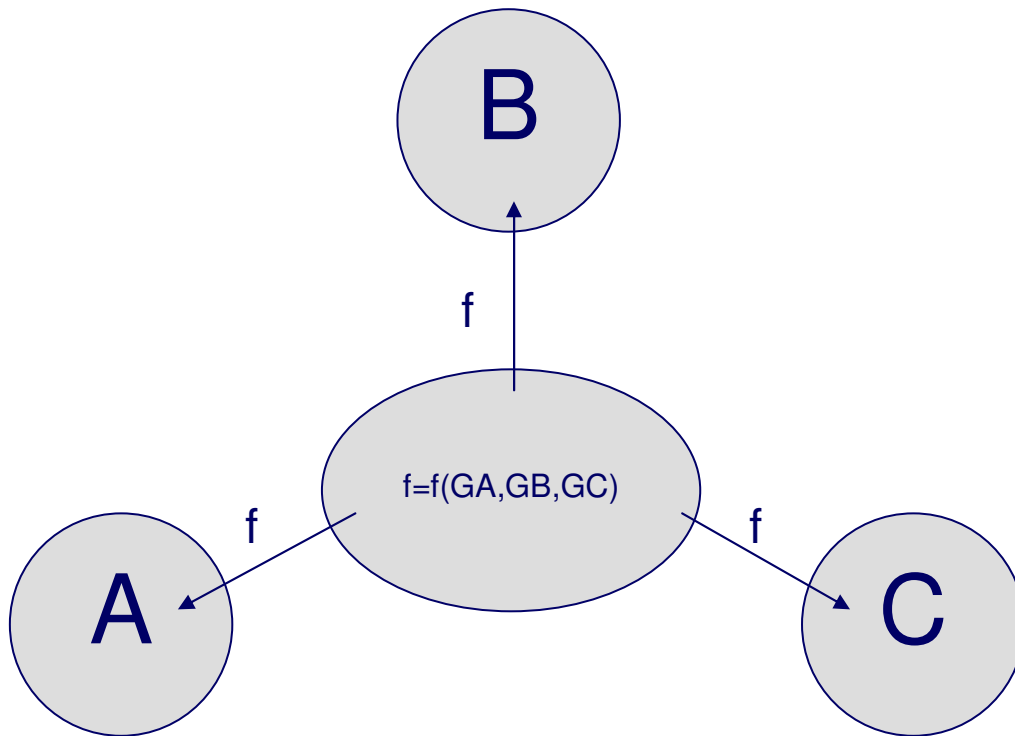


Abb.2: Idealisiertes Modell – Ausgabe (Das Orakel berechnet die einzelnen Ausgaben und sendet sie an die entsprechenden TN zurück.)

Das Orakel ist dabei nur ein theoretisches Konstrukt und kann in der Praxis zum Beispiel durch ein entsprechendes Protokoll ersetzt werden. Die entscheidende Frage lautet jedoch, inwieweit man das Orakel in einem realistischen Protokoll, in dem sich unter Umständen nicht alle TN korrekt verhalten, nachahmen kann.

Zur Beantwortung dieser Frage befassen wir uns für einen Moment mit der Sicherheit von Multiparty – Computations – Modellen:

Im allgemeinen ist ein System sicher, wenn ein effizienter Angreifer all das, was er nach seinem Angriff erreichen kann, auch vor seinem Angriff, also in einem idealisierten Modell, hätte erreichen können. Um die Sicherheit für eine Multiparty – Computation zu definieren, vergleicht man also die Möglichkeiten eines Angreifers in einem konkreten Protokoll mit denen im idealisierten Modell. Dies führt zu folgender Definition:

Ein Multiparty – Computations – Modell ist sicher, wenn ein effizienter Angreifer im realen Protokoll die gleichen Informationen erhält wie im idealisierten Modell.

Für sichere Multiparty- Computation soll also das konkrete Protokoll die gleichen Eigenschaften erfüllen wie das idealisierte.

Wenn ein Modell erstmal sicher ist, muss man sich nur noch gegen Angriffe absichern, die auch im idealisierten durchführbar sind.

Ob sich sichere Multiparty – Computations – Protokolle entwerfen lassen, hängt somit entscheidend von den Eigenschaften des zugrunde liegenden Modells ab.

Die folgenden Eigenschaften spielen dabei eine wichtige Rolle:

1. Kommunikation:

Die Standardannahme hierbei ist die Authentizität des Kommunikationskanals, d.h. der Angreifer kann die zwischen ehrlichen TN gesendeten Nachrichten nur lesen, aber nicht ändern. Eine stärkere Annahme ist die Existenz eines privaten Kommunikationskanals. In diesem Fall kann der Angreifer die von ehrlichen TN gesendeten Nachrichten nicht mal mehr lesen. Häufig wird auch von einem Broadcast – Kanal ausgegangen, bei dem die TN Nachrichten authentisch und gleichzeitig an alle anderen TN senden können.

2. Angreifer:

Angreifer können gesendete Nachrichten abhören oder verändern und (*unehrliche*) TN korrumpieren. Man unterscheidet zwischen *nicht-adaptiven* (Zahl der unehrlichen TN steht vor Beginn fest) und *adaptiven* (kann während des Protokolls weitere TN korrumpieren) Angreifern. Weiterhin kann ein *aktiver* Angreifer die unehrlichen TN zwingen, falsche Nachrichten zu schicken, während ein *passiver* Angreifer nur erhaltene Informationen verwenden kann.

3. Sicherheit:

Typische Sicherheitsziele sind...

- Geheimhaltung der Geheimnisse der ehrlichen TN
- alle ehrlichen TN sollen nach Durchführung des Protokolls das korrekte Ergebnis erhalten.

Manche Modelle lassen auch den Abbruch des Protokolls durch einen der unehrlichen TN zu. In diesem Fall kann es passieren, dass einige der ehrlichen TN ihr Ergebnis erhalten, andere ihr Ergebnis nicht erhalten, den Protokollabbruch aber bemerken.

Typische Annahmen für die Entwicklung eines sicheren Multiparty – Computation – Protokolls sind ein effizienter Angreifer, ein authentischer Kommunikationskanal und das korrekte Verhalten von mindestens der Hälfte der TN. Ein Angreifer soll also darauf beschränkt sein, die Eingaben der unehrlichen TN zu ändern und für seine Ein- und Ausgaben zu nutzen.

Im folgenden betrachte ich nur den Fall eines nicht-adaptiven Angreifers. Sichere Multiparty – Computation ist zwar auch bei einem adaptiven Angreifer möglich, benötigt dann aber andere Voraussetzungen und dies würde hier den Rahmen sprengen. Ich gehe also ab jetzt von einem effizienten, nicht-adaptivem Angreifer und einem authentischen Kommunikationskanal aus. Falls der Angreifer aktiv ist, so fordere ich weiter einen Broadcast-Kanal.

3. Secret – Sharing – Verfahren

Ziel: Aufteilung eines Geheimnisses unter mehreren Personen, so dass es später nur unter bestimmten Bedingungen rekonstruiert werden kann.

Einsatzgebiete:

- Zugangskontrolle: Das Ergebnis der Rekonstruktion wird mit dem im System gespeicherten Geheimnis verglichen. Bei einer Übereinstimmung wird der Zugang erteilt, sonst nicht.
- Geheimniserzeugung: Das rekonstruierte Geheimnis, das vorher nicht im System vorlag, wird kryptografisch weiterverarbeitet, z.B. als Signaturschlüssel.
- Als Baustein auch bei elektronischen Wahlen und elektronischen Münzen

Beispiele:

Secret – Sharing – Verfahren mit komplexen Zugriffsstrukturen:

Das Geheimnis kann nur dann rekonstruiert werden, wenn die Teilgeheimnisse aus einer zuvor festgelegten Zugriffsstruktur stammen. Beispielsweise könnte man verlangen, dass nur entweder der Präsident alleine oder drei seiner wichtigsten Minister zusammen eine bestimmte Rakete starten können. Alle Aufteilungsmöglichkeiten die man dabei im Kopf haben mag, lassen sich mathematisch durch den Begriff des (m, n) -Schwellenschemas modellieren. Falls auch jede Obermenge der zuvor festgelegten Zugriffsstruktur zur Geheimnisrekonstruktion zulässig ist, heißt die Zugriffsstruktur monoton.

(t, n) – Schwellenschemata (threshold schemes):

Idee: In einem (t, n) – Schwellenverfahren ($t \leq n, t, n \in \mathbb{N}$) wird das Geheimnis s in n Teilgeheimnisse s_1, \dots, s_n aufgeteilt, so dass gilt:

- Das Geheimnis kann aus t oder mehr Teilgeheimnissen rekonstruiert werden.
- Aus $t' < t$ Teilgeheimnissen kann das Geheimnis s nicht berechnet werden.

Man bezeichnet t dabei als die Schwelle des Verfahrens.

Verfahren: Die Aufteilung des Geheimnisses s wird durch eine vertrauenswürdige Instanz, dem so genannten Dealer, übernommen. Die so bestimmten Teilgeheimnisse werden dann über einen sicheren Kanal an die n TN verteilt. Zur Rekonstruktion von s müssen sich mindestens t der TN zusammenfinden.

Zur Veranschaulichung ein konkretes Beispiel:

3.1. Das Schwellenschema von Shamir

Als Grundlage dienen uns hier polynomiale Gleichungssysteme und das Wissen, dass ein Polynom vom Grad $t-1$ aus $K[x]$, K endlicher Körper, durch t verschiedene Punkte eindeutig bestimmt ist. Das Verfahren funktioniert nun wie folgt:

Wähle zufällig eine Primzahl p , so dass p größer ist als die maximale Anzahl von Geheimnisteilern und größer als die maximal mögliche Länge des Geheimnisses.

Alle weiteren Berechnungen finden in Z_p statt.

Der Dealer wählt zufällig Koeffizienten $c_{t-1}, \dots, c_1, c_{t-1} \neq 0$, aus Z_p aus und bildet so ein Polynom vom Grad $t-1$:

$$F(x) = c_{t-1} \cdot x^{t-1} + c_{t-2} \cdot x^{t-2} + \dots + c_1 x + c_0$$

Die Koeffizienten c_{t-1}, \dots, c_1 sind zufällig und geheim und c_0 ist das aufzuteilende Geheimnis s .

Nun erzeugt er die n Teilgeheimnisse s_i mit $i=1, \dots, n$ in der Weise, dass $s_i = F(x_i)$ gilt.

Die x_i werden vom Dealer beliebig gewählt und öffentlich bekannt gegeben.

Jeder Teilnehmer P_i erhält das Teilgeheimnis s_i .

Zur Rekonstruktion von s müssen sich mindestens t der n Teilnehmer zusammenfinden und bereit sein, ihr Teilgeheimnis preis zu geben.

Gemeinsam können sie dann, zum Beispiel mit Hilfe der Lagrangeschen Interpolationsformel

$$f(x) = \sum_{i=1}^t \frac{(x-x_1) \cdot \dots \cdot (x-x_{i-1}) \cdot (x-x_{i+1}) \cdot \dots \cdot (x-x_t)}{(x_i-x_1) \cdot \dots \cdot (x_i-x_{i-1}) \cdot (x_i-x_{i+1}) \cdot \dots \cdot (x_i-x_t)} \cdot s_i$$

oder durch Lösung des Gleichungssystems $f(x_1), f(x_2), \dots, f(x_t)$, das Geheimnis $s = f(0)$

berechnen. Finden sich zur Geheimnisaufdeckung nur $t' < t$ Teilnehmer zusammen, so können sie das Geheimnis nicht rekonstruieren, da jeder Punkt auf der y -Achse in Frage kommen kann. Mehr als t Teilnehmer haben keinen weiteren Einfluss auf das Ergebnis.

Um die Sicherheit einer Multiparty – Computation zu beschreiben, betrachtet man vor allem das mögliche Fehlverhalten unehrlicher TN, das sich selbst in einem idealisierten Modell nicht verhindern läßt. Das Schwellenschema von Shamir besteht aus zwei Phasen, in denen jeweils bestimmte Aktionen nicht ausgeschlossen werden können:

Geheimnisaufteilung:

- Der Dealer kann alle Teilgeheimnisse s_i falsch berechnen.
- Der Dealer kann die Teilgeheimnisse s_i veröffentlichen.
- Der Dealer kann $t' < t$ Teilgeheimnisse ausgeben.

Geheimnisrekonstruktion:

- Die TN weigern sich, die Teilgeheimnisse preiszugeben.
- Die TN geben einen falschen Wert an.

Folgerung: Für eine sichere Durchführung von Shamirs – Schwellenschema müssen sich Dealer und Teilnehmer korrekt verhalten, was im praktischen Einsatz jedoch eine unrealistische Forderung ist. Als Lösung fordert man statt dessen, dass ein solches System auch dann noch robust sein soll, wenn sich eine relativ kleine Menge von TN nicht korrekt verhält. Unter dieser Voraussetzung gibt es zahlreichen Varianten von Shamirs – Schwellenschema die unter anderem Schutz bieten gegen:

- Einen betrügerischen Dealer,
- Nicht zu viele betrügerische TN

3.2. Verifizierbare Geheimnisaufteilung

Das Ziel der verifizierbaren Geheimnisaufteilung ist die Kontrolle der Teilnehmer und des Dealers auf korrektes Verhalten. Als entscheidendes Hilfsmittel verwenden wir dabei homomorphe Einwegfunktionen $E : (G, +) \rightarrow (H, \oplus)$, G, H Gruppen.

Einfaches Beispiel: Sei Geheimnis $s \in G$, dann: $s = s_1 + s_2$. Dealer veröffentlicht $E(s), E(s_1), E(s_2)$ und sendet die s_i an TN A und B. A und B können nun den Dealer und den jeweils anderen TN überprüfen.

Dieses Verfahren läßt sich leicht auf n TN verallgemeinern:

Konkretes Beispiel:

- Verwende als Einwegfunktion die diskrete Exponentialfunktion in \mathbb{Z}_p zur Basis g .
- Dealer wählt zufällig Koeffizienten c_{t-1}, \dots, c_1 aus \mathbb{Z}_p , aus und bildet so ein Polynom f vom Grad $t-1$, so dass gilt: $s = f(0)$, wobei $c_0 = s$.
- Dealer veröffentlicht die $g^{c_j} \bmod p$ ($j=0, \dots, t-1$)

- Dealer berechnet die Teilgeheimnisse $s_i = f(x_i)$, wobei die x_i , $i=1, \dots, n$, beliebig gewählt werden. Veröffentlicht alle x_i und alle $g^{f(x_i)}$.
- Dealer sendet die $s_i = f(x_i)$ vertraulich an die TN

Jeder der TN kann Folgendes feststellen:

- Wurde $g^{f(x_i)}$ vom Dealer richtig berechnet?
(TN P_i kennt $f(x_i)$)
- Gilt für alle Teilgeheimnisse: $\prod_{j=0}^{t-1} (g^{c_j})^{x_i^j} \equiv g^{f(x_i)} \pmod{p}$?

Der Dealer hat damit praktisch keine Betrugsmöglichkeiten mehr, da die TN alle von ihm berechneten und ausgegebenen Werte überprüfen können.

Da für alle Teilgeheimnisse der Wert $g^{f(x_i)}$ öffentlich ist, können alle Beteiligten feststellen, ob ein TN das korrekte Teilgeheimnis preisgegeben hat oder nicht.

Fazit: Dealer und TN können durch verifizierbare Geheimnisaufteilung auf Ehrlichkeit überprüft werden.

4. (t,n) – Treshold – Signaturverfahren

Ein großer Nachteil von Secret – Sharing, ist die Notwendigkeit, zur Rekonstruktion des Geheimnisses die Teilgeheimnisse bekannt zu geben, wodurch das Verfahren ineffizient wird, da jede Geheimnisaufteilung jeweils nur einmal verwendet werden kann. Als Lösung für dieses Problem wurde das Treshold – Signaturverfahren entwickelt, dessen Ziel ein Signaturverfahren ist, bei dem die TN P_i ihr Teilgeheimnis s_i nicht preisgeben müssen.

Verfahren: Combiner und Dealer seien vertrauenswürdige Instanzen, dann...

- Dealer teilt geheimen Schlüssel auf n TN auf.
- Zum Signieren einer Nachricht sind $t < n$ TN nötig, von denen jeder eine Teilsignatur m_i erstellt.
- Combiner fügt die Teilsignaturen zu einer Signatur zusammen.
- Die Signatur der Nachricht m ist dann (B, R, σ) , wobei:
 - B die Gruppe der t Personen, die an der Signatur beteiligt waren, bezeichnet,
 - R dem Produkt über den von den Teilnehmern aus B zufällig für diese eine Signatur gewählten Zahlen entspricht und
 - σ gleich der Summe über die Teilsignaturen m_i ist.

Wichtig dabei ist, dass die TN aus B, die an der Signaturerstellung beteiligt waren, zur Verifizierung der Signatur bekannt sein müssen. Außerdem muss man sich in der Praxis gegen mögliche unehrliche Handlungen vom Dealer und vom Combiner schützen. Die Ehrlichkeit der TN kann durch den Combiner kontrolliert werden, weswegen dort unehrliche Handlungen ausgeschlossen werden können.

5. Oblivious – Transfer

Wir betrachten nun die folgende Situation: A sendet B eine Nachricht m unter der Forderung, dass B m nur zu einer Wahrscheinlichkeit von 0,5 erhalten soll. B akzeptiert die Forderung, wenn A dafür nicht erfährt, welcher der beiden Fälle eingetreten ist.

Protokolle, die diese Eigenschaft erfüllen, bezeichnet man als Oblivious – Transfer (OT), auf deutsch: „unbemerkte Übertragung“.

Eine praktische Variante des OT ist der 1-aus-2-Oblivious-Transfer ($O_{\frac{1}{2}}$), dessen Protokoll folgende Eigenschaften erfüllt:

- A ist im Besitz von zwei Geheimnissen s_0 und s_1 , von denen B am Ende des Protokolls genau eins erhält und über das andere nichts erfährt.
- A weiß nicht, welches Geheimnis B erhalten hat.

Ein anschauliches Beispiel hierzu ist das folgende:

- A kennt zwei Geheimnisse s_0 und s_1
- Öffentlich bekannt ist ein Generator g von \mathbb{Z}_p^* und ein Element c aus \mathbb{Z}_p^* .
- B berechnet $\beta_1 = g^x \bmod p$ und $\beta_0 = c(g^x)^{-1} \bmod p$ wobei x aus \mathbb{Z}_p^* zufällig gewählt, sendet β_0, β_1 an A.
- A prüft $\beta_0, \beta_1: \beta_0 \cdot \beta_1 \stackrel{?}{\equiv} c \bmod p$ und berechnet $\gamma_i = \beta_i^{y_i} \bmod p$, wobei y_i zufällig gewählt, verschlüsselt s_j zu $r_j = s_j \oplus \gamma_j$, berechnet $\alpha_i = g^{y_i} \bmod p$ und sendet $r_0, r_1, \alpha_0, \alpha_1$ an B.
- B entschlüsselt γ_1 mit $\alpha_1: \alpha_1^x = g^{xy_1} = \beta_1^{y_1} = \gamma_1$ dann $s_1 = r_1 \oplus \gamma_1$.
- B kann s_2 nicht entschlüsseln, da er γ_2 nicht berechnen kann.

Auch bei diesem Verfahren verwendet man also als Hilfsmittel Einwegfunktionen.

Man kann:

- aus jedem O_2^1 ein OT konstruieren und
- aus jedem OT ein O_2^1 konstruieren.

OTs gibt es in vielen Varianten, die aber alle zu einander äquivalent sind.

OTs eignen sich gut als Bausteine für andere kryptografische Verfahren und spielen daher eine wichtige Rolle. 1988 konnte sogar gezeigt werden, dass man die gesamte Kryptografie auf der Basis der OT-Protokolle aufbauen kann.

6. Bewertung

Der wohl größte Vorteil an Multiparty – Computations ist, dass es sehr sichere Verfahren gibt, die daher in vielen Bereichen eine wichtige Rolle spielen. Dafür gibt es allerdings kein allgemeines mathematisches Konzept, mit dem sich generell sichere Multiparty – Computation – Protokolle entwerfen lassen, weswegen man von Fall zu Fall entscheiden muss, ob sichere Multiparty – Computation überhaupt möglich ist und wie sie erreicht werden kann. Dies hat natürlich einigen Mehraufwand zur Folge, weswegen häufig auch auf andere Verfahren zurückgegriffen wird, wenn eigentlich auch Multiparty – Computation möglich wäre. Insgesamt stellt die Multiparty – Computation aber dennoch einen wichtigen Bereich der Kryptografie dar, der gerade auch in der heutigen Zeit, z.B. bei Online-Auktionen, aber auch in vielen anderen Gebieten zur Geltung kommt.

7. Literatur

- Beutelspacher, Neumann, Schwarzpaul: Kryptografie in Theorie und Praxis, Vieweg 2005.
- Buchmann: Einführung in die Kryptographie, 2.Auflage, Springer, 2001
- Christian Schmid: Kompakte Quelle verschränkter Photonen und Anwendungen in der Quantenkommunikation, Diplomarbeit an der Fakultät für Physik, Ludwig-Maximilians-Universität München, Januar 2004

8. Inhaltsverzeichnis

1. Was bedeutet Multiparty – Computations?	2
2. Zugrunde liegendes Modell	3
3. Secret – Sharing – Verfahren	6
3.1. Schwellenschemata	
3.2. Verifizierbare Geheimnisaufteilung	
4. Treshold – Signaturverfahren	9
5. Oblivious Transfer	10
6. Bewertung	11
7. Literatur	12
8. Inhaltsverzeichnis	12