Oblivious Transfer Amplification – Verstärkung vergesslicher Übertragung

Jürg Wullschleger

Department of Mathematics
University of Bristol, Grossbritannien
j.wullschleger@bris.ac.uk

Abstract: Sichere Mehrparteienberechnung erlaubt mehreren Spielern eine Berechnung auf sichere Art durchzuführen, das heisst, ohne dass irgendein Spieler zusätzliche Information über die Eingaben oder Ausgaben der anderen Spieler erhält. Da gezeigt wurde, dass mit Hilfe von *Oblivious Transfer (OT, vergessliche Übertragung)* jede Mehrparteienberechnung beweisbar auf sichere Art ausgeführt werden kann, kommt OT eine zentrale Rolle in Mehrparteienberechnung zu. In dieser Arbeit untersuchen wir, woraus OT hergestellt werden kann und vereinfachen, korrigieren, beziehungsweise verstärken bisheriger Protokolle.

Zuerst zeigen wir, dass es ein *sehr* einfaches Protokoll gibt, welches OT *verlustfrei* umdreht. Dieses Protokoll ist nicht nur viel effizienter als bisherige Protokolle, es gibt uns auch eine neue, vorher unbekannte Sicht auf OT: OT ist aus kryptographischer Sicht *symmetrisch*.

Schliesslich zeigen wir Protokolle zur *Verstärkung* von OT, das heisst zur Herstellung von OT aus verschiedenen, schwächeren Formen von OT, bei denen unehrliche Spieler Zusatzinformationen über die fremden Eingaben erhalten können.

1 Einführung

Am 16. Januar 1797 sandte Johann Wolfgang von Goethe (1749-1832) einen Brief an seinen Verleger Vieweg mit folgendem Inhalt [Jen84, Seite 651]:

> "Ich bin geneigt Herrn Vieweg [...] 'Herrmann und Dorothea' [...] zum Verlag zu überlassen. [...] Was das Honorar betrifft so stelle ich Herrn Ober-Consistorialrath Böttiger ein versiegeltes Billet zu, worinn meine Forderung enthalten ist und erwarte was Herr Vieweg mir für meine Arbeit anbieten zu können glaubt. Ist sein Anerbieten geringer als meine Forderung, so nehme ich meinen versiegelten Zettel ungeöffnet zurück [...], ist es höher, so verlange ich nicht mehr als in dem [...] Zettel verzeichnet ist."

Goethe wählte dieses komplizierte Verfahren nicht um seinen Profit zu maximieren. Bei einem höheren Angebot des Verlegers hätte er ja trotzdem nur den von ihm festgelegten Betrag verlangt. Sein Ziel war herauszufinden, wieviel Geld Vieweg bereit gewesen wäre zu zahlen, vermutlich um den Preis zukünftiger Arbeiten entsprechend anzupassen.

Um dies zu erreichen wandte Goethe ein Verfahren an, in welchem er seinen Preis zu einem bestimmten Zeitpunkt auf einen fixen Betrag festlegte, Vieweg hingegen sollte den Preis erst später erfahren, nachdem er sein Gebot abgegeben hatte. Heute wird dieses Verfahren *Bit-Commitment* genannt und bildet einen wichtigen Bestandteil sogenannter *sicherer Zwei- und Mehrparteienberechnung (Two- and Multiparty Computation)*, einem von Yao [Yao82] begründeten Zweig moderner Kryptographie. Goethe versuchte sein Bit-Commitment mit Hilfe einer dritten Person, Herrn Böttiger, zu erreichen. Leider stellte sich aber Herrn Böttiger als nicht so vertrauenswürdig wie von Goethe erhofft heraus: Böttiger gab Vieweg einen Tipp über den Inhalt des Briefes, und Vieweg bot schliesslich genau den von Goethe verlangten Betrag. Dieses Beispiel zeigt, dass es oft vorzuziehen ist, *nicht* einer dritten Person vertrauen zu müssen. In der Kryptographie ist genau dies unser Ziel: Wir versuchen, sichere Zwei- und Mehrparteienberechnung *ohne* die Hilfe einer dritten Partei herzustellen, der alle Parteien vertrauen müssen.

Sichere Mehrparteienberechnung findet in vielen Bereichen Anwendung, wenn auch manchmal auf versteckte Weise. Bei Wahlen werden die klassischen Systeme mit Stimmzetteln und Wahlurnen immer mehr durch Wahlcomputer ersetzt. Dies vereinfacht zwar den Wahlprozess, macht aber die Wahl auch erheblich anfälliger für Betrug. Sichere Mehrparteienberechnung kann helfen, dies zu verhindern, ohne die Privatsphäre der Bürger zu verletzen.

Leider ist es bis heute unmöglich sichere Zwei- und Mehrparteienberechnung ohne zusätzliche Annahmen auszuführen. Eine zentrale Frage ist deshalb, welche (minimalen) zusätzlichen Annahmen benötigt werden, um dies tun zu können. Man unterscheidet zwischen zwei Typen zusätzlicher Annahmen. Auf der einen Seite sind dies Annahmen über die Schwierigkeit gewisser Berechnungsprobleme (z.B. Faktorisieren), die jedoch nicht bewiesen sind. Auf der anderen Seite sind dies zusätzliche Funktionalitäten. Ein Beispiel solch einer zusätzlichen Funktionalität ist *Oblivious Transfer*.

1.1 Oblivious Transfer – Vergessliche Übertragung

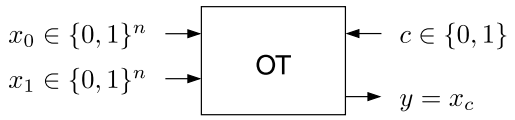

Abbildung 1: Oblivious Transfer (vergessliche Übertragung).

Kilian hat in [Kil88] gezeigt, dass *jede* Zweiparteienberechnung sicher gegen berechenmässig unbeschränkte Gegner ausgeführt werden kann, unter der Bedingung, dass die Spieler über eine sehr einfache zusätzliche Funktionalität mit dem Namen *Oblivious Transfer (OT, vergessliche Übertragung)* [Wie83, Rab81, EGL85] verfügen.

OT kann man als eine abstrakte Box (eine Blackbox) mit einem bestimmten Verhalten betrachten, die von zwei Spielern, nennen wir sie Heidi und Peter, benutzt werden kann (siehe Abbildung 1). Zuerst schickt Heidi der Box zwei Nachrichten x_0 und x_1, und Peter

ein Auswahlbit c, mit welchem er festlegt, welche der beiden Nachrichten er empfangen möchte. Nach Erhalt beider Eingaben sendet die Box Peter die von ihm gewünschte Nachricht, *aber nichts weiteres*. Peter erfährt nicht, was in der anderen Nachricht steht und Heidi erfährt nicht, welche Nachricht Peter ausgewählt hat.

Obwohl Oblivious Transfer etwas sehr einfaches ist, ist es doch sehr schwierig zu implementieren. Zum Beispiel ist es sehr viel schwieriger zu realisieren als ein Bit-Commitment. Während ein Bit-Commitment aus *Einwegfunktionen*[1] hergestellt werden kann [Nao91], so ist dies für OT vermultich nicht möglich. In [IR89] wurde gezeigt, dass es keine Implementierung geben kann, die Einwegfunktionen als Blackbox verwendet. Implementierungen von OT benötigen zum Beispiel die sehr viel stärkere Annahme von *Trapdoorpermutationen* [EGL85].

OT aus schwächeren Annahmen herzustellen, entweder aus schwachen Funktionalitäten, wie *unfairen, verrauschten Kanälen* [DKS99, DFMS04], oder aus schwächeren Annahmen über die Schwierigkeit gewisser Probleme [Hai04] kann recht mühsam sein. Man kann oft zuerst nur eine schwache Form von OT herstellen. In einem zweiten Schritt wird dieser *schwache OT* dann zu einem OT verstärkt. In der hier zusammengefassten Arbeit [Wul07] untersuchen wir diesen zweiten Schritt, aus welchen schwächeren Formen von OT ein sicherer OT konstruiert werden kann.

2 Was heisst ”sicher”?

In der Einführung haben wir die Sicherheit von OT-Protokollen bereits angesprochen. Der Sender darf die Eingabe des Empfängers nicht erhalten und der Empfänger darf nur eine einzige Meldung des Senders empfangen. Leider sind diese Bedingungen aber nur schwer in eine formale Definition zu fassen, es gibt viele Details zu beachten. Die so entstandene Definition ist dann recht unbefriedigend, denn man weiss nicht, ob nicht doch eines der Details vergessen wurde. Hinzu kommt, dass nicht klar ist, ob ein OT-Protokoll, welches diese Definition erfüllt, immer noch sicher ist, wenn es in ein anderes Protokoll integriert wird.

Micali und Rogaway [MR92], und Beaver [Bea92] haben einen natürlichen Weg gefunden, die Sicherheit von Protokollen zu definieren. Die Idee hinter dieser Definition ist sehr einfach: Wir vergleichen das OT-Protokoll mit einem *idealen Protokoll*, in welchem OT als Blackbox zur Verfügung steht und das per Definition sicher ist. Falls es nun für jede Attacke auf das OT-Protokoll eine Attacke auf das ideale Protokoll gibt, die dasselbe erreicht, dann ist das OT-Protokoll mindestens so sicher wie das ideale Protokoll, und deshalb ebenfalls sicher.
Die Attacke im idealen Protokoll wird auch Simulation genannt, denn sie simuliert die reale Attacke in der idealen Welt. Ein wichtiger Vorteil dieser simulationsbasierten Definition ist nun, dass sichere Protokolle *zusammengesteckt* werden können. Das heisst, man kann in einem Protokoll, welches OT als Blackbox verwendet und sicher ist, die OTs durch si-

[1] Eine Einwegfunktion kann einfach berechnet werden, die Berechnung des Inversen zu einem bekannten Funktionswert ist jedoch schwierig.

chere OT-Protokolle ersetzen, ohne die Sicherheit des Protokolles zu zerstören. Dies ist eine sehr wichtige Eigenschaft, denn sie erlaubt uns, grosse Protokolle in kleine Teile zu zerlegen, und diese einzeln zu beweisen.

Es gibt viele verschiedene Varianten von Sicherheitsdefinitionen, je nach dem wie stark der Gegner sein kann. Man unterscheidet die Gegner grundsätzlich in drei Punkten.

1. Es gibt sogenannte passive oder aktive Gegner.
2. Der Gegner ist berechenmässig unbeschränkte oder in seiner Rechenzeit polynomiell beschränkt.
3. Der Gegner kann beliebig oder beschränkt mit seiner Umgebung interagieren.

Ein passiver Gegner folgt dem vorgegebenen Protokoll, er versucht jedoch möglichst viel zu erfahren. Das heisst, er behält alle Informationen, die er während des Protokolls zusätzlich erfährt. Ein aktiver Gegner folgt hingegen nicht unbedingt dem Protokoll, sondern führt seine eigene Strategie aus.

Als Umgebung bezeichnen wir alles ausserhalb des Protokolls, z.B. andere Protokolle, die von den selben Spielern ausgeführt werden, oder auch solche anderer Spieler. Die Interaktion mit der Umgebung hat direkte Auswirkungen auf die Art und Weise, wie Protokolle zusammengesteckt werden können. Wir erlauben hier, dass der Gegner beliebig mit der Umgebung interagieren kann. Daraus folgt, dass Protokolle beliebig (zum Beispiel auch parallel) zusammengesteckt werden können. Man nennt dies *universal composability* [PW01, Can01].

Bei Sicherheitsbetrachtungen aus informationstheoretischer Sicht (das heisst, wenn der Gegner berechenmässig unbeschränkt ist) wird oft auf eine simulationsbasierte Definition der Sicherheit verzichtet, und stattdessen eine Liste von Bedingungen verwendet. Meist weil dies den Beweis, dass ein Protokoll sicher ist, vereinfacht. Es ist jedoch, wie oben schon erwähnt, nicht klar, ob diese Liste von Bedingungen ausreichend ist. Ausserdem ist nicht klar, wie man solche Protokolle zusammenstecken kann.

In [CSSW06] zeigten wir, dass man eine solche Listendefinition aus einer Simulationsdefinition herleiten kann. Es stellte sich tatsächlich heraus, dass fast alle bisherigen Listendefinitionen lückenhaft gewesen waren. Leider funktionierte unser Beweis aber nur für den fehlerfreien Fall, und gegen aktive Gegner. Dies bedeutet, dass unser Resultat oft nicht angewendet werden kann, da die meisten Protokolle einen kleinen Fehler machen oder nur gegen passive Gegner sicher sind. In dieser Arbeit beheben wir nun diese beiden Einschränkungen des Resultats von [CSSW06] für *randomisierten OT* (ROT). Dies ist eine Variation von OT, in der das Protokoll die Eingaben der ehrlichen Spieler zufällig wählt. Die Beschränkung auf ROT ist ausreichend, denn Beaver [Bea95] hat gezeigt dass sich OT sehr einfach aus ROT herstellen lässt, und umgekehrt. Diese Listendefinition für ROT war ausserdem sehr hilfreich für die Definition von *schwachem OT*, auf welchen wir in Abschnitt 4.2 genauer eingehen werden.

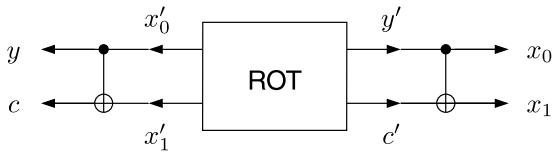

Abbildung 2: Protokoll zum Umdrehen von ROT.

3 OT ist symmetrisch

Wenn OT als Blackbox gegeben ist, dann ist festgelegt, welcher der beiden Spieler Sender und welcher Empfänger ist. Es kann nun aber auch sein, dass die beiden Spieler einen OT in die Gegenrichtung ausführen wollen. Sie wollen die Seiten tauschen und die Nachricht in die andere Richtung schicken.
Crépeau und Sántha [CS91], sowie Ostrovsky, Venkatesan und Yung [OVY93] haben gezeigt, dass man OT[2] in Gegenrichtung aus 2 OTs herstellen kann, sofern der Gegner nur passiv ist. Gegen aktive Gegner wird die Konstruktion wesentlich komplizierter, und benötigt $\Theta(k)$ OTs, um einen OT in Gegenrichtung mit Fehlerwahrscheinlichkeit von 2^{-k} herzustellen.
In dieser Arbeit zeigen wir, dass dies *sehr* viel einfacher geht. Sowohl bei passiven als auch bei aktiven Gegnern reicht ein einziger OT aus, um einen OT in Gegenrichtung herzustellen. Abbildung 2 zeigt das Protokoll, welches einen ROT in Gegenrichtung aus einem ROT herstellt, unter Benützung von nur 2 exklusiven ODER Operationen.

Dass dieses Protokoll korrekt ist, d.h. $y = x_c$, ist sehr einfach zu sehen. (Wie oben erwähnt, machen die beiden Spieler in ROT keine Eingaben, sondern die Funktion ROT setzt diese Werte zufällig. Dies wird in Abbildung 2 dadurch dargestellt, dass die Pfeile der Eingaben aus dem ROT herausgehen.) Aus der Definition von ROT folgt $y' := x'_{c'}$ oder

$$y' = x'_0 \oplus (x'_0 \oplus x'_1) \cdot c' \ . \tag{1}$$

Weiterhin gilt $y = x'_0$, $c = x'_0 \oplus x'_1$, $x_0 = y'$, und $x_1 = y' \oplus c'$. Daraus folgt $c' = x_0 \oplus x_1$. Durch einfaches Ersetzen dieser Ergebnisse in die Gleichung (1) folgt somit

$$x_0 = y \oplus c \cdot (x_0 \oplus x_1)$$

oder $y = x_c$. Die Sicherheit dieses Protokolls kann ebenfalls sehr einfach gezeigt werden.

Zusammen mit dem Protoll von Beaver [Bea95] kann man deshalb sehr einfach aus einem OT ein OT in Gegenrichtung herstellen, welcher sowohl gegen passive als auch gegen aktive Gegner sicher ist.

Da somit aus einem OT sehr einfach und *verlustfrei* ein OT in Gegenrichtung erzeugt werden kann, kann man deshalb auch sagen, dass OT aus kryptographischer Sicht *symmetrisch* ist.

[2]Wir nehmen in diesem Abschnitt an, dass alle OTs Länge 1 haben, das heisst $x_0, x_1 \in \{0, 1\}$

4 OT aus schwachen Formen von OT

Wir kommen nun zu den verschiedenen schwachen Formen von OT, die sich zu OT verstärken lassen.

4.1 OT aus universellem OT

Der sogenannte *Universelle OT*, der von Cachin [Cac98] eingeführt wurde, ist eine Form von OT, die nur eine schwache Form von Sicherheit bietet, falls der Empfänger ein Angreifer ist. Der Empfänger kann nicht nur eine der beiden Eingaben des Senders empfangen, sondern beliebige Informationen über beide Eingaben erhalten. Die einzige Bedingung ist, dass der Empfänger nach dem Protokoll immer noch eine gewisse Unsicherheit über die beiden Eingaben hat, wenn diese zufällig gewählt wurden. Die Unsicherheit wird *Min-Entropie* gemessen. Wie in [Cac98] vorgeschlagen kann die Sicherheit des Senders nun durch das Anwenden von zwei zufälligen Hashfunktionen auf die beiden Eingaben des Senders und auf die Ausgabe des Empfängers erhöht werden. Leider war aber der Sicherheitsbeweis in [Cac98] fehlerhaft, und der Sicherheitsbeweis in [DFSS06] zwar korrekt, aber ziemlich kompliziert und nur noch halb so effizient. In dieser Arbeit korrigieren wir nun den Fehler in [Cac98] ohne Einbussen an Effizienz.

Die Beweise in [Cac98] und in [DFSS06] benutzen das sogenannte *Leftover-Hash-Lemma* [BBR88, ILL89], welches auch unter dem Namen *Privacy Amplification* bekannt ist und in der Kryptographie sehr oft verwendet wird. Das Lemma sagt uns, wie viele fast perfekt zufällige Bits wir aus einer nicht gleichverteilten Zufallsvariable mit Hilfe einer Hashfunktion extrahieren können. Das Kernstück unseres Sicherheitsbeweises ist eine Verallgemeinerung dieses Lemmas, welches wir *verteiltes Leftover-Hash-Lemma* nennen. Es sagt aus, wie viele zufällige Bits aus zwei korrelierten Zufallsvariablen extrahiert werden können, wenn aus jeder der Zufallsvariablen unabhängig extrahiert wird.

4.2 OT aus schwachem OT

Schwacher OT (Weak OT, WOT) ist eine Variante von OT, in der entweder ein unehrlicher Sender oder auch ein unehrlicher Empfänger zu einem gewissen Grad zusätzliche Informationen erhalten können. Es ist aber auch möglich, dass die Ausgabe des Empfängers fehlerhaft ist, obwohl beide Parteien ehrlich sind. Da wir nun 3 verschiedene Arten von Fehlern haben, ist es viel schwieriger diese Form von OT zu verstärken. Wir beschränken uns deshalb darauf, dass die Eingaben des Senders nur einzelne Bits sind. WOT wird durch drei Parameter charakterisiert: p ist die Wahrscheinlichkeit, dass der Sender das Auswahlbit des Empfängers erfährt, q ist die Wahrscheinlichkeit, dass der Empfänger beide Eingabebits erfährt, und ε ist die Wahrscheinlichkeit dass die Ausgabe des Empfängers fehlerhaft ist. Dieses Model wurde bereits in [DKS99, DFMS04] untersucht. Es wurde gezeigt, dass falls die Fehler nicht allzu gross sind, aus polynomielle vielen WOT ein OT

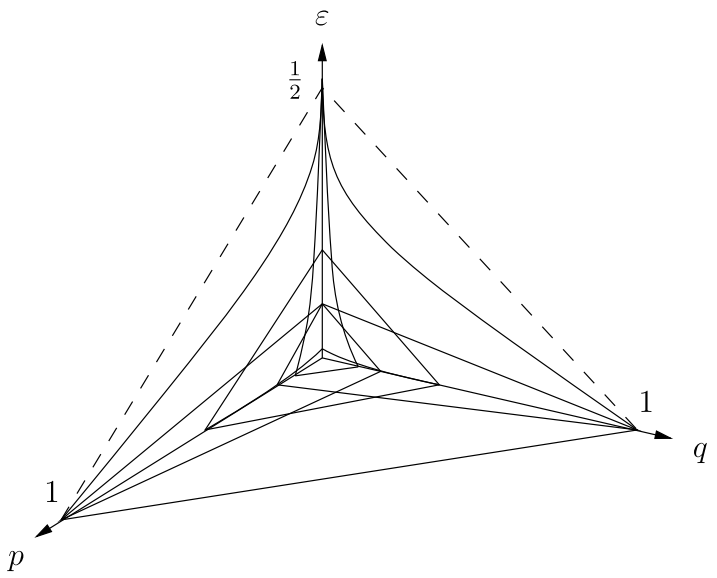

Abbildung 3: Region, für welche WOT zu OT verstärkt werden kann.

hergestellt werden kann, in dem alle drei Fehler exponentiell klein sind. Dies wird dadurch erreicht, dass man abwechslungsweise drei verschiedene Protokolle anwendet, welche aus mehreren WOT einen neuen WOT herstellen, indem einer der drei Fehler verkleinert wird, jedoch die anderen zwei vergrössert werden. Durch geschickte Kombination dieser Portokolle können alle drei Fehler gleichzeitig verkleinert werden. Leider hatten diese Resultate zwei Probleme. Erstens war die Definition von WOT in [DKS99] sehr ungenau, und in [DFMS04] zwar genau, jedoch viel zu stark, und deshalb für alle heute bekannten Anwendungen unbrauchbar. Zweitens war das Protokoll zur Reduktion von ε nur kompatibel mit dieser zu starken Definition von WOT.

In dieser Arbeit geben wir nun zuerst zwei neue, formale Definitionen von WOT, eine für den Fall von passiven Gegnern und eine für den Fall von aktiven Gegnern. Wir geben ein neues Protokoll zur Reduktion von ε an und zeigen, für welche Parameter von p, q und ε es möglich ist, alle Fehler gleichzeitig zu reduzieren. Wir erweitern den Bereich, für den dies möglich ist und geben eine explizite obere Schranke an, wie viele WOT zur Verstärkung benötigt werden. Wie in [DKS99, DFMS04] sind diese neuen Protokolle sicher gegen passive als auch aktive Gegner falls $\varepsilon = 0$, aber nur sicher gegen passive Gegner falls $\varepsilon > 0$. Hier zeigte sich nun ein erstes Mal, wie nützlich die Symmetrie von OT ist: Da WOT ebenfalls symmetrisch ist, brauchten wir die Sicherheit eines der drei Protokolle gar nicht zu beweisen, sie folgte direkt aus der Symmetrie von WOT.

Abbildung 3 zeigt schematisch die verschiedenen Bereiche, für die WOT verstärkt werden kann. Die gestrichelte Linie markiert die Schranke $p + q + 2\varepsilon = 1$, für welche WOT simuliert, das heisst durch Kommunikation hergestellt werden kann. Es ist unmöglich,

solche WOT zu verstärken.

4.3 OT aus berechenmässig schwachem OT

Alle bisherigen Protokolle funktionieren sowohl gegen berechenmässig beschränkte, als auch gegen berechenmässig unbeschränkte Gegner. Für den Fall von berechenmässig beschränkten Gegnern können wir aber eine neue, viel schwächere Definition von WOT geben, welche wir "computational WOT" oder *compWOT* nennen. Wir verlangen nun nicht mehr, dass *jeder* (das heisst auch ein unbeschränkter) Gegner die Eingabe des anderen Spielers nur mit einer gewissen Wahrscheinlichkeit raten kann, sondern nur, dass jeder *polynomiell beschränkte* Gegner dies nicht tun kann. Dies ist sehr wichtig, denn nur so können wir einen WOT herstellen der auf schwächeren berechenmässigen Annahmen beruht, wie zum Beispiel in [Hai04].

Erstaunlicherweise kann man für die Verstärkung von compWOT genau das gleiche Protokoll verwenden wie für WOT. Dies ist jedoch sehr viel schwieriger zu beweisen: Nun muss man zeigen, dass falls es einen polynomiell beschränkten Gegner gibt, welcher beim resultierenden OT die Eingabe des anderen Spielers mit einer etwas besseren Wahrscheinlichkeit als $\frac{1}{2}$ raten kann, so können wir für den darunterliegenden compWOT einen neuen Gegner herstellen, der auch polynomiell beschränkt ist, und welcher diesen compWOT mit einer höheren Wahrscheinlichkeit raten kann als angenommen.

Anders als in [Hai04] beweisen wir aber nicht nur einen Spezialfall, sondern, wie oben erwähnt, zeigen wir, dass *jedes* Protokoll, welches aus vielen Instanzen von WOT ein sicheren OT herstellt, ebenfalls ein (gegen berechenmässig beschränkte Gegner) sicheren OT herstellt, wenn alle WOT durch compWOT ersetzt werden. Unser Beweis benützt das *uniforme Hardcorelemma* von Holenstein [Hol05, Hol06], welches auf dem Hardcorelemma von Imagliazzo [Imp95] beruht.

Wie in [Hai04] beschränken auch wir uns auf den passiven Fall. Mit Hilfe des Kompliers von [GMW87] kann jedoch auch hier ein OT hergestellt werden, welcher gegen aktive Gegner sicher ist.

Literatur

[BBR88] C. H. Bennett, G. Brassard und J.-M. Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988.

[Bea92] D. Beaver. Foundations of Secure Interactive Computing. In *Advances in Cryptology — CRYPTO '91*, Jgg. 1233 of *LNCS*, Seiten 377–391. Springer-Verlag, 1992.

[Bea95] D. Beaver. Precomputing Oblivious Transfer. In *Advances in Cryptology — EUROCRYPT '95*, Jgg. 963 of *LNCS*, Seiten 97–109. Springer-Verlag, 1995.

[Cac98] C. Cachin. On the Foundations of Oblivious Transfer. In *Advances in Cryptology — EUROCRYPT '98*, Jgg. 1403 of *LNCS*, Seiten 361–374. Springer-Verlag, 1998.

[Can01] R. Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. In *Proceedings of the 42th Annual IEEE Symposium on Foundations of Computer Science (FOCS '01)*, Seiten 136–145, 2001. Aktuelle Version erhältlich unter http://eprint.iacr.org/2000/067.

[CS91] C. Crépeau und M. Sántha. On the Reversibility of Oblivious Transfer. In *Advances in Cryptology — EUROCRYPT '91*, Jgg. 547 of *LNCS*, Seiten 106–113. Springer, 1991.

[CSSW06] C. Crépeau, G. Savvides, C. Schaffner und J. Wullschleger. Information-Theoretic Conditions for Two-Party Secure Function Evaluation. In *Advances in Cryptology — EUROCRYPT '06*, Jgg. 4004 of *LNCS*, Seiten 538–554. Springer-Verlag, 2006.

[DFMS04] I. Damgård, S. Fehr, K. Morozov und L. Salvail. Unfair Noisy Channels and Oblivious Transfer. In *Theory of Cryptography Conference — TCC '04*, Jgg. 2951 of *LNCS*, Seiten 355–373. Springer-Verlag, 2004.

[DFSS06] I. Damgård, S. Fehr, L. Salvail und C. Schaffner. Oblivious Transfer and Linear Functions. In *Advances in Cryptology — CRYPTO '06*, Jgg. 4117 of *LNCS*. Springer-Verlag, 2006.

[DKS99] I. Damgård, J. Kilian und L. Salvail. On the (Im)possibility of Basing Oblivious Transfer and Bit Commitment on Weakened Security Assumptions. In *Advances in Cryptology — EUROCRYPT '99*, Jgg. 1592 of *LNCS*, Seiten 56–73. Springer-Verlag, 1999.

[EGL85] S. Even, O. Goldreich und A. Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, 1985.

[GMW87] O. Goldreich, S. Micali und A. Wigderson. How to Play any Mental Game. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC '87)*, Seiten 218–229. ACM Press, 1987.

[Hai04] I. Haitner. Implementing Oblivious Transfer Using Collection of Dense Trapdoor Permutations. In *Theory of Cryptography Conference — TCC '04*, Jgg. 2951 of *LNCS*, Seiten 394–409. Springer-Verlag, 2004.

[Hol05] T. Holenstein. Key Agreement from Weak Bit Agreement. In *Proceedings of the 37th ACM Symposium on Theory of Computing (STOC '05)*, Seiten 664–673. ACM Press, 2005.

[Hol06] T. Holenstein. *Strengthening key agreement using hard-core sets*. Dissertation, ETH Zurich, Switzerland, 2006. Reprint as vol. 7 of *ETH Series in Information Security and Cryptography*, Hartung-Gorre Verlag.

[ILL89] R. Impagliazzo, L. A. Levin und M. Luby. Pseudo-Random Generation from One-Way Functions. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC '89)*, Seiten 12–24. ACM Press, 1989.

[Imp95] R. Impagliazzo. Hard-core distributions for somewhat hard problems. In *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science (FOCS '95)*, Seiten 538–545. IEEE Computer Society, 1995.

[IR89] R. Impagliazzo und S. Rudich. Limits on The Provable Consequences of One-Way Permutations. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC '89)*, Seiten 186–208. ACM Press, 1989.

[Jen84] I. Jensen. *Quellen und Zeugnisse zur Druckgeschichte von Goethes Werken*. Akademie-Verlag Berlin, 1984.

[Kil88] J. Kilian. Founding Cryptography on Oblivious Transfer. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC '88)*, Seiten 20–31. ACM Press, 1988.

[MR92] S. Micali und P. Rogaway. Secure Computation (Abstract). In *Advances in Cryptology — CRYPTO '91*, Jgg. 576 of *LNCS*, Seiten 392–404. Springer-Verlag, 1992.

[Nao91] M. Naor. Bit Commitment Using Pseudorandomness. *Journal of Cryptology: the journal of the International Association for Cryptologic Research*, 4(2):151–158, 1991.

[OVY93] R. Ostrovsky, R. Venkatesan und M. Yung. Fair Games Against an All-Powerful Adversary. In *Advances in Computational Complexity Theory*, Jgg. 13 of *AMS DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Seiten 155–169. AMS, 1993.

[PW01] B. Pfitzmann und M. Waidner. A Model for Asynchronous Reactive Systems and its Application to Secure Message Transmission. In *Proceedings of the 2001 IEEE Symposium on Security and Privacy (SP '01)*, Seite 184, 2001. Auch erhältlich unter http://eprint.iacr.org/2000/066.

[Rab81] M. O. Rabin. How to exchange secrets by oblivious transfer. Bericht TR-81, Harvard Aiken Computation Laboratory, 1981.

[Wie83] S. Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, 1983.

[Wul07] J. Wullschleger. *Oblivious-Transfer Amplification*. Dissertation, ETH Zurich, Switzerland, 2007. Auch erhältlich unter http://arxiv.org/abs/cs.CR/0608076.

[Yao82] A. C. Yao. Protocols for Secure Computations. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science (FOCS '82)*, Seiten 160–164, 1982.

Jürg Wullschleger wurde am 5. Juli 1975 in Zug (Schweiz) geboren. Nach Abschluss der Mittelschule am Kantonalen Kollegium in Schwyz begann er im Jahre 1995 ein Informatikstudium an der Eidgenössischen Technischen Hochschule in Zürich (Schweiz), welches er im Jahre 2001 mit einem Diplom abschloss. 2007 promovierte er mit der hier zusammengefassten Arbeit, welche unter Aufsicht von Prof. Stefan Wolf teils an der Université de Montréal (Kanada), und teils an der ETH Zürich entstand. Seit 2007 ist er Forschungsmitarbeiter am Department of Mathematics an der University of Bristol (Grossbritannien). Seine derzeitige wissenschaftliche Arbeit befasst sich mit Kryptographie und Quanteninformationstheorie.