Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

SybilGuard: Defending Against Sybil Attacks via Social Networks

Seminar
„P2P-Netzwerke“
im Wintersemester 2006

Eingereicht von: **Christian Kretschmer**
kretschm@informatik.uni-freiburg.de
Matrikelnummer: 1335273

Eingereicht am: 07. März 2007

Betreuer: Arne Vater

Vorwort

Sybil-Attacken sind heutzutage keine Seltenheit mehr und stellen insbesondere in Peer-to-Peer Netzwerken, aufgrund der fehlenden zentralen Instanz (Server), eine ernst zu nehmende Bedrohung dar. Hier muss nun jeder teilnehmende Peer seinen Teil dazu beitragen diese Angriffe abzuwehren.

Bisherige Strategien versuchten durch Berechnungen auf den Peers die Tragweite zu beschränken. So mussten Puzzel oder Rechenaufgaben gelöst werden, um weiter im Netz bleiben zu dürfen in der Hoffnung, dass Angreifer solche Aufgaben nicht für alle ihre Sybil-Identitäten schnell genung lösen können. Doch wo setzt man die Grenze? Geht man davon aus, dass ehrliche Peers mit Pentium-Rechner auch teilnehmen dürfen, so hat ein Sybil-Angreifer mit einem Rechenzentrum unter seiner Kontrolle genug Rechenleistung, um das Netz ernsthaft zu gefährden.

Genau hier setzt nun *SybilGuard* an und stellt in der durch die dezentralisierte Struktur des Netzes erschwerten Umgebung ein Protokoll bereit, welches die Tragweite solcher Angriffe begrenzt.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

In dieser Arbeit werden zunächst einige grundlegenden Begriffe, sowie das Netzwerk (Kapitel 1), auf welches *SybilGuard* basiert, näher gebracht, um anschließend die Funktionsweise des neuen Protokolls in einer statischen (Kapitel 2) und später in der praxisnahen dynamischen Umgebung (Kaptiel 4) zu erläutern.
Kapitel 3 wird die Garantien von *SybilGuard* kurz erläutern und Kapitel 5 schließt mit der Evaluation der von *SybilGuard* zugesicherten Eigenschaften sowohl ohne als auch mit Sybil-Angreifern die Ausarbeitung ab.

2 Grundlegende Begriffe

Um die Ausführungen zu *SybilGuard* besser verstehen zu können bedarf es der Klärung einiger grundlegender Begriffe, die im folgenden näher beschrieben werden.

2.1 Sybil-Attacke

Unter einer *Sybil-Attacke* versteht man einen Angriff auf ein P2P-Netzwerk indem ein Angreifer viele Identitäten (Knoten) innerhalb des Netzes erstellt und so versucht Einfluss auf das Netz auszuüben, entweder um es für seine eigenen Absichten zu nützen oder um es unbrauchbar zu machen. Schafft er es den Großteil der Knoten zu kompromitieren, so ist das Netzwerk unter seiner Kontrolle.

Der Name *"Sybil"* leitet sich vom gleichnamigen Buch von Flora Rheta Schreiber aus dem Jahre 1973 ab, welches auf der Geschichte einer Frau mit multipler Persönlichkeitsstörung basiert.

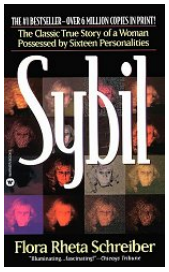

Abb. 1: *Das Buch "Sybil" von Flora Rheta Schreiber aus [1]*

2.2 Das soziale Netzwerk

Alle Knoten eines Netzes formen ein *soziales Netzwerk*. Eine ungerichtete Kante existiert zwischen den Knoten, falls sie eine starke soziale Verbindung haben (z.B. Verwandte, Kollegen). So verbundene Knoten werden als Freunde bezeichnet.

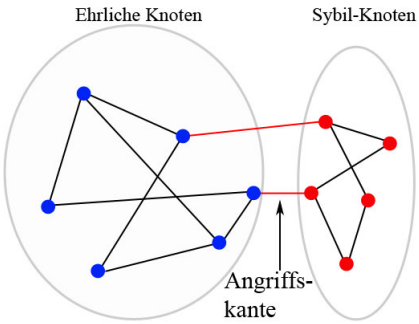

Abb. 2: *Das soziale Netzwerk mit Sybil-Knoten nach [2]*

Um das Netz durch einen Sybil-Angriff bedrohen zu können, versuchen die Sybil-Knoten eben solche Verbindungen zu den ehrlichen Knoten aufzubauen. Falls dies gelingt entstehen die so genannten *Angriffskanten* (g).

3 SybilGuard

Da das beschriebene *F2F-Netzwerk* einen sehr geringen Grad der Knoten und somit nur geringen praktischen Nutzen hat, stellt *SybilGuard* ein Protokoll bereit, welches den ehrlichen Knoten erlaubt eine Vielzahl anderer Knoten ebenso zu akzeptieren. Dadurch werden aber keine neuen Kanten aufgebaut oder gelöscht! Die dazu nötigen Mechanismen werden in diesem Kapitel näher erläutert.

3.1 Soziales Netzwerk von SybilGuard

Als erstes erweitert *SybilGuard* das in Abschnitt 2.2 beschriebene Netzwerk um Schlüssel an jeder Kante. Diesen eindeutigen symmetrischen *Kantenschlüssel* teilen sich die beiden Freunde, um ihre Mitteilungen zu authentifizieren. Da nur die beiden Freunde diesen Schlüssel kennen müssen, erfolgt ein Austausch außerhalb des Netzes beispielsweise per Telefon.
Desweiteren sollen sich die Freunde über etwaige IP-Änderungen informieren. Die IP-Adressen dienen hierbei lediglich als Hinweis. Wie wir später noch sehen werden können diese Aktualisierungen nachlässig erfolgen. Steht DNS zur Verfügung so wird auch der DNS-Name genutzt.

3.2 Zufallsrouten

Die *Zufallsrouten* stellen den wichtigsten Mechanismus von *SybilGuard* dar. Mit ihnen kann ein Knoten, im folgenden Verifier genannt, entscheiden, ob ein anderer Knoten (Subject) sein Vertrauen verdient. Überschneiden sich nämlich die *Zufallsrouten* des Subject mit der des Verifiers, so wird das Subject akzeptiert und ihm zukünftig das Vertrauen geschenkt.

Im Unterschied zu *Random Walks*, welche bei anderen P2P-Netzwerken wie z.B. *Gnutella* eingesetzt werden um die Suche zu verbessern, sind hier die Wege durch einen Knoten genau festgelegt. Es wird also einer *Eingangskante* genau eine *Ausgangskante* zugeordnet und nicht wie bei den *Random Walks* jedes Mal von neuem entschieden welche Ausgangskante gewählt werden soll.

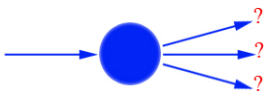

Abb. 3: *Freie Wegewahl der Random Walks nach [2]*

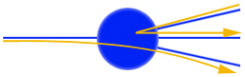

Abb. 4: *Festgelegte Wege der Zufallsrouten nach [2]*

3.2.1 Eigenschaften von Zufallsrouten Durch die Festlegung der Wege ergeben sich zwei wichtige Eigenschaften.

Zum einen die *Konvergenzeigenschaft*, die besagt, dass zwei Zufallsrouten, welche über die selbe Eingangskante in einen Knoten kommen, auf der gleichen Ausgangskante den Knoten wieder verlassen werden und auch den weiteren Weg gemeinsam gehen. Dabei wiederum muss eine der Zufallsrouten vor der anderen begonnen haben.

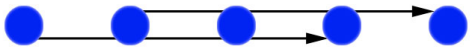

Abb. 5: *Zwei Zufallsrouten mit gemeinsamer Eingangskante nach [2]*

Umgekehrt kann, wenn die Ausgangskante bekannt ist, die Eingangskante eindeutig bestimmt werden. Diese Eigenschaft wird als *Zurückverfolgbarkeit* bezeichnet.Außerdem kann der genaue Knoten von dem die Route ausging anhand der bereits absolvierten Hops eindeutig bestimmt werden.

Diese beiden Eigenschaften der Zufallsrouten werden in Kapitel 4 wieder aufgegriffen und werden zur Begrenzung der Sybil-Gruppen und der Anzahl der Sybil-Knoten noch wichtig werden.

3.2.2 Schleifen in Zufallsrouten Damit ein Subject akzeptiert wird müssen sich die Zufallsrouten des Verifiers und des Subjects überschneiden. Schleifen in den Routen können die Effektivität der Zufallsroute verschlechtern, stellen jedoch kein Sicherheitsriskio dar. Entstehen können sie, wenn bei einem der Knoten nach dem 2. Sprung wieder der Startknoten als Ziel eingetragen ist und im Startknoten genau diese Kante wieder auf die bereits am Anfang gewählte Ausgangskante geroutet wird.
Abbildung 6 zeigt eine Schleife die durch Knoten Nr. 6 mit der Wahl des Startknotens als Ziel seiner Routingtabelle entsteht. Die kleinst mögliche Schleife hätte durch Knoten Nr. 3 erzeugt werden können, was im übrigen auch wahrscheinlicher gewesen wäre, da mit zunehmender Länge der Zufallsrouten die Wahrscheinlichkeit einer Schleifenbildung sinkt. Genauer gesagt beträgt die Wahrscheinlichkeit der Bildung einer Schleife im Knoten Nr. 3 $\frac{1}{Grad}$ dass der Knoten auf den Startknoten zeigt multipliziert mit $\frac{1}{Grad}$, dass die Route des Startknoten wieder auf die selbe Ausgangskante geht. Mit jedem weite-

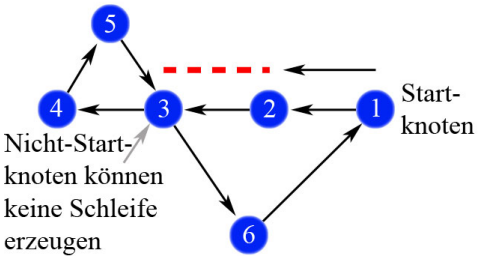

Abb. 6: *Eine Schleife in einer Zufallsroute nach [2]*

ren Knoten multipliziert sich das weiter, sodass es beispielsweise im Knoten Nr. 6 nur noch mit der Wahrscheinlichkeit $\frac{1}{Grad^5}$ zu einer Schleife kommen kann.

Während der Evaluation in Abschnitt 6.2 werden wir sehen, dass Schleifen keine Gefahr für die Verifizierung darstellen, zumal bei *SybilGuard* jeder Knoten entlang aller seiner Kanten eine Zufallsroute erzeugt und durch diese Redundanz der Einfluss von Schleifen und auch der im nächsten Kapitel behandelten Probleme bei dynamischen Netzen weiter verringert wird.

3.2.3 Registry- und Witness-Tabelle Um die Zufallsrouten und die Verifikation durchführen zu können müssen die Knoten nur zwei lokale Datenstrukturen unterhalten, die *Registry-* und die *Witness-Tabelle*. In der *Registry-Tabelle* werden die Knoten gespeichert, deren Zufallsrouten durch den Ersteller der Tabelle gehen, wohingegen in der *Witness-Tabelle* alle Knoten eingetragen sind, durch die die Routen des Erstellers selber gehen.

Damit ein Knoten nicht über seine Route lügen kann muss er sich bei allen Knoten auf seinen Zufallsrouten registrieren. Bei der Verifikation eines Knoten muss das Subject beim Knoten, bei dem die Überschneidung der Zufallsrouten von Verifier und Subjects zustande kommt, registriert sein. Hierzu wird ein *"Token"* genutzt, das nicht einfach von anderen Knoten gefälscht werden kann. Im ursprünglichen Design von *SybilGuard* wurde die IP-Adresse als *Token* benutzt. Da durch *IP-Spoofing* sehr leicht IP-Adressen ausspioniert werden können, wird nun im verbesserten Design die *"Public-Key"-Kryptographie* eingesetzt. Jeder Knoten hat somit seinen eigenen privaten, sowie einen öffentlichen Schlüssel, der als *Token* verschickt wird. Wichtig anzumerken ist, dass diese Schlüssel rein gar nichts mit den *Kantenschlüsseln* zu tun haben!

Diese öffentlichen Schlüssel werden nun für die *Registry-* und für die *Witness-Tabelle* benötigt.
Abbildung 7 zeigt eine solche *Registry-Tabelle*. Jeder Knoten unterhält damit für jede seiner Kanten eine Tabelle mit den öffentlichen Schlüsseln der Knoten deren Zufallsrouten durch ihn führen. So hat z.B. Knoten C die Schlüssel von B

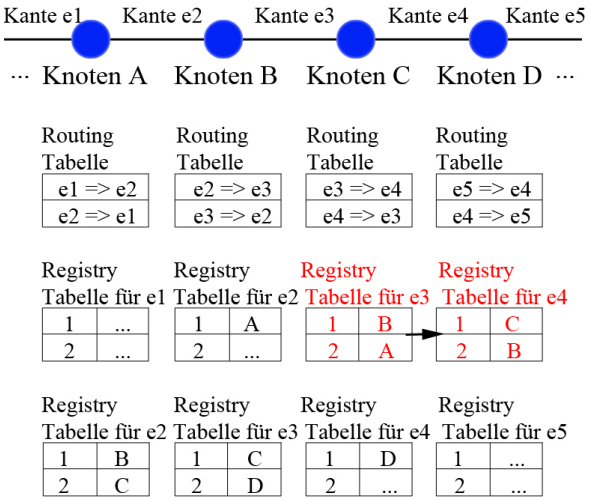

Abb. 7: *Beispiel einer Registry-Tabelle mit d=w=2 nach [2]*

und A an i-ter Stelle eingetragen, da sie ihn auf dem i-ten Hop über Kante e3 erreichen, sowie die Schlüssel von D und einem weiteren, die ihn über die Kante e4 erreichen. Die fertigen *Registry-Tabellen* schickt C dann aufsteigend, gemäß seiner Routingtabelle an die entsprechenden Nachbarn.
Dieser Registrierungsprozess verursacht bei 10 Nachbarknoten, einer Zufallsrouten-Länge von $w = 2000$ und 1024 bit Schlüsseln ca. 2.5 MB an Datenverkehr. Verbessern lässt sich dies auf ca. 400 KB bei Verwendung von 160 bit Hash-Werten der Schlüssel, was heutzutage eine annehmbare Größe darstellt.

In der *Witness-Tabelle* wiederum stehen nun die Knoten, welche auf einer Zufallsroute des unterhaltenden Knotens sind. Darin enthalten sind die öffentlichen Schlüssel und die IP-Adressen der Knoten auf seiner Route. Der öffentliche Schlüssel wird später für die Überschneidungssuche und Authentifizierung benötigt, wohingegen die IP-Adresse lediglich einen Hinweis zum Auffinden des gesuchten Schnittknotens darstellt. Da sich IP-Adresse ändern können sollte eine Änderung den Nachbarknoten mitgeteilt und die *Witness-Tabelle* abgeändert werden. Diese Änderungen können jedoch nachlässig behandelt werden.

Da sich Sybil-Knoten an keinerlei Vorgaben des Protokolls halten müssen können sie, falls sie eine Verbindung zu einem ehrlichen Knoten aufbauen konnten, ihre manipulierte Registry-Tabelle weitergeben.
Bei $n * d * w$ systemweiten Einträgen in Registry-Tabellen der erhlichen Knoten können bei einer Angriffskante jedoch nur $w + (w-1) + ... + 1 \approx \frac{w^2}{2}$ Einträge "verseucht" werden. Sind es g Angriffskanten so multipliziert sich der Wert um

diesen Faktor, was aber immer noch weniger als die Hälfte aller systemweiten Einträge darstellt.

3.2.4 Länge der Zufallsrouten Die Länge der Zufallsrouten (w) ist eine wichtige Designentscheidung. Sie sollte nicht zu lang sein, damit die Route nur mit sehr niedriger Wahrscheinlichkeit in die *Sybil-Region* gelangt. Sie sollte wiederum auch nicht zu kurz sein, damit sich die Routen mit hoher Wahrscheinlichkeit überschneiden.

$\Theta(\sqrt{n}logn); n = \#Knoten$ erfüllt diese beiden Bedingungen. Jedoch ist die Anzahl der Knoten unseres dezentralisierten Netzwerkes nicht bekannt. Aus diesem Grund muss *SybilGuard* die geeignete Länge der Zufallsrouten selbst ermitteln.

Um ein geeignetes w zu finden führt ein Knoten A einen *Random Walk* zufälliger Länge durch, welcher bei Knoten B endet. Während der Evaluation in Abschnitt 6.2 wird sich zeigen, dass drei Sprünge als *Random Walk* ausreichen. Nun führen A und B Zufallsrouten aus um herauszufinden wie lange ihre Routen sein müssen, damit es zu einer Überschneidung kommt. Diese Methode führt Knoten A mehrfach durch und berechnet aus diesen Stichproben den *Median* m. Multipliziert mit 2.1 ergibt sich somit: $w = 2.1m$ als Länge der Zufallsrouten. Der Vorfaktor sorgt für eine 95%ige Wahrscheinlichkeit einer Überschneidung.

3.3 Verifikation

Bei der *Verifikation* eines *Subjects* S sucht der *Verifier* V nach Überschneidungen ihrer Zufallsrouten. Um dies zu tun erhält V alle *Witness-Tabellen* von S zusammen mit seinem öffentlichen Schlüssel und sucht mit allen seinen Tabellen nach einem Knoten bei dem sich ihre Routen kreuzen. Hat V einen Knoten X gefunden, so nimmt er mit Hilfe der IP-Adresse, die in seiner *Witness-Tabelle* gespeichert ist kontakt mit ihm auf. Ist der Knoten unter dieser IP nicht zu erreichern, so fragt er bei den Nachbarn von X nach einer aktuelleren Adresse. Ist X schließlich gefunden, läßt V den Knoten X sich mit seinem privaten Schlüssel authentifizieren.

Falls nun X den selben öffentlichen Schlüssel von S in seiner *Registry-Tabelle* hat, so wird diese Route akzeptiert. Werden mehr als die Hälfte aller Routen von V akzeptiert, so wird auch S akzeptiert und von nun an muss S alle Nachrichten mit seinem privaten Schlüssel signieren.

4 Beschränkung der Anzahl und Größe der Sybil-Gruppen

SybilGuard garantiert sowohl die Beschränkung der Sybil-Gruppen als auch die Anzahl der darin enthaltenen Sybil-Knoten. Diese Garantien sind z.B. genau dann interessant, wenn ein User sicherstellen will, dass sein Datum auf mindestens einem ehrlichen Knoten abgelegt wird. Ist die Anzahl der Sybil-Gruppen (g) bekannt, so muss er lediglich $g + 1$ Kopien auf unterschiedlichen Gruppen verteilen. Ist hingegen die Anzahl und Größe bekannnt und diese kleiner als die Anzahl der erhlichen Knoten ($g * w < n$) , so steigt die Wahrscheinlichkeit exponentiell mit der Anzahl der zufällig verteilten Kopien.

4.1 Beschränkung der Anzahl der Sybil-Gruppen

Da alle Zufallsrouten der Sybil-Identitäten über die Angriffskante des Sybil-Knotens in die ehrliche Region gehen müssen (*Konvergenzeigenschaft*), um zu einem Schnitt mit dem Verifier zu kommen, somit die selbe Route nehmen und über die selbe Eingangskante (e1) in den Schnittknoten gelangen, ordnet der Verifier alle Knoten der selben *Equivalenzgruppe* zu. Diese *Equivalenzgruppe* kann sowohl aus Sybil-Knoten als auch aus erhlichen Knoten bestehen.
Somit ist die Anzahl der Sybil-Gruppen einfach durch die Anzahl der Angriffskanten beschränkt.

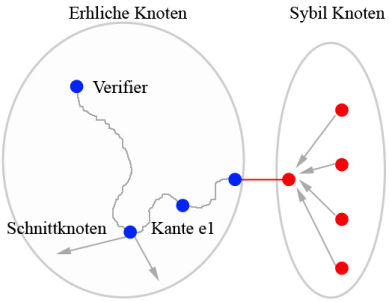

Abb. 8: *Beschränkung der Gruppen durch die Anzahl der Angriffskanten nach [2]*

4.2 Beschränkung der Größe der Sybil-Gruppen

SybilGuard beschränkt jedoch nicht nur die Anzahl der Gruppen sondern auch die Anzahl der Sybil-Knoten in einer Sybil-Gruppe und zwar durch die Länge (w) der Zufallsrouten.
Da der Verifier lediglich w Knoten, die über die selbe Kante den Schnittknoten gelangen, akzeptiert ist somit die Anzahl der Knoten in einer Sybil-Gruppe durch die Länge der Zufallsrouten beschränkt.

5 SybilGuard in einer dynamischen Umgebung

Kapitel 3 lies bisher jegliche Probleme im Zusammenhang mit dynamischen Netzwerken außer acht. Ein P2P-Netzwerk ist aber meist kein statisches Netz. Die Benutzer melden sich an, nutzen das Netz für eine Weile, melden sich wieder ab und kommen möglicherweise nie mehr wieder. Das führt im P2P-Netzwerk zu einigen Problemen die im Folgenden näher beschrieben werden.

5.1 Umgang mit Offline-Knoten

Bei *SybilGuard* kommunizieren Knoten nur bei der *Verifikation* und beim *Propagieren* ihrer Registry- und Witness-Tabellen.

Da sich die Zufallsrouten des Verifiers und des Subjects mehrfach überschneiden können reicht es aus, wenn einer der Schnittknoten auf der Route Online ist. Zudem werden entlang aller Kanten Zufallsrouten erzeugt wovon lediglich die Mehrzahl zum Schnitt kommen müssen.
Das Propagieren der Registry- und Witness-Tabellen geschieht, wenn eine neue Kante hinzukommt oder eine Kante gelöscht werden muss. Das Löschen einer Kante entspricht in unserem sozialen Netz dem Abbrechen einer Freundschaft, was als recht seltenes Ereignis angenommen wird. Der Aufbau einer neuen Freundschaft ist ebenso recht selten. Aus diesem Grund kann das System eine Verbreitungsdauer der neuen Registry- und Witness-Tabellen im Netzwerk von mehreren Tagen verkraften. Die Witness-Tabellen werden zusätzlich noch bei Änderung der IP-Adresse aktualiesiert und verbeitet. Da die IP aber lediglich als Hinweis genutzt wird kann auch hier eine längere Ausbreitungszeit verkraftet werden.

Als Verbesserung könnte eine einfache *Lookahead-Routingtabelle* die Möglichkeit bieten *Offline-Knoten* zu umgehen und so wenigstens allen *Online-Knoten* bereits die neuen Tabellen mitzuteilen.

5.2 Inkrementelle Routing Tabellen

Wenn Kanten in unserem sozialen Netzwerk gelöscht oder hinzugefügt werden müssen die Routing-Tabellen geändert werden.
Das Hinzufügen eines neuen Knoten erfolgt zuerst ohne jegliche Kante. Diese werden erst danach einzeln hinzugefügt. Beim Löschen eines Knoten wird umgekehrt vorgegangen. Deshalb wird nun nur das Hinzufügen der Kanten näher betrachtet.

Das Hinzufügen einer Kante ist exemplarisch in Abbildung 9 dargestellt. Zunächst hat der Knoten einen Grad von 3. Dann wird die Kante *e4* hinzugefügt,

was dazu führt, dass durch zufällige Auswahl, einer der bereits existierenden Routingeinträge diese Kante als neues Ziel einträgt. In unserem Beispiel erhält so *e2* als neues Ziel *e4*. Nun muss lediglich ein neuer Routingeintrag erstellt werden, der die neue Kante *e4* auf das alte Ziel von *e2* routet. Dieses Verfahren sorgt dafür, dass der *Overhead* möglichst gering gehalten werden kann.

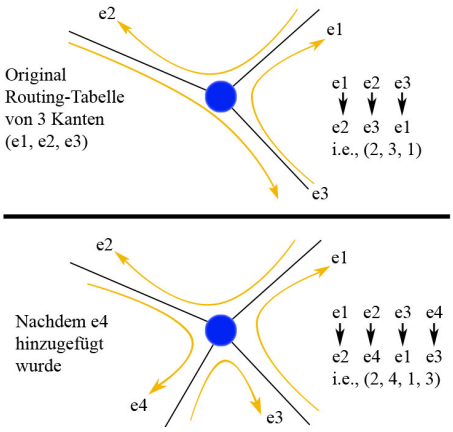

Abb. 9: *Inkrementelle Erweiterung der Routing-Tabelle nach [2]*

5.3 Angriffsmöglichkeiten

Dieser Abschnitt soll zeigen, warum Zufallsrouten entlang aller Kanten auch aus Sicherheitsgründen zur Abwehr von potentiellen Angriffen in einem dynamischen Netzwerk ausgeführt werden sollten.

Zunächst das Szenario: Angenommen jeder Knoten führt nur eine Zufallsroute aus. Abbildung 10 zeigt ein einfaches Beispiel mit $w = 3$ und einem Sybil-Knoten M. Da die Route von M über A, B und C geht haben diese Knoten auch den öffentlichen Schlüssel 1 von M gespeichert. Kommt nun ein neuer Knoten hinzu, der eine Kante zu A aufbaut, so muss A seine Routingtablle ändern. In unserem Beispiel führt die neue Route von M kommend nun nach D. Nun ändert M seinen öffentlichen Schlüssel. A, D und E speichern daraufhin den neuen öffentlichen Schlüssel 2. Zu diesem Zeitpunkt sind die Schlüssel von M bei w-1 (Schlüssel 1) bzw. w (Schlüssel 2) Knoten gespeichert und kann sich somit mit hoher Wahrscheinlichkeit mit beiden verifiziert. Es wird also ausgenutzt, dass der Schlüssel von M bei den Knoten B und C nicht gelöscht wird. Dies explizit zu tun würde sich sehr komplex gestalten, da B und C Offline sein könnten. Als Alternative könnte eine Art *"Time to live"* benutzt werden, was aber einen trade-off erzeugen würde.

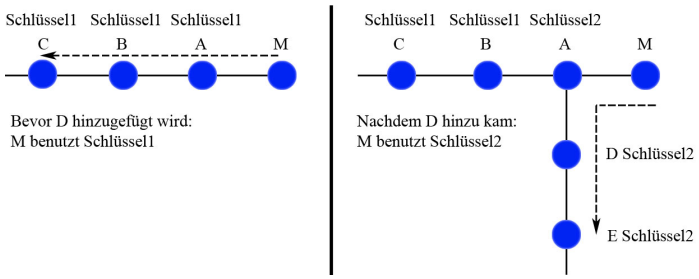

Abb. 10: *Eine potentielle Angriffsmöglichkeit nach [2]*

SybilGuard verhindert einen solchen Angriff dadurch, dass Zufallsrouten entlang aller Kanten ausgeführt werden. Dadurch kann es nur genau einen "Besitzer" des Registry-Eintrages geben. In unserem Beispiel würde D den Eintrag von M mit seinem öffentlichen Schlüssel überschreiben.

6 Evaluation

Um die Aussagen und Garantien aus *Kapitel 2*, *3* und *5* zu verifizieren wurde eine *Evaluation* mittels des synthetischen sozialen Netzwerkmodells von *"Kleinberg"* durchgeführt. Ein Test mit einem realen sozialen Netzwerk ist aufgrund der privaten Informationen innerhalb eines solchen Netzes schwer realisierbar.

6.1 Modelle

Das Modell wurde in drei Skalierungsstufen untersucht. Der Hauptaugenmerk lag auf einem eine Millionen Knoten umfassenden Graphen mit einem durchschnittlichen Grad von 24. Zu einem 10.000 Knoten und Grad 24, sowie einem kleinen 100 Knoten und Grad 12 großen Graphen wurden nur die Zusammenfassungen aufgeführt.

6.2 Ergebnisse

6.2.1 Ohne bösartige User Zunächst wurde die Simulation nur mit ehrlichen Knoten durchgeführt. Hier von Interesse war die Verifizierung der Knoten und die Zufallsrouten, insbesondere deren Länge und der Wahrscheinlichkeit der Bildung einer Schleife.

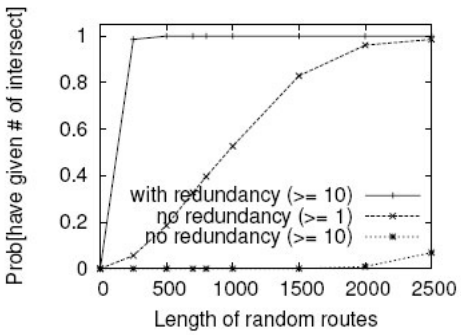

Abb. 11: *Wahrscheinlichkeit einer Überschneidung der Zufallsrouten aus [2]*

Abbildung 11 zeigt (with redundancy (>=10) = Zufallsrouten entlang aller Kanten), dass es bereits bei $w = 300$ fast sicher Überschneidungen gibt, welche auch bei einem dynamischen Netzwerk und bei Knoten, die min. 20% der Zeit Online sind, ausreichend sind, um die Verifizierung erfolgreich durchführen zu können.

Was natürlich gleich die Frage aufwirft, ob diese Länge der Random Routes auch durch das in Unterabschnitt 3.2.4 beschriebene Testverfahren der Knoten bei der Ermittlung von w auch erreicht wird.

Genau dies wurde in einer weiteren Simulation untersucht. Dabei stellte sich heraus, dass die Länge der Zufallsrouten gegen $w = 1906$ konvergierte je mehr Stichproben man hinzu nahm. Ein guter Wert von 1906 ± 300 wurde bereits mit 30 Proben erreicht. Somit wurde der benötigte Mindestwert von $w = 300$ bei weitem übertroffen.
Bei dem 10.000er Netz erwiesen sich 35 Proben für einen Wert von $w = 197 \pm 30$ und beim 100er Netz ein Wert von $w = 24 \pm 7$ bei 40 Proben, als ausreichend.

Desweiteren wurde untersucht, wielange die Random Walks des Algorithmus aus Unterabschnitt 3.2.4 zur Bestimmung der Länge der Zufallsrouten sein muss.

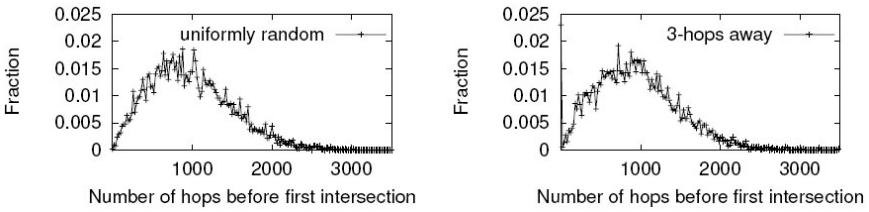

Abb. 12: *Vergleich der Random Walk Längen aus [2]*

Abbildung 12 zeigt einen Vergleich zwischen Random Walks zufälliger Länge und einer Länge von drei. Beide Graphen unterscheiden sich kaum, was zeigt, dass ein drei Sprünge langer Random Walk ausreicht.

Außerdem von Interesse war nun noch die Wahrscheinlichkeit einer Schleifenbildung, da dies wie bereits in Unterabschnitt 3.2.2 beschrieben die Wahrscheinlichkeit einer Überschneidung der Zufallsrouten verschlechtert. Es ergab sich jedoch, dass in 99,3% der Fälle keine Schleife innerhalb der ersten 2500 Sprünge vorkam. Somit waren auch keine Schleifen innerhalb der ermittelten Länge der Zufallsroute von $W = 1906 \pm 300$ zu befürchten. Ebenso verhielt es sich auch bei den beiden kleineren Netzen.

6.2.2 Mit Sybil-Angreifern Nachdem die Simulation ohne bösartige Knoten die erwarteten Ergebnisse lieferte, wurden nun auch *Sybil-Angreifer* simuliert, um deren Auswirkungen zu testen.
Sybil-Angreifer waren hierbei bösartige Knoten, die eine unendliche Anzahl von *Sybil-Knoten* erzeugen konnten. Die Anzahl der *Sybil-Angreifer* wurden zwischen 0 und 100 variiert. Diese Angreifer sollten 2500 Angriffskanten erzeugen,

was jedoch *SybilGuard* verhindern sollte.

Untersucht wurde nun wie hoch die Wahrscheinlichkeit war Sybil-Knoten zu akzeptieren, ob ehrliche Knoten immer noch mit einer hohen Wahrscheinlichkeit verifiziert werden können und wie lang die Zufallsrouten nun werden konnten, um nur mit einer kleinen Wahrscheinlichkeit in die Sybil-Region zu gelangen.

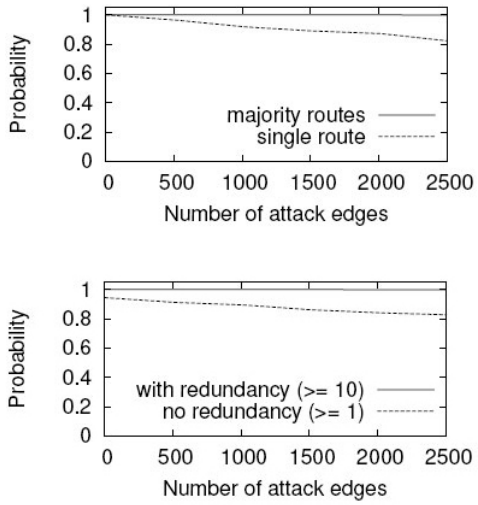

Abb. 13: *Zufallsroute in ehrlicher Region, ehrliche Knoten werden akzeptiert aus [2]*

Beide Bilder der Abbildung 13 zeigen einen sehr ähnlichen Verlauf. Weder die Akzeptanz von ehrlichen Knoten, noch die Ausbreitung der Random Routes in Sybil-Regionen stellt bei 2500 Angriffskanten ein Problem dar. 99,8% der ehrlichen Knoten wurden immer noch akzeptiert und mit dem selben Prozentsatz überschreitet keine der Zufallsrouten eine Angriffskante.

Die Simulation zeigte jedoch auch, dass die kleineren Netze mehr unter den Auswirkungen der Sybil-Attacken zu leiden hatten. So konnten beispielsweise bei dem kleinsten Netz bis zu 5,1% der Knoten nicht geschützt werden.

Auch das abschließenden Experiment zur Länge der Zufallsrouten konnte ohne schlechte Ergebnisse abgeschlossen werden. Es stellte sich heraus, dass die Wahrscheinlichkeit, dass ein Knoten beim Ermitteln der Länge der Zufallsrouten eine schlechte Stichprobe innerhalb der Sybil-Region wählt zwar linear ansteigt, jedoch selbst bei den beiden kleineren Netzwerken immer unterhalb von 20% bleibt.

Literaturverzeichnis

[1] Flora Rhea Schreiber. *Sybil*. Warner Books, 1973.

[2] Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, Abraham Flaxman. *SybilGuard: Defending Against Sybil Attacks via Social Networks*. ACM Press, New York NY, USA, 2006.

[3] John R. Douceur. *The Sybil Attack*. Microsoft Research - Press, London, UK, 2002.

[4] Peter Mahlmann, Christian Schindelhauer. *Peer-to-Peer Netzwerke*. Springer-Verlag, Freiburg, Germany, prerelease edition, 2007.

Index