

# Cookies

## Inhalt

- Inhalt
- Prolog
- Vorgeschichte
- Cookies (RFC-2109)
  - Definitionen
  - Ablauf
  - Inhalt des "set-cookie-headers"
  - Inhalt von Cookies
  - Regeln für das Zurückweisen von Cookies
  - Regeln für den Umgang mit Cookies
  - Beschränkungen für Clients
  - Beschränkungen für Server
- "Privacy"
  - Sicherheit
  - Sonstiges

## Prolog

Vor einer datenschutzrechtlichen Bewertung von "cookies" ist es notwendig, eine begriffliche Klärung vorzunehmen. Dabei ist zwischen der Spezifikation für "cookies" gemäß RFC-2109 und ihren tatsächlichen Implementierungen zu unterscheiden. Hinzu kommt der Umgang mit "cookies" von Seiten verschiedener Server, der diesbezüglich einer differenzierten Bewertung bedarf. In dieser Ausarbeitung liegt der Schwerpunkt auf der Darstellung des Standards, d.h. von RFC-2109.

An etlichen Stellen dieses Dokuments wird aus RFC-2109 zitiert oder übersetzt. Beim Durchlesen wird man feststellen, daß viele der Formulierungen unklar oder, schlimmer noch, widersprüchlich sind. Im RFC selbst werden konsequent etablierte Begrifflichkeiten vertauscht und – unter Datenschutzgesichtspunkten – schwächste Formulierungen eingesetzt (should, should not). Um den RFC selbst zu verstehen, sollte man folgende Hinweise im Kopf behalten:

Der Cookie-Austausch erfolgt zwischen Webserver und Webclient. Der Webserver ist der HTTP-Server, der auf die HTTP-Anfragen des Webclients (des Browsers) reagiert und ggf. HTML-Dokumente verschickt. Vor einem Cookies-Austausch steht eine HTTP-Anfrage. Der Webclient fragt beim Server nach (HTML-) Dokumenten. Diese Transaktionen entsprechen dem klassischen Schema einer Client-/Server-Interaktion, bei der ein Client um einen Dienst bittet und ein Server den Dienst ausführt.

Beim Cookie-Austausch sind die Aufgaben umgekehrt vergeben. Zuerst fragt ein Webserver

einen Webclient, ob dieser für ihn Cookies speichert und später zurückliefert. Übernimmt der Webclient diese Aufgabe, so wird aus dem Webserver de facto ein "Cookie-Client" und aus dem Webclient ein "Cookie-Server".

Dieser Aufgabentausch wird von den Autoren des RFC-2109 an keiner Stelle explizit benannt. Statt dessen verwenden sie für die Anfrage des Webservers an den Webclient betreffs der Cookie-Lagerung den Begriff "response", d.h. Antwort. Einer Antwort sollte aber eine Frage vorangegangen sein. Im Falle von Cookies erfolgt die Frage, d.h. der "request", nach der "response", wie aus RFC-2109 hervorgeht.

Für den von den Autoren eingeführten Begriffsgebrauch spricht lediglich die formale Einbettung des "set-cookie-header" in die HTTP-Antwort (http response) des Webservers, sowie die Einbettung des "cookie-request" in den HTTP-Anfrage (http request) des Webclients. Die inhaltlichen Argumente werden dadurch jedoch nicht hinfällig.

Eine dritte Begriffsverwirrung nehmen die Autoren vor, wenn sie die Anfrage zur Cookie-Lagerung als "set-cookie-header" bezeichnen. Die Verwirrung besteht darin, daß der Header gar kein Header ist, sondern die Cookies bereits enthält. Insofern gibt es nur "content", keinen Header. Der etablierte Gebrauch der Begriffe "header" und "content" wird dadurch konterkariert.

In der Summe scheinen die verbalen Mißgriffe kein Zufall mehr sein zu können, es ist wohl Absicht zu unterstellen. Eine Ursache dafür könnte in der Autorenschaft eines Netscape-Mitarbeiters zu sehen sein, der die Interessen seiner Firma mit allen Mitteln durchsetzen wollte.

## Vorgeschichte

"Cookies" wurden von der Firma Netscape im Jahr 1994 (?) als Ergänzung zum HTTP-Protokoll eingeführt. In der Spezifikation von HTTP/1.0 waren keine Mechanismen zur Dokumentation des Status' einer Verbindung zwischen HTTP-Server (server) und HTTP-Client (client, user agent) vorgesehen. Der Datenaustausch war als "Abruf" von Daten, die auf dem Server gespeichert oder erzeugt werden, durch den Client konzipiert. Der Server konnte nur auf Anfragen antworten und war nicht oder nur sehr begrenzt in der Lage, den Kontext solcher Anfragen zu bestimmen. Aus der Perspektive einer wünschenswerten Interaktion, wie es der Online-Einkaufsbummel für den Internet-Nutzer darstellt, war das Fehlen von Statusinformationen ein Mangel. Abhilfe schuf Netscape durch die Integration von "cookies".<sup>1</sup>

Cookies im Sinne dieser ersten Netscape-Implementierung waren kleine Datenpakete von einigen Kilobytes Größe, die ein Server an einen Client übermitteln konnte. Der Client speicherte empfangene Cookies auf der lokalen Festplatte und lieferte sie auf Anfrage an den Server zurück. In diesen Cookies konnten beliebige Informationen über den Status einer bestehenden oder vergangenen HTTP-Verbindung (session) abgelegt werden. Zu beachten waren dabei lediglich lexikalische Vorgaben.

Mit der Ausbreitung des World Wide Web (WWW) und neu entstehenden Implementierungen von HTTP-Servern und -Clients (web browsers) wurde eine Standardisierung erforderlich. Dazu wurde von der IETF (Internet Engineering Task Force) eine Arbeitsgruppe mit der Spezifikation eines Standards beauftragt. Die Ergebnisse des Entwicklungsprozesses liegen als RFC-2109 seit

---

<sup>1</sup> Anmerkung: Cookies wird im weiteren ohne Anführungszeichen und groß geschrieben.

Februar 1997 vor und wurden als Internet-Standard verabschiedet.

Abgesehen von der Spezifikation in RFC-2109 existieren mehrere Client-seitige Implementierungen des Cookie-Protokolls (Netscape Navigator, MS Internet Explorer u.a.), sowie eine unüberschaubare Vielzahl an Server-seitigen Implementierungen. Letzteres ist dem Umstand geschuldet, daß die Abwicklung großer Teile der Cookie-Bearbeitung durch CGI-Skripten erfolgt, die von den Server-Betreibern entwickelt werden und keiner unabhängigen Prüfung unterliegen. Eine Alternative zur CGI-Programmierung stellt der Einsatz von in die Webseite eingebettetem JavaScript-Code dar.<sup>2</sup>

Die notwendige Unterstützung des Cookie-Protokolls stellen dabei die Web-Server (Apache, Netscape FastTrack, MS Internet Information Server etc.) bereit. Über die CGI-Schnittstelle der Server kann auf diese Protokollelemente zugegriffen werden. Fast alle Server bieten von sich aus bereits einfache Cookie-Module<sup>3</sup> an.

## Cookies (RFC-2109)

RFC-2109 wurde als ein Vorschlag für die Spezifikation von HTTP-Verbindungen mit Austausch von Statusinformationen (stateful sessions) von D.Kristol (Bell Labs, Lucent Technologies) und L.Montulli (Netscape Communications) im Februar 1997 vorgelegt. Zu beziehen ist RFC-2109 vom WWW-Server der IETF aus dem RFC-Verzeichnis.

### Definitionen

Im Sinne von RFC-2109 ist ein *cookie* die Statusinformation die zwischen WWW-Server als Urheber und WWW-Client als Empfänger und Rücksender ausgetauscht wird.<sup>4</sup> Der Austausch von Statusinformationen führt zur Herstellung einer HTTP-Verbindung.<sup>5</sup> Eine solche Verbindung geht über den simplen Austausch von Informationen nach dem HTTP-Protokoll hinaus, das keine Statusinformationen vorsieht.<sup>6</sup>

Verbindungen mit Cookie-Austausch heißen im RFC-2109 *stateful*. Durch den Austausch von Statusinformationen wird ein Kontext hergestellt, der im reinen HTTP-Protokoll fehlt. Man

---

<sup>2</sup> Der Unterschied zwischen CGI- und JavaScript-Variante liegt primär in der Ausführungsumgebung. CGI-Programme werden auf dem Webserver ausgeführt, JavaScript-Programme vom Webclient. Daraus ergibt sich aus Benutzersicht eine unterschiedliche Transparenz: CGI-Programme bekommt der Benutzer nicht zu Gesicht, JavaScript-Programme lassen sich mit der "View page source"-Funktion des Web-Browsers inspizieren.

<sup>3</sup> Im Fall des Apache-Servers heißt dieses Modul beispielsweise *mod\_usertrack*. Vgl. Roßbach 1998, S.189.

<sup>4</sup> Vgl. RFC-2109, 2. Terminology: "*Because it was used in Netscape's original implementation of state management, we will use the term cookie to refer to the state information that passes between an origin server and user agent, and that gets stored by the user agent.*".

<sup>5</sup> In RFC-2109 "session" genannt. Die ursprüngliche Übersetzung von *session* lautet *Sitzung*. Im Deutschen stellt man sich unter einer Sitzung üblicherweise ein Treffen mehrerer Personen am selben Ort vor. Ein solches Bild würde irreführen, wenn man es für die Interaktion von Web-Server und Web-Browser verwenden würde. Hier erscheint es angebrachter, von *Verbindungen* zu reden, im Sinne einer Telefonverbindung.

<sup>6</sup> Statt dessen gibt es bei HTTP Anfragen und Antworten, wobei jede neue Anfrage als unabhängig von einer vorangegangenen Anfrage betrachtet werden kann. Um den Unterschied verbal zu illustrieren, könnte man hierbei von *connections* sprechen.

könnte somit auch von *kontextsensitiven Verbindungen* im Unterschied zu *kontextfreien Anfragen* (ohne Cookie-Austausch) sprechen.

Vier Kriterien werden in RFC-2109 aufgeführt, um eine Verbindung zu charakterisieren:

1. Verbindungen haben einen Anfang und ein Ende.
2. Verbindungen sind verhältnismäßig kurz befristet.
3. Verbindungen können vom Server oder vom Client beendet werden.
4. Verbindungen entstehen implizit durch den Austausch von Statusinformationen.

Dies sind Definitionskriterien, die von den Cookie-Implementationen nicht unbedingt erfüllt werden.

### **Ablauf**

Der Austausch der Statutinformatoren wird durch zwei komplementäre Aktionen realisiert:

- Cookie-Anfrage (*cookie-request*)
- Cookie-Antwort (*set-cookie-response*)

Will ein Server eine Verbindung zu einem Client herstellen, schickt er im Rahmen der Abwicklung des normalen HTTP-Protokolls eine Cookie-Antwort (*set-cookie-response*). Die Cookie-Anfrage (*cookie-request*) sendet der Client (Web-Browser) dem Server, wenn er bereit ist, Cookies zu akzeptieren. Erhält der Server eine positive Antwort<sup>7</sup>, d.h. einen *cookie-request*, so schickt er dem Client den Cookie, zur Abspeicherung.

Aus RFC-2109 geht nicht klar hervor, ob der Server dem Client die Cookies sozusagen "präventiv" mitschickt, oder ob erst eine positive Antwort abgewartet werden muß. Die dort enthaltenen Erläuterungen lassen beide Schlüsse zu. Die Spezifikation des Headers schließt jedenfalls die Cookies selbst mit ein, so daß die Bezeichnung *header* irreführend ist.<sup>8</sup>

Die Initiative zur Herstellung der Verbindung geht vom Web-Server aus. Will der Server die Verbindung beenden (s.o.), schickt er eine Cookie-Antwort mit einer Verbindungsdauer von *Max-Age=0* im Kopfteil.<sup>9</sup>

---

<sup>7</sup> Die Verwendung von *response* und *request* in RFC-2109 ist irreführend. Eigentlich ist es der Server, der um die Erlaubnis zur Übermittlung des Cookies nachsucht. Der Client gibt darauf eine positive oder negative Antwort und signalisiert so seine Bereitschaft, Cookies zu speichern.

Dieser widersprüchliche Gebrauch der Begriffe wird durch den Rollentausch von Client und Server verursacht. Im klassischen Sinne ist ein Client die Partei, die eine Anfrage stellt an den Server. Server nehmen Anfragen von Clients entgegen und beantworten sie.

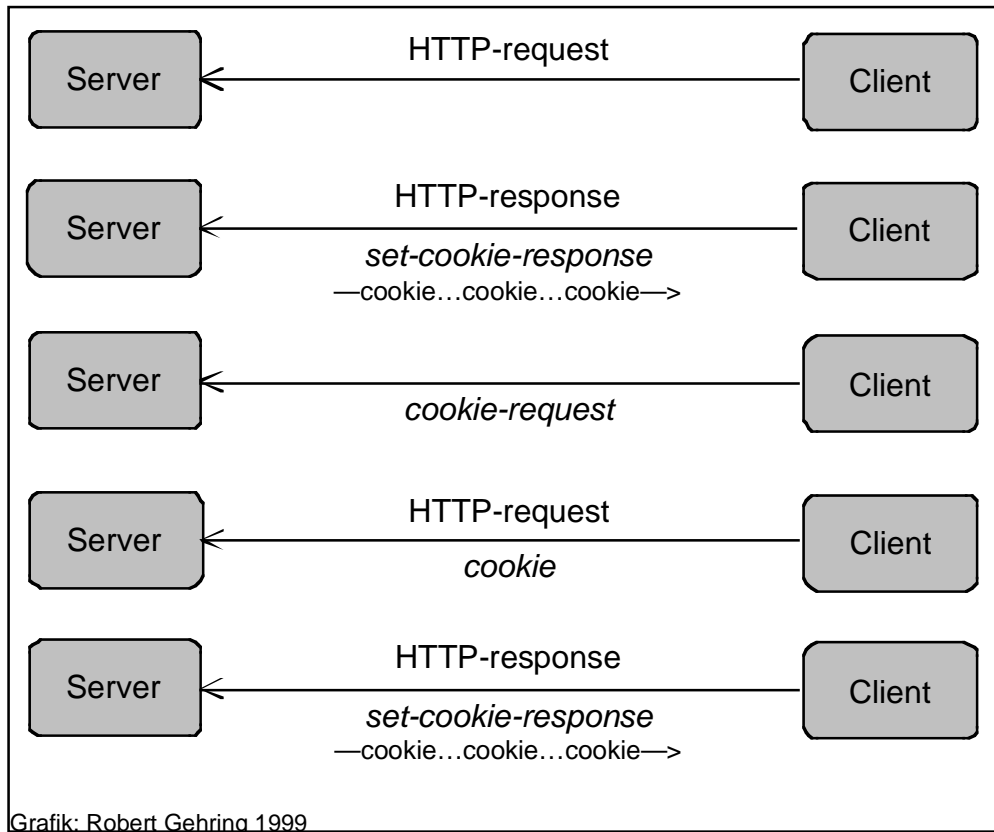
In Bezug auf Cookies stellt sich die Lage umgekehrt dar: Der Server will, daß der Client ihm einen Dienst erweist. Der Client erfüllt den Wunsch oder lehnt die Erfüllung ab. So gesehen wird der Computer mit dem HTTP-Client (Browser) zum "Cookie-Server" und der HTTP-(Web-) Server zum "Cookie-Client".

<sup>8</sup> Üblicherweise unterscheidet man zwei Teile eines Nachrichtenpakets: *header* und *content*. Dabei enthält der *header*-Teil Adreß- und Statusangaben und der *content*-Teil den für den Adressaten bestimmten Inhalt. In diesem Sinne gleicht ein Nachrichtenpaket dem Postpaket: der *header* ist die Aufschrift auf der Verpackung, der *content* ist der eigentliche Paketinhalt.

<sup>9</sup> RFC-2109, 4.2.1.

In der folgenden Abbildung ist die Initiierung einer Verbindung und der sich vollziehende Cookie-Transfer zu sehen. Die beiden letzten Schritte werden bis zum Abbruch der Verbindung wiederholt.

**Abbildung: Verbindungsaufbau und Cookie-Transfer**



Eine Verbindung bleibt bestehen, solange entweder der Server sie nicht beendet oder der Client sie nicht beendet. Ein Client beendet die Verbindung durch *Nichtübermittlung* der Statusinformation. Die Verbindung wird Client-seitig unterbrochen, wenn die Lebensdauer eines Cookies abgelaufen ist (cookie expired) oder die Übermittlung eines Cookies, dessen Lebensdauer noch nicht abgelaufen ist, unterbunden wird. Im letzteren Fall muß der Benutzer des Web-Browsers von sich aus aktiv werden und die Übermittlung von Cookies abschalten.

### **Inhalt des "set-cookie-headers"**

Die Spezifikation von RFC-2109 sieht für den "set-cookie-header" folgenden Inhalt vor<sup>10</sup>:

**"Set-Cookie:" cookies**

Der *cookies*-Teil enthält *einen oder mehrere* Cookies<sup>11</sup>, kann also *nicht leer* sein.

<sup>10</sup> RFC-2109, 4.2.2.

<sup>11</sup> RFC-2109, 4.4.2: "Informationally, the Set-Cookie response header comprises the token Set-Cookie: followed by

## Inhalt von Cookies

Cookies schließen sich unmittelbar an den "set-cookie-header" an. Der Name des Cookies wird vom Server festgelegt und sein Inhalt ist für den Benutzer unverständlich<sup>12</sup>. Um ein Beispiel aus Abschnitt 5 von RFC-2109 aufzugreifen, könnte der Name des Cookies "Customer" lauten und sein Inhalt "WILE\_E\_COYOTE". Das Cookie-Name-Cookie-Wert-Paar könnte aber auch lauten:<sup>13</sup>

```
NGUserID="cc98a714-258-891212008-3"
```

Im Falle des ersten Beispiels ist klar, worum es geht: Identifizierung eines Kunden. Im zweiten Beispiel ist der Benutzer auf Mutmaßungen angewiesen. Das dargestellte Cookie wurde vom Netscape-Webserver (<http://home.netscape.com>) übermittelt. Das *N* in *NGUserID* könnte also für *Netscape* stehen. In Erinnerung an die von Microsoft eingeführte Global User ID (GUID)<sup>14</sup> könnte man *GUserID* als *Global User ID* interpretieren. Die Bedeutung des Cookies wäre also die Übertragung einer Netscape-eigenen Benutzeridentifizierung.

Außer Name und Inhalt, gehören zu einem "set-cookie-response" noch optionale Parameter wie Kommentar, Domain-Angabe für die empfangsberechtigten Server, Versionsangabe, eine Pfadangabe, eine Lebensdauer (Max-Age) und einem Sicherheitsparameter. Letzterer informiert den Webbrowser, den Cookie nur über eine gesicherte Verbindung<sup>15</sup> an der Server zurückzuliefern.

Parameter	Einstufung
Name	required
Comment	optional
Domain	optional
Max-Age	optional
Path	optional
Secure	optional
Version	optional

Der Web-Browser ist für die Verwaltung der empfangenen Cookies zuständig und ergänzt fehlende Parameter selbständig. Die Parameter werden entsprechend der folgenden Tabelle ggf. aufgefüllt:

Parameter	Standardwert
Version	"old cookie"

---

*a comma-separated list of one or more cookies."*

<sup>12</sup> Im Text von RFC-2109 heißt es "*opaque*". Das Duden Oxford Großwörterbuch Englisch, Dudenverlag 1990, S. 493 schlägt für "*opaque*" folgende Übersetzungen vor: lichtundurchlässig; opak; dunkel; unverständlich.

<sup>13</sup> Das Beispiel stammt aus Abbildung 17.1 des Buches "*Der Apache Webserver*" von Stephan Roßbach, ADDISON WESLEY LONGMAN Verlag GmbH, 1998 (Roßbach 1998).

<sup>14</sup> Vgl. "*Windows 98 spöht die Anwender aus*" in Computer Zeitung 10/11.3.1999.

<sup>15</sup> Die Art der Sicherung bleibt in der Spezifikation offen.

Domain	request host
Max-Age	verfällt bei Programmende
Path	path of the request URL
Secure	ungesicherte Verbindung ist zulässig

Es ist nicht klar, ob mit der Belegung "*request host*" für *Domain* der Cookie-sende Webserver oder der Client-Computer gemeint ist.<sup>16</sup> Ähnlich unklar bleibt, ob mit "path of the request URL" etwas anderes als der Pfad der angeforderten URL gemeint ist.

Für die Größe des Inhalts und die Anzahl vom Server zum Client übermittelbarer Cookies gibt es in RFC-2109 Regeln, die an dieser Stelle nicht näher beschrieben werden sollen.

### **Regeln für das Zurückweisen von Cookies**

Clients (Web-Browser) können Cookies zurückweisen. Gründe dafür sehen die Autoren von RFC-2109 in Sicherheits- und Datenschutzerwägungen.<sup>17</sup> Es gibt vier Bedingungen für das Zurückweisen von Cookies:

1. Die Pfadangabe im Path-Parameter ist nicht Bestandteil der angeforderten URI<sup>18</sup>.
2. Der Wert des Domain-Parameters enthält keine eingebetteten Punkte oder beginnt nicht mit einem Punkt.
3. Die Angabe des Servers paßt nicht zu der Domain aus dem Domain-Parameter.
4. Die Serverangabe ist eine FQDN<sup>19</sup>-Angabe mit den Bestandteilen *Host.Domain*, wobei *Domain* mit dem Inhalt des Domain-Parameters übereinstimmt und *Host* einen oder mehrere Punkte enthält<sup>20</sup>.

Es genügt, daß eine der Bedingungen erfüllt ist, damit ein Client einen Cookie ablehnen soll. Andere Regeln zur Ablehnung schreibt RFC-2109 nicht vor.

### **Regeln für den Umgang mit Cookies**

Die Regeln in diesem Abschnitt stellen Zusammenfassungen von Abschnitt 4.3.3, RFC-2109 dar.<sup>21</sup>

- Neue Cookies haben Vorrang vor älteren Cookies, wenn
  - Der Name des neuen Cookies exakt mit dem Namen eines existierenden Cookies

---

<sup>16</sup> In der Praxis wird die Domain des Webservers gewählt.

<sup>17</sup> "To prevent possible security or privacy violations ...", RFC-2109, 4.3.2.

<sup>18</sup> URI = Universal Resource Identifier.

<sup>19</sup> Fully Qualified Domain Name.

<sup>20</sup> Z.B. `www.werbung.web.de` oder `junk.mail.at.web.de`, wobei `web.` die Domainangabe wäre.

<sup>21</sup> Um den Sinn von RFC-2109, Abschnitt 4.3.3, möglichst nicht zu verfälschen, wurde die Formulierung der Regeln mit "*sollte*" vorgenommen, da es im Originaltext "*should*" heißt. Insofern sind diese Regeln als Empfehlungen und nicht als verbindlich zu werten.

übereinstimmt und

- der Inhalt des Domain-Parameters bei beiden Cookies exakt übereinstimmt und
- der Inhalt des Path-Parameters bei beiden Cookies exakt übereinstimmt.
- Sollte der Speicherplatz für Cookies erschöpft sein, kann der Client alte Cookies löschen, z.B. wenn sie lange nicht benutzt wurden.
- Der Inhalt des Kommentar-Parameters eines empfangenen Cookies sollte vom Client angezeigt werden. Dafür sollte der Client ein "cookie inspection user interface" verwenden.
- Clients sollten dem Benutzer eine kontrollierte Zerstörung von Cookies ermöglichen.
- Überlegungen zum Schutz der Privatsphäre machen es erforderlich, daß Anwender über nennenswerte Möglichkeiten zum Umgang mit Cookies verfügen.

Neben diesen Empfehlungen für den lokalen Umgang mit Cookies gibt es solche für die Umstände, unter denen die Cookies an den Ursprungsserver zurückgeschickt werden. Das sind im einzelnen:

- Die Cookies sollten von derselben Version sein, wie im "set-cookie-header" angegeben.
- Andere Parameter sollten nur übermittelt werden, wenn sie mit denen aus dem "set-cookie-header" übereinstimmen. Andernfalls sollten sie weggelassen werden.

Die Auswahl der zu sendenden Cookies aus der Menge der gespeicherten Cookies erfolgt gemäß dem folgenden Schema:

- Domainauswahl
  - Der FQHN des Ursprungsservers muß auf den Inhalt des Domain-Parameters passen.  
*Beispiel:* `http://ig.cs.tu-berlin.de` paßt auf `Domain=.cs.tu-berlin.de`
- Pfadauswahl
  - Der Inhalt des Pfad-Parameters muß einen linken Ausschnitt aus der angeforderten URI darstellen.  
*Beispiel:* `http://www.newsnews.de/news/today` paßt auf `Path=/news.`

### **Beschränkungen für Clients**

Abschnitt 6.3 von RFC-2109 legt einige Randbedingungen für den Umgang von Clients (Browsern) mit Cookies fest:

- Mindestens 300 Cookies sollen verwendbar sein.
- Cookies soll mindestens 4096 Bytes groß sein können.
- Jedem einzelnen Server oder jeder einzelnen (Server-)Domain sollen mindestens 20 Cookies eingeräumt werden.

- Sollte es sich um einen Client mit geringen Ressourcen handeln, sollte er trotzdem mindestens 20 Cookies zu je 4096 Bytes zulassen.
- Die Cookie-Information muß entweder vollständig und unverändert sein oder darf gar nicht gespeichert werden.

### **Beschränkungen für Server**

Einige Beschränkungen für Server werden ebenfalls in Abschnitt 6.3 von RFC-2109 festgelegt:

- Applikationen sollten so wenig Cookies wie möglich verwenden.
- Verwendete Cookies sollten so klein wie möglich sein.
- Auf den Verlust eines Cookies sollten Applikationen tolerant<sup>22</sup> reagieren.

### **Privacy<sup>23</sup>**

Zum Schutz der Privatsphäre finden sich Aussagen insbesondere in Abschnitt 7 von RFC-2109. Da es sich um – unter Datenschutzgesichtspunkten – besonders wichtige Aussagen handelt, folgt an dieser Stelle eine Rohübersetzung des gesamten Kapitels 7 von RFC-2109.

#### ***"7. Privacy***

##### ***7.1 "User Agent"-Kontrolle***

*Ein Ursprungsserver ist in der Lage einen "set-cookie-header" zu erzeugen, um den Weg eines Anwenders durch die Pfade des Dokumentenbaumes auf dem Server zu verfolgen. Anwender könnten dieses Verhalten als eine unerwünschte Form von Informationsbeschaffung ablehnen, selbst wenn ihre Identität nicht bekannt würde. (Ihre Identität könnte zu einem späteren Zeitpunkt bekannt werden, z.B. durch das Ausfüllen eines entsprechenden Formulars.) Diese Spezifikation für eine Statusverwaltung fordert deshalb, daß ein "user agent"<sup>24</sup> dem Anwender die Kontrolle über ein solches Eindringverhalten ermöglicht. Die Art und Weise der Umsetzung dieser Forderung bleibt unspezifiziert. Der Kontrollmechanismus soll mindestens folgende Aktionen durch den Anwender unterstützen:*

- *Das Abschalten des Sendens und Speicherns von Cookies.*
- *Das Feststellen, ob eine Verbindung<sup>25</sup> besteht.*

---

<sup>22</sup> Im Text von RFC-2109, 6.3 heißt es "graceful".

<sup>23</sup> Der Originalterminus Privacy wird beibehalten, um einer Verwechslung mit dem Datenschutzproblem vorzubeugen. Den Autoren von RFC-2109 geht es Abschnitt 7 um Überlegungen zum Schutz der Privatsphäre, weniger um den Schutz der Daten des Client- (Browser-) Benutzers. Die in den USA zugrundegelegte Bedeutung von "Privacy" geht auf S.Warren und L.Brandeis (1890) zurück: „...—the right to be let alone...“ Zitat nach Diffie/Landau 1998, S.131.

<sup>24</sup> "user agent" = Client = Browser.

<sup>25</sup> Original: "session".

- *Die Entscheidung, die Speicherung eines Cookies zuzulassen, muß in Abhängigkeit von der angegebenen Domain erfolgen können.*
- *Wenn ein Client einen Cookie an einen Server schicken will, um eine Verbindung herzustellen, wird der Benutzer darüber informiert und die Möglichkeit angeboten, daß er die Verbindungsaufnahme ablehnen kann.*
- *Wenn eine Verbindung besteht, wird der Benutzer durch eine visuelle Anzeige darauf hingewiesen.*
- *Wenn der Benutzer ein Fenster oder den Browser schließt, wird ihm die Möglichkeit zur Auswahl der Cookies angeboten, die er speichern will.*
- *Der Benutzer sollte den Inhalt eines Cookies jederzeit in Augenschein nehmen können.*

*Ein Client beginnt bei der Ausführung üblicherweise ohne Statusinformationen. Es sollte möglich sein, den Client so zu konfigurieren, daß er niemals Cookies verschickt, wodurch keine statusgesteuerte Verbindung zu einem Ursprungsrechner aufgebaut werden kann. (Der Client verhält sich dann wie einer, der unfähig ist, mit Cookies umzugehen.)*

*Wenn der Client seine Ausführung beendet, sollte dem Benutzer die Möglichkeit angeboten werden, alle Statusinformationen zu löschen. Alternativ kann der Benutzer befragt werden, ob Statusinformationen beibehalten werden sollen. Die Vorgabe dafür sollte "Nein" sein. Wenn der Anwender beschließt, Statusinformationen zu behalten, wird diese bei der nächsten Ausführung des Clients wiederhergestellt.*

*Hinweis: Clients sollten vermutlich vorsichtig im Umgang mit Files zur Langzeitspeicherung von Cookies sein. Sollte ein Anwender mehr als eine Instanz des Clients zur Ausführung bringen, könnten Cookies vermischt oder anderweitig "verspeist"<sup>26</sup> werden.*

## **7.2 Protokolldesign**

*Die Restriktionen bezüglich des Domain-Parameters und die Regeln für den Umgang mit nicht überprüfbaren Transaktionen sind vorgesehen, um die Möglichkeiten von Informationsabgaben durch den Cookie an "falsche" Rechner einzuschränken. In der Intention geht es darum, Cookies auf einen Host oder eine eng verbundene Menge von Hostrechnern einzuschränken. Aus diesem Grunde ist ein "request-host" auf die Werte beschränkt, die er im Domain-Parameter setzen kann. Wir betrachten es für die Hostrechner `host1.foo.com` und `host2.foo.com` als akzeptabel, sich Cookies zu teilen. Das gilt nicht für `a.com` und `b.com`.*

*Entsprechend kann ein Server den Pfad-Parameter nur für diejenigen Cookies setzen, die in Beziehung zur "request-URL" stehen."*

---

<sup>26</sup> Im Originaltext: "messed up".

### **Sicherheit**

Die Informationen in den Cookies sind ungeschützt und können durch einen böswilligen "Zwischenbesitzer"<sup>27</sup> manipuliert werden. Aus diesem Grunde sollten private und/oder Finanzinformationen nicht über ungeschützte Kanäle übermittelt werden. Werden Cookies über ungeschützte Kanäle übertragen, sollten Server-seitig Vorkehrungen für den Fall einer Manipulation getroffen werden (RFC-2109, 8.1). Zu solchen Vorkehrungen gehören auch Überprüfungen, die eine "cookie spoofing"-Manipulation erkennen lassen (RFC-2109, 8.2).

### **Sonstiges**

RFC-2109 enthält noch diverse Empfehlungen, z.B. für den Umgang mit Cookies durch Proxies, Caches etc. Diese sind im Zusammenhang mit der Datenschutzproblematik von untergeordneter Bedeutung und wurden hier nicht behandelt. Bei Bedarf sei auf den Text von RFC-2109 verwiesen, zu finden unter <http://www.ietf.org/rfc>.

---

<sup>27</sup> Solch ein Zwischenbesitz stellt sich "Schleusen" wie Proxies oder Caches ein. Diewser "Zwischenbesitz" kann praktisch nicht überprüft werden.