DKIM – Kryptographie auf wackligem Boden

Steffen Ullrich[1]

Kurzfassung:
Ein großer Teil der heutigen Angriffe auf Endanwender erfolgt über Phishing-Mails bzw. Spear-Phishing-Mails, in denen ein Angreifer versucht, das Opfer zur Übermittlung von Zugangsdaten oder zum Öffnen von bösartigen Anhängen zu überreden. Um die Erfolgsrate der Angriffe zu erhöhen, wird typischerweise suggeriert, dass die Mail von einem erwarteten und vertrauenswürdigen Absender kommt. Dabei wird sich zu Nutze gemacht, dass, analog zu klassischen Briefen, die Kodierung und der Transport von Mail keinen eingebauten Schutz gegen eine Fälschung des Absenders haben.
Um einen gefälschten Absender zu entdecken, hat sich eine Kombination aus SPF, DKIM und DMARC etabliert. SPF erlaubt dabei zu verifizieren, dass Mails nur von für die Absenderdomain erlaubten IP-Adressen verschickt werden. Bei DKIM hingegen signiert der ausgehende Mailserver Mails kryptographisch und liefert damit eine überprüfbare Herkunft einer Mail. Mit DMARC wiederum wird der vom Mail-Client dargestellte Absender einer Mail mit der behaupteten Absenderdomain entweder aus der DKIM-Signatur oder aus dem SMTP-Dialog (bei SPF) abgeglichen. Während aber SPF Probleme bei der Weiterleitung von Mail durch Sammeldienste oder Mailinglisten verursacht, hat DKIM diese Probleme nicht und gilt daher als die robustere Basis zur Verifikation des Absenders. Darüberhinaus verspricht die kryptographische Signatur bei DKIM auch einen Schutz der Mail vor Modifikation während des Transports.
Diese Versprechen können in der Praxis jedoch nicht gehalten werden, weil DKIM in unsicherer Weise mit MIME interagiert. MIME ist ein Standard, der für praktisch alle heutigen Mails genutzt wird, um Umlaute und Attachments innerhalb der ursprünglichen Restriktionen von Mails abzubilden. Bedingt durch die Komplexität und Flexibilität von MIME ist es vielfach möglich, den dargestellten Inhalt von Mails signifikant zu ändern, ohne die DKIM-Signatur der Mail zu invalidieren. Ebenso kann es bedingt durch die Flexibilität und Abwärtskompatibilität beim Transport über SMTP passieren, dass aus Versehen eine bestehende DKIM-Signatur invalidiert wird, obwohl der eigentliche Inhalt der Mail gleich bleibt. Durch diese Fehler beim praktischen Einsatz von Kryptographie kann einer DKIM-Signatur nur bedingt bei der Erkennung von gefälschten Absendern oder Manipulation von Mails vertraut werden.
Zwar sind diese Probleme im DKIM-Standard prinzipiell erkannt und mögliche Vorkehrungen vorgeschlagen. Diese Vorschläge sind jedoch weder präzise genug noch verpflichtend, sodass sie in der Praxis überwiegend nicht bzw. nicht ausreichend umgesetzt werden. Entsprechend geben wir detaillierte Empfehlungen ab, unter welchen Bedingungen und Konfigurationen DKIM in der Praxis eingesetzt werden sollte.

Stichworte: DKIM, MIME, Kryptographie, Signaturen, Phishing

1. Einleitung

SPF, DKIM und DMARC werden als wesentliche Elemente bei der Erkennung von Mails mit gefälschten Absendern angesehen. Bei SPF (Sender Policy Framework, [1]) wird dazu mittels DNS-Einträgen festgelegt, von welchen IP-Adressen aus im Namen der Domain Mails verschickt werden dürfen. Bei DKIM (DomainKeys Identified Mail, [2]) wiederum signiert der für eine Mail verantwortliche Mailserver eine ausgehende Mail kryptographisch, wobei die Signatur mit einem im DNS hinterlegten Public Key verifiziert werden kann. Die Signatur schließt dabei wesentliche Bestandteile des

1 genua GmbH, Kirchheim

Mail-Headers wie auch den Inhalt der Mail ein und ermöglicht dadurch zu erkennen, wenn eine Mail signifikant modifiziert wurde.

Sowohl SPF wie auch DKIM betrachten dabei nicht den im Mail-Client dargestellten Absender: bei SPF wird der beim Mailtransport über SMTP angegebenen Absender (SMTP.MAILFROM) benutzt, bei DKIM hingegen die mit der Signatur verbundene Domain. Erst mit DMARC (Domain-based Message Authentication, Reporting and Conformance, [3]) wird überprüft, ob die von SPF bzw. DKIM gelieferte Domain zu dem im Mail-Client sichtbaren Absender aus dem Mail-Header (RFC822.From) passt. Dabei ist es ausreichend, dass entweder ein erfolgreiches SPF oder eine der gültigen DKIM-Signaturen die passende Domain haben. Gibt es keine Übereinstimmung, so wird je nach der im DNS hinterlegten DMARC-Policy die Mail abgelehnt, in Quarantäne verschoben oder trotzdem angenommen. Zusätzlich kann in der Policy auch ein Reporting erbeten werden, über welches Domaininhaber Verletzungen der DMARC-Policy detailliert nachvollziehen können.

SPF ist die am einfachsten zu nutzende Technologie, benötigt sie doch nur einen DNS-Eintrag, welcher die zum Versenden erlaubten IP-Adressen spezifiziert. Wir sehen daher für ca. 42% der untersuchten Mails[2] (45% der Absender-Domains) SPF-Einträge an der Domain. 6% dieser Mails sind jedoch für DMARC nicht nutzbar, da die Domain aus dem SMTP-Dialog nicht zum Mail-Header passt. Schwierigkeiten bereitet SPF auch beim Weiterleiten von Mails, wie bei Sammeldiensten von Mail-Providern oder bei Mailinglisten. Da die IP-Adressen der weiterleitenden Server nicht in der SPF-Policy der Domain enthalten sind, ist es in der Vergangenheit schon öfter zu Problemen bei der Zustellung eigentlich gutartiger Mails gekommen ([4]).

DKIM-Signaturen sind dagegen robuster. Durch ihr Design bleiben sie auch gültig, wenn Mails durch Mailinglisten oder Sammeldienste weitergeleitet werden und selbst bei einer Weiterleitung als Mail-Attachment. Aber im Gegensatz zu SPF sind zur Umsetzung neben DNS-Einträgen auch Änderungen an den für die Domain verantwortlichen Mailservern nötig, einschließlich eventuell der Installation neuer Software zum Signieren der Mails. Demzufolge haben nur 14% der untersuchten Mails (26% der Absender-Domains) eine DKIM-Signatur. 20% von diesen haben aber keine zum Absender im Mail-Header passende Signatur und sind dadurch für DMARC nicht verwendbar.

Die Aussagekraft von DKIM alleine ist jedoch begrenzt. Entsprechend dem Design ergibt sich der für die Validierung einer DKIM-Signatur benötigte DNS-Eintrag nur aus der Signatur selber. Damit ist es für eine bestimmte Mail nicht vorab klar, ob eine DKIM-Signatur existieren sollte oder nicht. Selbst wenn bisherige Mails eines Absenders eine DKIM-Signatur besessen haben, ist ein Fehlen der DKIM-Signatur kein solides Indiz für eine gefälschte Mail. Tatsächlich gibt es größere Absender, welche teilweise Mails mit DKIM-Signatur und teilweise ohne senden. Richtig nutzbar werden DKIM und SPF daher erst im Zusammenhang mit einer DMARC-Policy.

2 ca. 24.000 Mails aus 1.500 verschiedenen Domains, sowohl kundenspezifische Trouble-Tickets wie auch generische Anfragen an info@ oder sales@. Spam und Phishing ist bereits aussortiert. Zeitraum 9/2017 bis 8/2018.

Aber auch wenn für DMARC wie bei SPF nur ein einzelner DNS-Eintrag konfiguriert werden muss, hat DMARC mit 12% der untersuchten Mails (11% der Absender-Domains) eine nochmals geringere Verbreitung. Und man kann auch bei Fehlen einer Policy nicht einfach von einer Default-Policy ausgehen: würde man annehmen, dass alle SPF- bzw. DKIM-nutzenden Absender auch eine DMARC-Policy besitzen, dann würde es in 5% der entsprechenden Mails keine Übereinstimmung bezüglich der Domain aus der DKIM-Signatur bzw. dem Absender aus dem Mail-Dialog (bei SPF) geben. Ein beträchtlicher Teil gutartiger Mails würde damit als Problem klassifiziert werden.

Aus Sicht des Angreifers ist jedoch ein maximales Vertrauen des Empfängers in die Validität der Phishing-Mail erwünscht. Der sicherste Weg, dieses zu erreichen, wäre es, die Mail über die Infrastruktur des zu fälschenden Absenders zu versenden. Das ist aber typischerweise nur über eine Kompromittierung des Absenders oder dessen Infrastruktur machbar. Wir zeigen dagegen im Weiteren, wie ein Angreifer eine frühere Mail des zu fälschenden Absenders unter Beibehaltung der validen DKIM-Signatur so modifizieren kann, dass der dargestellte Inhalt signifikant verändert wird und unter Umständen sogar andere Anhänge möglich sind ([5]). Diese Mail kann dann über die Infrastruktur des Angreifers versendet werden und besteht, da mit einer validen und passenden DKIM-Signatur ausgestattet, trotzdem die Überprüfung mittels DMARC. Anschließend wird betrachtet, wie das Vertrauen in DKIM zusätzlich durch sporadische Fehler untergraben wird, bei denen beim Transport zwischen Mail-Servern eine gültige DKIM-Signatur trotz gleichbleibendem Inhalt der Mail invalidiert wird. In beiden Fällen werden Empfehlungen zum Einsatz von DKIM gegeben, welche die beschriebenen Sicherheits- und Vertrauensprobleme beseitigen.

2. Änderung des Inhalts bei Beibehaltung der Signatur

Bei DKIM wird durch den sendenden Mail-Server ein strukturiertes Header-Feld *DKIM-Signature* der Mail hinzugefügt. Dieses Feld beinhaltet im Parameter *b* eine kryptographische Signatur, welche die im Parameter *h* angebenen Header-Felder der Mail, einen Hash *bh* über den Body der Mail sowie andere Teile des *DKIM-Signature* Header-Feldes umfasst. Die Domain des Mail-Servers wird im Parameter *d* gegeben und über den Parameter *s* wird bestimmt, über welchem DNS-Eintrag der zur Validierung der Mail nötige öffentliche Schlüssel ermittelt werden kann. Im Falle der in Abbildung 1 gezeigten DKIM-Signatur würde der öffentliche Schlüssel als TXT-Eintrag unter dem Namen *20161025._domainkey.google.com* zu finden sein, wie in Abbildung 2 zu sehen.

Wie bereits aus früheren Untersuchungen ([6]) bekannt, können leichte Manipulationen an kritischen Feldern im Mail-Header als auch die Definitionen zusätzlicher Felder die Interpretation und damit die Darstellung einer Mail durch den Mail-Client signifikant verändern. Der DKIM-Standard ([2]) bietet theoretisch einen Schutz gegen derartige Manipulationen: so können gegen eine Manipulation vorhandener Header-Felder diese in die Signatur eingeschlossen werden. Ebenso können zur Verhinderung von zusätzlichen Definitionen fehlende Felder zur Signatur

```
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;
  d=google.com; s=20161025;
  h=mime-version:reply-to:auto-submitted:message-id:date:subject:from
  :to;
  bh=kWpZXyyQS+AccSQ2p0JxC7ykVQYhPTw6AmYbIHqsOdU=;
  b=tmyGeak...
```

Abbildung 1: Beispiel für einen *DKIM-Signature* Mail-Header

```
$ dig txt 20161025._domainkey.google.com
...
20161025._domainkey.google.com. 300 IN  TXT "k=rsa; p=MIIBIjANBgk..."
```

Abbildung 2: Nachschlagen des öffentliches Schlüssels zu einer DKIM-Signatur

hinzugefügt werden. Jedoch liefert der Standard nur rudimentäre Empfehlungen, welche Felder von der Signatur umfasst sein sollten. Weder werden alle kritischen Felder erwähnt, noch wird darauf hingewiesen, dass diese zwingend vor zusätzlichen Definition zu schützen sind. Und auch in keinem der im Standard enthaltenen Beispiele (Kapitel 3.5, A.2, A.3) werden die kritischen Felder ausreichend geschützt. Allerdings zeigen die Ausführungen in Kapitel 5.4 („Determine the Header Fields to Sign“) und die Überlegungen zur Sicherheit in Kapitel 8.15 („Attacks Involving Extra Header Fields“), dass sich die Autoren der prinzipiellen Problematik durchaus bewusst waren.

Durch Ausnutzen von Modifikationen ungeschützter Felder oder Definitionen zusätzlicher Felder kann es einem Angreifer gelingen, den dargestellten Inhalt der Mail signifikant zu ändern, ohne die Signatur ungültig zu machen. Ein Beispiel dafür ist in Abbildung 3 zu sehen: die ursprüngliche Mail befindet sich hier in den Zeilen 1-12, die vom Angreifer hinzugefügten zusätzlichen Felder in A und B. Der Parameter *h* in der DKIM-Signatur gibt zwar an, das die Felder *Subject* und *Content-Type* von der Signatur einbezogen werden, allerdings nur in jeweils einer Instanz und zwar der untersten. Ein Hinzufügen neuer Felder der gleichen Art oberhalb, wie in den Zeilen A und B getan, hat damit keinen Einfluß auf die Valididierung der Signatur. Wie bereits in [6] beschrieben definieren die Mail-Standards kein eindeutiges Verhalten bei der Mehrfachdefinition von singulären Header-Feldern, aber in der Praxis benutzen viele Mail-Clients die oberste Definition der Felder und ignorieren den Rest. Zur Darstellung der Mail werden dabei also häufig genau die Felder genutzt, welche nicht von der Signatur umfasst sind. Im konkreten Fall führen die Änderungen bei der Darstellung im populären Mail-Client Thunderbird zu der Anzeige des *Subject*'s aus Zeile A und Nutzung des *Content-Type*'s aus Zeile B, welcher mangels passendem MIME-Boundary „bar“ im Body der Mail zur Anzeige eines leeren Inhalts führt. Wie in Abbildung 4 zu sehen, zeigt das Thunderbird-Plugin zur DKIM-Verifikation[3] dabei wie erwartet weiterhin eine gültige Signatur an.

Deutlich kritischer ist das Problem, wenn der ursprüngliche Sender in der DKIM-Signatur den Parameter *l* verwendet hat. Während sich ohne diesen die Signatur über

3 Thunderbird 60.2.1, DKIM Verifier 2.0.0

```
A  Subject: Urgent Update at http://foo
B  Content-type: multipart/mixed; boundary=bar
1  DKIM-Signature: v=1; h=from:to:cc:subject:content-type; ...
2  From: <...>
3  To: <...>
4  Subject: 20170920:1755 - good
5  Content-type: multipart/mixed; boundary=foo
6  Date: Wed, 20 Sep 2017 17:55:18 +0200
7
8  --foo
9  Content-type: text/plain
10
11 some text
12 --foo--
```

Abbildung 3: Manipulation von Subject und Content-Type

Abbildung 4: Ergebnis der Manipulation in Thunderbird – rechts die originale Mail und links die modifizierte mit anderem Subject, leerem Inhalt aber gültiger Signatur

den kompletten Body erstreckt, kann mittels *l* der Umfang des zu betrachtenden Body beschränkt werden. Dies ist gedacht um zum Beispiel von Virenscannern generierte Texte an die Nachricht anhängen zu können, ohne die Signatur ungültig zu machen. Wie in Abbildung 6 zu sehen, kann es jedoch auch genutzt werden, um die Mail komplett zu verändern, einschließlich dem Hinzufügen von Attachments. Dieses Problem war den Autoren des DKIM-Standards offensichtlich bewusst, wie explizite Hinweise auf die Gefährlichkeit in Kapitel 8.2 („Misuse of Body Length Limits“) zeigen. Nichtsdestotrotz ist dieser Parameter Teil des Standards geworden und wird in der Praxis genutzt.

Die originale Mail ist in Abbildung 5 in den Zeilen 1-16. Wie in Zeile 3 zu sehen, umfasst die Signatur einige, aber nicht alle, wichtigen Felder und verhindert vor allem auch keine zusätzlichen Definitionen. Entsprechend ist es möglich, dieser Felder in den Zeilen A-C neu zu definieren und auch passend zur Neudefinition des Multipart-Boundary in Zeile C einen komplett neuen Inhalt einschließlich eines Anhanges für die Mail in Zeilen D-L zu erzeugen. Der bisherige Mail-Body in Zeilen 13-16 wird bei der Anzeige als MIME-Preamble interpretiert und ignoriert.

Dieses Beispiel beruht auf einer tatsächlich von DHL bekommenen Mail. Kritisch daran ist nicht nur, dass die Mail komplett verändert werden konnte und die Signatur

```
A    Subject: Zollpapiere zur Sendung 123456789
B    Date: Mon, 28 Jan 2019 07:48:54 +0000
C    Content-Type: multipart/mixed; boundary="foo"
     ...
1    DKIM-Signature:   v=1; a=rsa-sha256; c=relaxed/relaxed;
2           d=dpdhl.com; l=87336; s=20140901; t=1496389754;
3           h=from:to:subject:date:message-id:mime-version;
4           bh=3lnyE3...;
5           b=fpw+kBAdLa4J...
6    ...
7    From: "FHD, Zollrechnung DE" <zollrechnung-de@dpdhl.com>
8    To:   ...
9    Subject:    Zollpapiere zur Sendung 5689501322
10   ...
11   Content-Type:     multipart/mixed;
12         boundary="_005_ED90D8841FCE174D9988646A02083ECF314A044ACZCHOWS1342prgd_"
13   ...
14   Der originale Inhalt der Mail
15   ...
16   --_005_ED90D8841FCE174D9988646A02083ECF314A044ACZCHOWS1342prgd_--
D    --foo
E    Content-Type: text/plain
F
G    Bitte ausfuellen und zurueckschicken.
H    --foo
I    Content-Type: application/octet-stream; name="Zoll.pdf"
J
K    ...
L    --foo--
```

Abbildung 5: Ändern von dargestelltem Inhalt und Attachments

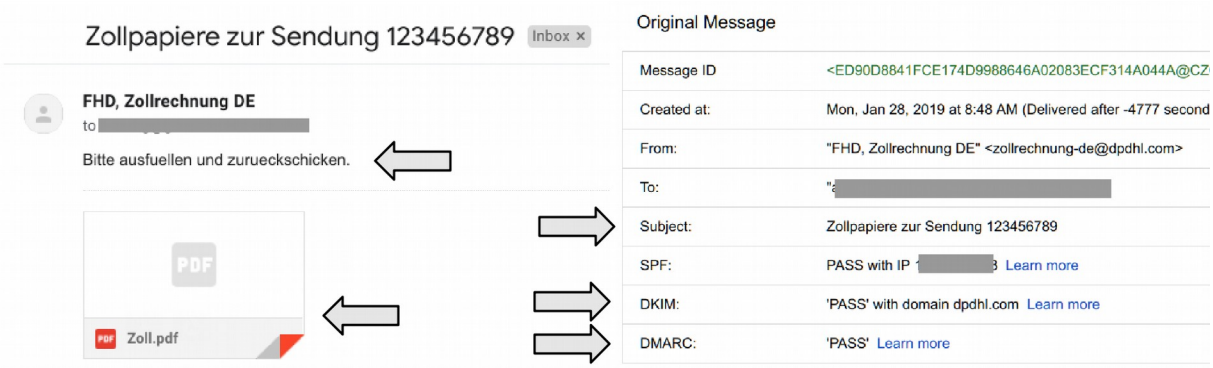

Abbildung 6: Darstellung in GMail - Subject und Inhalt sind verändert und neue Anhänge hinzugefügt. DKIM und DMARC sind aber weiterhin PASS.

gültig bleibt. Dazu kommt, dass DHL ein typisches Ziel von Absender-Spoofing in Phishing-Mails ist und zum Schutz davor eine restriktive DMARC-Policy hat. Da die DKIM-Signatur aber trotz Manipulation an der Mail weiterhin gültig ist und es bei DMARC ausreicht, dass entweder DKIM oder SPF passen, ist es einem Angreifer auch Jahre, nachdem die ursprüngliche Mail verschickt wurde, noch möglich, diese als Phishing-Mail mit neuem Betreff, Datum, Inhalt und Anhängen versehen über die eigene Infrastruktur zu verschicken, ohne als Phishing eingestuft zu werden.

Eine Auswertung der DKIM-Signaturen in den untersuchten Mails zeigte, dass in 98% der Signaturen kein Schutz gegen Hinzufügen eines neuen *Subject*'s, *Content-Transfer-Encoding*- und *From*-Feldes und bei 97% kein Schutz gegen Hinzufügen von *Content-Type* bestand. Damit ist bei nahezu allen Mails zumindest eine Änderung des

```
 1   DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/simple;
 2    d=example.com; s=m1; bh=...; b=...;
 3    h=from:from:to:to:date:date:subject:subject
 4    :content-type:content-type:content-transfer-encoding:cc
 5   From: ...
 6   To: ...
 7   Subject: ...
 8   Date: ...
 9   Content-Type: multipart/mixed; boundary=foo
10
11   --foo
12   ...
13   --foo--
```

Abbildung 7: Beispiel für eine DKIM-Signatur, welche mit passendem Parameter *h* kritische Manipulationen des Mail-Headers ausreichend verhindert

Betreffs und ein Verstecken des originalen Inhalts möglich, wie in Abbildungen 3 und 4 gezeigt. 8% aller Mails signierten sogar nur einen Teil des Mail-Body durch Nutzung des Parameter *l* und hatten außerdem keinen Schutz gegen Hinzufügen von *Content-Type*, was zusätzlich Änderungen des Inhalts und Hinzufügen von Anhängen wie in Abbildungen 5 und 6 ermöglicht.

3. Best Practices zur Erstellung und Überprüfung von DKIM-Signaturen

Wenn die Problematik erstmal erkannt ist, ergibt sich eine offensichtliche Lösung: der Parameter *h* in *DKIM-Signature* sollte durch den signierenden Mail-Server so konstruiert werden, dass alle für die Darstellung der Mail signifikanten Felder von der Signatur umfasst sind und auch zusätzliche Definitionen verhindert werden. Diese kritischen Felder sind zumindest *From*, *Subject*, *Content-Type* und *Content-Transfer-Encoding*, wobei zusätzlich ein Schutz von *Date*, *To* und *Cc* zu empfehlen ist. Eine entsprechend abgesicherte Mail ist in Abbildung 7 zu sehen: die im Mail-Header enthaltenen kritischen Felder sind offensichtlich geschützt gegen Modifikation und durch das Hinzufügen zusätzlicher Instanzen der Feldnamen zum Parameter *h* werden auch zusätzliche Definition der betreffenden Felder sicher verhindert. Beim Empfang einer Mail wiederum muss geprüft werden, ob es kritische Felder im Header gibt, die nicht von der Signatur umfasst sind. Ist das der Fall, kann die DKIM-Signatur nicht als vertrauenswürdige Basis für Entscheidungen angesehen werden. Im Zusammenhang mit DMARC bedeutet dies, dass eine positive Entscheidung basierend auf einer derartigen unsicherer Signatur ebenfalls als unsicher anzusehen ist.

Zusätzlich zur Härtung von Parameter *h* sollte die Nutzung des Parameters *l* in der Signatur vermieden werden. Wird dieser aber in einer empfangenen Mail verwendet und ist der Mail-Body länger als die in *l* angegebene Länge, so darf dieser überschüssige Teil nicht in die Anzeige der Mail einbezogen werden und sollte eventuell direkt beim Empfang im DKIM-verifizierenden Mail-Server entfernt werden. Allerdings sind in der Praxis auch Mails beobachtet worden, bei denen offensichtlich durch Fehlverhalten des sendenden Mail-Servers der Parameter *l* nur

einen kleinen Teil der eigentlichen Mail umfasste und ein Abschneiden des über *l* hinausgehenden Teils damit zu einer fehlerhaften Darstellung der Mail führen würde.

4. Brechen der Signature beim Transport

Neben der Möglichkeit, eine Mail gezielt zu manipulieren, ohne die DKIM-Signatur zu invalidieren, wird das Vertrauen in die Robustheit von DKIM zusätzlich durch sporadisches Brechen von Signaturen während des Transports der Mail geschwächt. Die Ursache dafür ist ebenfalls in einer unerwünschten Interaktion zwischen DKIM und MIME zu finden.

Ursprünglich waren alle Mails nur ASCII mit einer Zeilenlänge von maximal 1000 Zeichen. Mit den MIME-Standards wurde eine Abbildung von strukturierten und potentiell binären Inhalten auf ASCII definiert, die es ermöglichte, strukturierte Mails mit binären Attachments sowie beliebigen Zeichensätzen zu übertragen. Damit nicht jedesmal eine Kodierung aller Inhalte für den Transport und Dekodierung am Ziel vorgenommen werden muss, wurde mit 8BITMIME eine Erweiterung für den Mailtransport SMTP definiert, bei der bestimmte Inhalte unkodiert übertragen werden können ([7]). Um eine Kompatibilität zwischen verschiedenen Systemen zu gewährleisten, weisst der empfangende Mailserver die Unterstützung für diese Erweiterung explizit aus. Bietet der empfangende Mailserver kein 8BITMIME an, so muss der Sender eine Mail mit binären Daten passend nach ASCII umkodieren, indem 8bit-kodierte Teile nach Quoted-Printable oder Base64 umgesetzt werden. Unsere Untersuchungen deuten dabei darauf hin, dass ca. 20% der öffentlichen Mailserver keine Unterstützung für 8BITMIME ausweisen[4].

Für den Body-Hash (*bh*) unterstützt der DKIM-Standard eine Kanonisierung des Inhaltes vor der Berechnung des Hashes über den Parameter *c*, um einer Invalidierung der Signatur durch transportbedingte Änderungen an der Mail vorzubeugen. Dabei wird bei *simple* praktisch keine Kanonisierung vorgenommen, bei *relaxed* werden hingegen verschiedene Variationen von Leerstellen (Leerzeichen, Tabulatoren, Leerzeilen) normalisiert. Nicht berücksichtigt wird hingegen eine potentielle nötige Transformation des Mail-Body in eine ASCII-kompatible Kodierung. Konträr zu der treibenden Idee hinter der Kanonisierung wird im DKIM-Standard davon ausgegangen, dass die Mail während des Transports nicht mehr umkodiert wird, das heisst bereits in der finalen Kodierung vorliegt. Wird doch eine Umkodierung der Mail beim Transport erforderlich, so passt die bisherige DKIM-Signatur nicht mehr auf die neue Kodierung der Mail, auch wenn sich der eigentliche Inhalt der Mail nicht geändert hat. Abbildung 8 veranschaulicht dieses Problem. Obwohl im Standard in Kapitel 5.3 explizit auf dieses Problem hingewiesen und eine Konvertierung in eine ASCII-kompatible Kodierung vor der Berechung der Signatur empfohlen wird, ist diese nicht explizit gefordert. Entsprechend finden sich in der Praxis genügend Fälle, wo der Sender sich dieser Probleme nicht bewusst ist und die DKIM-Signatur durch den Transport für einzelne Empfänger entsprechend ungültig wird.

4 Untersuchungen in 09/2017. Zu diesem Zeitpunkt unterstützten weder GMX.de noch AOL.com 8BITMIME, tun dieses aber in 01/2019.

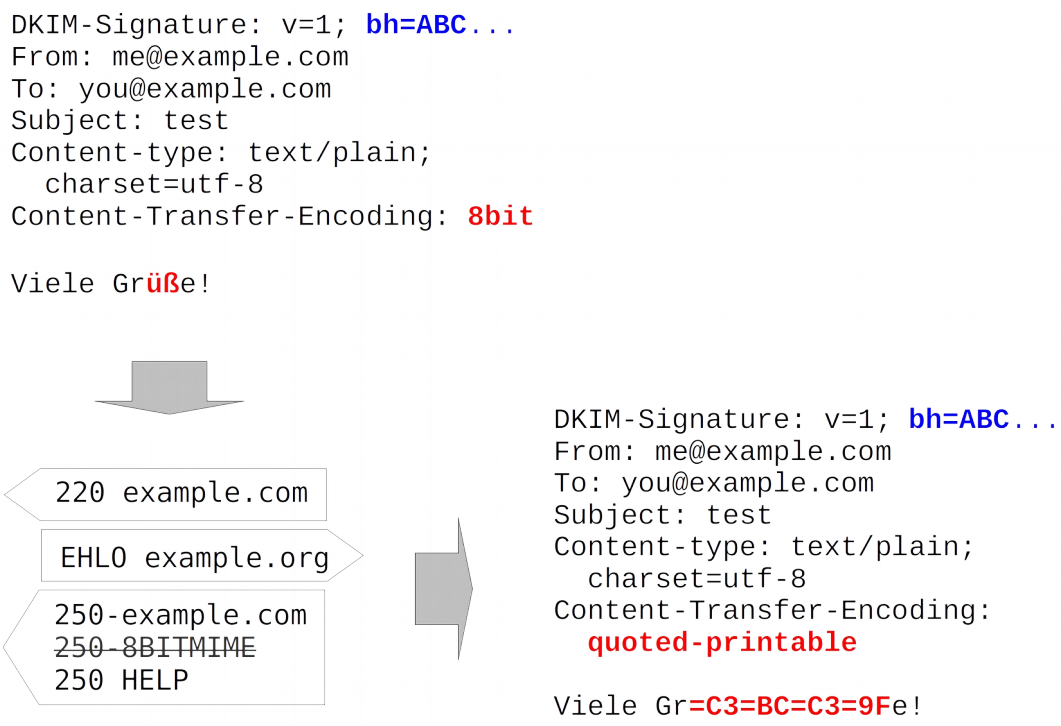

Abbildung 8: Wenn der empfangende Mail-Server kein 8BITMIME unterstützt, muss die Mail passend umkodiert werden. Dieses invalidiert die DKIM-Signatur.

5. Best Practices, um ein Brechen der Mail beim Transport zu verhindern

In der Praxis gibt es eine einfache aber nicht wirklich zufriedenstellende Lösung für dieses Problem. Gerade in größeren Infrastrukturen erfolgt die Zustellung einer Mail vom Sender zum Empfänger typischerweise über eine Kette von Mail-Servern, wobei ein einzelner Server immer nur die Möglichkeiten des nächsten Servers in der Kette ermitteln kann, nicht aber der gesamten Kette bis zum Ziel. Daher kann aus Sicht eines signierenden Mail-Servers nicht festgestellt werden, ob der Rest der Zustellungskette 8BITMIME unterstützt. Bei einer korrekten Umsetzung bleibt daher nichts anderes übrig, als die Mail vorab in eine ASCII-kompatible Darstellung umzuwandeln. Dieses kann der Einfachheit halber auch dem Sender überlassen werden, indem der signierende Mail-Server selber keine Unterstützung für 8BITMIME annonciert. Ein flächendeckender und korrekter Einsatz von DKIM bedeutet daher quasi das Aus für einen effizienteren Transport von Mails mittels 8BITMIME.

6. Zusammenfassung

Die Nutzung kryptographischer Signaturen in DKIM suggeriert eine Unfälschbarkeit, die in der Praxis nicht existiert. Ursache dafür sind unerwünschte Interaktionen mit MIME, einem Standard, dessen Unzulänglichkeiten und Schwächen auch den Autoren von DKIM offensichtlich bekannt waren. Trotzdem wurde es versäumt, durch klare Regeln zur Signierung das Missbrauchspotential einzudämmen sowie einen robusten Transport signierter Mails über vorhandene Infrastrukturen zu gewährleisten.

Eine sichere Nutzung von DKIM erlauben erst die in Kapitel 3 vorgestellten Best Practices. Diese ermöglichen es, eine DKIM-Signatur so aufzusetzen, dass ein Sender eine nachträgliche aber unentdeckte Manipulation der Mail sicher verhindern, als auch das ein Empfänger die Manipulationswahrscheinlichkeit bei einer empfangenen signierten Mail beurteilen kann. Best Practices zum Schutz vor einer Invalidierung der DKIM-Signatur beim Transport wurden in Kapitel 5 aufgezeigt. Zwar funktionieren diese zuverlässig, sind aber nicht wirklich zufriedenstellend, da sie praktisch die optimierte 8bit-Übertragung von Mails abschalten. Dieses ist jedoch nötig bedingt durch das Design von DKIM und SMTP. Verbesserungen wären nur durch tiefgreifende Veränderungen an einem dieser Standards möglich.

7. Literatur

[1] Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1
RFC 7372, https://tools.ietf.org/html/rfc7208

[2] DomainKeys Identified Mail (DKIM) Signatures
RFC 6376, https://tools.ietf.org/html/rfc6376

[3] Domain-based Message Authentication, Reporting, and Conformance (DMARC)
RFC 7489, https://tools.ietf.org/html/rfc7489

[4] gmx.de und web.de haben Mail-Rejects durch SPF
https://www.heinlein-support.de/blog/news/gmx-de-und-web-de-haben-mail-rejects-durch-spf/

[5] Breaking DKIM - on Purpose and by Chance
https://noxxi.de/research/breaking-dkim-on-purpose-and-by-chance.html

[6] Dubious MIME - Conflicting Multipart Boundaries
https://noxxi.de/research/mime-conflicting-boundary.html

[7] SMTP Service Extension for 8-bit MIME Transport
RFC 6152, https://tools.ietf.org/html/rfc6152