Sichere Datenübertragung mit IDL.KONSIS und IDL.XLSLINK

14.09.2021

Inhaltsverzeichnis

1 Vorwort

Das Handbuch richtet sich sowohl an IT-Administrator*innen als auch an die technisch interessierten Anwender*innen der Applikationen IDL.KONSIS und IDL.XLSLINK.

Im vorliegenden Handbuch finden Sie sicherheitsrelevante Information, so zum Beispiel über den HTTPS-Verbindungsaufbau. Weiterhin werden digitale Signaturen und Sicherheitszertifikate erläutert. Verweise zu IDL.KONSIS sind farblich in **Grün** hervorgehoben.

1 Fachlicher und Technischer Support

Bitte fragen Sie bei aufkommenden Verständnisschwierigkeiten und bei auftretenden Problemen nach Unterstützung. Hilfestellung bekommen Sie durch den Support der IDL.

Telefon (Anruf ins deutsche Festnetz):

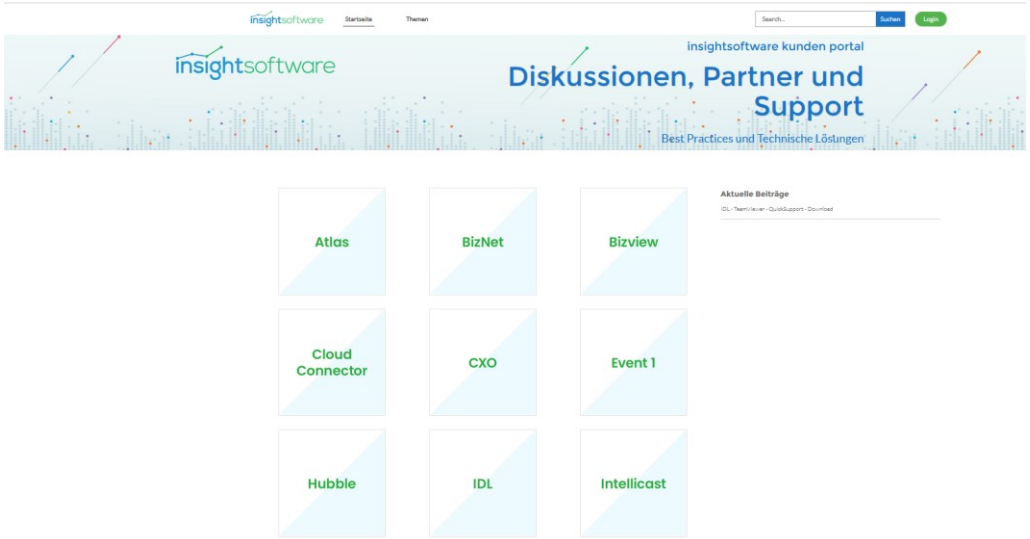
- Fachlicher Support +49 4102-4785-10
- Technischer Support +49 4102-4785-11

E-Mail (fachlicher und technischer Support): idlsupport@insightsoftware.com

Bei Nutzung dieser E-Mail wird automatisch ein neues Ticket zu Ihrer Anfrage erstellt. Sie bekommen eine Eingangsbestätigung mit einer Ticket-Nummer. Bei schriftlichen Rückfragen oder auch beim Versenden weiterer Informationen zum Ticket nutzen Sie bitte **immer** die Antwortfunktion in Ihrem E-Mail-Programm. Nur so kann sichergestellt werden, dass Ihre Fragen und weiteren Informationen eindeutig Ihrem Ticket zugeordnet werden können.

Remote-Zugang

Mit dem **IDL-QuickSupport** von TeamViewer können wir uns mit Ihrem Computer verbinden und Ihnen schnell und effizient Hilfestellung leisten. IDL-QuickSupport finden Sie im Kundenportal von Insightsoftware unter der URL: https://help.insightsoftware.com/s/?language=de



Für den Download vom IDL QuickSupport benötigen Sie keinen Login. Eine Installation ist nicht erforderlich, Sie müssen nur das Programm starten und uns die Kenndaten bekanntgeben.

Alternativ ist **Microsoft Teams** möglich, um Ihnen eine schnelle Remote-Hilfestellung zu geben.

Kunden-Community

Loggen Sie sich in die Kunden-Community ein, um Wissensartikel zu lesen, um Installationssoftware herunterzuladen, um einen neuen Support-Fall zu eröffnen oder um ihre offenen (oder auch ihre bereits geschlossenen) Anfragen einzusehen. Wenn Sie noch keinen Login für die Kunden-Community von Insightsoftware haben sollten, dann wenden Sie sich bitte vertrauensvoll an den Technischen Support der IDL.

2 HTTPS (Hypertext Transfer Protocol Secure)

Um Daten im Netzwerk sicher zu übertragen, wird oft das Protokoll HTTPS benutzt.

HTTPS wird hauptsächlich verwendet, um Webseiten und dynamische Inhalte einer Webseite in einem Internet-Browser zu laden. Es ist jedoch nicht darauf beschränkt. HTTPS kann auch in der Kommunikation zwischen einem Application Server und einem Client verwendet werden.

Die Sicherheit der Datenübertragung mit HTTPS wird durch eine Authentifizierung des Servers mit einem SSL-Zertifikat und außerdem durch eine abhörsichere Transportverschlüsselung erreicht.

Der IDL Application Server kommuniziert mit den beiden IDL Client-Anwendungen IDL.KONSIS und IDL.XLSLINK über das Protokoll HTTPS.

Was geschieht bei einer HTTPS-Verbindung im Einzelnen?

Ein Internetbrowser fragt eine https-Verbindung zur URL https://www.insightsoftware.com an.

Die Bezeichnung https wird in der URL[1] immer dann angezeigt, wenn eine Website durch ein Sicherheitszertifikat[2] abgesichert ist.

Der Web Server, auf dem die Domain www.insightsoftware.com gehostet ist, authentisiert sich gegenüber dem Internetbrowser mit einem Sicherheitszertifikat. Der Internetbrowser überprüft nun, ob das vom Webserver gesendete Zertifikat vertrauenswürdig ist. Ist das Zertifikat vertrauenswürdig, sendet der Internetbrowser eine Nachricht an den Server und bestätigt damit, dass der Internetbrowser den Aussteller des Zertifikats erkennt und ihm vertraut. Der Internetbrowser überprüft die Vertrauenswürdigkeit der angefragten Verbindung anhand einer Liste mit öffentlichen Schlüsseln. Diese stammen von Root-Zertifizierungsstellen. Jeder Internetbrowser kann auf so eine Liste mit öffentlichen Schlüsseln zugreifen.

Ist diese Überprüfung der Vertrauenswürdigkeit negativ, dann zeigt der Internetbrowser eine Warnungsmeldung an, dass die Verbindung nicht vertrauenswürdig ist. Der Anwender kann dann mitunter[3] selbst entscheiden, wie in diesem Fall weiter verfahren werden soll.

Ist diese Überprüfung der Vertrauenswürdigkeit im Internetbrowser positiv, dann wird in der Adressleiste direkt vor der URL ein kleines Vorhängeschloss angezeigt. Alle Angaben zum Zertifikat wie der Aussteller, der Inhaber der Domain, der zugehörige öffentliche Schlüssel, der Fingerabdruck des Zertifikates, die Signatur, das Signaturverfahren, aber auch die Gültigkeitsdauer können durch Klicken auf das Vorhängeschloss angezeigt werden.

[1] Das Kürzel URL steht für „Uniform Ressource Locator“ und identifiziert eine Website über das Kommunikationsprotokoll (i.d.R. http; https) und den zugehörigen Ort.

[2] Der Aussteller eines Zertifikates ist i.d.R. eine Zertifizierungsstelle für digitale Zertifikate. Diese überprüft bei einer Zertifizierungsanfrage die Inhaberschaft des öffentlichen Schlüssels und signiert bei erfolgreicher Überprüfung das angefragte Sicherheitszertifikat. Der Anfrager ist oft ein Betreiber des Servers, eine Person oder eine Organisation.

[3] Dies hängt sehr stark von den Richtlinien ihres Unternehmens ab, wie ein Browser auf nicht vertrauenswürdige Verbindungen reagieren soll. Manche Unternehmen erlauben keinen Verbindungsaufbau zu nicht vertrauenswürdigen Adressen.

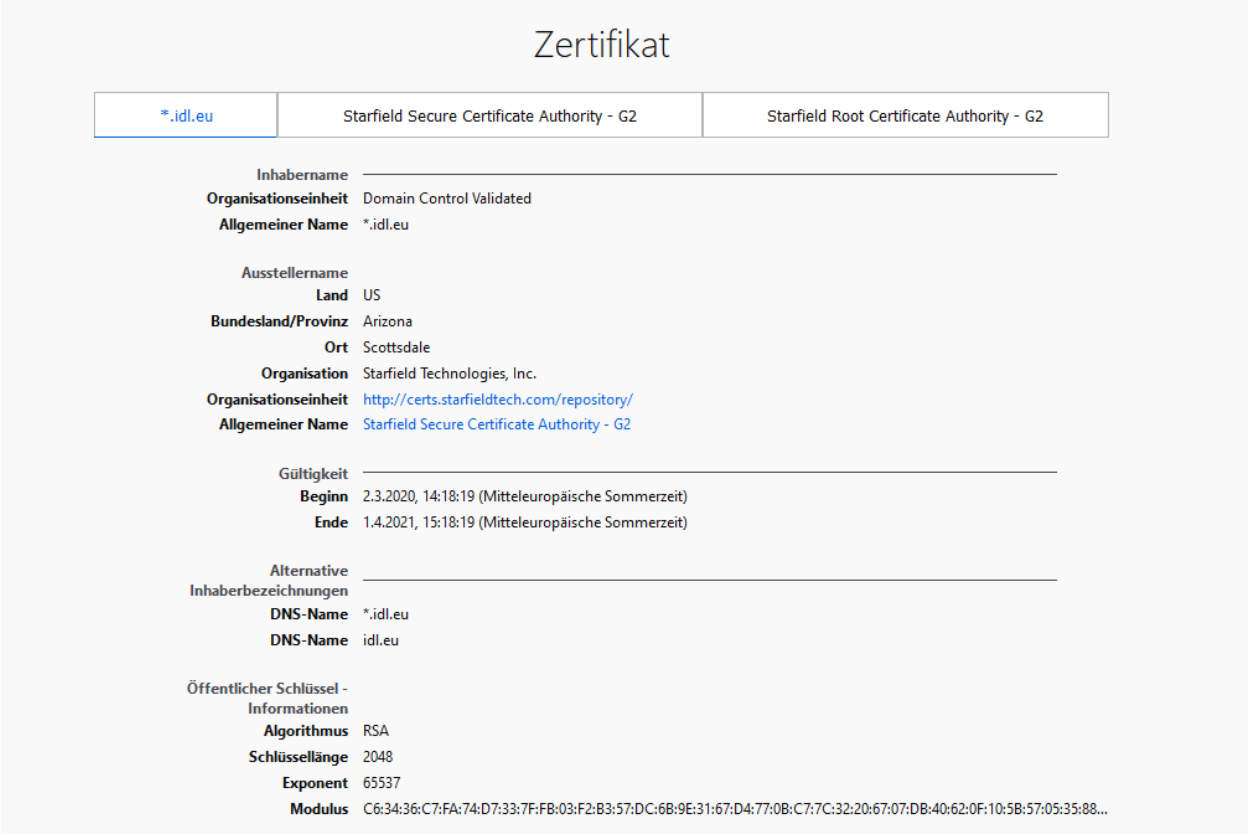


*Abbildung 1 Zertifikat für *.idl.eu mit einigen erweiterten Angaben: Gültigkeit, Aussteller, Öffentlicher Schlüssel, …*

Verschlüsselte Sitzung einleiten

Wenn die https-Verbindung weiter bestehen bleibt, dann handeln nun der Internetbrowser und der Webserver ein Verfahren aus, um eine verschlüsselte Sitzung einzuleiten. Damit wird eine sichere Verbindung gestartet.

Bevor eine verschlüsselte Kommunikation zwischen zwei Partnern erfolgen kann, wird zunächst eine Chiffren-Sammlung (Cipher Suite[4]) zwischen den Teilnehmern ausgehandelt. Dazu schickt der Internetbrowser (zukünftig als Client bezeichnet) eine Liste der möglichen Cipher Suiten an den Server. Der Webserver wiederum selektiert aus diesem Angebot die sicherste Cipher Suite, die er ebenfalls beherrscht. Je nach verwendeter Cipher Suite ergeben sich Abweichungen bezüglich der Sicherheit. Nach dem Aushandeln der Cipher Suite einigen sich Client und Server auf einen Verschlüsselungscode, den sogenannten Sitzungsschlüssel. Entweder schickt der Client dem Server eine mit dem öffentlichen Schlüssel verschlüsselte geheime Zufallszahl, oder die beiden Parteien (Client und Server) berechnen ein gemeinsames Geheimnis, zum Beispiel über das sogenannte Diffie Hellman Schlüsselaustausch-Verfahren. Aus dem Geheimnis wird der Sitzungsschlüssel abgeleitet.

Der Sitzungsschlüssel wird in der Folge für diese eine Sitzung benutzt, um alle Nachrichten der Verbindung zwischen Client und Server zu verschlüsseln. Nachdem Client und Server der Sitzungsschlüssel bekannt sind, fließen zwischen Client und Server verschlüsselte Informationspakete, die der Empfänger wieder entschlüsselt und zusammensetzt.

Außerdem wird der Sitzungsschlüssel benutzt, um die Nachricht durch einen Message Authentication Code (MAC) abzusichern. Dies geschieht zum Schutz der Integrität. Wird die bestehende

[4] Eine Cipher Suite ist eine standardisierte Sammlung kryptographischer Verfahren zum Schlüsselaustausch und zur Verschlüsselung, aber auch zur Überprüfung der Authentizität, Integrität und Verbindlichkeit der gesendeten Daten.

Sitzung unterbrochen und neu aufgebaut, wiederholt sich der gesamte Prozess, der Webserver bzw. Applikationsserver authentifiziert sich gegenüber dem Client erneut mit dem Zertifikat. Es wird nach erfolgreicher Überprüfung eine neue Sitzung aufgebaut und ein neuer Sitzungsschlüssel erzeugt.

Nachfolgend sind einige Sicherheitsaspekte zusammengefaßt:

- **Sicherheit durch Validierung der Authentizität**
- **Sicherheit durch Verwendung aktueller Protokolle auf Client und Server**
- **Sicherheit durch Integrität der Nutzdaten**

Sicherheit durch Validierung der Authentizität

Damit eine Verbindung wirklich sicher ist, genügt die Verwendung aktueller Protokolle und eine reine Verschlüsselung nicht. Schließlich muss auch sichergestellt sein, dass eine Webseite vom richtigen Anbieter kommt. Die Authentizität wird durch sogenannte Sicherheitszertifikate gewährleistet, die autorisierte Zertifizierungsstellen (CA) innerhalb einer Public-Key-Infrastruktur (PKI) ausstellen. Internetbrowser prüfen anhand der Zertifikate die Vertrauenswürdigkeit und lehnen eine sichere Verbindung ab, wenn diese Browser dem Zertifikat nicht vertrauen. Für die Prüfung der Authentizität haben Internetbrowser Listen von vertrauenswürdigen Rootzertifikaten, die regelmäßig aktualisiert werden. Erst wenn die Prüfung erfolgreich war, erscheint bei den Browsern in der Adresszeile das geschlossene Sicherheitsschloss neben der URL. Mitunter entscheidet der Benutzer selbst, ob er das Sicherheitszertifikat in dieser HTTPS Verbindung akzeptiert und ihm vertraut. Das bedeutet aber, wenn ein Angreifer in der Lage ist, die Validierung (Prüfung) des Browsers zu behindern, kann er den Verbindungsaufbau manipulieren und sich womöglich als Man-in-the-Middle in die Verbindung zwischen Client und Webserver einklinken.

Sicherheit durch Verwendung aktueller Protokolle auf Client und Server

Um sichere HTTPS-Verbindungen zu gewährleisten, sollte man versuchen, sowohl auf Server- als auch auf Clientseite immer aktuelle Sicherheitsprotokolle zu verwenden. Für eine HTTPS-Verbindung wird das hybride Verschlüsselungsprotokoll TLS verwendet. Für die Verschlüsselung werden bei den TLS-Versionen verschiedene Verschlüsselungsverfahren benutzt, die sich sehr stark in ihrer kryptographischen Leistungsfähigkeit unterscheiden. Die unterschiedlichen Verfahren werden durch verschiedene Cipher Suiten bereitgestellt.

Vereinfacht ausgedrückt besteht eine Cipher Suite aus den folgenden Elementen:

- Eine asymmetrische Verschlüsselung, zum sicheren Austausch des Schlüssels zwischen Client und Server und zur Authentifikation
- Ein symmetrisches Verfahren für die Verschlüsselung der eigentlichen Kommunikation
- Eine Hashfunktion, welche die Integrität der Daten sicherstellt

Die Cipher Suite ECDHE-RSA-AES256-GCM-SHA384 benutzt folgende Elemente:

Zum sicheren Austausch des Schlüssels zwischen Client und Server wird auf das Diffie-Hellman Schlüsselaustauschverfahren auf Basis elliptischer Kurven gesetzt, zur Authentifikation wird RSA verwendet, für die Verschlüsselung der Kommunikation wird AES (256 Bit) mit GCM eingesetzt. Die Integrität der Daten wird mit SHA384 (SHA-2) sichergestellt. SHA384 verwendet 64-Bit-Wörter und 1024-Bit-Nachrichtenblöcke und verschlüsselt in 80 Runden.

Sichere HTTPS-Verbindungen werden aktuell bei Verwendung von TLS in der Version 1.2 und 1.3 gewährleistet. Diese werden weltweit von vielen Webservern und Standardbrowsern unterstützt.

In IDL.KONSIS wird TLS 1.2 für die Kommunikation zwischen IDL.KONSIS-Client und dem IDL Application Server benutzt.

Die Sicherheit wird durch Verwendung von hochkomplexen, kryptografischen Schlüsseln mit sehr großen Schlüssellängen erhöht. Ein Angriff durch reines Ausprobieren (Brut Force) eines möglichen Schlüssels steht dabei in direkter Abhängigkeit zur verwendeten Schlüssellänge.

Sicherheit durch Integrität der Nutzdaten

Die Integrität einer zu übertragenden Nachricht lässt sich mittels **Message Authentication Code** (MAC) prüfen. Der MAC dient dazu, Gewissheit über den Ursprung von Daten oder Nachrichten zu erhalten und ihre Integrität zu überprüfen. MAC-Algorithmen erfordern genau zwei Eingabeparameter, erstens die zu schützenden Daten und zweitens einen Sitzungsschlüssel, und berechnen aus beidem eine Prüfsumme, den Message Authentication Code. Der Absender berechnet so den MAC und sendet die Nachricht sowie den MAC an den Empfänger. Dieser berechnet ebenfalls den MAC und vergleicht nun den berechneten MAC mit dem empfangenen MAC. Die Übereinstimmung beider Werte interpretiert der Empfänger als erfolgreichen Integritätstest. Die Nachricht wurde von einer Partei abgeschickt, die den Sitzungsschlüssel kennt, und die Nachricht wurde während der Übertragung nicht verändert.

Sicherheitsrelevanter Hinweis für den Administrator

Der IDL Applikationsserver sollte in einer sicheren Umgebung untergebracht und durch Firewalls geschützt sein, sodass Unbefugte darauf keinen Zugriff haben.

3 Digitale Signaturen

Zum besseren Verständnis von digitalen Signaturen ist es hilfreich, den Begriff der Hashfunktion zu erwähnen. Eine Hashfunktion erzeugt aus einer beliebig großen Datenmenge, zum Beispiel aus einem Textdokument, einen sogenannten Hash-Code definierter Länge. Der Hash-Code einer Nachricht ist der **digitale Fingerabdruck** dieser Nachricht. Einweg-Hash-Algorithmen sind ein wichtiger Teil vieler Verschlüsselungsprotokolle.

Das digitale Signieren einer Nachricht

Um nun eine Nachricht zu signieren, erzeugt im ersten Schritt der Server über eine Einweg-Hashfunktion einen digitalen Fingerabdruck dieser Nachricht und verschlüsselt dann im zweiten Schritt diesen Fingerabdruck mit seinem privaten Schlüssel. Dieser verschlüsselte Fingerabdruck wird als **digitale Signatur** bezeichnet. Der Server sendet die digitale Signatur zusammen mit der Nachricht an den Client. Der Client berechnet seinerseits mit dem gleichen Hash-Algorithmus den digitalen Fingerabdruck der gesendeten Nachricht. Außerdem entschlüsselt der Client die mitgeschickte digitale Signatur mit dem öffentlichen Schlüssel des Servers.

Stimmt der selbst berechnete digitale Fingerabdruck mit dem entschlüsselten Wert überein, dann ist die Überprüfung der Signatur erfolgreich.

Durch den Einsatz digitaler Signaturen in Kombination mit dem asymmetrischen Verschlüsselungsverfahren kann sichergestellt werden, dass Nachrichten auf dem Übertragungsweg nicht verfälscht werden können. Bei digitalen Signaturen soll es praktisch unmöglich sein, eine zweite Nachricht zu erzeugen, für die diese Signatur ebenfalls gültig ist.

Der auf dem IDL Applikationsserver hinterlegte private Schlüssel wird benutzt, um ein auf dem Applikationsserver erzeugtes Zertifikat selbst zu signieren (selbstsigniertes Zertifikat).

Wird das Zertifikat von einer vertrauenswürdigen Zertifizierungsstelle (CA) erzeugt, dann wird dieses Zertifikat mit dem privaten Schlüssel der CA signiert.

4 Sicherheitszertifikate (Public-Key-Zertifikate)

Durch die Verwendung von Sicherheitszertifikate soll Vertrauen zwischen dem Prüfer bzw. der prüfenden Software (zum Beispiel dem Internetbrowser) und dem Besitzer eines öffentlichen Schlüssels geschaffen werden.

Sicherheitszertifikate bestätigen somit die Authentizität (Zugehörigkeit des Schlüssels zu einer Identität) und darüber hinaus die Integrität (Unverfälschtheit) dieses öffentlichen Schlüssels.

Sicherheitszertifikate werden synonym als Public-Key-Zertifikate, als digitale Zertifikate oder aber auch als SSL-Zertifikate bezeichnet. Da SSL mehr und mehr auf Server- und Clientseite durch TLS ersetzt ist, ist die Bezeichnung SSL-Zertifikat veraltet und auch nicht richtig, denn die Sicherheitszertifikate sind nicht vom verwendeten Verschlüsselungsprotokoll abhängig.

Bei IDL.KONSIS wird während der Konfiguration des IDL Applikationsservers ein Sicherheitszertifikat installiert bzw. importiert. Dieses Zertifikat liegt in einem Keystore Container. Der Keystore Container ist passwortgeschützt. Der zum Sicherheitszertifikat zugehörige private Schlüssel befindet sich ebenfalls im Keystore. Auch dieser Schlüssel ist passwortgeschützt.

Zertifizierungsstellen für Sicherheitszertifikate

Wenn Sie ein Sicherheitszertifikat von einer vertrauenswürdigen Zertifizierungsstelle (Certificate Authority, CA) anfordern, dann überprüft (validiert) die Zertifizierungsstelle Ihre Identität auf der Basis der von Ihnen angegebenen Daten und gibt bei einer positiven Überprüfung ein nur für Sie bestimmtes Sicherheitszertifikat aus. Eine Zertifizierungsstelle (CA) kann ein Unternehmen, eine Institution innerhalb eines Unternehmens, eine Organisation oder eine Regierungsstelle sein.

Zertifizierungsstellen können sich in der Zuverlässigkeit der im Zertifikat enthaltenen Informationen stark unterscheiden. So hängt die Verlässlichkeit des Zertifikates und damit die Zuordnung des öffentlichen Schlüssels zu dem Eigentümer sehr von der Sorgfalt und Gründlichkeit des Ausstellers bei der Überprüfung der eingereichten Daten ab. Die eingesetzten Validierungsprozesse zur Identifizierung der Schlüsseleigentümer unterscheiden sich. Einige Zertifizierungsstellen identifizieren ihre Antragsteller nur persönlich und gegen Vorlage eines amtlichen Ausweises.

Validierungsprozesse für Sicherheitszertifikate

Verschiedene Zertifikatsarten basieren auf einen unterschiedlichen Validierungsprozess. Hierbei geht es darum, dem Nutzer ein hohes Maß an Vertrauen zum Webseitenbetreiber oder zum Applikationsserver-Betreiber zur Verfügung zu stellen, indem mit einer hohen Validierung und deshalb mit entsprechend hohem Aufwand die Unternehmensdaten von einer vertrauenswürdigen Instanz geprüft werden. Die sorgfältigste Prüfung aller Sicherheitszertifikate erfährt ein sogenanntes Extended Validation-Zertifikat, dass durch eine grün eingefärbte Adresszeile im Internetbrowser auch für Anwender ohne IT-Kenntnisse sichtbar wirkt.

Folgende Zertifikatsarten werden benutzt:

- Domain-Validation-Zertifikat (DV)
- Organisation-Validation-Zertifikat (OV)
- Extended-Validation-Zertifikat (EV)

Beim Domain-Validation Zertifikat (DV) wird lediglich die Domain der Seite bestätigt. Das bildet die einfachste Vertrauensstufe und ist auf einen Domainnamen ausgelegt. Diese Sicherheitszertifikate werden lediglich durch eine E-Mail Nachricht validiert und sind dementsprechend in wenigen Minuten verfügbar. Diese Zertifikate eignen sich für Webprojekte, für die keine Unternehmensvalidierung

notwendig ist. Durch das Schloss-Symbol im Internetbrowser wird dem Nutzer eine sichere Verbindung angezeigt. Eine besondere Art des DV-Zertifikates bildet das Wildcard Zertifikat. Dieses ist vor allem dann eine sinnvolle Anschaffung, wenn mehrere verschiedene Hostnamen unter einer Hauptdomain betrieben werden. Mit Wildcard-Zertifikaten lassen sich alle Subdomains integrieren und verschlüsselt verbinden. Wildcard-Zertifikate verringern den Aufwand, den man sonst hätte, um in die PKI einzelne Zertifikate einzubinden und zu verwalten.

Das Organisation-Validation-Zertifikat (OV) bestätigt neben dem Inhaber der Domain auch, für welches Unternehmen das Sicherheitszertifikat ausgestellt wurde. Im Schloss-Symbol findet man neben Laufzeit und Zertifizierungsstelle auch, für welches Unternehmen das Zertifikat ausgestellt wurde. Somit bietet dieses Sicherheitszertifikat einen deutlich höheren Validierungsgrad und schafft zusätzliches Vertrauen der Nutzer, die so den Betreiber einer Website jederzeit überprüfen können. OV-Zertifikate eignet sich für Webprojekte, die Login-Daten von Nutzern abfragen.

Das Extended-Validation-Zertifikat (EV) bietet die umfangreichste Validierung an und schafft somit maximales Vertrauen beim Nutzer. Hier wird ganz konkret geprüft, ob und wie Unternehmen registriert sind. Auch Adressen und Telefonnummern werden überprüft, sowie die Personen, die dieses Zertifikat beantragen wollen. Dieses Sicherheitszertifikat eignet sich für Unternehmen, die im Finanzsektor aktiv sind oder zum Beispiel Kreditkartendaten von Kunden abfragen.

Das Zertifizierungsverfahren dauert bei EV-Zertifikaten logischerweise am längsten, da Unternehmensdaten von der Zertifizierungsstelle genauer geprüft werden. Teilweise meldet sich die Stelle sogar telefonisch bei den Unternehmen, um Daten zu validieren. Beim EV-Zertifikat ist in der Adresszeile des Browsers zusätzlich der Firmenname in grün zu finden.

Public Key Infrastructure (PKI) und Wurzelzertifikate (Root-Zertifikate)

Mit Public-Key-Infrastruktur (PKI) bezeichnet man ein System, das Sicherheitszertifikate ausstellen, verteilen und prüfen kann. Die innerhalb einer PKI ausgestellten Sicherheitszertifikate werden zur Herstellung von Vertraulichkeit verwendet. Bei einer Hierarchie von mehreren Sicherheitszertifikaten wird ein Wurzelzertifikat mit dem zugehörigen Schlüsselpaar bei einer für alle Teilnehmer vertrauenswürdigen Wurzelzertifizierungsstelle (Policy Certificate Authority, kurz PCA)) erstellt. Weitere Sicherheitszertifikate innerhalb dieser PKI können von der Wurzelzertifizierungsstelle mit dem zum Wurzelzertifikat zugehörigen privaten Schlüssel signiert werden.

Nicht jedes Zertifikat einer PKI muss mit dem privaten Schlüssel des Wurzelzertifikats signiert werden. Zum Signieren eines Sicherheitszertifikates können auch private Schlüssel verwendet werden, deren zugehörige Zertifikate wiederum mit einem privaten Schlüssel eines Zertifikates innerhalb der PKI signiert wurden sind. Diese Zertifikate mit den zugehörigen privaten Schlüsseln bilden innerhalb einer PKI eine sogenannte Zertifikatskette. Theoretisch kann eine solche, hierarchisch aufgebaute Zertifikatskette beliebig lang werden, diese Kette muss nur immer beim Wurzelzertifikat beginnen. Eine Zertifikatskette darf nicht unterbrochen sein.

Mit dem IDL.KONSIS-Konfigurationstool (configure.exe) lässt sich ein von einer vertrauenswürdigen Zertifizierungsstelle ausgestelltes Zertifikat importieren.

Wird bei IDL.KONSIS ein selbstsigniertes Sicherheitszertifikat benutzt, kommt beim Aufruf der IDL.KONSIS-Startseite (https://hostname:Portnummer) im Browser eine Warnmeldung, dass das vorhandene Sicherheitszertifikat von einer nicht vertrauenswürdigen Zertifizierungsstelle (CA) ausgestellt worden ist. Möchte man diese Warnung nicht sehen, dann sollte auf dem IDL Applikationsserver ein vertrauenswürdiges Sicherheitszertifikat installiert werden.

Zum Zertifikatimport über das Konfigurationstool ist empfohlen, das Dateiformat pfx zu verwenden. Die Importdatei sollte neben dem Zertifikat den privaten Schlüssel enthalten, deshalb wird die Datei auch als sogenannte Schlüsseldatei bezeichnet. Der private Schlüssel ist passwortgeschützt hinterlegt. Beim Zertifikatimport wählt man den Schlüsseltyp PKCS#12 aus, gibt im Feld Schlüsseldatei den Pfad und Dateinamen der Importdatei und in dem darüber liegenden Feld das Passwort des privaten Schlüssels ein.

Zum Importieren bestätigt man mit OK.

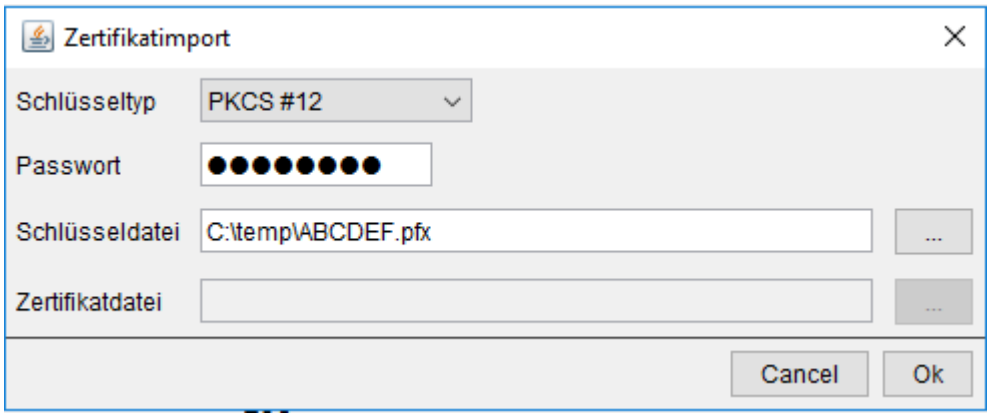


Abbildung 2 KONSIS-Server-Verzeichnis -> Configure.exe -> 4.SSL-Zertifikat-keyStore -> Import „Zertifikatimport“

Selbstsignierte Sicherheitszertifikate

Im Unterschied zu Sicherheitszertifikaten, die von einer CA signiert wurden sind, gibt es Zertifikate, welche vom Ersteller selbst mit seinem eigenen privaten Schlüssel signiert sind. Browser und auch Java reagieren auf selbst signierte Zertifikate mit einer Sicherheits-Warnmeldung, da diese über keine Signatur einer bekannten CA verfügen.

Dem Zertifikat wird nicht vertraut, weil es vom Aussteller selbst signiert wurde.

Obwohl selbstsignierte Sicherheitszertifikate bei https-Verbindungen generell unterstützt werden, veranlassen diese Zertifikate den Client, eine Sicherheitswarnung anzuzeigen, da das Zertifikat nicht von einer vertrauenswürdigen Zertifizierungsstelle verifiziert wurde. Oft raten die Warnungen im Internetbrowser den Besuchern aus Sicherheitsgründen zum Abbrechen der Verbindung.

Während die Gefahren der Verwendung von selbstsignierten Zertifikaten auf öffentlichen Websites offensichtlich sein mögen, besteht auch ein Risiko, sie intern zu nutzen. Selbstsignierte Zertifikate für interne Websites (z.B. Mitarbeiterportale) führen ebenfalls zu Warnungen im Browser. Viele Unternehmen raten deshalb ihren Mitarbeitern, diese Browser-Warnungen einfach zu ignorieren, da sie genau wissen, dass die interne Website sicher ist. Aber dies kann ein gefährliches öffentliches Internet-Surfverhalten fördern. Mitarbeiter, die daran gewöhnt sind, Warnungen auf internen Websites zu ignorieren, können dazu neigen, auch die Warnungen auf öffentlichen Websites ebenso zu ignorieren. Dies macht diese Mitarbeiter und damit das Unternehmen anfällig für Malware und andere Bedrohungen aus dem Internet.

Die Verwendung von Sicherheitszertifikaten, die von einer vertrauenswürdigen Zertifizierungsstelle (CA) ausgestellt wurden, zeigt keine Sicherheitswarnungen im Browser an. Damit fördern sie ein sicheres Internet-Verhalten der Mitarbeiter. Unternehmen, die Websites hosten, können von einem vertrauenswürdigem Fremdanbieter Sicherheitszertifikate für die einzelnen Webserver bzw. auch ein Wildcard-Zertifikat für die gesamte Domäne erwerben. Alternativ kann das Unternehmen aber

auch eine Microsoft Enterprise-Zertifizierungsstelle (CA) in der „Active Directory-Gesamtstruktur“ installieren. Danach kann das Unternehmen mithilfe dieser internen Zertifizierungsstelle (CA) für die einzelnen Webserver Sicherheitszertifikate erstellen und verteilen.

Bei IDL.KONSIS kann über das Konfigurationstool (configure.exe) ein selbstsigniertes Sicherheitszertifikat erstellt werden.

Erforderliche Eingaben zum Erstellen eines selbstsignierten Sicherheitszertifikates:

- **Host (CN): Servername, unter dem der Server von den Clients erreichbar ist.**
- **Schlüssellänge (RSA): Der Default ist eine Schlüssellänge von 2048 bit (ca. 2^{2048}).**
- **Gültigkeit (Tage): Gültigkeitsdauer des Zertifikats, z.B. 365 für 1 Jahr.**
- **Organisationseinheit (OU): Name der Abteilung**
- **Organisation (O): Name des Unternehmens**
- **Ort (L): Standort des Unternehmens**
- **Bundesland (ST): Name des Bundeslands**
- **Land (C): Ländercode, z.B. in Deutschland DE, in England EN**
- **Passwort für den privaten Schlüssel**

Mit dem Button "Erstellen" wird das Zertifikat im Keystore erzeugt. Beim Beenden des Konfigurations-Programms oder beim Speichern werden alle Parameter des Zertifikates und der Private Schlüssel im Keystore gespeichert. IDL.KONSIS-Client überprüft mit dem öffentlichen Schlüssel die digitale Signatur im Zertifikat und damit die Vertrauenswürdigkeit der https-Verbindung. Beim Installiere eines Zertifikates und des zugehörigen Schlüssels unterstützt der IDL Applikationsserver die Schlüssellängen RSA-2048 und RSA-4096.

Der PKI-Standard X.509

X.509 ist ein Standard für eine Public-Key-Infrastruktur zum Erstellen digitaler Sicherheitszertifikate. Dieser Standard setzt ein strikt hierarchisches System von vertrauenswürdigen Zertifizierungsstellen voraus, die Zertifikate erteilen können. X.509 beinhaltet auch einen Standard, mittels dessen Sicherheitszertifikate seitens der Zertifizierungsstelle wieder ungültig gemacht werden können, wenn deren Sicherheit nicht mehr gegeben ist. Die Zertifizierungsstelle kann hierfür ungültige Zertifikate in Zertifikatsperrlisten führen.

X.509v1, X.509v2 und X.509v3

Die ersten X.509-Zertifikate wurden im Jahre 1988 ausgestellt. Einem Zertifikat im X509.v1 -Format konnte man jedoch nicht ansehen, wie aufwendig der Validierungsprozess durchgeführt worden ist. Sie ließen auch keine Rückschlüsse auf den Verwendungszweck zu. Im Jahre 1993 wurde die X509.v2 mit zwei zusätzlichen Feldern ausgegeben, die jedoch selten verwendet werden, die „Issuer Unique ID“ als eindeutige Kennung einer Zertifizierungsstelle sowie die „Subject Unique ID“ als eindeutige Kennung des Zertifikatsinhabers. X.509v3 wurde im Jahre 1996 veröffentlicht. In der Version X.509v3 definierte man die Formatierung von Erweiterungen, die für eine Zertifizierungserweiterung zu verwenden ist. Jeder Erweiterung enthält dabei eine Information, die angibt, ob die jeweilige Erweiterung kritisch oder unkritisch ist. Wenn eine Software eine kritische X.509v3 Erweiterung entdeckt, welche der Software unbekannt ist, wird das Sicherheitszertifikat als ungültig betrachtet. Hingegen stellt eine der Software unbekannte unkritische Zertifikatserweiterung kein Problem dar. Im X.509v3 gibt es einige Standard-Erweiterungen.

Die Erweiterung „Private Internet Extensions“ ermöglicht es einer Zertifizierungsstelle, weitere Informationen im Sicherheitszertifikat zu hinterlegen.

Das über das Konfigurationstool erstellte Sicherheitszertifikat hat das Format X.509v3.

X.509 Sicherheitszertifikate enthalten folgende Informationen:

- Version: es ist wichtig zu wissen, welche X.509-Version für das Zertifikat verwendet wurde. Die Version wiederum bestimmt, welche Daten das Zertifikat enthalten muss.
- Seriennummer: Die CA, die das Zertifikat ausstellt, muss diesem eine Seriennummer zuweisen, damit man es von anderen Zertifikaten unterscheiden kann.
- Algorithmus-Information (zum Bsp. RSA)
- Eindeutiger Namen (oder eine andere eindeutige Bezeichnung) des Ausstellers
- Informationen zur Gültigkeitsdauer des Zertifikates
- den öffentlichen Schlüssel, zu dem das Zertifikat Angaben macht
- den Namen des Eigentümers des öffentlichen Schlüssels
- weitere Informationen zum Eigentümer des öffentlichen Schlüssels
- Angaben zum zulässigen Anwendungs- und Geltungsbereich des öffentlichen Schlüssels
- eine digitale Signatur des Ausstellers über alle anderen Informationen

Zurückgerufene Sicherheitszertifikate

Zertifizierungsstellen müssen in regelmäßigen Abständen Listen mit zurückgerufenen Zertifikaten (Certification Revocation Lists) veröffentlichen. Jedes Zertifikat hat eine eindeutige Seriennummer. Taucht eine Seriennummer in der Liste der gesperrten Zertifikate auf, so ist das zur Seriennummer zugehörige Zertifikat ungültig.

Browser-Warnmeldungen bei Sicherheitszertifikaten

Browser-Warnmeldungen sollten nicht ignoriert werden, denn sie weisen darauf hin, dass das Sicherheitszertifikat nicht vertrauenswürdig ist. Die Angaben über den Inhaber des öffentlichen Schlüssels könnten womöglich nicht stimmen. Ein Zertifikat kann aus den folgenden Gründen ungültig und somit nicht vertrauenswürdig sein**:**

- Das Zertifikat oder seine Signatur wurden widerrufen.
- Das Zertifikat wurde illegal ausgestellt.
- Die Struktur des Zertifikates ist beschädigt.
- Bei der Überprüfung der Signatur ist ein Fehler aufgetreten.
- Die im Zertifikat angegebene Domain entspricht nicht der Website bzw. dem Server (Host), zu der die Verbindung hergestellt wird.

Die Liste der vertrauenswürdigen Stammzertifizierungsstellen, sowie die Liste der vertrauenswürdigen, aber auch die der nichtvertrauenswürdigen Herausgeber von Sicherheitszertifikaten ist in den Einstellungen eines Internetbrowsers einsehbar und auch editierbar. Ist das Root-Zertifikat der PKI im Browser nicht enthalten, kommt beim Öffnen der Internetseite eine Warnung. Die Zertifikatskette ist unterbrochen. Sie ist in folgenden Fällen unterbrochen:

- Die Zertifikatskette besteht aus einem einzigen selbstsignierten Zertifikat.
- Die Zertifikatskette endet nicht mit einem vertrauenswürdigen Stammzertifikat.
- Die Kette enthält Zertifikate, die nicht zum Signieren anderer Zertifikate bestimmt sind.

- Das Stamm- oder Zwischenzertifikat ist abgelaufen oder noch nicht gültig.
- Die Zertifikatskette kann nicht gebildet werden.

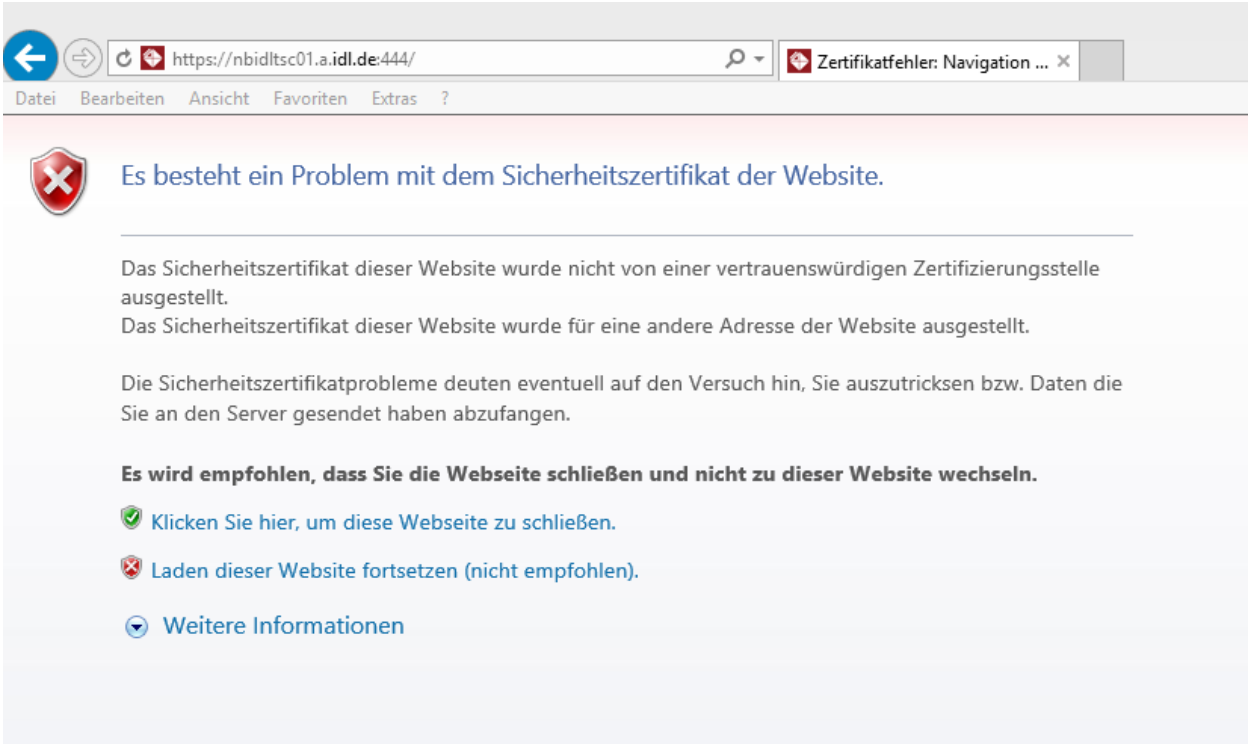


Abbildung 3: Internet Explorer -> Das Zertifikat wurde nicht von einer vertrauenswürden Zertifizierungsstelle ausgestellt

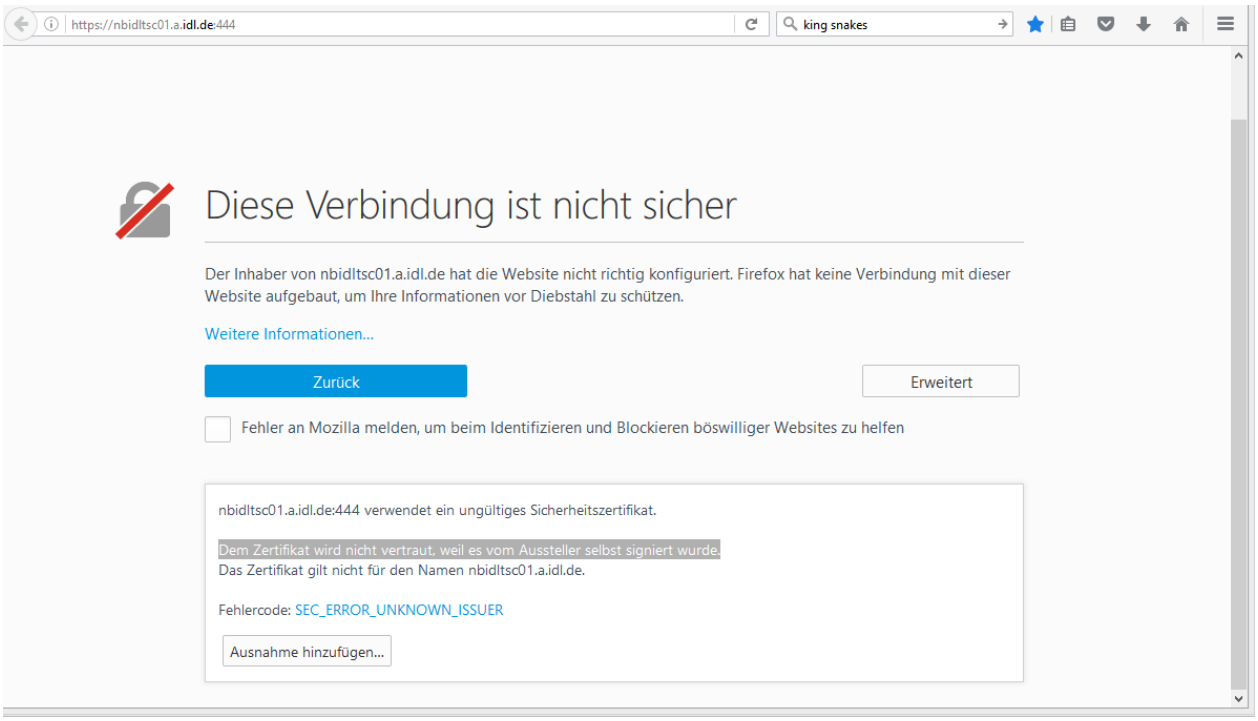


Abbildung 4: Mozilla Firefox -> Dem Zertifikat wird nicht vertraut, weil es selbst signiert wurde

5 Kryptografischer Anhang

Bei der Erstellung von Verschlüsselungssystemen verwenden Programmierer und Software Architekten sogenannte kryptographische Primitive als ihre elementarsten Bausteine.

Kryptographische Primitive sind Verschlüsselungsalgorithmen, die häufig verwendet werden, um kryptographische Protokolle für Sicherheitssysteme innerhalb einer Anwendung oder App zu bauen. Diese Routinen umfassen zum Beispiel Einweg-Hash - und Verschlüsselungsfunktionen.

Symmetrische Kryptosysteme

Die symmetrischen Verfahren zeichnen sich dadurch aus, dass sowohl für die Verschlüsselung in Geheimtext als auch für die Entschlüsselung in Klartext der exakt selbe Schlüssel verwendet wird. Die größte Problematik der symmetrischen Verfahren besteht beim unsicheren Schlüsselaustausch. Eine verschlüsselte Nachricht kann zwar gefahrlos über einen unsicheren Kanal verschickt werden, nicht aber der Schlüssel selbst. Um vor Angriffen geschützt zu sein, muss für den Transport des Schlüssels also unbedingt ein sicherer Kanal verwendet werden.

Der **Advanced Encryption Standard** (**AES**) verschlüsselt Datenblöcke. Nach seinen Entwicklern Joan Daemen und Vincent Rijmen wird AES auch Rijndael-Algorithmus genannt.

Bei AES handelt sich um ein symmetrisches Verschlüsselungsverfahren, d. h. der Schlüssel zum Ver- und Entschlüsseln ist identisch. AES schränkt die Blocklänge auf 128 Bit und die Wahl der Schlüssellänge auf 128, 192 oder 256 Bit ein. Die Bezeichnungen der drei AES-Varianten AES-128, AES-192 und AES-256 beziehen sich jeweils auf die gewählte Schlüssellänge.

Die Funktionsweise des Rijndael-Algorithmus beruht auf Byte-Ersetzungen (Substitutionen), Verwürfelungen (bzw. Permutationen) und auf linearen Transformationen, die auf Datenblöcken von 16 Byte ausgeführt werden – daher auch die Bezeichnung Blockverschlüsselung. Diese Operationen werden mehrmals wiederholt, wobei in jeder dieser Runden ein individueller, aus dem Schlüssel berechneter Rundenschlüssel in die Berechnungen einfließt. Wird nur ein einziges Bit im Schlüssel oder im Datenblock verändert, entsteht ein komplett anderer Chiffreblock.

Der Rijndael-Algorithmus bietet ein sehr hohes Maß an Sicherheit und daher ist AES der bevorzugte Verschlüsselungsstandard für Regierungen, Banken und High-Security Systeme weltweit.

Asymmetrische Kryptosysteme

Das asymmetrisches Kryptosystem oder Public-Key-Kryptosystem ist ein Verfahren, bei dem die kommunizierenden Parteien zwei unterschiedliche Schlüssel benutzen. Der private Schlüssel, oft auch geheimer Schlüssel genannt, wird unter keinen Umständen weitergegeben. Der öffentliche Schlüssel hingegen wird an die Clients weitergegeben.

Der öffentliche Schlüssel ermöglicht es jedem Benutzer, Nutzdaten zu verschlüsseln, die nur der Besitzer des privaten Schlüssels wieder entschlüsseln kann. Mit dem öffentlichem Schlüssel kann außerdem die Signatur eines Zertifikates und damit deren Verauenswürdigkeit überprüft werden.

Der private Schlüssel ermöglicht es seinem Besitzer, mit dem öffentlichen Schlüssel verschlüsselte Daten wieder zu entschlüsseln und außerdem digitale Signaturen zu erzeugen.

Privater und öffentlicher Schlüssel bilden das gemeinsame Schlüsselpaar für die asymmetrische Verschlüsselung. Kennt ein Angreifer den öffentlichen Schlüssel und außerdem den Chifre, so kann

er daraus weder auf die Nachricht noch auf den privaten Schlüssel schließen. Der öffentliche Schlüssel und auch die Chifre können ohne Bedenken über unsichere Kanäle verschickt werden.

Diffie-Hellman-Schlüsselaustausch

1976 entwarfen die Mathematiker Whitfield Diffie und Martin Hellman die Grundzüge des Public-Key-Verfahrens, mit dem das Problem der Geheimhaltung des Schlüssels gelöst werden konnte. Diese wurden im November 1976 unter dem Titel „New directions in Cryptography“ veröffentlicht. Später wurde der Algorithmus, der zum Austausch des öfferntlichen Schlüssels verwendet wird, als Diffie-Hellman-Schlüsselaustausch oder Diffie-Hellman-Merkle-Schlüsselaustausch bezeichnet.

Der RSA-Verschlüsselungsstandard als Beispiel für eine asymmetrische Verschlüsselung

RSA ist ein asymmetrisches kryptographisches Verfahren, das sowohl zum Verschlüsseln als auch zum digitalen Signieren verwendet werden kann. Das RSA-Verfahren benutzt zum Verschlüsseln und zum Überprüfen der Signatur einen öffentlichen Schlüssel, den Public Key. Zum Entschlüsseln sowie zum digitalen Signieren benutzt RSA einen privaten Schlüssel, den Private Key.

Der Informatik-Professor **R**onald Linn Rivest (Theoretische Informatik) und die beiden Mathematik-Professoren Adi **S**hamir und Leonard **A**dleman entwickelten 1977 am MIT (Massachusetts Institute of Technology in Cambridge, USA) ein kryptografisches Verfahren zum digitalen Signieren und zum Verschlüsseln. Sie veröffentlichten ihre Ideen im April 1977 unter dem Titel:

„A method for obtaining Digital Signatures and Public Key Cryptosystems“.

Der öffentliche Schlüssel (Public Key) besteht aus einem Zahlenpaar (e , N). Der private Schlüssel (Private Key) besteht ebenfalls aus einem Zahlenpaar (d , N). Die Zahl **e** ist der sogenannte Verschlüsselungsexponent. Die Zahl **d** ist der Entschlüsselungsexponent und **N** ist der RSA-Modul.

N soll das Produkt zweier zufällig gewählter, sehr großer Primzahlen p und q sein. Die beiden Primzahlen sollen eine Größe von 1024 bit nicht unterschreiten.

Die Zahlen **e, d** und **N** sind über eine mathematische Formel miteinander verbunden. Es ist jedoch nicht ohne weitere Zusatzinformation (ohne Kenntnis von p oder q) möglich, aus dem öffentlichen Schlüssel **e** und dem Modul **N** den Entschlüsselungsexponenten **d** zu berechnen (Falltürfunktion). Wäre aber p und q bekannt, dann könnte man den privaten Schlüssel **d** einfach berechnen:

e * **d** = 1 mod ((p-1) (q-1)) -> d ist invers zu e

Die Sicherheit des RSA-Algorithmus basiert dabei auf einem schwierigen mathematischen Problem. Es wird vermutet, dass es auch in Zukunft kein intelligentes, mathematisches Verfahren geben wird, welches jede, noch so große, natürliche Zahl in ihre Primfaktoren zerlegen kann. Diese Vermutung wurde allerdings noch nicht bewiesen.

Es ist derzeit nicht möglich, in einer annehmbaren Zeit aus dem Modul N mit einer Schlüssellänge von 2048 bit die beiden Primzahlen p und q mathematisch zu berechnen.

Hybride Kryptosysteme

Da asymmetrische Krypto-Verfahren wie das RSA Verfahren extrem rechenaufwändig sind und auch noch andere Nachteile[5] besitzen, benutzt man in der Praxis fast immer eine Kombination aus symmetrischer und aus asymmetrischer Kryptographie. Diese Kombination nennt man Hybrid-Kryptographie bzw. hybride Verschlüsselung, und diese Systeme hybride Kryptosysteme. Als

[5] Zum Beispiel das Verteilungsproblem mit dem Mittelsmann-Angriff /Man-In-The-Middle

Anwendungsbeispiel für die Kombination aus symmetrischer und asymmetrischer Kryptographie wäre das Verschlüsselungsprotokoll **TLS** zu nennen. Bei diesem Sicherheitsprotokoll ist es üblich, den Sitzungsschlüssel nach verhältnismäßig kurzer Zeit wieder neu auszuhandeln. TLS ist ein kryptographisches Protokoll zur Etablierung eines sicheren Kanals. Im TCP/IP-Modell ist TLS oberhalb der Transportschicht (zum Beispiel TCP) und unterhalb Anwendungsprotokollen wie HTTP oder SMTP angesiedelt. TLS dient der Sicherstellung von Vertraulichkeit, Authentizität, Integrität und Verbindlichkeit bei der Übertragung von Daten in unsicheren Netzwerken.

TLS nutzt verschiedene Verschlüsselungsverfahren, die sich in ihrer kryptographischen Leistungsfähigkeit stark unterscheiden. Bei TLS wird zu Beginn einer jeden Sitzung zwischen Client und dem Server das Verschlüsselungs-Verfahren ausgehandelt. Erst danach wird der sogenannte **Sitzungsschlüssel** (englisch: **Session Key)** erstellt. Dieser Sitzungsschlüssel wird asymmetrisch verschlüsselt zum Emfpänger übertragen. Der Sitzungsschlüssel ist ein zufällig generierter Schlüssel, der nur ein einziges Mal für eine einzelne Sitzung (Session) zwischen Client und Server verwendet wird. Dieser Sitzungsschlüssel kann von beiden Seiten für eine ressourcenschonende symmetrische Verschlüsselung und Entschlüsselung genutzt wird.

Mit dem erstellten Sitzungsschlüssel ist es möglich, größere Mengen an Nutzdaten performant zu verschlüsseln und zu entschlüsseln. TLS-Verschlüsselung wird heute vor allem bei HTTPS eingesetzt. TLS schiebt sich zwischen HTTP und dem Transportprotokoll TCP. TLS arbeitet dabei für Anwender nahezu unsichtbar.

Heutzutage werden viele Anwendungen und Apps benutzt, die Daten über das Internet Daten übertragen. Bei diesen Anwendungen und Apps ist es wichtig, dass insbesondere Zugangsdaten, PIN‘s, Passwörter, aber auch andere Nutzdaten sicher übertragen werden können. Hier spielt das TLS-Protokoll eine sehr wichtige Rolle. TLS dient dazu, einen sicheren Kanal zwischen Sender und Empfänger aufzubauen und alle Nutzdaten sicher über diesen Kanal zu übertragen.

Anmerkung: Seit 2011 sind einige Angriffe gegen TLS bekannt geworden. Die Schwachstellen von TLS 1.1 können durch Nutzung entsprechender Cipher-Suiten der Version TLS 1.2 behoben werden. Das IEFT hat inzwischen TLS 1.3 zum neuen Standard erklärt.

So unterstützt TLS 1.3 eine noch stärkere Verschlüsselung als der Vorgänger TLS 1.2.