DER „MAN-IN-THE-MIDDLE“-ANGRIFF SIMULIERT AM BEISPIEL DER RSA - VERSCHLÜSSELUNG MIT EINER GRAFISCHEN BENUTZEROBERFLÄCHE IN JAVA

Facharbeit im Fach Informatik

Geschrieben von: Christopher Neugebauer
Betreuungslehrer: Rolf Faßbender
Informatik – Leistungskurs Q1
Schuljahr 2019/2020

Inhalt

Einleitung

Das Internet ist aus der modernen Gesellschaft nicht mehr weg zu denken. Menschen nutzen es nicht nur zum Suchen von Informationen. Häufiger wird das Internet für den Austausch von Informationen genutzt. Das kann von der kurzen E-Mail an Freunde bis zum Onlinebanking gehen, bei dem sensible Daten wie Kontonummern oder PINs weiter gegeben werden. Spätestens bei den Kontodaten möchte jeder verhindern, dass Unbefugte Zugriff auf diese Daten haben. Aus diesem Grund ist Kryptografie im Internet zwingend notwendig.

Diese Facharbeit beschäftigt sich mit dem asymmetrischen Verschlüsselungsverfahren RSA. Das RSA – Verfahren wurde in den 1970er Jahren entwickelt und ermöglicht eine Verschlüsselung, deren Sicherheit ausschließlich auf der Geheimhaltung bestimmter Schlüsselteile basiert. Da es natürlich auch Angreifer gibt, welche Daten ausspionieren wollen – ansonsten bräuchte niemand seine Daten zu verschlüsseln – behandelt diese Facharbeit auch eine sehr prominente Form von Angriffen auf asymmetrische Verschlüsselungsverfahren, den Man-in-The-Middle Angriff. Abschließend wird beschrieben, wie diese Angriffe verhindert werden können.

Zusätzlich liegt dieser Facharbeit ein selbstgeschriebenes Simulationsprogramm bei. Es ist mit der Programmiersprache Java geschrieben und im Stande, Man-in-The-Middle Angriffe sowie deren Verteidigung zu simulieren.

Begriffserklärung

Alice: In der Kryptografie wird Kommunikationspartner A traditionellerweise Alice genannt. In dieser Facharbeit wird ebenfalls von Alice die Rede sein, wenn es um Kommunikationspartner A geht.

Asymmetrisches Verschlüsselungsverfahren: Ein asymmetrisches Verschlüsselungsverfahren arbeitet mit zwei Schlüsseln. Einer wird zum Verschlüsseln, der andere zum Entschlüsseln verwendet. Da der Schlüssel zum Verschlüsseln für alle Kommunikationspartner zugänglich gemacht werden muss, wird er öffentlicher Schlüssel genannt. Da der Schlüssel zum Entschlüsseln selbstverständlich geheim gehalten werden muss, wird er privater Schlüssel genannt. Wichtig ist, dass ein außenstehender Angreifer nicht in der Lage ist, aus dem öffentlichen Schlüssel den privaten Schlüssel abzuleiten, da er sonst den Geheimtext entschlüsseln könnte.

Bob: In der Kryptografie ist Bob traditionellerweise der Kommunikationspartner von Alice. So auch in dieser Facharbeit.

Bruteforce: Als Bruteforce Angriff wird ein Angriff auf Kryptosysteme bezeichnet welcher sich ausschließlich einer großen Menge zur Verfügung stehenden Rechenleistung bedient. Analytische Herangehensweisen wie Wahrscheinlichkeiten werden dabei weitgehend ignoriert. Stattdessen wird jede mögliche Kombination ausprobiert.

Eve: Eve wird traditionell der Angreifer auf eine verschlüsselte Kommunikation genannt. So auch in dieser Facharbeit.

Kryptosystem: Ein Kryptosystem ist eine Methode, ein Protokoll oder eine Maschine, mit der Nachrichten ver- sowie entschlüsselt werden können. Dazu zählt RSA genauso wie die Enigma.

Schlüssel: Als Schlüssel wird in der Kryptografie eine Information bezeichnet, welche zum Ver- oder Entschlüsseln einer Nachricht benötigt wird.

Symmetrische Verschlüsselung: Ein symmetrisches Verschlüsselungsverfahren nutzt im Gegensatz zur asymmetrischen Verschlüsselungogra denselben Schlüssel für die Ver- und Entschlüsselung.

Einführung – was wird als sicher bezeichnet?

Bevor eine Möglichkeit zur sicheren Kommunikation erklärt wird sollte geklärt werden, was in der Kryptografie überhaupt als sicher bezeichnet wird. Die im 19. Jahrhundert von Auguste Kerkhoff formulierten Anforderungen bezüglich der Sicherheit von Kryptosystemen gelten heute noch. Laut Kerkhoff ist ein Kryptosystem sicher, wenn, obwohl das Verschlüsselungsverfahren öffentlich bekannt ist, ohne Kenntnis über den Schlüssel der Klartext aus dem Geheimtext nicht zu entschlüsseln ist[1]. Dieses Prinzip hat allerdings die Schwachstelle, das in der Theorie jeder mögliche Schlüssel des bekannten Kryptosystems ausprobiert werden könnte. Deswegen wird sich darauf beschränkt, ein Kryptosystem als sicher zu bezeichnen, wenn ein Bruteforce Angriff in praktikabler Zeit nicht zielführend ist. In Zeiten von Computern wäre dies der Fall,

[1] Feiermuth K. et al., „Einführung in die Informatik“, S. 38

wenn alle Rechenleistung der Welt nicht ausreichen würde um in einem angemessenen Zeitraum durch Ausprobieren den Schlüssel zu erfahren.

Das RSA – Verfahren

Das RSA – Verfahren ist das erste asymmetrische Verschlüsselungsverfahren, welches in den 1970er Jahren entwickelt wurde. Für die Öffentlichkeit wurde das Verfahren 1977 von Ron Rivest, Adi Shamir und Leonard Adelman entwickelt. Aus den Anfangsbuchstaben der Nachnamen der Entwickler leitet sich auch der Name für das Verfahren ab. Diese drei arbeiteten zu der Zeit am Massachusetts Institute of Technology und lösten zusammen mit Dieffie – Hellman eine Revolution in der Kryptografie aus. Interessanterweise wurde das RSA Verfahren bereits 1973 vom britischen Kommunikationsgeheimdienst GCHQ entdeckt, aber nie veröffentlicht. Da der britische Geheimdienst nicht wusste ob RSA überhaupt sicher wäre wurde RSA auch nicht für interne Zwecke genutzt[2].

Die Schlüsselgenerierung

Das Wichtigste bei dem RSA – Verfahren ist das Erzeugen der Schlüsselpaare. RSA nutzt zwei Schlüssel, privater und öffentlicher Schlüssel genannt. Der öffentliche Schlüssel ist für jeden zugänglich, während der private Schlüssel unbedingt geheim gehalten werden muss. Die Schlüssel müssen bestimmte Anforderungen erfüllen, um das geforderte Maß an Sicherheit gewährleisten zu können. Jeder Schlüssel besteht aus zwei Zahlen: N und dem Exponenten d für das Entschlüsseln oder dem Exponenten e für das Verschlüsseln. Alle drei Zahlen müssen positiv sein[3].

Zuerst wird N berechnet. N ist die Summe zweier Primzahlen p und q. Da N öffentlich gemacht wird und seine Summanden p und q, um Sicherheit gewährleisten zu können, geheim bleiben müssen, sollten p und q große Primzahlen sein[3]. Ansonsten könnten mit der Primfaktorzerlegung p und q korrekt berechnet werden. Das Bundesamt für Sicherheit in der Informationstechnik empfahl Anfang 2019 für N mindestens 2000 Bit, ab 2023 sogar 3000 Bit, um eine sichere Verschlüsselung zu gewährleisten[4]. Als nächstes wird die Phi Funktion von N berechnet. Diese gibt an, wie viele teilerfremde Zahlen es für N im Zahlenraum 1-N gibt. Da für Primzahlen $\varphi(p) = p - 1$ gilt, ist

[2] Nach wired.de – The Open Secret
[3] Nach „A Method for Obtaining Digital Signatures and Public-Key Cryptosystems“ von R.L. Rivest, A. Shamir und L. Adleman
[4] BSI: Kryptographische Verfahren: Empfehlungen und Schlüssellängen

$\varphi(N) = (p-1) * (q-1)$. Mithilfe der Phi – Funktion wird dann *d* berechnet. Dabei wird *d* so gewählt, das $d = kgv(\varphi(N), d) = 1$ gilt. In Worten: *D* und $\varphi(N)$ müssen teilerfremd zueinander sein. Zum Schluss wird *e* berechnet. *E* ist das multiplikative Inverse von *d* modulo $\varphi(N)$. Als Formel ausgedrückt ist $e = e * d(mod\varphi(N))$ [3]. In der Praxis sollte e nicht klein sein, da dafür so genannte Low – Exponent Angriffe bekannt sind. Näheres im Abschnitt zur Verschlüsselung mit RSA.

Beispiel zur Schlüsselgenerierung

Zum Abschluss dieses Unterthemas noch ein kleines Beispiel zur Veranschaulichung. Die hier gewählten Zahlen sollen auch im Kopf nachvollziehbar sein und sind weit von einer sicheren RSA – Verschlüsselung entfernt.

1) *P* und *q* wählen: $p = 7$ und $q = 11$
2) *N* berechnen: $N = p * q = 7 * 11 = 77$
3) Phi von *N* berechnen: $\varphi(N) = (p-1) * (q-1) = 60$
4) *D* wählen: $d = 23$ (kgv(60, 23) = 1)
5) *E* berechnen: $e = d * e\big(mod\ \varphi(N)\big) = 23 * e(mod\ 60) = 47$

Somit ist der öffentliche Schlüssel (23, 77) und der private Schlüssel (47, 77).

Ver- und Entschlüsselung

Im Vergleich zur Schlüsselgenerierung und erst recht im Vergleich zu anderen Verschlüsselungsverfahren ist die Ver- und Entschlüsselung mit RSA ziemlich simpel gehalten, da RSA ein rein mathematisches Verfahren ist. Der Geheimtext (G) berechnet sich wie folgt, wenn M der Klartext und N sowie d die Bestandteile des Schlüssels sind: $G = M^d(mod\ N)$. Der Klartext lässt sich aus dem Geheimtext unter Verwendung des anderen Schlüsselpaares zurückgewinnen: $M = G^e(mod\ N)$ [3].Dieses Verfahren ist auch als Lehrbuch – RSA bekannt, da die Entwickler das RSA Verfahren in dieser Form vorgeschlagen hatten. Es ist allerdings an mehreren Stellen unsicher, weswegen heute Einschränkungen gelten und ein weiteres Verfahren vor der Verschlüsselung angewendet wird. Beide sollen hier kurz der Vollständigkeit halber Erwähnung finden. Zum einen darf der Verschlüsselungsexponent nicht zu klein gewählt werden. Dann ist eine so genannte Low – Exponent Attack möglich[5].

[5] Nach Uni Flensburg – RSA Verschlüsselung

Blockung

Bei dem RSA – Verfahren darf die zu verschlüsselnde Nachricht *K* nicht die Größe des RSA – Modulus *N* überschreiten. Wenn dem so wäre, würde die Nachricht bereits vor dem Verschlüsseln verändert. Außerdem könnten mehrere Klartexte *K* auf einen Geheimtext *G* abgebildet werden, wie dieses Beispiel zeigen soll:

$$40^1(mod\ 10) = 30^1(mod\ 10) = 20^1(mod\ 10) = 0$$

Aus diesem Grund wird eine Nachricht in Blöcke unterteilt, deren Wert unter dem des Modulus *N* liegt. Ein fahrlässiger Implementationsfehler wäre es, je ein Zeichen als einzelnen Block zu verschlüsseln. Mit dieser Methode würden gleichen Zeichen im Geheimtext immer den gleichen Zeichen im Klartext entsprechen. Damit wäre dann eine simple Häufigkeitsanalyse möglich und RSA nicht mehr sicher.

Nutzung von RSA in der Realität

Für den alltäglichen Gebrauch eignet sich RSA nur bedingt, da RSA sehr große Schlüssel braucht, um Sicherheit gewährleisten zu können. Das macht RSA zu einem sehr rechenaufwändigen und damit langsamen Verschlüsselungsverfahren. Außerdem wird die verschlüsselte Nachricht in fast allen Fällen größer als der ursprüngliche Klartext. Dafür bietet RSA als Asymmetrisches Verfahren den großen Vorteil, von Anfang an sichere Kommunikation über ein unsicheres Medium zu ermöglichen. Deswegen wird RSA in der Realität meistens für die Authentifizierung zu Beginn einer Kommunikation und für den Austausch eines Symmetrischen Schlüssels verwendet. Das Symmetrische Verschlüsselungsverfahren ist der Advanced Encryption Standard (AES)[6]. Diese Kombination wird auf HTTPS Seiten und bei mehreren Verschlüsselungsverfahren für den Email Verkehr genutzt[6].

Der „Man-in-The-Middle“ Angriff

Wenn Daten verschlüsselt werden, gibt es dazu immer einen Grund, nämlich zu verhindern, dass die Daten öffentlich einsehbar sind. Selbstverständlich gibt es auch Methoden, eine eigentlich sichere Verbindung abzuhören. Dazu zählt der so genannte Man-in-The-Middle Angriff (im folgenden MiTM Angriff), welcher es darauf abgesehen hat, sich in die Kommunikation zweier Parteien einzuklinken.

[6] Nach IONIS Digital Guide: Verschlüsselungsverfahren im Überblick

Wie funktioniert ein MiTM Angriff?

Bei einem MiTM Angriff klinkt sich der Angreifer in eine Kommunikation ein. Dies kann in einer Netzwerkverbindung oder zwischen zwei Programmen geschehen. Meistens nutzt der Angreifer einen W-Lan Router, welchen er entweder selbst aufgesetzt hat oder den er hacken konnte, oder Access Points. Dieser Router oder Access Point gibt sich gegenüber dem Opfer als vertrauenswürdig aus, sodass das Opfer ihn als Kommunikationsmittel wählt. Wenn das Opfer dann die Verbindung erstellt kann der Angreifer jede Kommunikation mitlesen und verändern. Normalerweise werden die Nachrichten weitergegeben, damit der Angreifer nicht bemerkt wird[7]. Weitere Möglichkeiten eines MiTM Angriffes sind das Fälschen einer IP – Adresse, das Fälschen einer DNS – Adresse, das Fälschen einer Website, zum Beispiel von Banken, das Fälschen von Email – Adressen und das Auslesen von Browser Cookies, welche sensible Daten wie Login Informationen enthalten[8]. Auch verschlüsselte Kommunikation hilft nicht, wenn sich der Angreifer am Schlüsselaustausch beteiligt. Dazu ein Beispiel, welches von einer asymmetrisch verschlüsselten Kommunikation ausgeht:

Beispiel eines MiTM Angriffes

In diesem Beispiel möchte Alice Bob eine Nachricht zukommen lassen, während Eve versucht, den Inhalt dieser Nachricht zu erhalten. Um eine sichere Kommunikation zu ermöglichen, hat Alice Bob ihren öffentlichen Schlüssel über das Internet geschickt. Was weder Alice noch Bob wissen, ist, dass Eve den Router von Alice gehackt hat und deswegen alles was Alice verschickt oder bekommt, abhören sowie verändern kann. Das macht sich Eve nun zunutze, indem sie den öffentlichen Schlüssel von Alice speichert und sich bei Bob als Alice ausgibt.

[7] Nach Security Insider – Was ist ein Man-in-the-Middle-Angriff?
[8] Nach Norton – What is a man-in-the-middle attack?

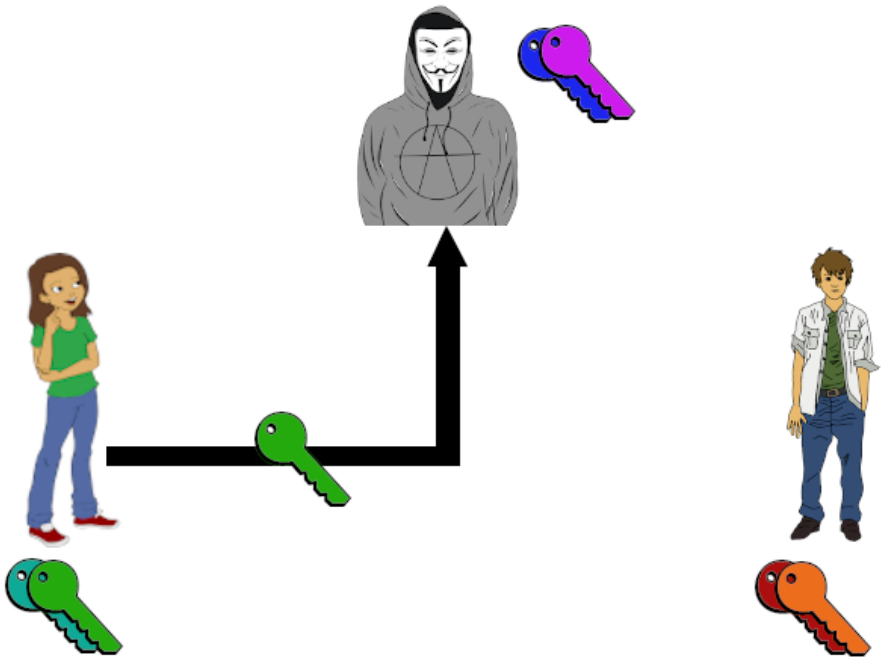

Abb. 1: Alice sendet unwissentlich Eve ihren privaten Schlüssel, da Eve sich in die Kommunikation eingeklinkt hat

So gibt Eve Bob ihren eigenen öffentlichen Schlüssel und behauptet, er würde von Alice stammen.

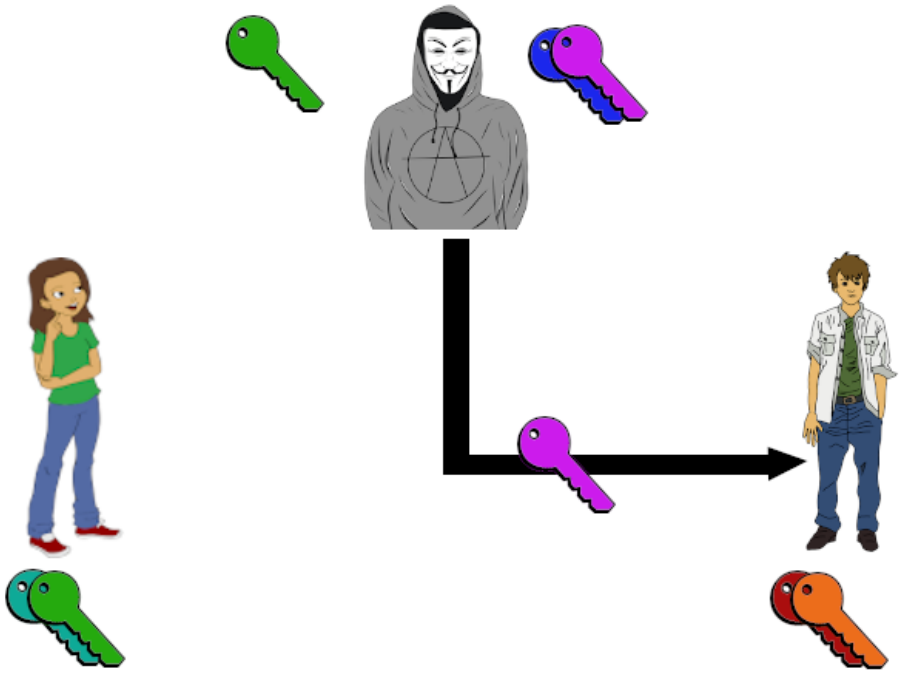

Abb. 2: Eve sendet Bob ihren öffentlichen Schlüssel und tut so als käme er von Alice

Dieses Verfahren führt sie auch für den öffentlichen Schlüssel von Bob durch, welcher seinen öffentlichen Schlüssel Alice schicken möchte. Am Ende verfügt Eve über die

öffentlichen Schlüssel von Alice und Bob. Alice und Bob haben jeweils den öffentlichen Schlüssel von Eve.

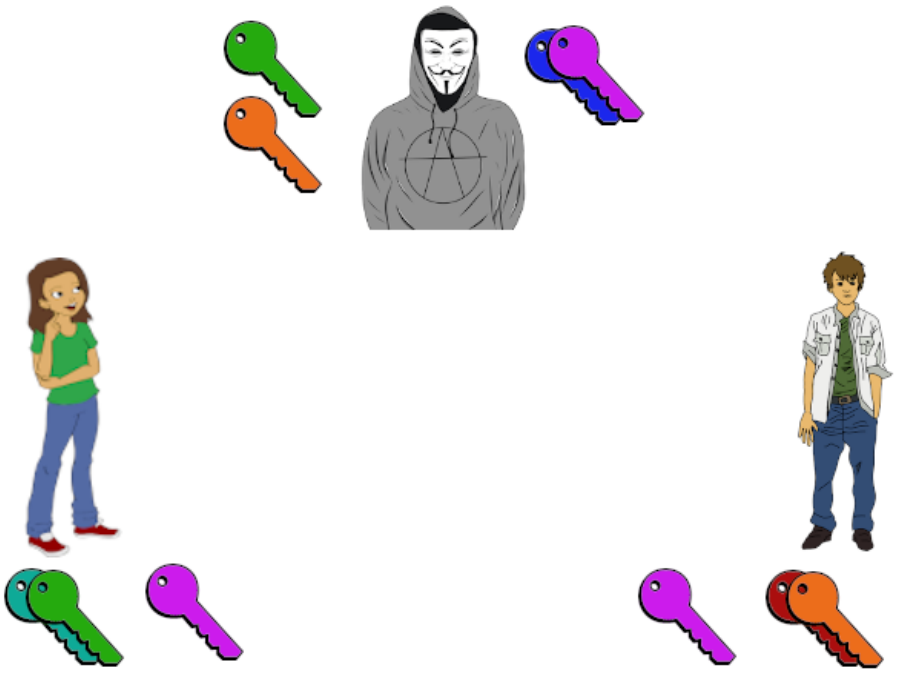

Abb. 3: Am Ende verfügt Eve über die öffentlichen Schlüssel von beiden und Alice sowie Bob haben jeweils den öffentlichen Schlüssel von Eve. Sie denken aber er wäre von Bob bzw. Alice

Alice denkt aber, der öffentliche Schlüssel wäre von Bob, und Bob denkt, er wäre von Alice. Wenn Alice Bob nun eine Nachricht schicken möchte, verschlüsselt sie die Nachricht mit dem öffentlichen Schlüssel von Eve.

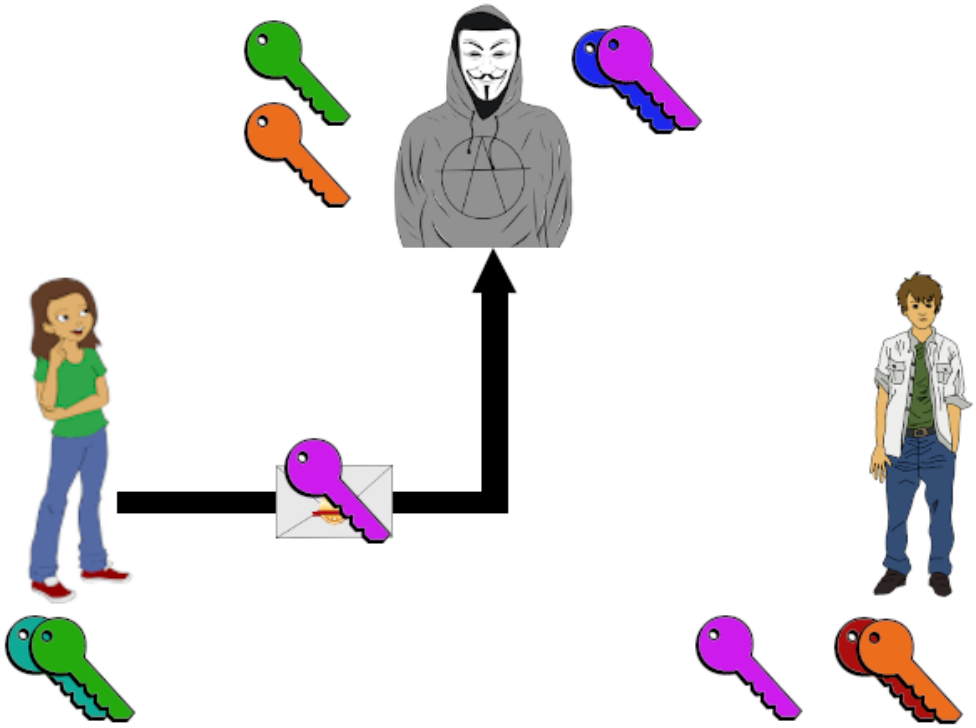

Abb. 4: Alice möchte Bob eine Nachricht schicken, verschlüsselt diese aber mit Eves Schlüssel. Eve fängt die Nachricht ab

Eve fängt diese Nachricht ab und entschlüsselt sie mit ihrem privaten Schlüssel. Nun hat sie die unverschlüsselte Nachricht, welche Alice eigentlich Bob schicken wollte.

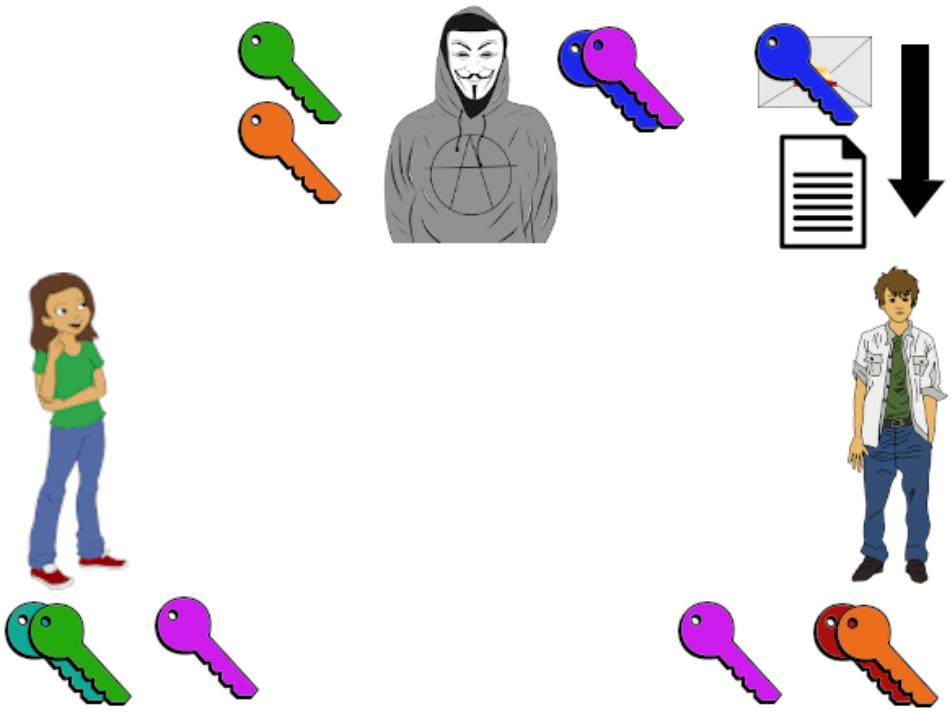

Abb. 5: Eve kann die Nachricht mit ihrem privaten Schlüssel entschlüsseln

Damit der Schwindel nicht auffällt kann, Eve nun den öffentlichen Schlüssel von Bob nehmen, die Nachricht wieder verschlüsseln und an Bob versenden. Wenn Bob die Nachricht mit seinem privaten Schlüssel entschlüsselt, wird er nicht bemerken können dass Eve sich in seine Kommunikation mit Alice eingeklinkt hat und jede Nachricht mitliest.

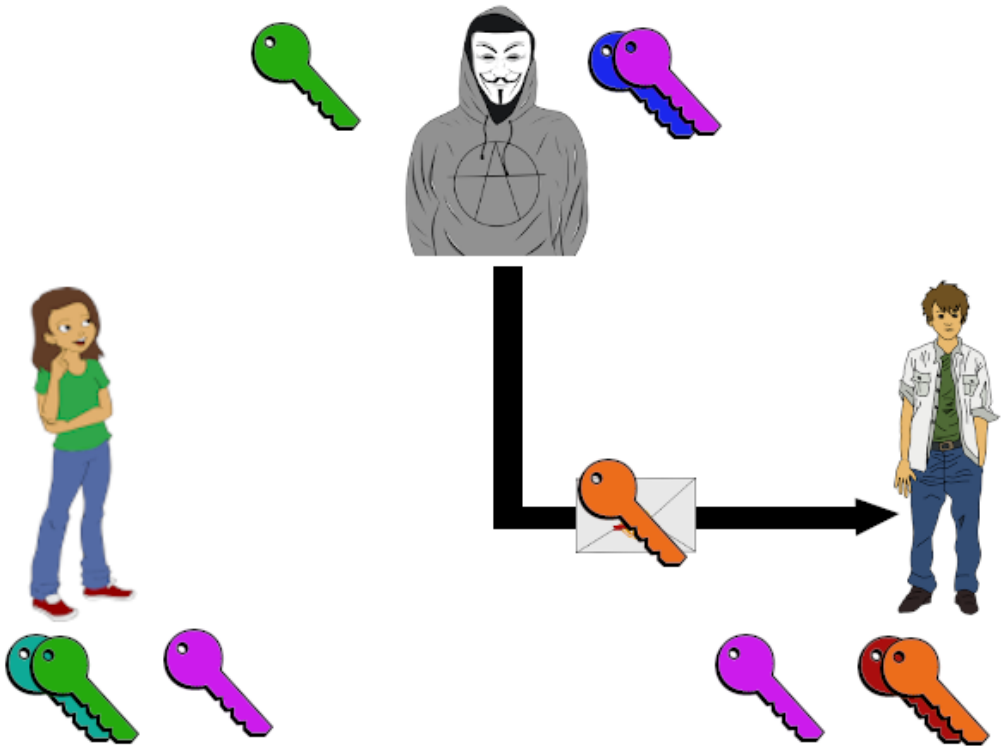

Abb. 6: Eve sendet die Nachricht mit Bobs öffentlichen Schlüssel verschlüsselt an Bob, um Alice und Bob in Sicherheit zu wiegen

Auch Alice hat von dem ganzen Vorgang nichts bemerkt. Wenn Eve möchte, kann sie eine ganz andere Nachricht an Bob schicken als die, welche Alice ihm eigentlich schicken wollte. Natürlich funktioniert der MiTM Angriff auch in die andere Richtung, wenn Bob Alice eine Antwort auf die Nachricht schickt.

Auf diese Weise ist es möglich, die Kommunikation zweier Kommunikationspartner zu überwachen. Um Daten, zum Beispiel den Login einer Website für Online Banking eines Opfers zu bekommen, kann ein MiTM Angriff ebenfalls verwendet werden. Dafür ist eine Kombination aus einer gefälschten Email Adresse sowie einer gefälschten Website möglich. Hacker könnten eine E-Mail-Adresse fälschen, in der sie ihr Opfer auffordern seine Identität zu bestätigen, indem sie einen mitgeschickten Link verwenden, um sich auf der Website der Bank einzuloggen. Der Link führt allerdings nicht zur tatsächlichen Website der Bank, sondern zu einer gefälschten Website, welche

der tatsächlichen Website zum Verwechseln ähnlich sieht. Wenn das Opfer seine Logindaten auf der gefälschten Website eingibt, erhält der Angreifer die Logindaten und kann sich nun mit diesen Daten auf der echten Website einloggen[7].

Alternativ können gefälschte Websites allerlei Schad- oder Spionagesoftware auf dem Gerät des Opfers installieren.

Wie wird ein MiTM Angriff verhindert?

Es ist also klar, mit einem MiTM Angriff kann viel Schaden angerichtet werden. Glücklicherweise gibt es Systeme, welche die Echtheit von Websites sowie die Sicherheit der Schlüsselübertragung gewährleisten. Im Folgenden wird der Aufbau und die Funktion von Public Key Infrastrukturen, kurz PKIs, beschrieben.

Public Key Infrastructure

Public Key Infrastrukturen (kurz: PKI) sind Institutionen, die einen sicheren Austausch von Daten über das Internet ermöglichen sollen. Wie im Abschnitt zum Man-in-The-Middle Angriff beschrieben, liegt das Problem bei der Datenübertragung. Entweder die Übertragung wird abgehört und ist nicht verschlüsselt oder ein Nutzer kommuniziert unwissentlich direkt mit einem Angreifer. Diese Probleme kann eine PKI lösen. Meistens sind PKIs hierarchisch aufgebaut, beginnend bei so genannten Root-Certification Authorities (zu dt. Wurzelzertifikatinstanz). Sie können Stammzertifikate ausstellen, welche in Browsern oder Betriebssystemen vom Softwareentwickler eingespeichert werden[9].

Von diesen Stammzertifikaten ausgehend können weitere Certification Authorities (zu dt. Zertifizierungsinstanz) zertifiziert werden. Diese können dann Zertifikate an Endnutzer ausstellen und sicher öffentliche Schlüssel verteilen, ohne dass ein MiTM die Kommunikation erfolgreich manipulieren oder belauschen kann. Dafür ist allerdings eine Kette an Vertrauen nötig. Die Wurzelzertifikatinstanzen können weitere Zertifikate für so genannte Registration Authorities ausstellen, welche wiederum Zertifikate für Nutzer ausstellen. Theoretisch ließe sich diese Kette endlos erweitern, solange sie bei einem Root-Zertifikat anfängt. Sollte sich ein Glied in dieser Kette plötzlich als unsicher herausstellen, gelten alle in der Kette nachfolgenden Zertifikate als unsicher[9].

[9] Nach Security Insider – Was ist eine PKI?

Im speziellen Fall der Schulhomepage des Städtischen Gymnasium Rheinbachs stammt das Root Zertifikat von der amerikanischen Organisation DigiCert Inc. Der Name der Wurzelzertifikatinstanz lautet DigiCert Global Root CA. Mit diesem Root Zertifikat hat DigiCert ein Zertifikat ihrer eigenen Zertifizierungsinstanz signiert, welche wiederum ein Endzertifikat für die Schulhomepage des Städtische Gymnasium Rheinbach ausgestellt hat.

Digitale Zertifikate

Digitale Zertifikate sind das Verifizierungsmedium einer PKI, quasi der Beweis, dass alles mit rechten Dingen zugeht. Eines der meist genutzten Zertifkatformen ist das X.509 Zertifikat. Aus Gründen des Umfanges werden Andere an dieser Stelle außen vorgelassen und nur die Bestandteile eines X.509 Zertifikats werden genauer erläutert.

Ein X.509 Zertifikat enthält Informationen über die genaue Version des Zertifikats sowie eine eindeutige Seriennummer, um es von anderen Zertifikaten unterscheiden zu können. Des Weiteren beinhaltet es Informationen über die genutzten Verschlüsselungsalgorithmen, darüber wer das Zertifikat ausgestellt hat, von wann bis wann das Zertifikat gültig ist und für wen es ausgestellt wurde. Zusätzlich beinhaltet es den öffentlichen Schlüssel des Zertifikatinhabers[10].

Wie funktionieren digitale Zertifikate?

Am Beginn eines digitalen Zertifikates steht ein asymmetrisches Schlüsselpaar, heutzutage fast immer ein RSA Schlüsselpaar mit 2048 oder 4096 Bit. Wenn ein Nutzer für seine Website – egal ob Privatmensch, Websitenbetreiber oder Firma – möchte, dass seine Schlüssel und damit sein ganzes Zertifikat als sicher betrachtet werden können, wendet er sich an eine Zertifizierungsstelle, um das Zertifikat signieren zu lassen. Je nach Art des Zertifikates, zum Beispiel, ob von ihm aus weitere Zertifikate ausgestellt werden sollen oder nicht, werden dabei unterschiedliche Überprüfungen durchgeführt[11]. Wird ein Antrag angenommen, signiert die Zertifizierungsstelle das Zertifikat mit ihrem eigenen privaten Schlüssel und speichert es in seiner Datenbank[12]. Dadurch, dass das Zertifikat mit dem privaten Schlüssel der Zertifizierungsstelle signiert wurde, kann von nun an immer nachgewiesen werden, dass dieses Zertifikat tatsächlich von dieser Zertifizierungsstelle signiert wurde. Da es bei einem asymmetrischen Schlüsselpaar egal

[10] Nach computerweekly.com – X.509 Zertifikat
[11] Nach GlobalSign: Zertifizierungsstellen und Vertrauenshierachien
[12] Gallenbacher J., „Abenteuer Informatik“, S. 297 - 307

ist, welcher Schlüssel zum Ver- und welcher zum Entschlüsseln verwendet wurde, kann jeder, der im Besitz des öffentlichen Schlüssels der Zertifizierungsstelle ist, das Zertifikat entschlüsseln. Wenn die Entschlüsselung funktioniert, ist klar, von wem das Zertifikat ausgestellt wurde, da nur die Zertifizierungsstelle im Besitz ihres privaten Schlüssels ist[12]. Man sagt, die Zertifizierungsstelle hat das Zertifikat signiert. Da das Vertrauen in das Zertifikat nun hergestellt ist, kann der im Zertifikat enthaltene Schlüssel nun für die Kommunikation mit der Partei, welche das Zertifikat beantragt hat, genutzt werden.

Wie sichern PKIs die Kommunikation im Internet?

Nachdem nun klar geworden ist, wie PKIs, Zertifikate und Zertifizierungsstellen funktionieren, stellt sich die Frage, wie genau Angreifern ein Angriff unmöglich gemacht wird? Im Abschnitt zum MiTM Angriff wurden bereits die größten Schwachpunkte der Kommunikation über ein erstmal unsicheres Medium wie das Internet angesprochen. Einem Man-in-The-Middle ist es möglich, eine – zumindest anfangs – unverschlüsselte Kommunikation abzuhören und so auch den Schlüsselaustausch abzuhören, oder sich aktiv in die Kommunikation einzumischen und sich bei beiden Parteien als der gewünschte Kommunikationspartner zu präsentieren. So kann er jedenfalls sensible Informationen abgreifen, was eine PKI verhindern soll.

Um die Sicherheit zu demonstrieren, bemühen wir wieder Alice, Bob und Eve aus unserem vorherigen Beispiel. Zusätzlich gibt es noch eine Zertifizierungsinstanz, bei welcher Alice sowie Bob ihren öffentlichen Schlüssel mittels eines Zertifikates hinterlegt haben. Gestartet wird wieder am Anfang, Alice und Bob kennen sich also nicht persönlich, möchten nun aber sicher miteinander kommunizieren können. Eve hört jede Kommunikation ab und kann diese auch manipulieren. Es beginnt mit Alice, welche an den öffentlichen Schlüssel von Bob kommen möchte, um ihm eine verschlüsselte Nachricht zu senden. Dafür fragt sie Bob nach seinem Zertifikat, welches ja den öffentlichen Schlüssel enthält. Diese Bitte kann unverschlüsselt geschehen, da sie dem Angreifer Eve keine weiteren Informationen gibt, als das Alice mit Bob kommunizieren möchte. Dies kann Eve aber schon daran erkennen, dass überhaupt eine Datenübertragung zwischen den beiden stattfindet. Als Reaktion auf die Anfrage sendet Bob Alice sein Zertifikat, welches auch Bobs öffentlichen Schlüssel enthält.

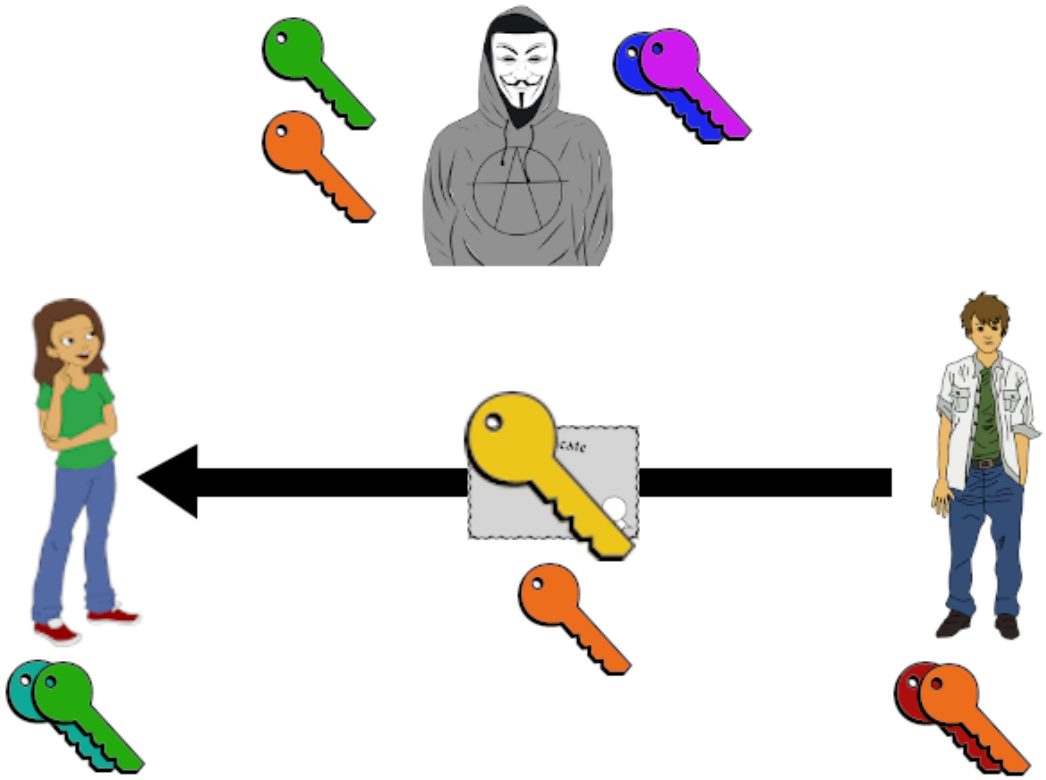

Abb. 7: Bob sendet Alice sein Zertifikat, welches mit dem privaten Schlüssel der Zertifizierungsinstanz signiert ist. Das Zertifikat enthält unter anderem Bobs öffentlichen Schlüssel

Hier ergibt sich für Eve das erste Mal die Möglichkeit, einen sinnvollen MiTM – Angriff zu probieren. Sie könnte – analog zum ungesicherten Schlüsselaustausch ohne Zertifizierungsinstanz – versuchen, Alice ihr selbst erstelltes Zertifikat zu übermitteln und Bobs verschwinden lassen. Dabei wird Eve jedoch Schwierigkeiten bekommen, da ihr Zertifikat nicht von einer Zertifizierungsinstanz signiert wurde. Zumindest im Normalfall. Sollte die PKI an einer Stelle unsicher sein und falsche Zertifikate signieren, kann es natürlich vorkommen, dass ein „falsches“ Zertifikat signiert wurde. Deswegen ist eine sichere PKI auch so wichtig für sichere Kommunikation. Normalerweise merkt Alice also durch das falsche Zertifikat, das ein Angreifer in die Kommunikation eingreift und bricht diese ab.

Jetzt, da Alice den öffentlichen Schlüssel von Bob hat, kann sie Bob eine verschlüsselte Nachricht senden, in der sie ihm einen symmetrischen Schlüssel für die weitere Kommunikation übermittelt.

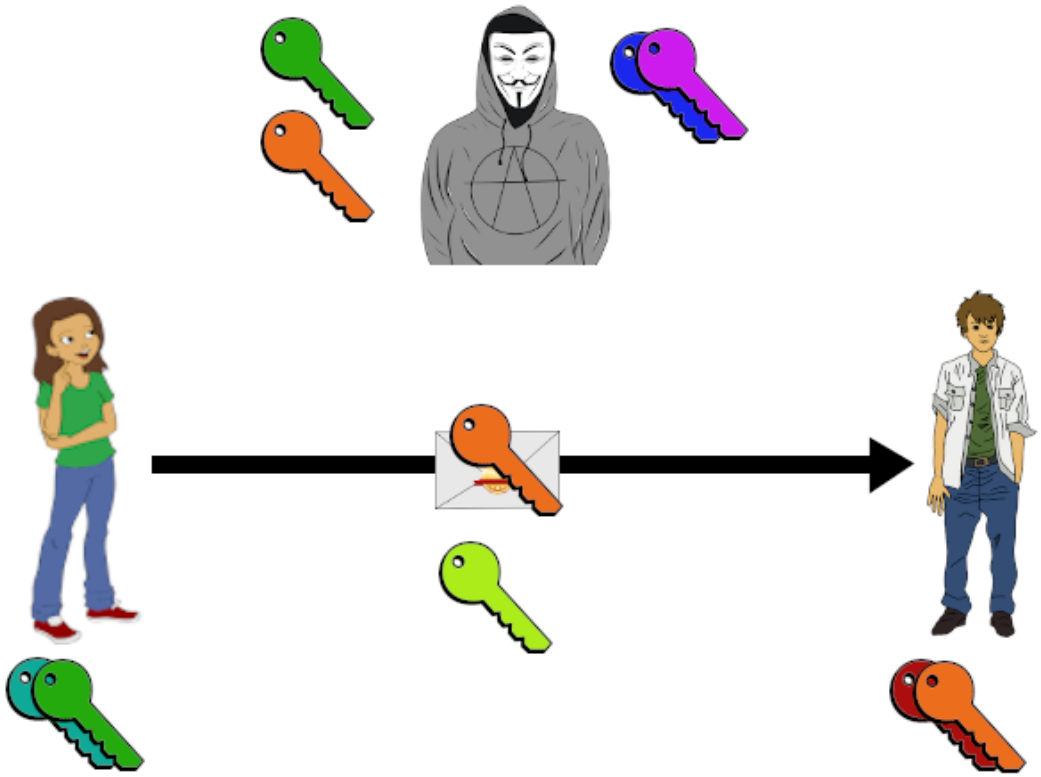

Abb. 8: Alice schickt Bob eine Nachricht, in der sie Bob einen Schlüssel für eine symmetrische Verschlüsselung vorschlägt. Diese Nachricht verschlüsselt sie mit Bobs öffentlichem Schlüssel

Eve kann diese Nachricht nicht lesen, da Eve nicht im Besitz von Bobs privaten Schlüssel ist. Eve kann wieder versuchen die Nachricht zu ersetzen, da sie ebenfalls ganz legal Bob nach seinem Zertifikat und damit seinem öffentlichen Schlüssel fragen kann. Somit kann Eve sich bei Bob als Alice ausgeben und Bob einen anderen symmetrischen Schlüssel übermitteln. Wenn Bob den Erhalt des symmetrischen Schlüssels an Alice bestätigen will, verschickt er diese Bestätigung bereits verschlüsselt mit dem symmetrischen Schlüssel.

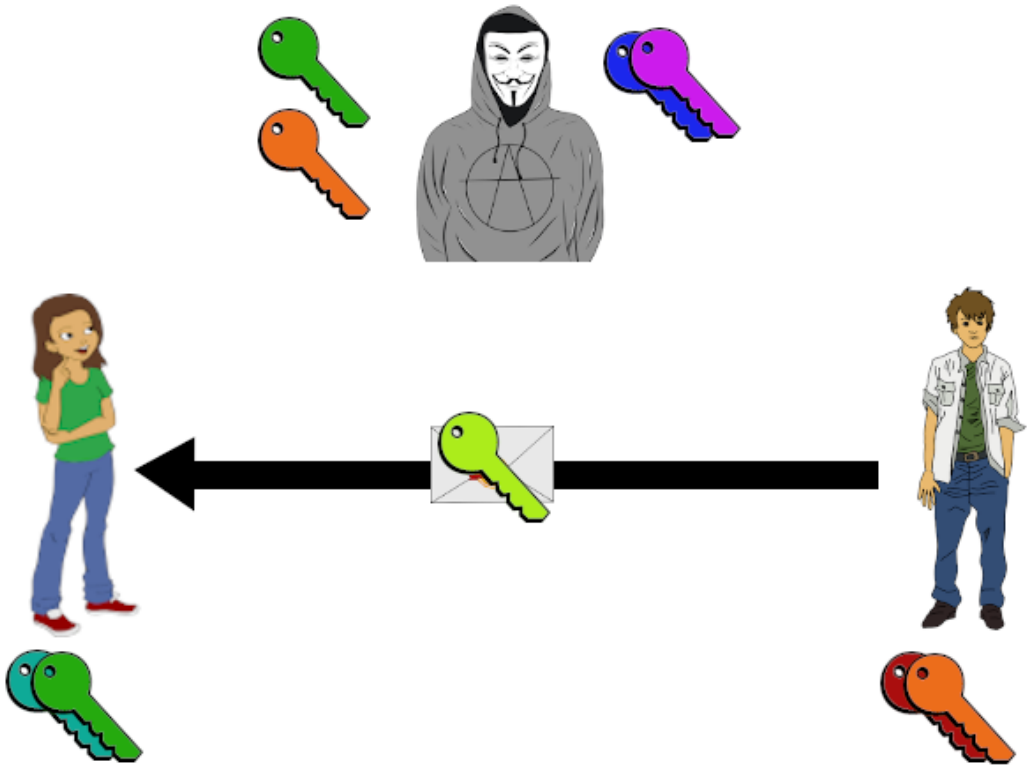

Abb. 9: Bob bestätigt den Erhalt der Nachricht und benutzt dabei den von Alice vorgeschlagenen symmetrischen Schlüssel

Diese Nachricht kann Eve zwar entschlüsseln, da sie aber nicht wissen kann, welchen Schlüssel Alice ursprünglich für die Kommunikation vorgesehen hat, kann Eve Alice die erwartete Bestätigung nicht zukommen lassen. Alice merkt wieder, dass ihre Kommunikation abgehört werden soll und verzichtet auf weiteren Datenaustausch.

Somit ist also klar, dass eine sichere PKI es einem Angreifer unmöglich macht, einen MiTM Angriff durchzuführen. Dies gilt aber nur, wenn die Kommunikationspartner sich korrekt verhalten. Oft benachrichtigt der Browser seinen Nutzer, wenn er eine unsichere Kommunikation eingeht. Wenn jeder Nutzer darauf achtet, haben Angreifer kaum eine Chance, sich in einen Datenaustausch einzuklinken und so an sensible Daten wie Bankdaten zu kommen – vorausgesetzt, die Implementation der Sicherheitsprotokolle wurde korrekt ausgeführt[13].

Simulation in Java

Dieser Facharbeit liegt ein umfangreiches Simulationsprogramm bei, mit dem MiTM Angriffe sowie die Arbeit von Zertifizierungsstellen gut nachvollzogen werden können.

[13] Beispiel frei konstruiert nach Gallenbacher J., „Abenteuer Informatik“, Kapitel 12

Dem Nutzer stehen vier Parteien zur Verfügung: Alice, Bob, Eve und eine Zertifizierungsinstanz. Alice und Bob sind jeder im Besitz eines RSA – Schlüsselpaares und sollen verschlüsselt miteinander kommunizieren. Da sich diese Facharbeit mit RSA und damit einhergehende MiTM Angriffe sowie den nötigen Sicherheitsvorkehrungen beschäftigt, wurde auf die Implementation eines Symmetrischen Schlüsselverfahrens zur weiterführenden Kommunikation, wie es in der Realität genutzt werden würde, verzichtet. Ebenfalls um die Implementation in einem gewissen Rahmen zu halten, wurde auf eine korrekte Blockung und auf korrektes Padding verzichtet. Die vorgenommenen Änderungen ändern nichts an der Funktion, am Ablauf oder Ähnlichem. In der Realität wäre die Implementation aber sehr unsicher und leicht zu knacken. Die Simulation wurde in der Programmiersprache Java verfasst und für die Grafische Benutzeroberfläche (kurz GUI) wurde die Java Bibliothek Swing verwendet. Um RSA durchzuführen, wird die Klasse BigInteger verwendet.

Die Figuren

Alice: Alice besitzt ein eigenes RSA Schlüsselpaar und möchte mit Bob verschlüsselt kommunizieren. Zu Beginn der Simulation kennt sie Bobs öffentlichen Schlüssel nicht.

Bob: Bob besitzt ein eigenes RSA Schlüsselpaar und möchte mit Alice verschlüsselt kommunizieren. Zu Beginn der Simulation kennt er Alices öffentlichen Schlüssel nicht.

Eve: Eve ist die Angreiferin. Sie besitzt ebenfalls ein eigenes RSA Schlüsselpaar. Wenn sie über die Checkbox oben rechts in der GUI aktiviert wurde, fängt sie jede Nachricht, die verschickt wird, ab. Eve kann mit allen ihr zur Verfügung stehenden Schlüssel versuchen eine Nachricht zu entschlüsseln und die Nachrichten weiterleiten. Eve wird automatisch aktiv geschaltet, wenn die Zertifizierungsinstanz aktiv ist.

Zertifizierungsinstanz: Die Zertifizierungsinstanz befindet sich nicht nur im Besitz eines eigenen RSA Schlüsselpaares, Alice und Bob haben bei ihr ein Zertifikat beantragt, sodass sie sich im Besitz der öffentlichen Schlüssel der beiden befindet. Diese kann sie dem jeweiligen gewünschten Kommunikationspartner zusenden. Dabei signiert sie die Nachricht mit ihrem eigenen privaten Schlüssel, sodass Eve sie zwar mit dem öffentlichen Schlüssel der Zertifizierungsinstanz entschlüsseln kann, nicht aber verändern. Alice und Bob können sich von der Echtheit der Nachricht überzeugen, indem sie den öffentlichen Schlüssel der Zertifizierungsinstanz verwenden, um Nachrichten zu entschlüsseln.

Fazit

Die Erfindung von asymmetrischer Kryptographie und RSA haben sichere Kommunikation über das Internet überhaupt erst ermöglicht. Ohne diese Erfindungen wäre das Internet, wie wir es heute kennen, nicht möglich, da keine Verbindung sicher wäre, wenn nicht vorher ein physischer Schlüsselaustausch stattgefunden hätte. Was das Internet aber in letzter Instanz sicher macht, sind die Zertifizierungsstellen. Wie in dieser Facharbeit gezeigt, wären ohne vertrauenswürdige Public Key Infrastrukturen eine Reihe von Angriffen möglich, welche die Sicherheit der asymmetrischen Verschlüsselung aushebeln könnten.

Anhang

Weiteres Material

Relevante Anhänge befinden sich im beiliegenden Ordner „Anhang“. Hier ist eine Auflistung des Anhangs:

1.) Simulation_RSA_und_MiTM, das Simulationsprogramm
2.) Dokumentation_Alice, die Klassendokumentation der Klasse Alice
3.) Dokumentation_Bob, die Klassendokumentation der Klasse Bob
4.) Dokumentation_Ca, die Klassendokumentation der Klasse Ca
5.) Dokumentation_CryptoHandler, die Klassendokumentation der Klasse CryptoHandler
6.) Dokumentation_Eve, die Klassendokumentation der Klasse Eve
7.) Dokumentation_GUI, die Klassendokumentation der Klasse GUI
8.) Dokumentation_Schluessel, die Klassendokumentation der Klasse Schluessel
9.) UML_Diagramm, das UML – Implementationsdiagramm zum Simulationsprogramm
10.) Ordner „Websites“, in welchem alle verwendeten Internetquellen als .html oder, im Falle der pdf-Dokumente, als .pdf gespeichert sind.

Literaturquellen

1.) Freiermuth K. et al., Einführung in die Kryptologie – Lehrbuch für Unterricht und Selbststudium, Wiesbaden [1] 2010
2.) Gallenbacher J., Abenteuer Informatik – IT zum Anfassen für alle von 9 bis 99- vom Navi bis Social Media, o.O.[4] 2017

Internetquellen

1.) A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, http://people.csail.mit.edu/rivest/Rsapaper.pdf , 01.04.2020
2.) BSI – Technische Richtlinie: Kryptografische Verfahren: Empfehlungen und Schlüssellängen, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf?__blob=publicationFile , 01.04.2020

3.) RSA-Verschlüsselung, http://www.inf.fh-flensburg.de/lang/krypto/grund/index.htm , 01.04.2020
4.) The Open Secret, https://www.wired.com/1999/04/crypto/ , 02.04.2020
5.) Verschlüsselungsverfahren im Überblick, https://www.ionos.de/digitalguide/server/sicherheit/verschluesselungsverfahren-ein-ueberblick/ , 02.04.2020
6.) Was ist ein Man-in-the-Middle-Angriff?, https://www.security-insider.de/was-ist-ein-man-in-the-middle-angriff-a-775391/ , 02.04.2020
7.) What is a man-in-the-middle attack?, https://us.norton.com/internetsecurity-wifi-what-is-a-man-in-the-middle-attack.html , 02.04.2020
8.) Was ist eine PKI (Public-Key-Infrastruktur)?, https://www.security-insider.de/was-ist-eine-pki-public-key-infrastruktur-a-696659/ , 02.04.2020
9.) Zertifizierungsstellen & Vertrauenshierachien, https://www.globalsign.com/de-de/ssl-information-center/zertifizierungsstellen-vertrauenshierarchien/ , 02.04.2020
10.) X.509-Zertifikat, https://www.computerweekly.com/de/definition/X509-Zertifikat , 02.04.2020

Bildquellen

Folgende Bildquellen wurden als Vorlage in dieser Facharbeit und dem beiliegenden Simulationsprogramm verwendet:

1.) Vorlage für Alice von https://pixabay.com/users/openclipart-vectors-30363/
2.) Vorlage für Bob von https://pixabay.com/users/openclipart-vectors-30363/
3.) Vorlage für den Brief von https://pixabay.com/de/users/mediengestalter-420494/
4.) Vorlage für die Zertifizierungsstelle von https://pixabay.com/users/clker-free-vector-images-3736/
5.) Vorlage für Eve von https://pixabay.com/users/worstwanted-4392060/
6.) Vorlage für die Schlüssel von https://pixabay.com/de/users/clker-free-vector-images-3736/
7.) Vorlage für das Siegel auf dem Brief von https://pixabay.com/de/users/creative_designer-1973841/
8.) Vorlage für das Zertifikat von https://pixabay.com/de/users/clker-free-vector-images-3736/

9.) Vorlage für das entschlüsselte Dokument von https://pixabay.com/de/users/openclipart-vectors-30363/

Alle Bilder sind unter der Pixabay Lizenz (https://pixabay.com/de/service/terms/#license) lizensiert.

Eigenständigkeitserklärung

Ich versichere, dass ich die vorliegende Arbeit einschließlich evtl. beigefügter Zeichnungen, Kartenskizzen, Darstellungen u. ä. m. selbstständig angefertigt und keine anderen als die angegebenen Hilfsmittel benutzt habe. Alle Stellen, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen sind, habe ich in jedem Fall unter genauer Angabe der Quelle deutlich als Entlehnung kenntlich gemacht.

Morenhoven, den 02.04.2020

Gez. Christopher Neugebauer