Asymmetrische Verfahren basierend auf dem diskreten Logarithmusproblem

8

Im vorherigen Kapitel wurde das RSA-Verfahren eingeführt und gezeigt, dass RSA auf der Schwierigkeit basiert, große Zahlen zu faktorisieren. Man sagt auch, dass das Faktorisierungsproblem die Einwegfunktion von RSA ist, vgl. auch Abschn. 6.1.2. Die Frage lautet nun: Können wir andere Einwegfunktionen finden, mit denen asymmetrische Verfahren konstruiert werden können? Es stellt sich heraus, dass neben RSA die meisten praktisch bedeutsamen asymmetrischen Algorithmen auf dem *diskreten Logarithmusproblem* (DLP) basieren.

In diesem Kapitel erlernen Sie

- den Diffie-Hellman-Schlüsselaustausch (DHKE, Diffy Hellmann Key Exchange),
- zyklische Gruppen, die für ein tieferes Verständnis des DHKE notwendig sind,
- das DLP, das von fundamentaler Bedeutung für viele praktische asymmetrische Verfahren ist,
- Verschlüsselung mit dem Elgamal-Verfahren.

Die Sicherheit vieler asymmetrischer Verfahren basiert darauf, dass es rechnerisch nicht möglich ist, Lösungen zum DLP zu finden. Bekannte Beispiele sind der DHKE und die Elgamal-Verschlüsselung, die wir beide in diesem Kapitel einführen werden. Auch die digitale Signatur nach Elgamal (vgl. Abschn. 8.5.1) und der digitale Signaturalgorithmus (vgl. Abschn. 10.2) basieren auf dem DLP, und Kryptoverfahren mit elliptischen Kurven (vgl. Abschn. 9.3) sind eine Verallgemeinerung des DLP.

Wir beginnen mit dem klassischen Diffie-Hellman-Protokoll, das erstaunlich einfach und mächtig ist. Das DLP ist in sog. *zyklischen Gruppen* definiert. Diese algebraischen Strukturen werden in Abschn. 8.2 eingeführt. Wir geben eine formale Definition des DLP zusammen mit einigen anschaulichen Beispielen und beschreiben Algorithmen für Angriffe auf das DLP. Mit diesem theoretischen Hintergrund schauen wir uns das Diffie-Hellman-Protokoll noch einmal genauer an und untersuchen dessen Sicherheit etwas formaler. Am Kapitelende wird eine DLP-Methode vorgestellt, mit der man auch Daten verschlüsseln kann, das Kryptosystem von Elgamal.

C. Paar, J. Pelzl, *Kryptografie verständlich*, eXamen.press, DOI 10.1007/978-3-662-49297-0_8

8.1 Diffie-Hellman-Schlüsselaustausch

Der DHKE wurde von Whitfield Diffie und Martin Hellman im Jahr 1976 [8] vorgestellt und war das erste veröffentlichte asymmetrische Verfahren überhaupt. Die beiden Erfinder wurden auch von den Arbeiten von Ralph Merkle beeinflusst. Der DHKE bietet eine praktische Lösung des Schlüsselverteilungsproblems, d. h. er ermöglicht zwei Parteien den Austausch eines gemeinsamen Geheimnisses über einen unsicheren Kanal[1]. Der DHKE ist eine sehr eindrucksvolle Anwendung des DLP, das wir in den nachfolgenden Abschnitten untersuchen werden. Der DHKE kommt in vielen weit verbreiteten Sicherheitsstandards wie dem Transport-Layer-Security- (TLS) bzw. SSL-Protokoll oder bei Internet Protocol Security (IPsec) zum Einsatz. Die wesentliche Idee hinter dem DHKE ist, dass das Potenzieren in $\mathbb{Z}_p^*$ mit p prim eine Einwegfunktion darstellt, und dass die Exponentiation kommutativ ist, d. h.

$$k = (\alpha^x)^y \equiv (\alpha^y)^x \bmod p.$$

Der Wert $k \equiv (\alpha^x)^y \equiv (\alpha^y)^x \bmod p$ ist das gemeinsame Geheimnis, das anschließend als Sitzungsschlüssel zwischen den beiden Parteien verwendet werden kann.

Schauen wir uns nun an, wie das DHKE-Protokoll über $\mathbb{Z}_p^*$ funktioniert. Ziel des Protokolls ist es, dass zwei Teilnehmer, Alice und Bob, einen gemeinsamen geheimen Schlüssel über einen unsicheren Kanal vereinbaren. Optional kann es eine weitere vertrauenswürdige Partei geben, die die öffentlichen Parameter für den Schlüsselaustausch wählt. Es ist aber auch möglich, dass Alice oder Bob die öffentlichen Parameter erzeugen. Streng genommen besteht der DHKE aus zwei Protokollen: dem Set-up-Protokoll und dem Hauptprotokoll, das den eigentlichen Schlüsselaustausch durchführt. Das Set-up-Protokoll besteht aus den folgenden Schritten:

Diffie-Hellman-Set-up

1. Wähle eine große Primzahl p
2. Wähle eine ganze Zahl $\alpha \in \{2, 3, \ldots, p-2\}$
3. Veröffentliche p und α

Diese beiden Werte werden manchmal als *Domain-Parameter* bezeichnet. Wenn Alice und Bob beide die öffentlichen Parameter p und α aus der Set-up-Phase erhalten haben, können sie einen gemeinsamen geheimen Schlüssel k mit dem folgenden Protokoll berechnen:

[1] Der Kanal muss noch authentisiert werden, was später in diesem Buch besprochen wird.

Diffie-Hellman-Schlüsselaustausch

Alice		**Bob**
Wähle $a = k_{\text{pr},A} \in \{2, \ldots, p-2\}$		Wähle $b = k_{\text{pr},B} \in \{2, \ldots, p-2\}$
Berechne $A = k_{\text{pub},A} \equiv \alpha^a \bmod p$		Berechne $B = k_{\text{pub},B} \equiv \alpha^b \bmod p$
	$\xrightarrow{k_{\text{pub},A}=A}$	
	$\xleftarrow{k_{\text{pub},B}=B}$	
$k_{AB} = k_{\text{pub},B}^{k_{\text{pr},A}} \equiv B^a \bmod p$		$k_{AB} = k_{\text{pub},A}^{k_{\text{pr},B}} \equiv A^b \bmod p$

Es folgt nun der Beweis, dass dieses überraschend einfache Protokoll korrekt ist, d. h. dass Alice und Bob tatsächlich den gleichen Sitzungsschlüssel k_{AB} berechnen.

Beweis Alice berechnet:

$$B^a \equiv (\alpha^b)^a \equiv \alpha^{ab} \bmod p,$$

während Bob die folgende Berechnung durchführt:

$$A^b \equiv (\alpha^a)^b \equiv \alpha^{ab} \bmod p,$$

und daher besitzen Alice und Bob beide den gleichen Sitzungsschlüssel $k_{AB} \equiv \alpha^{ab} \bmod p$. Der Schlüssel kann nun für die sichere Kommunikation zwischen Alice und Bob verwendet werden, z. B. kann k_{AB} der Schlüssel für eine symmetrische Chiffre wie AES oder 3DES sein. □

Wir schauen uns nun ein einfaches Beispiel mit kleinen Zahlen an.

Beispiel 8.1 Die Diffie-Hellman-Domain-Parameter sind $p = 29$ und $\alpha = 2$. Das Protokoll läuft wie folgt ab:

Alice		**Bob**
Wähle $a = k_{\text{pr},A} = 5$		Wähle $b = k_{\text{pr},B} = 12$
$A = k_{\text{pub},A} = 2^5 \equiv 3 \bmod 29$		$B = k_{\text{pub},B} = 2^{12} \equiv 7 \bmod 29$
	$\xrightarrow{A=3}$	
	$\xleftarrow{B=7}$	
$k_{AB} = B^a \equiv 7^5 = 16 \bmod 29$		$k_{AB} = A^b = 3^{12} \equiv 16 \bmod 29$

Wie man sehen kann, haben beide Parteien den Wert $k_{AB} = 16$ berechnet, der nun z. B. der Sitzungsschlüssel für eine symmetrische Verschlüsselung sein kann.

Die Berechnungen, die für den DHKE notwendig sind, ähneln stark denen von RSA. Während der Set-up-Phase erzeugen wir p mithilfe von probabilistischen Suchalgorithmen für Primzahlen (vgl. Abschn. 7.6). Wie in Tab. 6.1 dargestellt, sollte p ebenso wie der RSA-Modul n eine Länge von mindestens 2048 Bit haben, um Langzeitsicherheit zu bieten. Die Zahl α muss eine spezielle Eigenschaft erfüllen: Sie muss ein sog. primitives Element sein. Was dies bedeutet, besprechen wir in den folgenden Abschnitten. Der in dem Protokoll berechnete Sitzungsschlüssel k_{AB} hat die gleiche Bitlänge wie p. Wenn weniger Bit benötigt werden, z. B. für einen AES-Schlüssel, werden oft nur die 128 höchstwertigen Bits von k_{AB} genutzt. Als Alternative hierzu wendet man manchmal auch eine Hash-Funktion auf k_{AB} an und verwendet die Ausgabe als symmetrischen Schlüssel.

Während des eigentlichen DHKE-Protokolls müssen zunächst die privaten Schlüssel a und b gewählt werden. Diese sollten aus einem echten Zufallszahlengenerator stammen, um zu verhindern, dass ein Angreifer diese erraten kann. Für die Berechnung der öffentlichen Schlüssel A und B sowie für die Berechnung des Sitzungsschlüssels können beide Parteien den Square-and-Multiply-Algorithmus verwenden. Die öffentlichen Schlüssel werden üblicherweise vorausberechnet. Während eines Schlüsselaustauschs besteht der Hauptrechenaufwand daher aus der Exponentiation für den Sitzungsschlüssel. Im Allgemeinen ist der Rechenaufwand für RSA und DHKE vergleichbar, da die Bitlängen ähnlich sind und beide Exponentationen benötigen. Die Beschleunigung von RSA durch Verwendung kurzer Exponenten, die in Abschn. 7.5 beschrieben ist, ist jedoch nicht auf den DHKE anwendbar.

Bisher haben wir das klassische DHKE-Protokoll in der Gruppe $\mathbb{Z}_p^*$ gezeigt, wobei p eine Primzahl ist. Das Protokoll kann verallgemeinert werden, insbesondere auf Gruppen von Punkten auf elliptischen Kurven. Man spricht hier von Elliptische-Kurven-Kryptografie („elliptic curve cryptography“, ECC), die insbesondere in den letzten 10 Jahren in der Praxis immer beliebter geworden ist und in Kap. 9 eingeführt werden wird. Um DHKE und verwandte Verfahren wie ECC oder die Elgamal-Verschlüsselung besser zu verstehen, wird in den nachfolgenden Abschnitten das DLP eingeführt. Dieses Problem bildet die mathematische Grundlage für den DHKE. Nachdem wir das DLP besprochen haben, werden wir uns noch einmal dem DHKE widmen und dessen Sicherheit diskutieren.

8.2 Ein wenig abstrakte Algebra

In diesem Abschnitt führen wir einige Grundlagen der abstrakten Algebra ein, insbesondere Gruppen, Untergruppen, endliche Körper und zyklische Gruppen. Diese sind wesentlich für das Verständnis des DLP und von Kryptoverfahren, die auf ihm basieren.