Fortgeschrittene Techniken der Kryptographie

Jürgen Fuß

Episode 14: CRYSTALS

HAGENBERG | LINZ | STEYR | WELS

FH OBERÖSTERREICH UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Polynome und Restklassen

Restklassen und Polynome

- Rechnen in $\mathcal{R} := \mathbb{Z}_q[x]/(x^{256}+1)$.

$x^{256} + 1 = 0$
$x^{256} = -1$

$q \in \mathbb{P}$

- Restklassen modulo q nicht als ganze Zahlen zwischen 0 und $q-1$ angeschrieben, sondern als ganze Zahlen zwischen $-q/2$ und $q/2$.

$q = 7$ $\quad 0, 1, 2, \dots 6$

$-3, -2, -1, 0, 1, 2, 3$
$4 \quad 5 \quad 6$

- $x^{256}+1$ ist nicht irreduzibel, wir brauchen aber keine Divisionen.
- Wir bilden Vektoren ($\mathcal{R}^n$) und Matrizen ($\mathcal{R}^{m\times n}$) aus solchen „Zahlen".
- Ist $p := a_0 + a_1x + \cdots + a_{255}x^{255}$ ein Element von $\mathcal{R}$, dann wird dessen **Norm** definiert als der Betrag des Koeffizienten a_i mit dem größten Betrag.
- Bildet man einen Vektor solcher Polynome, so wird dessen **Norm** (Länge) als die größte auftauchende Norm in den Koordinaten des Vektors definiert.[1]
- In den folgenden Abschnitten werden für **kurze** Vektoren (Vektoren mit kleiner Norm) griechische Buchstaben verwendet, damit diese einfacher zu erkennen sind.

[1] Das ist also der größte Koeffizient, der irgendwo auftaucht.

Das Problem

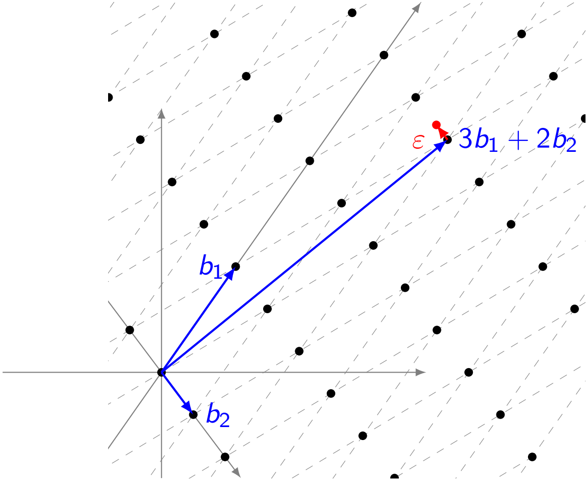
Gegeben sei eine Matrix M.
Wählt man zwei Vektoren α und ε mit kleiner Norm (also kurze Vektoren), so lässt sich einfach

$$a := M \cdot \alpha + \varepsilon$$

berechnen.
Umgekehrt ist es aber sehr schwierig, kurze Vektoren α und ε zu finden, so dass $a = M \cdot \alpha + \varepsilon$, wenn M und a gegeben sind.[2]

[2] Beachte: Irgendwelche Vektoren x und e zu finden, so dass $a = M \cdot x + e$ gilt, ist sehr einfach: man wähle irgendeinen kurzen Vektor x und berechne $e := a - M \cdot x$. Nur ist der Vektor e dann meist nicht kurz. Die Aufgabe, zwei kurze Vektoren α und ε zu finden, führt auf ein sogenanntes Gitterproblem. Diese Art von Problemen scheint selbst für Quantencomputer schwierig zu lösen zu sein.

Graphische Veranschaulichung



Ist M die Matrix $\begin{pmatrix} | & | \\ b_1 & b_2 \\ | & | \end{pmatrix}$ und $\alpha = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$, dann ist $M \cdot \alpha$ der Vektor $3b_1 + 2b_2$ und $M \cdot \alpha + \varepsilon$ liegt ein klein wenig daneben.

CRYSTALS Dilithium – Post-quantum Signatures

CRYSTALS Dilithium – Schlüsselerzeugung

In diesem Verfahren ist $q := 8380417$ und $\mathcal{R} := \mathbb{Z}_q[x]/(x^{256}+1)$.

Schlüsselerzeugung: Alice wählt eine zufällige Matrix $M \in \mathcal{R}^{6\times 5}$. Weiterhin wählt sie zwei Vektoren $\alpha \in \mathcal{R}^5$ und $\varepsilon \in \mathcal{R}^6$ zufällig, deren Norm höchstens 4 ist. Schließlich berechnet sie

Matrix Vektor

Vektor

$$a := M \cdot \alpha + \varepsilon.$$

Der **Public Key** ist dann (M, a). Der dazugehörige **Private Key** ist α.

CRYSTALS Dilithium – Signieren

Signieren: Alice wählt einen Vektor $k \in \mathcal{R}^5$ zufällig, dessen Norm höchstens 2^{19} ist. Alice berechnet nun $r := \mathrm{high}(M \cdot k)$, die höchstwertigen Bits aller Koordinaten des Vektors $M \cdot k$ (direkt als Bitfolge interpretiert). Nun werden r und m zusammen gehasht, das Ergebnis wird kodiert als ein Polynom $\zeta \in \mathcal{R}$, das genau 49 Koeffizienten hat, die den Wert 1 oder -1 haben und dessen restliche Koeffizienten 0 sind.[3] Schließlich wird $s := k + \zeta \cdot \alpha$ berechnet. Kompakter also:

die zu signierende Nachricht gehasht:
$h := h(m)$

$$r := \mathrm{high}(M \cdot k), \quad \to \text{Bits}$$
$$\zeta := H(r, m), \quad \to \text{kleines Polynom}$$
$$s := \underbrace{k}_{\text{Vektor}} + \underbrace{\zeta \cdot \alpha}_{\text{kurzer Vektor}}. \quad \to \text{Vektor}$$

Die **Signatur** ist (ζ, s).

[3] Es wird hier nicht darauf eingegangen, wie dies genau geschieht.

CRYSTALS Dilithium – Verifizieren

Matrix Vektor Polynom Vektor

Verifizieren: Bob prüft, dass die Norm von s nicht zu groß ist und berechnet dann $r := \text{high}(M \cdot s - \zeta \cdot a)$ und $h := H(m)$. Damit wird $H(r, h)$ berechnet und abschließend mit ζ verglichen.

Tatsächlich erhält man beim Verifizieren dasselbe a wie beim Signieren, denn

$$M \cdot s - \zeta \cdot a = M \cdot (k + \zeta \cdot \alpha) - \zeta \cdot (M \cdot \alpha + \varepsilon) = M \cdot k + \zeta \cdot M \cdot \alpha - \zeta \cdot M \cdot \alpha - \zeta \cdot \varepsilon = M \cdot k - \zeta \cdot \varepsilon.$$

Da sowohl in ζ als auch in ε nur kleine Koeffizienten vorkommen, beeinflussen diese die höchstwertigen Bits nicht. Daher ist

$$r = \text{high}(M \cdot s - \zeta \cdot t) = \text{high}(M \cdot k - \zeta\varepsilon) = \text{high}(M \cdot k).$$

CRYSTALS Kyber – Post-quantum Key Encapsulation

CRYSTALS Kyber – Schlüsselerzeugung

In diesem Verfahren ist $q := 3329$ und $\mathcal{R} := \mathbb{Z}_q[x]/(x^{256} + 1)$.

Schlüsselerzeugung: Alice wählt eine zufällige Matrix $M \in \mathcal{R}^{3\times 3}$. Weiterhin wählt sie zwei Vektoren $\alpha \in \mathcal{R}^3$ und $\varepsilon \in \mathcal{R}^3$ zufällig, deren Norm höchstens 2 ist. Schließlich berechnet sie

$$a := M \cdot \alpha + \varepsilon.$$

Der **Public Key** ist dann (M, a). Der dazugehörige **Private Key** ist α.

CRYSTALS Kyber – Verschlüsseln

Verschlüsseln: Um einen 256 Bit langen Schlüssel zu verschlüsseln, wird dieser zunächst als ein Element $\kappa \in \mathcal{R}$ dargestellt, nämlich als jenes Polynom, dessen Koeffizienten die Schlüsselbits sind. Daraus erhält man $k := \lfloor q/2 \rfloor \cdot \kappa$.[4]

Bob wählt Vektoren $\beta, \zeta \in \mathcal{R}^3$ sowie $\gamma \in \mathcal{R}$ zufällig, deren Norm höchstens 2 ist. Er berechnet nun

Vektoren

$$u := M^\mathsf{T} \cdot \zeta + \beta,$$

Polynome

$$v := \underbrace{a^\mathsf{T} \cdot \zeta}_{\text{Skalarprodukt}} + k + \gamma.$$

Es ergibt sich als **Chiffrat** das Paar (u, v).

[4] Die Multiplikation von k mit $\lfloor q/2 \rfloor$ führt dazu, dass die Koeffizienten dieses Polynoms entweder den (betragsmäßig) kleinsten Wert 0 oder oder größten Wert $\lfloor q/2 \rfloor = 1664$ haben.

CRYSTALS Kyber – Entschlüsseln

Entschlüsseln: Alice berechnet

$$k' := v - \alpha^\intercal \cdot u.$$

CRYSTALS Kyber – Korrektheit

Beim Entschlüsseln ergibt sich die ursprüngliche Nachricht, denn

$$
\begin{aligned}
k' &= v - \alpha^\intercal \cdot u \\
&= a^\intercal \cdot \zeta + k + \gamma - \alpha^\intercal \cdot (M^\intercal \cdot \zeta + \beta) && \text{(Einsetzen von } u \text{ und } v\text{)} \\
&= a^\intercal \cdot \zeta + k + \gamma - \alpha^\intercal \cdot M^\intercal \cdot \zeta - \alpha^\intercal \cdot \beta && \text{(Ausmultiplizieren)} \\
&= a^\intercal \cdot \zeta + k + \gamma - (M \cdot \alpha)^\intercal \cdot \zeta - \alpha^\intercal \cdot \beta && (\alpha^\intercal M^\intercal = (M\alpha)^\intercal) \\
&= a^\intercal \cdot \zeta + k + \gamma - (a - \varepsilon)^\intercal \cdot \zeta - \alpha^\intercal \cdot \beta && (a = M\alpha + \varepsilon) \\
&= a^\intercal \cdot \zeta + k + \gamma - a^\intercal \cdot \zeta + \varepsilon^\intercal \cdot \zeta - \alpha^\intercal \cdot \beta && \text{(Ausmultiplizieren)} \\
&= \lfloor q/2 \rfloor \cdot \kappa + \underbrace{\gamma + \varepsilon^\intercal \cdot \zeta - \alpha^\intercal \cdot \beta}_{\text{nur kleine Koeffizienten}} .
\end{aligned}
$$

In den Ausdrücken β, $\varepsilon^\intercal \cdot \zeta$ und $\alpha^\intercal \cdot \beta$ kommen nur kleine Koeffizienten vor. Werden diese Werte zum Polynom $\lfloor q/2 \rfloor \cdot \kappa$ addiert oder von diesem subtrahiert, können diese kleinen „Fehler“ sehr einfach korrigiert werden, um die Schlüsselbits zu erhalten.