Implementierung des McEliece-Kryptosystems auf P4-programmierbaren Switches

Roman Wetenkamp[1]

Abstract: This is a brief overview of the paper, which should be 70 to 150 words long and include the most relevant points. This has to be a single paragraph.

Keywords: McEliece; Code-basierte Kryptography; Programmierbare Switches

1 Einführung

Seitdem die Aussicht auf einen Quantencomputer, der bisher in der Kryptographie verwendete Probleme wie jenes des *diskreten Logarithmus* oder der *Primfaktorzerlegung* für große Zahlen deutlich effizienter brechen kann als Computer mit herkömmlichen Prozessoren, immer realistischer wird, nimmt die Forschung zur *Post-Quanten-Kryptographie* stetig zu. Neben Verfahren, die auf Gittern oder multivariaten Polynomen basieren, gelten auch einige *code-basierte Verfahren* als quantensicher und sind somit beispielsweise Teil eines Standardisierungsprozesses des National Institute of Standards and Technology (NIST) in den USA [Al22a]. Diese Kryptosysteme nutzen lineare, fehlerkorrigierende Codes wie *verallgemeinerte Reed-Solomon-Codes* oder *Goppa-Codes* und deren *Dekodierproblem* aus, das darin besteht, dass es hinreichend schwierig ist, von einem durch eine permutierte und verwürfelte Generatormatrix erzeugten Codewort auf die ursprünglich verwendete Generatormatrix zurückzuschließen [BMT78]. Eines der ersten und bekanntesten code-basierten Kryptosysteme ist jenes von McEliece [Mc78], dessen Implementierung die erste Hauptkomponente dieser Arbeit darstellt.
Die zweite Komponente ergibt sich durch *programmierbare Switches*. Während Verschlüsselungsoperationen von Netzwerkdatenpaketen typischerweise im Endgerät (beispielsweise einem Server oder einem Client) ausgeführt werden, ergibt sich mit programmierbaren Switches eine Möglichkeit, die Daten im Switch zu verschlüsseln.
Diese Arbeit widmet sich der Fragestellung, inwiefern das McEliece-Kryptosystem in programmierbaren Switches implementiert werden kann. Zunächst werden in Abschnitt **??** die Hintergründe beider Technologien erläutert, bevor in Abschnitt **??** die Umsetzung in der Domänen-spezifischen und nicht Turing-vollständigen Sprache *P4* [Bo14] erfolgt. Die Ergebnisse werden abschließend anhand einer Switch-Emulation reflektiert und zusammengefasst.

[1] Duale Hochschule Baden-Württemberg Mannheim
s200376@student.dhbw-mannheim.de

2 Ausgangslage und Beitrag

Fundament dieser Arbeit in kryptographischer Hinsicht sind die Darstellung des McEliece-Kryptosystems in [Mc78], die Definition von *Goppa*-Codes in [Be73] und die Implementierung des McEliece-Kryptosystems basierend auf quasi-zyklischen MDPC-Codes in [MOG15], die auf [Mi13] basiert. Die Idee, Ver- und Entschlüsselung auf programmierbaren Switches durchzuführen, wird in [Ch20] anhand des AES-Verfahrens dargestellt. [Al22b] liefert eine Übersicht über bisherige Arbeiten zu diesem Ansatz, die keine Verfahren enthält, die Teil des *Post Quantum Cryptography*-Standardisierungsprozesses sind.

Diese Arbeit entwirft eine Implementierung des ursprünglichen McEliece-Kryptosystems in der Sprache *P4*, evaluiert jene anhand der Switch-Emulation *bmv2* und diskutiert Limitationen neben der praktischen Umsetzbarkeit des Entwurfs.

3 Hintergründe

3.1 McEliece-Kryptosystem

Das McEliece-Kryptosystem ist eine Blockchiffre auf der Basis von linearen fehlerkorrigierenden Codes, wobei ursprünglich *Goppa*-Codes, in aktuellen Verfahren meist quasi-zyklische MDPC-Codes vorgeschlagen werden [Mc78] [Mi13].

Die zu einem gewählten $[n, k]$-Code C mit Fehlerkorrekturkapazität t gehörige Generatormatrix wird als G und die Paritätsprüfmatrix als H bezeichnet. Dann ist der öffentliche Schlüssel K_{pub} definiert durch das Matrixprodukt aus einer nicht-singulären und dichten $k \times k$-Verwürfelungsmatrix S, der Generatormatrix G und einer $n \times n$-Permutationsmatrix P, also $K_{pub} = S \cdot G \cdot P$ [Mc78, S. 114].

Die Verschlüsselungsoperation erfolgt nun, in dem jeder Klartextblock der Länge k mit K_{pub} multipliziert und ein zufälliger Störvektor z mit Länge n und Gewicht t addiert wird [Mc78].

Algorithm 1 Verschlüsselungsalgorithmus des McEliece-Kryptosystems (nach [Mc78])

Require: K_{pub}, Eingabedaten m mit l Komponenten aus $\mathbb{F}_{2^f}$
1: Teile m in Blöcke der Länge k (ggf. mit Nullen auffüllen)
2: **for** u in $m[]$ **do**
3: Wähle zufälligen Störvektor z mit Länge n und Gewicht t
4: $x_u \leftarrow u \cdot K_{pub} + z$
5: **end for**
6: **return** x

Ein Chiffrat wird entschlüsselt, indem die Permutation P invertiert, ein zur Code-Klasse gehöriger Dekodierlagorithmus angewandt und dessen Ergebnis abschließend mit dem

Algorithm 2 Entschlüsselungsalgorithmus des McEliece-Kryptosystems (nach [Mc78])

Require: Übertragenes Wort x, *Goppa*-Code-Parameter, P, S
1: $(x' \leftarrow x \cdot P^{-1}) \Rightarrow x' \in C$
2: $u' \leftarrow$ Ergebnis des Dekodieralgorithmus von Patterson auf x'
3: $u = u' \cdot S^{-1}$
4: **return** u

Inversen der Verwürfelungsmatrix S multipliziert wird [Mc78]. Für die Implementierung der Verschlüsselung sind entsprechend Additionen, Multiplikationen und die Inversionenbildung bezüglich endlichen Körpern erforderlich. Für die Entschlüsselung müssen zudem lineare Gleichungssysteme gelöst werden können.

3.2 Programmierbare Switches

Ein Switch empfängt Datenpakete und trifft anhand verschiedener Prüfungen Entscheidungen zur Weiterleitung („Match-Action“) des Pakets über die Ausgänge des Switches. Er ist dann programmierbar, wenn die Verarbeitungslogik („Data Plane“) systematisch, schnell und vollumfänglich durch die Kontrolllogik („Control Plane“) rekonfiguriert werden kann [BR18, S. 2], die Wege zur Weiterleitungsentscheidung folglich programmiert werden können. Abbildung 1 zeigt einen schematischen Ablauf. Die logische Trennung beider Ebenen ist Voraussetzung für die Programmierbarkeit.

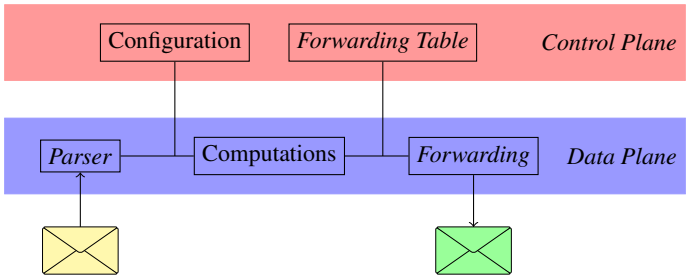

Abb. 1: Schematische Darstellung der *Control Plane* und *Data Plane* eines Switches (eigene Darstellung nach [BR18, S. 2])

Literatur

[Al22a] Alagic, G.; Apon, D.; Cooper, D.; Dang, Q.; Dang, T.; Kelsey, J.; Lichtinger, J.; Liu, Y.-K.; Miller, C.; Moody, D.; Peralta, R.; Perlner, R.; Robinson, A.; Smith-Tone, D.: Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process, Bericht, National Institute of Standards und Technology, 2022.

[Al22b] AlSabeh, A.; Khoury, J.; Kfoury, E.; Crichigno, J.; Bou-Harb, E.: A survey on security applications of P4 programmable switches and a STRIDE-based vulnerability assessment. Computer Networks 207/, S. 108800, 2022, ISSN: 1389-1286, URL: https://www.sciencedirect.com/science/article/pii/S1389128622000287.

[Be73] Berlekamp, E.: Goppa codes. IEEE Transactions on Information Theory 19/5, S. 590–592, 1973.

[BMT78] Berlekamp, E.; McEliece, R.; van Tilborg, H.: On the inherent intractability of certain coding problems (Corresp.) IEEE Transactions on Information Theory 24/3, S. 384–386, 1978.

[Bo14] Bosshart, P.; Daly, D.; Gibb, G.; Izzard, M.; McKeown, N.; Rexford, J.; Schlesinger, C.; Talayco, D.; Vahdat, A.; Varghese, G.; Walker, D.: P4: Programming Protocol-Independent Packet Processors. SIGCOMM Comput. Commun. Rev. 44/3, S. 87–95, Juli 2014, ISSN: 0146-4833, URL: https://doi.org/10.1145/2656877.2656890.

[BR18] Bifulco, R.; Rétvári, G.: A Survey on the Programmable Data Plane: Abstractions, Architectures, and Open Problems. In: 2018 IEEE 19th International Conference on High Performance Switching and Routing (HPSR). S. 1–7, 2018.

[Ch20] Chen, X.: Implementing AES Encryption on Programmable Switches via Scrambled Lookup Tables. In: Proceedings of the Workshop on Secure Programmable Network Infrastructure. SPIN '20, Association for Computing Machinery, Virtual Event, USA, S. 8–14, 2020, ISBN: 9781450380416, URL: https://doi.org/10.1145/3405669.3405819.

[Mc78] McEliece, R. J.: A Public-Key Cryptosystem Based On Algebraic Coding Theory. The Deep Space Network Progress Report 42-44/, NASA Code 310-10-67-11, 1978, URL: https://ntrs.nasa.gov/api/citations/19780016269/downloads/19780016269.pdf, Stand: 15. 02. 2023.

[Mi13] Misoczki, R.; Tillich, J.-P.; Sendrier, N.; Barreto, P. S. L. M.: MDPC-McEliece: New McEliece variants from Moderate Density Parity-Check codes. In: 2013 IEEE International Symposium on Information Theory. S. 2069–2073, 2013.

[MOG15] Maurich, I. V.; Oder, T.; Güneysu, T.: Implementing QC-MDPC McEliece Encryption. ACM Trans. Embed. Comput. Syst. 14/3, Apr. 2015, ISSN: 1539-9087, URL: https://doi.org/10.1145/2700102.