# Hypercomplex: abstract & fast header-only C++ template library for lattice-based cryptosystems in high-dimensional algebras

Maciek Bak [iD] [1]

1 Independent Researcher, Switzerland

Summary

The following work presents a C++ library that is dedicated to performing arbitrary-precise calculations on hypercomplex numbers from the Cayley-Dickson algebras (Schafer, 2017). Basic arithmetical operations as well as a few miscellaneous functions are implemented. Its most important feature is the support for encryption/decryption procedures for public-key lattice-based cryptosystems in high-dimensional algebras over truncated polynomial rings.

Statement of need

This is a highly specialised software package aimed mostly at computational mathematicians and scientists who operate on high-dimensional numbers, need to carry out arbitrary-precise calculations, or whose focus is the study of lattice-based, post-quantum cryptography (PQC). The library is well suited for a wide range of computationally-challenging projects, from investigating general algebraic properties per se to applied research where a hypercomplex framework serves merely as a mean to an end (as in previously mentioned cryptosystems).

Key features

- As a header-only C++ template code, its greatest advantage is the combination of speed, generic programming, and convenience for the end user. An open source license together with a template specialisation mechanism allows contributors to add support for custom objects, define specific functions, and extend the scope of the library.
- The most important specialisation, already included in the library itself, is the introduction of operations in hypercomplex algebras over truncated polynomial rings. These allow for many cryptographic applications as described in a dedicated section below.
- A template class specialisation introduces support for arbitrary high precision of calculations via the GNU MPFR library (Fousse et al., 2005), for which the operators have been overloaded such that all the instructions are carried out on specific data structures.
- State of the art technology for software engineering:
  - CI/CD mechanism set up with GitHub Actions: automatic tests for library installation, source code inclusion, compilation, and execution,
  - extensive unit testing with the Catch2 framework (Hořeňovský, 2020) alongside code coverage measurement uploaded to Codecov; current coverage: 100%,
  - source code linting with cpplint (Google Inc., last accessed: 16 Feb 2021) - Google code style enforced, and
  - automatic documentation generation and hosting on GitHub Pages: building via Doxygen (Heesch, 2021), publishing via GitHub Actions.

Cryptographic applications

In the following section we shall describe the mathematical foundations for the previously mentioned family of cyptosystems. Consider a polynomial convolution ring $\mathcal{R} = \mathbb{Z}[x]/(x^N - 1)$ with $N > 2$ being prime. Let $\mathcal{R}_p$ and $\mathcal{R}_q$ denote derived modular structures with coefficients from $\mathbb{Z}/\mathbb{Z}_p$ and $\mathbb{Z}/\mathbb{Z}_q$, respectively. Every element of $\mathcal{R}$, $\mathcal{R}_p$, $\mathcal{R}_q$ may be writted down as:

$$f = \sum_{i=0}^{N-1} f_i x_i \equiv [f_0, \ldots, f_{N-1}] \tag{1}$$

where the addition operation $+$ refers to a regular element-wise addition of coefficients (modular for $\mathcal{R}_p$ and $\mathcal{R}_q$). Multiplication $\star$ within this structure is defined as:

$$f \star g = \sum_{i=0}^{k} f_i g_{k-i} + \sum_{i=k+1}^{N-1} f_i g_{N+k-i} \tag{2}$$

with a final reduction modulo p or q in the modular quotient rings.

Based on the above, let us pick an integer $\lambda \geq 0$ and define three corresponding algebras, generated by the Cayley-Dickson process:

$$\mathcal{A}^\lambda = \{x_0 + \sum_{i=1}^{2^\lambda - 1} x_i e_i | x_i \in \mathcal{R}\}$$

$$\mathcal{A}_p^\lambda = \{x_0 + \sum_{i=1}^{2^\lambda - 1} x_i e_i | x_i \in \mathcal{R}_p\} \tag{3}$$

$$\mathcal{A}_q^\lambda = \{x_0 + \sum_{i=1}^{2^\lambda - 1} x_i e_i | x_i \in \mathcal{R}_q\}$$

where the addition operation $+$ refers to ring addition defined above.

Note that $\forall x \in \mathcal{A}^\lambda : x = (a, b)$, where $a, b \in \mathcal{A}^{\lambda-1}$.

Multiplication $\times$ is defined recursively based on the conjugation operation $*$ as below:

$$\mathcal{A}^\lambda \ni x^* = \begin{cases} x, & \lambda = 0 \\ (a, b)^* = (a^*, -b), & \lambda > 0 \end{cases} \tag{4}$$

$$(a, b) \times (c, d) = (ac - d^* b, da + bc^*)$$

which as well holds for the modular algebras, given a final reduction modulus p or q.

Based on the above, let us define a general scheme for hypercomplex-based cyptosystems. Having agreed on (N, p, q), Bob selects $F, G \in \mathcal{A}^\lambda : \exists F_p^{-1} \in \mathcal{A}_p^\lambda \wedge \exists F_q^{-1} \in \mathcal{A}_q^\lambda$.

A procedure to generate the public key $H \in \mathcal{A}_q^\lambda$ is then given by:

$$H = F_q^{-1} \times G \mod q \tag{5}$$

Alice encrypts her message $M \in \mathcal{A}_p^\lambda$ into $E \in \mathcal{A}_q^\lambda$ with the use of a blinding element $\Phi \in \mathcal{A}_q^\lambda$ according to:
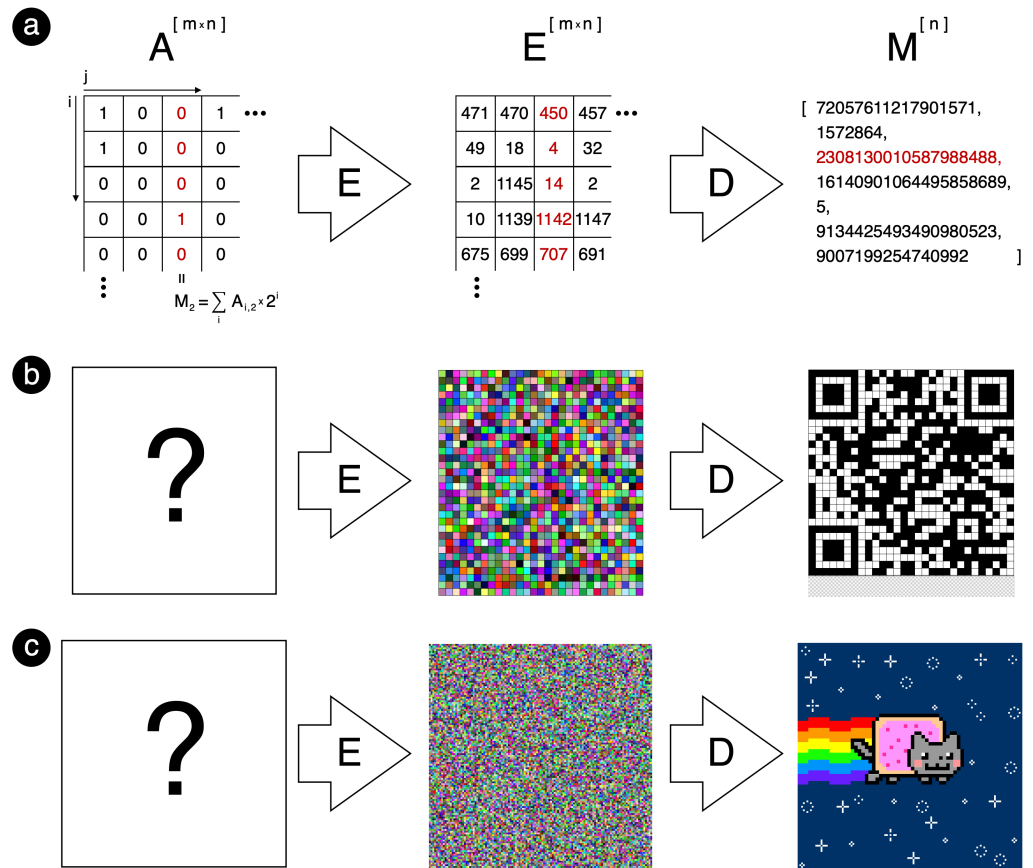
$$E = p \otimes H \times \Phi + M \mod q \tag{6}$$

The following decryption consist of three steps:

$$\mathcal{A}_q^\lambda \ni D_1 = (F \times E) \times F \mod q$$
$$\mathcal{A}_p^\lambda \ni D_2 = D_1 \mod p \qquad (7)$$
$$\mathcal{A}_p^\lambda \ni D_3 = F_p^{-1} \times (D_2 \times F_p^{-1}) \mod p$$

If the decryption was successfull, Bob receives $D_3 = M$ (up to coefficients' centered lift in $\mathcal{A}_p^\lambda$). Please remember that the lattice-based cryptography is always burdened with a chance of decryption failure due to an incorrect recovery of polynomial's coefficients. Also, for $\lambda \geq 4$ note that $\mathcal{A}^\lambda$ is neither alternative nor associative; thus successful decryption relies on a careful initial choice of F (e.g. $F : \exists! i \in \{0, \dots, 2^\lambda - 1\} : F_i \neq 0$). For a more detailed coverage of similar cryptosystems please see publications presenting QTRU (Malekian et al., 2009) and OTRU (Malekian & Zakerolhosseini, 2010).

Three examples of matrix encryption-decryption are presented in the Figure 1. All of the data and code required to reproduce these results is available in the code repository.

Figure 1: Examples of *Hypercomplex* applications for cryptography. **(a)** Message (M) composed of seven, relatively-big secret numbers is encoded in binary in a [64x7] matrix. This is later encrypted (E) and decrypted (D) with a public-key cryptosystem, allowing to transfer the numbers in a secure manner; $p = 2, q = 1151$. **(b)** Graphical representation of an encrypted/decrypted QR code, encoded in a [32x29] matrix (padded image); $p = 3, q = 1723$. **(c)** Encrypted/Decrypted [128x127] meme; $p = 17, q = 16777213$; credits: www.nyan.cat.

State of the field

When it comes to a general hypercomplex framework, the well-known *boost C++* libraries deserve the most notable mention here (Koranne, 2011). Unfortunately their scope is limited as they only implement classes for quaterions and octonions (however, as an upside, all the operations are well optimised). Moreover, these libraries do not support operations on MPFR types natively. It may also be worth mentioning the existence of smaller projects like (Girard, 2007) and (Hoppe, last accessed: 16 Feb 2021), but, unlike our work, they often lack proper test suites, code coverage reports, and documentation, and are also significantly restricted in functionality, which is a major drawback.

However, (most importantly) to our best knowledge, there is currently no high-quality open-source library that natively supports cryptosystems based on truncated polynomial rings. Previous research described distinct versions of NTRU (Hoffstein et al., 1998), among others: 4-dimensional QTRU (Malekian et al., 2009), 8-dimensional OTRU (Malekian & Zakerolhosseini, 2010); and a proposed 16-dimenisional STRU (Singh et al., 2021), the correctness of which has not yet been verified. Despite these efforts, no generalization has been provided yet. Our work is the first to present that these procedures are vaild in arbitrarily high-dimensional Cayley-Dickson algebras (provided a careful choice of parameters of the system) and to provide reproducible examples of a successful encryption/decryption procedures. Finally, it has not escaped our notice that the specific polynomial-based hypercomplex multiplication scheme we presented immediately suggests a possible hashing mechanism for string messages.

Acknowledgments

References

Fousse, L., Hanrot, G., Lefèvre, V., Pélissier, P., & Zimmermann, P. (2005). *MPFR: A multiple-precision binary floating-point library with correct rounding* (Research Report RR-5753; p. 15). INRIA. https://hal.inria.fr/inria-00070266

Girard, P. R. (2007). *Quaternions* (pp. 3–17). Birkhäuser Basel. https://doi.org/10.1007/978-3-7643-7791-5_2

Google Inc. (last accessed: 16 Feb 2021). cpplint. In *GitHub repository*. GitHub. https://github.com/google/styleguide

Heesch, D. van. (2021). Doxygen. In *GitHub repository* (Version 1.9.1). GitHub. https://github.com/doxygen/doxygen

Hoffstein, J., Pipher, J., & Silverman, J. H. (1998). NTRU: A ring-based public key cryptosystem. In J. P. Buhler (Ed.), *Algorithmic number theory* (pp. 267–288). Springer Berlin Heidelberg. https://doi.org/10.1007/BFb0054868

Hoppe, T. (last accessed: 16 Feb 2021). Cayley Dickson. In *GitHub repository*. GitHub. https://github.com/thoppe/Cayley-Dickson

Hořeňovský, M. (2020). Catch2. In *GitHub repository* (Version 2.13.2). GitHub. https://github.com/catchorg/Catch2

Koranne, S. (2011). *Boost C++ libraries* (pp. 127–143). Springer US. https://doi.org/10.1007/978-1-4419-7719-9_6

Malekian, E., & Zakerolhosseini, A. (2010). OTRU: A non-associative and high speed public key cryptosystem. *2010 15th CSI International Symposium on Computer Architecture and Digital Systems*, 83–90. https://doi.org/10.1109/CADS.2010.5623536

Malekian, E., Zakerolhosseini, A., & Mashatan, A. (2009). *QTRU: A lattice attack resistant version of NTRU*. Cryptology ePrint Archive, Paper 2009/386. https://eprint.iacr.org/2009/386

Schafer, R. D. (2017). *An introduction to nonassociative algebras*. Dover Publications. ISBN: 9780486688138

Singh, S., Padhye, S., & Pal, A. (2021). Generalization of lattice-based cryptography on hypercomplex algebras. In P. Stănică, S. Gangopadhyay, & S. K. Debnath (Eds.), *Security and privacy* (pp. 67–79). Springer Singapore. https://doi.org/10.1007/978-981-33-6781-4_6