

Understanding the Magic: Teaching Cryptography with Just the Right Amount of Mathematics

Joshua Holden

Cryptography is a field which has recently attracted a great deal of attention from both students and teachers of mathematics and of computer science. Students of computer science see cryptography as something which is not only “cool” but may be necessary for them to know in their future careers. Many of them do not realize, however, just how much mathematics they need to know in order to understand the algorithms which lie at the heart of modern cryptography.

For the past two years I have co-taught a course in cryptography at the Rose-Hulman Institute of Technology with David Mutchler, a colleague from the Computer Science department. The course is cross-listed in both the Computer Science and Mathematics departments, but most of the students are CS majors rather than math majors. Two of my goals as a representative of the mathematics department are to show the students the mathematics behind the cryptography and to (hopefully) make them see why they need to understand it.

This requires a careful balance between giving them enough mathematics to truly understand why the algorithms work and avoiding overwhelming them with too many formal proofs. Thus for each topic a professor in my position needs to carefully choose how much use will be made of abstract theory as opposed to concrete examples and of rigorous proofs as opposed to non-technical explanations of concepts.

As an example, I’d like to show part of the presentation I use in class for Euler’s Theorem and its application to RSA public-key cryptography. In fact, I don’t start this topic with either Euler’s Theorem or RSA, but rather with the Pohlig-Hellman exponentiation cipher, which was developed slightly after RSA but is conceptually more natural. The decryption of the Pohlig-Hellman cipher requires the extraction of e -th roots modulo p , which needs naturally to a discussion of Fermat’s Little Theorem. I don’t prove Fermat’s Little Theorem in complete generality, but rather prove it only in a small case such as $p = 11$ which illustrates all of the necessary ideas but doesn’t require a large amount of abstraction, such as the use of variables indexed by subscripts.

Fermat’s Little Theorem leads naturally to a discussion of why it doesn’t work for composite moduli and how it can be fixed — thus leading naturally to the definition of the Euler phi function and the statement of Euler’s Theorem. I omit the proof of Euler’s Theorem almost completely, since the students can now see that it is exactly the same as the proof of Fermat’s Little Theorem! Now we need to show that $\phi(pq) = (p - 1)(q - 1)$ if p and q are prime, which is short enough to do with full rigor if one does not attempt to extend it to more general products of primes. At this point students are ready to see and appreciate the RSA algorithm, which I generally present in full abstraction. When I show the students that a complicated series of operations on the plaintext gets us exactly back where we started, I can act like a magician sawing an assistant in half and putting him back together. But hopefully the students are not left in the position of a bewildered audience — now they are complicit in the act, since they know the secret!