Blockchains and Stealth Tactics for Teaching Security

Paul C. van Oorschot
Associate Editor in Chief

As a university professor, I am in the habit of encouraging all computer science and engineering students to take courses on, and if truly interested follow up with research in, computer security and applied cryptography. Of course, given that students do not have an infinite amount of available time, this often means, in essence, choosing security courses ahead of alternatives. And some of these alternatives are at times very appealing; higher on student priority lists; or appear to offer greater excitement, adventure, or coolness. Is computer security in a position to compete? As it turns out, we are in a fantastic position, compared to almost all other technical subject areas.

Practical applications of security and privacy technology are numerous, compelling, and easily explained. There is a deep—and sadly, ever-growing—pool of case studies to draw on, offering real-world examples of how security failures have had severe adverse impacts. And there are also many examples of how security has enabled new services and capabilities. These points are easy to convey after students are already *in* our courses but may not always be clear *before* students make a choice to take a security and privacy course. So there is a danger (from the viewpoint of those of us trying to save the world, i.e., teach security) that we may never get the opportunity to present our case to students, should they predecide to bypass security-focused courses. Can we still save them, and the rest of the world, by finding a way to teach core aspects of security?

The approach I discuss here is what we might call *teaching security by stealth*—that is, working security into the curriculum so that students get it as a side effect of their desire to learn something else. We might liken this to mixing vitamins into their food—for their own benefit, of course—without them even knowing it. If we are to do this, however, we must specifically tailor the security content, trimming it down to the core parts specifically needed to support a primary subject matter and, ideally, deliver the security in modular, just-in-time fashion.

I will give one example for concreteness, to convey the idea and method. Rather than advertising your new course on the fundamentals of applied cryptography, consider instead convincing your department chair to allow you to offer an undergrad course on blockchains and smart contracts. (Caution: be sure to reserve some spots for the computer science and engineering students, lest the business students fill all the seats.) Let's set aside for the moment that blockchains have been overhyped to the point that they are viewed as the single answer to world hunger and every other open problem known to humankind. Instead, we will join the chorus and leverage this situation to advance our own agenda.

In our example, we first engage students with the idea of inventing a new cryptocurrency. Has that ever been done successfully? Indeed it has. Dispensing with the early academic history of Chaum and others, we can cut straight to the chase with the story of a mythical character, intellectually named Satoshi Nakamoto.[1] Does this person exist? No one seems to know, even after 12 years, but whoever he or she is, this person seems certain to be rich. We can ask students to look into the pizza bought in May 2010 with 10,000 bitcoins, setting up a two-part exam question. 1) What were the toppings? 2) Discuss whether the buyer or the seller got a better deal, given that 1 bitcoin is now worth on the order of US$10,000.

Leaving discussion of Bitcoin's genesis block to the Department of Religious Studies, we can move directly to more interesting topics, like what modern tools are used for

Digital Object Identifier 10.1109/MSEC.2020.3004974
Date of current version: 3 September 2020

Digital Object Identifier 10.1109/MSEC.2020.2981217

mining coins? Hint: they are not shovels. Rather, this involves something called *hash functions*. It seems we will need a small side detour for a few minutes to learn about the algorithm SHA-256 and what properties are needed from these special functions. But I promise to return directly to the main topic of our course.

Many students interested in Bitcoin would, as they say, rather eat chopped liver than sign up for a mathematics course. And that is fine. The design of the Bitcoin system is remarkably interesting and innovative—and math and cryptography are at our service here. We view them like electricity: absolutely essential as well as important to know how to use safely, lest you electrocute yourself. But thankfully, we can outsource the wiring of our electrical panel to master electricians, rather than spend 10 years learning how for ourselves (unless, of course, it becomes our true passion). This approach allows us to become familiar with useful tools without making them our prime focus. A subset of students may also decide to learn more in a dedicated course on cryptography or security.

But let's get back to the majority of students. They want to know Bitcoin. Its design cleverly uses incentives to align stakeholder interests, resulting in a system that offers properties previously unattained. It works surprisingly well in practice, despite eluding precise theoretical analysis—this is even viewed as a badge of honor by some students. Of course, along the way, we need another detour to teach just a few minor technical details about the general concept of digital signatures (since we need them to secure our transfer of coins) and the properties of public and private keys so that we understand Bitcoin better.

Oh yes, and because Bitcoin user wallets, in which coins are stored (through private keys), rely on some efficiency tricks, it turns out that we must spend a few minutes discussing *Merkle hash trees*. Since these were really just a part of one chapter of Merkle's 1979 thesis,[2] it need not take long to discuss. (You say these data structures actually interest you? Well then, perhaps we can spend a bit more time on them, as an extra project, when we move on to Ethereum, where *Merkle Patricia tries* are used. But let's remember, we are not here to learn about algorithms but rather about cryptocurrencies, and perhaps the Silk Road, the underground economy, and how to become rich founding new startup companies. Of course, if you are also interested in a tiny bit of algorithm analysis, we will not deny you that pleasure.)

We will likely also want to mention to our students the difference between SHA-256 and RIPEMD-160 since both algorithms are used in deriving Bitcoin addresses from public keys. And Bitcoin scripting is of course an opportunity to remind students how simple stack-based machines work. And if some students believe that their future lies in mining their own bitcoins, it will be helpful to teach them that efficient Bitcoin hashing is best done on customized hardware and that there are important differences between CPUs and graphics processing units (GPUs) and those things called application-specific integrated circuits (ASICs) that the professional miners use.

In other words, with apologies, we will have to take a short detour to review a little bit about hardware.[3] At this point, we may as well also mention *memory-hard hash functions*, such as Ethereum's Ethash.[4] By the way, perhaps we should take a minute to review the simple idea of *proof of work* based on practical computation as this tends to come up fairly often. But let me remind you, we're doing this only to understand the Bitcoin economy.

Now, moving ahead, some people are more excited about Ethereum[5] than Bitcoin, so we'll discuss the idea of *smart contracts*.[6] Ethereum extends the idea of Bitcoin scripts to general computability while still relying on the basis of a blockchain to avoid the need to rely on specific trusted authorities (and thus gaining protection from possibly untrustworthy authorities). But now that we are interested in these cryptocurrencies, blockchains, and smart contracts, it seems that some details about trust and threat models, security, and, yes, cryptographic algorithms are worth understanding a bit better. What is it that delivers the confidence in currency transactions being authentic? How do these things called *hash pointers*, a fundamental part of blockchain security, really work? And why does Ethereum use a *heaviest-chain mining model*,[7] instead of a *longest-chain model*, to produce new blocks every 13–14 seconds, instead of every 10 minutes for Bitcoin? Perhaps we should take a few minutes to see how *distributed consensus* is achieved in practice.[8]

Now that we have a basic understanding, further questions arise. What happens if an Ethereum contract has programmatic flaws in it or, indeed, the Ethereum system has flaws—can we lose money? (Yes.) Can we lose a lot of money? (Well that depends—do you consider US$50 or 100 million to be a lot of money? Don't worry, losses that large happen only occasionally, not every day.[9])

This may motivate us to learn how to improve the security of individual smart contracts and smart-contract systems, given that, by design, smart contracts are meant to be self-governing and take autonomous decisions once deployed. Perhaps we should learn how to prove various properties of these short programs? I didn't plan to spend much time on fundamentals and theory in this cryptocurrency course, but if you are really so interested, perhaps you would like to come back next term so that we can study those things in greater detail. We offer some interesting security courses that may help satisfy your newfound curiosity. And by the way, there are some excellent careers in this field and a shortage of experts. I really had no idea that you would find these topics so interesting.

And thus, stealth tactics for teaching computer security were born. We should add to this a few notes.[10] The first is that students who learn via the stealthy path (path one) may be overconfident in their ability to understand security, without the benefit of follow-up, dedicated security courses (path two); path one allows a high-level understanding but is insufficient to build a system or analyze technical risks. A second comment is that computer science instructors of a cryptocurrency course may find that they themselves lack background in the broader aspects of a strongly interdisciplinary subject area, including economic and business models, aside from human factors, networking, and systems engineering.

Beyond our running cryptocurrency example, among numerous other candidate application topics on which to base a stealthy security course is *contact tracing* in the context of COVID-19. This is itself a highly interdisciplinary topic spanning privacy, social issues, and epidemiology. Our belief, however, is that both students and professors have much to gain from studying real systems and challenges beyond single-pillar, artificially constrained academic problems. ■

References

1. S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf
2. R. C. Merkle, "Secrecy, Authentication, and Public Key Systems," Ph.D. thesis, Elect. Eng., Stanford Univ., CA, 1979.
3. M. B. Taylor, "The evolution of Bitcoin hardware," *Computer*, vol. 50, no. 9, pp. 58–66, 2017. doi: 10.1109/MC.2017.3571056.
4. G. Wood, "Ethereum: A secure decentralized generalised transaction ledger," Ethereum, Yellow Paper, version June 8, 2020. [Online]. Available: https://ethereum.github.io/yellowpaper/paper.pdf
5. V. Buterin, "A next generation smart contract and decentralized application platform," Ethereum, White Paper, 2013. [Online]. Available: https://ethereum.org/en/whitepaper/
6. N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, 1997. doi: 10.5210/fm.v2i9.548.
7. Y. Sompolinsky and A. Zohar, "Secure high-rate transaction processing in Bitcoin," in *Financial Cryptography and Security*, R. Böhme and T. Okamoto, Eds. Berlin: Springer-Verlag, 2015, pp. 507–527.
8. A. M. Antonopoulos, *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*. Sebastopol, CA: O'Reilly, 2014.
9. N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on Ethereum smart contracts," in *Proc. 6th Int. Conf. Principles of Security and Trust*, 2017, pp. 164–186. doi: 10.1007/978-3-662-54455-6_8.
10. R. Böhme, private communication, June 14, 2020.