CSEP 590E: Practical Aspects of Modern Cryptography

Homework #5 Due: 6:30pm Nov 15, 2016

1. Suppose that you are designing a tokenization service for Primary Account Numbers (PAN) of 16 decimal digits. The service accepts a PAN as input, and generates and stores a token in a database. Your colleague, who heard that hash functions could be used as random oracles, recommended that you use SHA-256 to generate a token, and you followed his recommendation. Your service computes the SHA-256 hash of the 16-digit PAN, and stores the hash output *token* in a publicly readable block chain database. Devise an attack to reveal all of the PANs with corresponding tokens in the database, and compute its complexity in number of hash computations. Assuming you have one 4-core machine, each core capable of $3.14 * 10^6$ SHA-256 computations per second, compute the amount of time it would take to compute all the PANs with a corresponding token in the database. As before, rather unrealistically, ignore all I/O costs; only focus on the CPU time. *Hint: You can assume the last-4 digits of the PAN publicly available, stored on the same database in plaintext form following PCI (Payment Card Industry) regulations.*
2. Recall the magnetic stripe with two tracks on credit cards from the lecture. We'd like to encrypt the contents (PAN and CVV, only) of those tracks with a symmetric key, and we want the POS (Point Of Sale) terminals to decrypt the contents after reading it from the card in the old style swipe-to-pay terminals. In this case, we want the user to enter the PIN (4 or 6 digit PIN) on the POS terminal's key pad. You can change the code on the terminal. What cryptographic algorithm and mode of operation would you use for encryption? *Hint: Derive as many keys as you need from the user-entered PIN. There may be multiple answers to this question.*
3. [Continuing on the previous question.] When the encrypted track contents are stored on the card, we want to make it harder to clone the card, but not necessarily impossible. Towards that goal, we want to remove the ability to detect incorrect PINs in offline PIN search attacks, so that an attacker can't find the correct PIN by attempting to decrypt the tracks with all possible PINs. We don't try to counter attacks when the decrypted

contents of tracks are sent to an acquirer, card network, issuer, or any other entity that can vouch for correctness of the decrypted track data. The user still enters the PIN on the terminal, and you can change the code that runs on the terminal. What cryptographic algorithm and mode of operation would you use to encrypt the track contents? *Hint: You can leave the last four digits of the PAN in cleartext, and assume the last digit is the checksum digit computed by Luhn's algorithm.*

4. Suppose that you run a bitcoin mining pool which controls 1/3 of the resources generally being used for mining. (For simplicity, let's say that things are tuned so that a single bitcoin is mined every 10 minutes on average and that no "re-tuning" is expected for at least the next 24 hours.) Suppose further that a virus suddenly disables *all* competing miners so that your pool members are the only miners for the next hour. Determine how many additional coins you expect to mine during the one-hour outage – above and beyond what you would expect to mine if your competitors were all online and working normally. (Be careful, your initial instinct might not be correct.)
5. Suppose now that you run a bitcoin mining pool that controls 4/9 of the resources generally being used for mining and that, as in the previous problem, things are tuned so that a single bitcoin is mined every 10 minutes on average. Suppose further that transaction fees are offered to miners averaging a total of 6 bitcoins per hour. Now, imagine that a powerful cloud computing service is available that can mine coins almost instantaneously but that it is not economical because the cost of the computation required to produce a new bitcoin is equivalent to the value of 2.5 bitcoins.

   Show how you can use this cloud resource to produce a net gain over a three-hour period. You may make the simplifying assumption that groups of miners produce coins at a steady rate proportional to their computing power.