15-440/15-640: Homework 4 Solutions

1 Virtualization [20 points]

Virtualization technology enables the public cloud infrastructure many companies use today. In the lecture, we discussed two approaches to implement virtualization: system virtualization (using VMMs) and process virtualization (using containers).

1. One important motivation for the use of VMMs is that they can allocate and assign more than the actual physical resources available (e.g., processors). Tushar is wondering if this can be achieved using process virtualization. Can it? Please answer in no more than two sentences. **[4 pts]**.

If we consider oversubscription of resources, both VMs and containers can oversubscribe CPU (time sharing) and memory (virtual memory, paging) across many virtualized tenants. Some of us considered allocating more than the actual resources available to a single virtualized tenant. That's technically possible for CPUs with VMs but not for containers. For memory, both virtualization techniques can allocate more than available.

2. Daniel wants to use a high-performance Linux server for building and deploying a native Windows binary. Should he use (i) a VM or (ii) a container? Why? Please answer in no more than two sentences. **[4 pts]**.

A VM, because he needs a different OS.

3. Yuvraj is building an airline ticketing service where he wants to backup/snapshot the full system state on each node in his distributed system. These snapshots can then be used to create replicas or to recover from failures. Should Yuvraj use (i) a VM or (ii) a container? Why? Please answer in no more than two sentences. **[4 pts]**.

To capture the full system state (e.g., including current memory content, current program counter, etc.), we need VMs. If we were to only capture a limited amount of state, a container backup could be faster (since the OS and other system files are not backed up).

4. Tushar and Chelsea believe they built a secure and fast trap-and-emulate VMM. To benchmark their VMM performance, they decide to run a benchmark application. Which benchmark out of the following two should they use, and why? (i) A CPU-bound microbenchmark which performs linear algebra computations, or, (ii) an I/O bound microbenchmark which performs random reads and writes. Please answer in no more than four sentences. **[8 pts]**.

I/O benchmark – because I/O instructions are trapped and emulated, which will test their hypervisors. Arithmetic computations generally go directly to the host processor.

2 Coin Flip [20 points]

Daniel and Yuvraj are arguing over telephone about who will teach the next 15-440/640 lecture. They decide to settle this fairly by "flipping a coin" (i.e., choosing a random bit) over the telephone. Yuvraj suggests that he can flip a coin and just tell Daniel the answer, but Daniel does not trust Yuvraj to flip the coin and reveal the result honestly. In this problem, you will develop a protocol that uses a cryptographic hash function to prevent cheating.

1. Yuvraj suggests the following protocol: Daniel and Yuvraj each choose their random bits x_d, x_y (x_d and x_y are each 1 bit), and Daniel sends x_d to Yuvraj. Yuvraj then tells Daniel x_y, and they both compute $x_y \oplus x_d$ to determine the answer. Argue that this protocol is correct if Yuvraj sends a truly random x_y, but that Yuvraj can cheat to manipulate the outcome. Please answer in no more than three sentences. **[6 pts]**

   If Yuvraj sends a random x_y, then $x_y \oplus x_d$ will be random regardless of the choice of x_d. Once Yuvraj observes x_d, however, he can choose x_y non-randomly to achieve any desired outcome.

2. Devious Yuvraj tries again: Daniel and Yuvraj choose x_d and x_y as before, and Daniel uses the SHA256 cryptographic hash function (https://en.wikipedia.org/wiki/SHA-2) to compute $h_d = \texttt{SHA256}(x_d)$, which he sends to Yuvraj. Yuvraj then sends $h_y = \texttt{SHA256}(x_y)$ to Daniel. They now both reveal x_y, x_d (in any order) and again compute $x_y \oplus x_d$. Daniel can use h_y to confirm that Yuvraj did not change x_y after sending h_y, and Yuvraj can similarly confirm that Daniel did not change x_d after sending h_d. Explain how Yuvraj can still cheat to manipulate the outcome. Please answer in no more than three sentences. **[6 pts]**.

   Even though Daniel hashes x_d, x_d is one of two values – 0 or 1 – and SHA256 is a public function. Yuvraj can determine Daniel's value simply by computing both values.

3. Improve the protocol from (2) to prevent both Yuvraj and Daniel from cheating. Your protocol should use only SHA256, the ability to send and receive messages between Yuvraj and Daniel and the ability to generate random numbers. Please answer in no more than five sentences. **[8 pts]**.

   Many correct answers.
   They pick random secret keys K_y and K_d such that there are too many possible key values to determine K from SHA256(K). (For instance, a random 256-bit number is sufficient.) Daniel and Yuvraj also choose any function $f(K_y, K_d)$ that is unbiased if either input key is truly random. (For instance, simply XORing the first bit of K_d with the first bit of K_y is sufficient). Then they can exchange SHA'd values in any order. Once they have exchanged the hashes of their keys, they can reveal their keys (also in any order). They can now confirm that no one changed their key after they sent the hash. They now compute $f(K_y, K_d)$ to determine the output value of the coin-flip.
   Another correct protocol: Daniel and Yuvraj each choose random bits x_d and x_y as well as secret keys K_d and K_y from a large set of possible key values, as above. Daniel uses K_d as a secret key for private-key-based digital signatures, sending Yuvraj SHA256(x_d + K_d) and SHA256(K_d). Yuvraj does likewise. They can now reveal their random bits and secret keys. Yuvraj can verify that Daniel did not alter his key K_d after sending its hash, and also verify that Daniel used his key to correctly sign his secret bit. Yuvraj can do likewise. Daniel and Yuvraj can finally compute $x_y \uplus x_d$ to determine the outcome of the coin-flip.

3 Asymmetric Key Cryptography [20 points]

First Part

Yuvraj and Daniel want to discuss the final exam questions through email. They decide to encrypt their emails to avoid being intercepted. The first thing they have to do is to agree on a secret key. A TA suggests that they can use the basic Diffie-Hellman key exchange protocol.

1. Suppose they have agreed on $g = 17$ and $p = 5$ (g and p are public). Now Yuvraj picks his secret number 9, and Daniel picks his secret number 6. What should Yuvraj send to Daniel? What should Daniel send to Yuvraj? Moreover, what is secret key on which they agree? **[5 pts]**

   Yuvraj sends to Daniel: $g^a \texttt{ mod } p = 17^9 \texttt{ mod } 5 = 2$ Daniel sends to Yuvraj: $g^b \texttt{ mod } p = 17^6 \texttt{ mod } 5 = 4$ The secret key they agree on is $4^9 \texttt{ mod } 5(2^6 \texttt{ mod } 5) = 4$

2. What is the fundamental problem of this protocol (i.e., what security property is missing)? What different protocol or variation in this protocol can protect the final exam? Please answer in no more than two sentences. **[5 pts]**.

   The problem is that DH protocol itself does not provide authentication. One solution is to use authentication protocol along with this key exchange protocol. Specifically, both Yuvraj and Daniel choose asymmetric key pairs usable for digital signatures. Yuvraj signs whatever he sends to Daniel, and Daniel verifies that signature (using Yuvraj's public key) after receiving the email and vice versa. Any email with invalid signature should be discard. Another solution is to use public key cryptography instead so that they do not need the shared secret key.

Second Part

Yuvraj and Daniel need to communicate to decide which TA is going to grade the next homework. They have a shared secret key, k_{yd}, that allows them to create unforgeable message authentication codes so that Daniel can verify that Yuvraj created any message that is received.

Yuvraj and Daniel have a simple protocol. They have been using this protocol for grading all the previous homework sets believing that it's secure. Yuvraj would send to Daniel "Who grades homework 4?" in plain text and Daniel would reply with one of two messages: M1 = $MAC_{k_{yd}}$("Tushar") or M2 = $MAC_{k_{yd}}$("Chelsea"). When Daniel receives either M1 or M2, he verifies the MAC using k_{yd} and knows who will grade homework 4.

1. This protocol is insecure. Either Tushar or Chelsea on a router between Daniel and Yuvraj might be able to avoid having to grade homework 4. In one sentence, describe the attack. **[5 pts]**

   It is subject to a replay attack. Tushar or Chelsea could simply replay an earlier answer from a different TA's name.

2. What simple change could Daniel and Yuvraj add to the protocol to defend this attack? Explain briefly. Please answer in no more than three sentences. **[5 pts]**.

   They could use a nonce. Along with "Who grades homework 4?", they could send a random string that must be included in the MAC to ensure the answer is unique.

4 Byzantine Fault Tolerance [25 points]

Three bank robbers tunnel towards a bank vault and can only stay in contact via ham radio. Their elaborate plan involves them staying in their separate tunnels over Thanksgiving and to simultaneously light explosives under the vault, which will then sink into their midst. However, having tunneled for four days, they are not sure if the other robbers are in place. They adopt a quorum-based voting approach to reach a consensus on when they will light the explosives on Friday morning. They need at least two explosives (better three) to go off at precisely the same time. Unfortunately, several kinds of misfortunes befall the robbers.

1. Random radio interference means that a random robber loses connection for several minutes at a time. The robbers, therefore, adopt Paxos to come to a decision. In this case, explain the quorum property that ensures that Paxos guarantees that the robbers will decide on the same explosion time. Please answer in no more than two sentences. **[5 pts]**

Correctness property: every quorum will always include a member of a previous quorum.

2. The bank officials catch hold of one robber and pretend to be that robber. Explain a scenario where the bank robbers will fail to explode at least two explosives at the same time. Please answer in no more than two sentences. **[5 pts]**.

Several possible scenarios (think about the Generals from the lectures slides). Essentially, Police/bank communicate different values to the two remaining robbers.

3. Another robber joins. The true robbers are sure that only exactly one of the tunnelers is collaborating with the bank officials. Can a quorum-based consensus algorithm ensure the robbers' success? Explain why or why not, using the BFT quorum property discussed in class. Please answer in no more than three sentences. **[5 pts]**.

BFT quorum needs to include an honest node. Which can be achieved with four nodes.

4. The local mob boss, Karan, breathes down on everyone's neck and wants to make the explosion timing decision himself. Using PBFT, Karan sends the timing message, signed with a private key, to the leader among the robbers (the leader is also determined using PBFT). For how many responses from the robbers does Karan need to wait and why? Please answer in no more than three sentences. **[5 pts]**.

This asks about PBFT and for how many responses the PBFT client needs to wait. The answer is $f + 1$, so 2 in this case.

5. The robbers are down to three after Daniel (the robber who joined the last) got hungry and went for burgers. The new idea is to use a distributed ledger (blockchain) instead of a quorum-based system. Exactly one of the tunnelers is still collaborating with the police. Does the argument that four robbers are required still apply to this system? If yes, why? If no, why not? Please answer in no more than three sentences. **[5 pts]**.

Quorum-based systems require $3f + 1$ nodes (4) to deal with Byzantine failures. Distributed ledgers/blockchains require $2f + 1$ nodes (3).
An alternative answer is that the bank and the government may use many more CPUs than the mob has available and thus a distributed ledger/blockchain will never be secure.

5 The rise and fall of SZA-Coin [15 points]

Sam, Zeleena, and Amadou plan to establish a new cryptocurrency called SZA-Coin. SZA hopes to differentiate itself from all the other cryptocurrencies and attract investor money for their upcoming initial coin offering.

1. SZA considers privacy as a differentiating factor. Can SZA-Coin hide/conceal transactions (hide/conceal a transaction itself, not just its metadata)? Please answer in no more than two sentences. [**5 pts**].

No, as cryptocurrency relies on distributed ledgers. Using a distributed ledger there is no way to conceal transactions.

2. SZA wants to rethink the fundamentals of blockchain technology. Can you help the SZA team recall what kind of properties any proof-of-work needs to have? Please answer in no more than three sentences. [**5 pts**].

Generally, it should be a hard though feasible to solve the problem. A solution must be verifiable quickly and with few calculations.

3. SZA wants people to start using SZA-Coin quickly. If a new user wants to *join* the SZA network, which data do they need to have on their server and what do they need to do before they perform any transactions? Please answer briefly in two to three sentences. [**5 pts**].

All blocks, and they have to compute the hashes to verify that everything is correct, and then cache.