They received more points for stealthy backdoors, ranging from one point (for frequently-active backdoors) to five points (for backdoors stealthy enough to go unnoticed even by high quality validation testing). Tiebreaks were determined by a panel of judges based on how interesting the design choices were.

4. RESULTS AND EFFICACY OF BLUE TEAM ANALYSIS

The results of the blue team analysis were overall better than anticipated. While many of FANCI's axioms were targeted by attacks, the tool held up well. The stealthy attacks were detected with relative ease. We found that the best way to attack FANCI in this contest was to either use a frequently-active backdoor or to use a 'pathological pipeline.'[1] These are the two cases where a backdoor is more likely to bypass FANCI. In the first case, frequently-on backdoors are not stealthy. Hence, FANCI by design ignores them. In the second case, extreme pipelining obfuscates the results and stress-tests those portions of the tool prototype that are least developed, though some of these attacks were detected by FANCI nonetheless. We include further details for the three red teams that were deemed the best by the judges.

UCF Team: The UCF team created three backdoors, which highlight different aspects of the FANCI prototype. They created two frequent-action backdoors (which were not detected) and one stealthy backdoor (which was detected). The stealthy backdoor uses a 66-bit cheat code that comes from a pseudo-random input source. This backdoor highlights the importance of using intermediate wires, which was one of the main contributions of FANCI. If FANCI only computed the impact of module inputs on module outputs, this type of backdoor might not be caught, but since we analyze all intermediate wires, noticing a 66-input AND gate is easy. The other two backdoors were frequent-action backdoors. One is turned on by the reset signal, and one is turned on only during the negative clock edge. Both of these are frequently enough active that they are not detectable. These are the types of backdoors that emphasize the need to use minimal validation testing in addition to FANCI. They also illuminate the fact that signals like reset and the clock, which might be given less attention during cursory validation testing, are still relevant, especially when dealing with an adversary (as opposed to simply looking for unintentional bugs).

Team NYU-Poly1 (NYU AD): The NYU AD team submitted three backdoors on independent baseline designs, one that was submitted for the contest and two that we looked at post-competition. While FANCI detected the three attacks, two of them required non-trivial manual analysis. The first backdoor was designed into a DES accelerator and was built off of an aggregation of finite state machines (FSMs), somewhat similar to UCONN's approach[2]. They used the state machines to control round scheduling and weaken the strength of encryption, which would expose the ciphertext to software-based attacks upon release. This was a case where we were able to detect the trigger circuit because of the combinational logic that controls the state machines, though we note that this backdoor took more manual analysis to track down than the other ones we detected. The second backdoor was a similar round-scheduling attack, this time on XTEA. This one acquires its stealth by counting the number of one-bits in the input. The combinational circuit for doing the counting was flagged. The third backdoor also attacked crypto hardware, this time a cellular automata PRNG. This one was a sequence cheat code where the user supplies one bit of the trigger at a time. This attack would be difficult for validation to catch, because it hides on the negative edge of the clock (similar to one of the UCF attacks). However, since FANCI ignores clock timing, it can detect the trigger logic.

[1]Pathological pipeline is the term used in the original FANCI paper [10] to describe backdoor triggers that rely on heavy pipelining to obfuscate combinational logic.

[2]UCONN constructed an FSM-based, heavily pipelined backdoor that took a long time to find. The various teams worked independently and simultaneously.

Figure 2: Trade-off between lines of code analyzed and runtime. The darker points denote designs that finished completely in the contest time frame. The lighter points represent designs that were analyzed partially for the contest. For those lighter points, the full runtime is estimated.

Team UT Tyler: The UT Tyler team produced a backdoor in a stream cipher module. This attack is essentially one of the attacks suggested in the original FANCI paper [10]. The attack is a combination of a frequent-action backdoor and a pathological pipeline backdoor. The trigger fires frequently but not always, meaning the stealth scores are not particularly low. The design is also heavily pipelined, with roughly one flip flop between each pair of complex logic gates for critical portions of the design. Looking at the gatelist, it is immediately obvious that the design has been compromised, but identifying what the exact backdoor payload is can be quite hard, and FANCI does not easily detect this type of attack. As mentioned in the original paper, going after this type of attack reliably requires either validation tests or oversight from an integration engineer to notice the pathological nature of the design.

5. OBSERVATIONS ON RESULTS

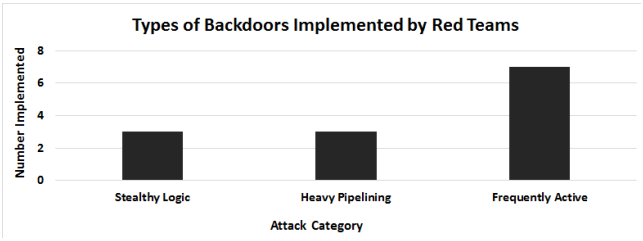

Figure 3: The amount of each type of backdoor implemented by the red teams.

We include a few observations and takeaways based on the results of the contest and our experiences.

Runtime and Scability Figure 2 shows the runtime of the tool as a function of the number of lines of code in the various designs, using one primary design from each red team. Naturally, FANCI runs slower on larger designs, but the slowdown is more or less linear, which makes analysis feasible. These tests were performed