Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

HS 2010

Prof. R. Wattenhofer / R. Eidenbenz

Distributed Systems Part II

Solution to Exercise Sheet 3

1 Authentication

a) The new algorithm looks like this:

```
if I am P then
  values ← {input}
  broadcast "P has input"
else
  values ← {}
end if
for r = 0 to f + 1 do
  for all received values x do
    if |values| < 2 and accepted r messages "P has x" with x ∉ values then
      values ← values ∪ {x}
      broadcast "P has x"
    end if
  end for
end for
if |values| = 1 then
  decide item in values
else
  decide "sender faulty"
end if
```

b) If P is correct: there is only one message in the system, which is accepted in the first round. There are no other messages, hence for all processes $|values| = 1$.

If P is Byzantine:

- Assume that a correct process `p` adds x to its value set in a round $r < f + 1$: Process `p` has accepted r messages including the message from P. Therefore all other correct processes accept the same r messages plus `p`'s message and add x to their value set as well in round $r + 1$.
- Assume a correct process `p` adds x to its value set in round $f + 1$: In this case, `p` accepted $f + 1$ messages. At least one of those is sent by a correct process, which must have added x to its set in an earlier round. We are again in the previous case, i.e., all correct processes added x to its value set.

2 Randomization

a) The algorithm can handle $f < n/8$ failures. To find this result we check the proofs for the validity condition, agreement, and termination for numbers that change:

Validity condition Nothing changes

Agreement Nothing changes

Termination If some process does not set its value randomly, all processes must set the same value, i.e. there must not be $n - 4f$ proposals for 0 and $n - 4f$ proposals for 1. This means that $2 * (n - 4f) > n$, or $f < n/8$.

The reason why this property changes is that Byzantine processes can create two different messages, while simple crashing processes cannot.

b) One solution is to replace "if at least n-4f proposals" by "if at least n-3f proposals". There are other correct solutions which will not be discussed in this master solution.

c) We modify the proof from the lecture. For the agreement property we replace "Every other correct process must have received x at least $n-4f$ times." by "... at least $n-3f$" times.

Why? Because $n-3f$ correct processes had to send their proposal in order for one process to decide. Meaning $n - 3f$ correct processes sent their proposal to any process.

Rewriting the termination property leads to $2 * (n - 3f) > n$, or $f < n/6$.