Solution 9

Program Analysis for System Security and Reliability
ETH Zürich

May 14, 2018

Problem 1.
1. We can encode the privacy enforcement ξ as:

```
def leak (password){
    // original program leaks full password
    val := password
    if 1<=val && val<=3{
        val = pick([1,2,3]);
    }else if 4<=val && val<=6{
        val = pick([4,5,6]);
    }else if 7<=val && val<=9{
        val = pick([7,8,9]);
    }else{
        val = 9;
    }
    return val;
}
```

2. With probability $\frac{9}{10}$, the output lies in $\{1, \ldots, 9\}$, in which case the attacker can infer that the password is one of 3 values, yielding a probability of $\frac{1}{3}$ of guessing correctly. With probability $\frac{1}{10}$, the output is 10, and the attacker knows the password with probability 1. Overall, the probability of guessing correctly is

$$\frac{9}{10} \cdot \frac{1}{3} + \frac{1}{10} \cdot 1 = 40\%$$

3. The permissiveness is 4 and the precision is 1.

4. No matter the enforcement, given privacy policy is always enforced, as any probability p must satisfy $p \in [0, 1]$. Hence, the simplest policy is $\xi_{\text{best}} = \{\{n\} \mid n \in \mathbb{N}, 1 \leq n \leq 10\}$. This is also the most permissive (and the most precise) enforcement.

5. Without the output of `leak`, the attacker believes that $\Pr[I \in \{1\}] = \frac{1}{10}$. Any additional knowledge on the output of `leak` will either decrease or increase this belief.

Hence, our enforcement must avoid leaking any information on the password. We achieve this by using

$$\xi_{\text{no leak}} = \{\{n \mid n \in \mathbb{N}, 1 \leq n \leq 10\}\}$$

Because this is the only enforcement for the given policy, it is also the most permissive (and most precise).

6. The most permissive enforcement is not unique. As an example, consider the privacy policy

$$\forall o \in O. \Pr[I \in \{1\} \mid O \in [o]_\xi] \in \left[0, \frac{1}{2}\right]$$

Two possible enforcements are

$$\xi_6 = \{\{1,2\},\{3\}\} \cup \{\{n\} \mid n \in \mathbb{N}, 4 \leq n \leq 10\}$$

and

$$\xi_6' = \{\{1,3\},\{2\}\} \cup \{\{n\} \mid n \in \mathbb{N}, 4 \leq n \leq 10\}$$

Note that both ξ_6 and ξ_6' enforce the policy, but they are equally permissive. Note also that the only more permissive enforcement is ξ_{best} (see above), but it does not enforce the policy.

7. The most precise enforcement is not unique. The counterexample is analogous to the one for the most permissive enforcement.

Problem 2.
1.

$$\begin{aligned}
\Pr[P = [0,0]] &= 99\% \cdot 90\% = 89.1\% \\
\Pr[P = [0,1]] &= 99\% \cdot 10\% = 9.9\% \\
\Pr[P = [1,0]] &= 1\% \cdot 50\% = 0.5\% \\
\Pr[P = [1,1]] &= 1\% \cdot 50\% = 0.5\%
\end{aligned}$$

2.

$$\begin{aligned}
\Pr[C=0] =& \Pr[P=[0,0]]\cdot\Pr[C=0\mid P=[0,0]]+\\
&\Pr[P=[0,1]]\cdot\Pr[C=0\mid P=[0,1]]\\
=&89.1\%\cdot 1+9.9\%\cdot 50\%=94.05\%\\
\Pr[C=70] =& \Pr[P=[0,1]]\cdot\Pr[C=70\mid P=[0,1]]\\
=&9.9\%\cdot 50\%=4.95\%\\
\Pr[C=10000] =& \Pr[P=[1,0]]\cdot\Pr[C=10000\mid P=[1,0]]+\\
&\Pr[P=[1,1]]\cdot\Pr[C=10000\mid P=[1,1]]\\
=&0.5\%\cdot 1+0.5\%\cdot 50\%=0.75\%\\
\Pr[C=10070] =& \Pr[P=[1,1]]\cdot\Pr[C=10070\mid P=[1,1]]\\
=&0.5\%\cdot 50\%=0.25\%
\end{aligned}$$

3. We only provide the probabilities that are not 0.

$$\begin{aligned}
\Pr[P=[0,0]\mid C=0] &= \frac{\Pr[P=[0,0]\wedge C=0]}{\Pr[C=0]}=\frac{89.1\%\cdot 1}{94.05\%}\approx 94.74\%\\
\Pr[P=[0,1]\mid C=0] &= \frac{\Pr[P=[0,1]\wedge C=0]}{\Pr[C=0]}=\frac{9.9\%\cdot 50\%}{94.05\%}\approx 5.26\%\\
\Pr[P=[0,1]\mid C=70] &= 1\\
\Pr[P=[1,0]\mid C=10000] &= \frac{\Pr[P=[1,0]\wedge C=10000]}{\Pr[C=10000]}=\frac{0.5\%\cdot 1}{0.75\%}\approx 66.7\%\\
\Pr[P=[1,1]\mid C=10000] &= \frac{\Pr[P=[1,1]\wedge C=10000]}{\Pr[C=10000]}=\frac{0.5\%\cdot 50\%}{0.75\%}\approx 33.3\%\\
\Pr[P=[1,1]\mid C=10070] &= 1
\end{aligned}$$

4.

Step 1: Originally, we get the enforcement $\xi=\{\{0\},\{70\},\{10000\},\{10070\}\}$. There are two violating classes: $C_1=\{10000\}$ and $C_2=\{10070\}$, as

$$\begin{aligned}
&\Pr[P=[1,0]\vee P=[1,1]\mid C=10000]\\
=&\frac{\Pr[(P=[1,0]\vee P=[1,1])\wedge C=10000]}{\Pr[C=10000]}\\
=&\frac{1\%\cdot 75\%}{0.75\%}=1\geq\frac{1}{2}
\end{aligned}$$

and

$$\begin{aligned}
&\Pr[P = [1,0] \vee P = [1,1] \mid C = 10070] \\
=&\frac{\Pr[(P = [1,0] \vee P = [1,1]) \wedge C = 10070]}{\Pr[C = 10070]} \\
=&\frac{1\% \cdot 25\%}{0.25\%} = 1 \geq \frac{1}{2}
\end{aligned}$$

We pick C_1, as it contains the smaller cost ($c = 10000$). As a merge candidate, we pick $\{0\}$ (this is the merge candidate that contains the smallest cost $c = 0$). This gives us the enforcement

$$\xi' = \{\{0, 10000\}, \{70\}, \{10070\}\}$$

Now, the output $\{0, 10000\}$ *does* satisfy the policy, as

$$\begin{aligned}
&\Pr[P = [1,0] \vee P = [1,1] \mid C = 0 \vee C = 10000] \\
=&\frac{\Pr[(P = [1,0] \vee P = [1,1]) \wedge (C = 0 \vee C = 10000)]}{\Pr[C = 0 \vee C = 10000]} \\
=&\frac{1\% \cdot 75\%}{94.05\% + 0.75\%} \approx 0.8\% \leq \frac{1}{2}
\end{aligned}$$

The next violating class is $\{10070\}$. We merge it with $\{0, 10000\}$. This gives us the enforcement

$$\xi' = \{\{0, 10000, 10070\}, \{70\}\}$$

The merged partition $\{0, 10000, 10070\}$ *does* satisfy the policy, as

$$\begin{aligned}
&\Pr[P = [1,0] \vee P = [1,1] \mid C = 0 \vee C = 10000 \vee C = 10070] \\
=&\frac{\Pr[(P = [1,0] \vee P = [1,1]) \wedge (C = 0 \vee C = 10000 \vee C = 10070)]}{\Pr[C = 0 \vee C = 10000 \vee C = 10070]} \\
=&\frac{1\% \cdot 1}{94.05\% + 0.75\% + 0.25\%} \approx 1\% \leq \frac{1}{2}
\end{aligned}$$

Since the class $\{70\}$ also satisfies the policy, we are done.