Exercise (Security of simple liveness proof). Entity authentication protocols are often used to prove liveness of a device or a person. For instance, ATM machines normally ask PIN codes several times during long transactions to assure that the person is still present. Such liveness proofs can be implemented with one-way functions. Let \( f: \mathcal{X} \to \mathcal{Y} \) be a one-way function and let \( n \) be the maximal number of protocol invocations. Then a secret key \( \text{sk} \) can be chosen as a tuple of random values \( x_1, \ldots, x_n \leftarrow \mathcal{X} \) and the corresponding public key \( \text{pk} \) as a tuple of hash values \( f(x_1), \ldots, f(x_n) \). Each time when a party wants to prove liveness he or she will release non-published sub-key \( x_i \). The proof is successful if \( f(x_i) = y_i \) where \( y_i \) is the \( i \)th component of the public key \( \text{pk} \). Prove that if \( f \) is \((t, \varepsilon_1)\)-secure one-way function and protocols are executed sequentially, then the probability that a \( t \)-time adversary succeeds in the \( i \)th authentication without seeing \( x_i \) is at most \( \varepsilon \).

Solution. Recall that one-wayness of a function \( f \) is defined through the following security game:

\[
\begin{align*}
Q & \quad x \leftarrow \mathcal{X} \\
& \quad y \leftarrow f(x) \\
& \quad \hat{x} \leftarrow B(y) \\
& \quad \text{return } [y \neq f(\hat{x})].
\end{align*}
\]

The function \( f \) is \((t, \varepsilon)\)-secure one-way function if for any \( t \)-time adversary \( B \) the corresponding advantage is bounded:

\[
\text{Adv}_0^\text{ow}(B) = \Pr[Q^B = 1] \leq \varepsilon.
\]

Now the scenario of guessing an \( i \)th subkey \( \hat{x}_i \) such that \( f(\hat{x}_i) = f(x_i) \) can be modelled in the following game:

\[
\begin{align*}
\mathcal{G}_i^A & \quad x_1 \leftarrow \mathcal{X} \\
& \quad y_1 \leftarrow f(x_1) \\
& \quad \ldots \\
& \quad x_n \leftarrow \mathcal{X} \\
& \quad y_n \leftarrow f(x_n) \\
& \quad \hat{x}_i \leftarrow A(y_1, \ldots, y_i-1, y_i+1, \ldots, y_n, x_1, \ldots, x_{i-1}) \\
& \quad \text{return } [y_i = f(\hat{x}_i)].
\end{align*}
\]

where the inputs \( y_1, \ldots, y_n \) for \( A \) correspond to the public key used in the liveness proof and inputs \( x_1, \ldots, x_{i-1} \) correspond to secrets leaked during previous protocol instances. Recall that in each liveness proof the honest prover reveals the corresponding sub-secret \( x_j \). Since the communication between the prover and verifier is not secured a malicious adversary can snatch corresponding values. Moreover, the verifier itself might become malicious at some time-point. Hence, we cannot assume that the adversary does not know \( x_1, \ldots, x_{i-1} \) during the attack even if communication channels are indeed secure.

To bound the success of an adversary \( A \) in the game \( \mathcal{G}_i \), note that we can use a simple wrapper:

\[
\mathcal{B}(y) \\
\quad x_1 \leftarrow \mathcal{X} \\
\quad y_1 \leftarrow f(x_1) \\
\quad \ldots \\
\quad x_n \leftarrow \mathcal{X} \\
\quad y_n \leftarrow f(x_n) \\
\quad \hat{x}_i \leftarrow A(y_1, \ldots, y_{i-1}, y, y_{i+1}, \ldots, y_n, x_1, \ldots, x_{i-1}) \\
\quad \text{return } \hat{x}_i
\]
to convert the adversary against the game $G_i$ to the adversary against the game $Q$. Simple inlining of the adversary construction $B$ into the game $Q$ yields:

```
Q
\[
\begin{align*}
x & \leftarrow \mathcal{X} \\
y & \leftarrow f(x) \\
x_1 & \leftarrow \mathcal{X} \\
y_1 & \leftarrow f(x_1) \\
\vdots \\
x_n & \leftarrow \mathcal{X} \\
y_n & \leftarrow f(x_n) \\
\hat{x}_i & \leftarrow A(y_1, \ldots, y_{i-1}, y, y_{i+1}, \ldots, y_n, x_1, \ldots, x_{i-1}) \\
\text{return} \ [y = f(\hat{x})]
\end{align*}
\]
```

which is completely equivalent to the game $G^A_i$. Indeed, instead of $x_i$ and $y_i$ the game $Q^A$ uses $x$ and $y$. However, these have exactly the same distribution. Thus, we have established that

$$
\Pr [G^A_i = 1] = \Pr [Q^B = 1] \leq \varepsilon
$$

as long as the running-time of $B$ is smaller or equal to $t$. As the overhead of $B$ compared to the running-time of $A$ is $\Theta(n)$, we get the desired security claim. Note that the extra penalty $\Theta(n)$ is small but still worth noting – the bound on the running-time of $A$ decreases linearly if we increase the number of sub-secrets $n$.

Finally, note that the overall probability that an adversary manages to succeed in any of the liveness proofs is bounded by $n\varepsilon$. Although the adversary might adaptively choose which liveness proofs it tries to attack, we can still consider probabilities that it succeeds against the $i$th liveness proof. As success means that the adversary succeeds against some proof, union bound gives the desired result:

$$
\Pr [A \text{ succeeds in some protocol}] \leq \sum_{i=1}^{n} \Pr [G^A_i = 1] \leq n\varepsilon
$$