Problem Set 3

1. Consider the following security definition for pseudorandom generator.

> Let m and n be positive integers. Let $G : \{0,1\}^m \to \{0,1\}^n$ be a pseudorandom generator, and let A be an adversary against G. We define the following subroutines, experiment, and advantage function.
>
> Subroutine $\texttt{Initialize}(w)$
>   If $w = 0$
>   then $y \xleftarrow{\$} \{0,1\}^n$
>   else $s \xleftarrow{\$} \{0,1\}^m$; $y \leftarrow G(s)$
>   Return y
>
> Experiment $\mathbf{Exp}_G^{\text{prg-}w}(A)$
> $y \xleftarrow{\$} \texttt{Initialize}(w)$
> $d \xleftarrow{\$} A(y)$
>   Return d
>
> We define the *prg* advantage* of an adversary A attacking G as
>
> $$\mathbf{Adv}_G^{\text{prg}*}(A) = \Pr\left[\mathbf{Exp}_G^{\text{prg-1}}(A) \Rightarrow 1\right] - \Pr\left[\mathbf{Exp}_G^{\text{prg-0}}(A) \Rightarrow 1\right] .$$

Recall the definition of $\mathbf{Adv}^{\text{prg}}$ defined in the textbook and studied in class. Prove that, for all G and A,

$$\mathbf{Adv}_G^{\text{prg}*}(A) = \mathbf{Adv}_G^{\text{prg}}(A) .$$

> **Solution:** Recall the definition of the advantage of an adversary in the PRG game as studied in class:
>
> $$\mathbf{Adv}_G^{\text{prg}}(A) = 2 \cdot \Pr[\,\mathbf{Exp}_G^{\text{prg}}(A) \Rightarrow \mathbf{T}\,] - 1 . \tag{1}$$
>
> Consider the expression $\Pr[\,\mathbf{Exp}_G^{\text{prg}}(A) \Rightarrow \mathbf{T}\,]$. Recognizing that the challenger picks the bit b for the experiment $\mathbf{Exp}_G^{\text{prg}}(A)$ uniform randomly and applying Bayes' theorem, we obtain
>
> $$\begin{aligned}\Pr[\,\mathbf{Exp}_G^{\text{prg}}(A) \Rightarrow \mathbf{T}\,] = {} & \Pr[\,\mathbf{Exp}_G^{\text{prg}}(A) \Rightarrow \mathbf{T} \mid b = 1\,] \cdot \frac{1}{2} \\ & + \Pr[\,\mathbf{Exp}_G^{\text{prg}}(A) \Rightarrow \mathbf{T} \mid b = 0\,] \cdot \frac{1}{2} .\end{aligned}$$
>
> Now, since
>
> $$\begin{aligned}\Pr[\,\mathbf{Exp}_G^{\text{prg}}(A) \Rightarrow \mathbf{T} \mid b = 1\,] &= \Pr\left[\mathbf{Exp}_G^{\text{prg-1}}(A) \Rightarrow 1\right] \text{ and} \\ \Pr[\,\mathbf{Exp}_G^{\text{prg}}(A) \Rightarrow \mathbf{T} \mid b = 0\,] &= 1 - \Pr\left[\mathbf{Exp}_G^{\text{prg-0}}(A) \Rightarrow 1\right] ,\end{aligned}$$
>
> we have
>
> $$\Pr[\,\mathbf{Exp}_G^{\text{prg}}(A) \Rightarrow \mathbf{T}\,] = \Pr[\,\mathbf{Exp}_G^{\text{prg-1}}(A) \Rightarrow 1\,] \cdot \frac{1}{2} + (1 - \Pr[\,\mathbf{Exp}_G^{\text{prg-0}}(A) \Rightarrow 1\,]) \cdot \frac{1}{2} . \tag{2}$$

Substituting the quantity in Equation (2) into Equation (1), we obtain

$$\begin{aligned}\mathbf{Adv}_G^{\mathrm{prg}}(A) &= 2 \cdot \left(\Pr\left[\,\mathbf{Exp}_G^{\text{prg-1}}(A) \Rightarrow 1\,\right] \cdot \frac{1}{2} + (1 - \Pr\left[\,\mathbf{Exp}_G^{\text{prg-0}}(A) \Rightarrow 1\,\right]) \cdot \frac{1}{2}\right) - 1 \\ &= \Pr\left[\,\mathbf{Exp}_G^{\text{prg-1}}(A) \Rightarrow 1\,\right] - \Pr\left[\,\mathbf{Exp}_G^{\text{prg-0}}(A) \Rightarrow 1\,\right] \\ &= \mathbf{Adv}_G^{\mathrm{prg}*}(A) \text{ as desired.}\end{aligned}$$

2. Let m and n be positive integers, and let $G_1 : \{0,1\}^m \to \{0,1\}^n$ and $G_2 : \{0,1\}^m \to \{0,1\}^n$ be pseudorandom generators. Define a pseudorandom generator $G : \{0,1\}^m \to \{0,1\}^{2n}$ as follows. For any $s \in \{0,1\}^m$,

$$G(s) \;=\; G_1(s) \| G_2(s)\ .$$

Suppose that G_1 and G_2 are secure under the PRG security notion. Is G necessarily a secure PRG? Prove your answer. Be sure to provide a complete proof. Specifically, if you answer yes, specify a reduction along with an analysis relating the advantages of relevant adversaries and their resource usage. If you answer no, specify a counterexample, an attack, and an analysis of the adversary's advantage and resource usage. As always, an adversary requiring a minimal amount of resources while achieving a high advantage value is better.

Solution: No, G is not necessarily a secure PRG. One counterexample is when $G_1 = G_2$.

Construction: We define an adversary A attacking G as follows.

Adversary $A(y)$:
Parse y as $y_1 y_2$ where $|y_1| = |y_2|$
If $y_1 = y_2$ then return 1 else return 0

Analysis:
We compute the advantage of A. Let b be the bit that A is supposed to guess in the PRG game.

Case $\mathbf{b = 1}$: Suppose the seed initialized in the game is s. In this case, we know that $y_1 = G_1(s) = y_2$. Thus, A returns 1 and is always correct. Thus, $\Pr\left[\,\mathbf{Exp}_G^{\text{prg-1}}(A) \Rightarrow 1\,\right] = 1$.

Case $\mathbf{b = 0}$: In this case, we know that y_1 and y_2 are chosen uniform randomly from $\{0,1\}^n$. Thus, A mistakenly returns 1 when the values of y_1 and y_2 happen to collide. This happens with probability $1/2^n$. Thus, $\Pr\left[\,\mathbf{Exp}_G^{\text{prg-0}}(A) \Rightarrow 1\,\right] = 1/2^n$.

From the previous question, we know that

$$\begin{aligned}\mathbf{Adv}_G^{\mathrm{prg}}(A) &= \mathbf{Adv}_G^{\mathrm{prg}*}(A) \\ &= \Pr\left[\,\mathbf{Exp}_G^{\text{prg-1}}(A) \Rightarrow 1\,\right] - \Pr\left[\,\mathbf{Exp}_G^{\text{prg-0}}(A) \Rightarrow 1\,\right] \\ &= 1 - \frac{1}{2^n}\ .\end{aligned}$$

When n is large, this value is close to 1. Thus, A has a high advantage value. So, G is insecure.

Resources: From the code, A has no oracle access and mainly reads strings and performs string comparison. This takes constant time since the sizes of the strings are fixed.

3. Let m and n be positive integers, and let $G_1 : \{0,1\}^m \to \{0,1\}^n$ and $G_2 : \{0,1\}^m \to \{0,1\}^n$ be pseudorandom generators. Define a pseudorandom generator $G : \{0,1\}^{2m} \to \{0,1\}^{2n}$ as follows. For any $s_1, s_2 \in \{0,1\}^m$,

$$G(s_1 s_2) \;=\; G_1(s_1) \| G_2(s_2) \; .$$

Suppose that G_1 and G_2 are secure under the PRG security notion. Is G necessarily a secure PRG? Prove your answer. Be sure to provide a complete proof. Specifically, if you answer yes, specify a reduction along with an analysis relating the advantages of relevant adversaries and their resource usage. If you answer no, specify a counterexample, an attack, and an analysis of the adversary's advantage and resource usage. As always, an adversary requiring a minimal amount of resources while achieving a high advantage value is better.

Solution: Let A be an adversary. Let Games $G0, G1$, and $G2$ be the following games:

Game $G0(A)$:	**Game** $G1(A)$:	**Game** $G2(A)$:
$s_1 \xleftarrow{\$} \{0,1\}^m$; $s_2 \xleftarrow{\$} \{0,1\}^m$	$y_1 \xleftarrow{\$} \{0,1\}^n$; $s_2 \xleftarrow{\$} \{0,1\}^m$	$y_1 \xleftarrow{\$} \{0,1\}^n$; $y_2 \xleftarrow{\$} \{0,1\}^n$
$y \leftarrow G_1(s_1)\|G_2(s_2)$	$y \leftarrow y_1\|G_2(s_2)$	$y \leftarrow y_1\|y_2$
$d \xleftarrow{\$} A(y)$	$d \xleftarrow{\$} A(y)$	$d \xleftarrow{\$} A(y)$
Return d	Return d	Return d

Notice that

$$\mathbf{Adv}_G^{\text{prg*}}(A) = \Pr[\,G0(A) \Rightarrow 1\,] - \Pr[\,G2(A) \Rightarrow 1\,]\ . \tag{3}$$

We construct adversaries B_1 attacking G_1 and B_2 attacking G_2 each with access to adversary A as follows:

Adversary $B_1(y_1)$:	**Adversary** $B_2(y_2)$:
$s_2 \xleftarrow{\$} \{0,1\}^m$; $y_2 \leftarrow G_2(s_2)$	$y_1 \xleftarrow{\$} \{0,1\}^n$
Run $A(y_1\|y_2)$ until A halts and returns d	Run $A(y_1\|y_2)$ until A halts and returns d
Return d	Return d

We make the following observations.

1. The environment of $\mathbf{Exp}_G^{\text{prg-1}}(A)$ is the same as $G0$.
2. The environment of $\mathbf{Exp}_G^{\text{prg-0}}(A)$ is the same as $G2$.
3. In the experiment $\mathbf{Exp}_{G_1}^{\text{prg-1}}(B_1)$, the input of B_1 is a real output of G_1. Here, the environment in which B_1 simulates A is exactly the same as $G0$. Notice from the code of B_2 that it outputs whatever A outputs.
4. In the experiment $\mathbf{Exp}_{G_1}^{\text{prg-0}}(B_1)$, the input of B_1 is a value chosen uniform-randomly from $\{0,1\}^n$. Here, the environment in which B_1 simulates A is $G1$. Notice from the code of B_1 that it outputs whatever A outputs.
5. In the experiment $\mathbf{Exp}_{G_2}^{\text{prg-1}}(B_2)$, the input of B_2 is a real output of G_2. Here, the environment in which B_2 simulates A is $G1$. Notice from the code of B_2 that it outputs whatever A outputs.
6. In the experiment $\mathbf{Exp}_{G_2}^{\text{prg-0}}(B_2)$, the input of B_2 is a value chosen uniform-randomly from $\{0,1\}^n$. Here, the environment in which B_2 simulates A is exactly the same as $G2$. Notice from the code of B_2 that it outputs whatever A outputs.

Thus, we have that

$$\begin{aligned}
\mathbf{Adv}_G^{\text{prg}}(A) &= \mathbf{Adv}_G^{\text{prg*}}(A) && (4)\\
&= \Pr\left[\,\mathbf{Exp}_G^{\text{prg-1}}(A) \Rightarrow 1\,\right] - \Pr\left[\,\mathbf{Exp}_G^{\text{prg-0}}(A) \Rightarrow 1\,\right] \\
&= \Pr[\,G0(A) \Rightarrow 1\,] - \Pr[\,G2(A) \Rightarrow 1\,] && (5)\\
&= \Pr[\,G0(A) \Rightarrow 1\,] - \Pr[\,G1(A) \Rightarrow 1\,] + \Pr[\,G1(A) \Rightarrow 1\,] - \Pr[\,G2(A) \Rightarrow 1\,] && (6)\\
&\le \Pr\left[\,\mathbf{Exp}_{G_1}^{\text{prg-1}}(B_1) \Rightarrow 1\,\right] - \Pr\left[\,\mathbf{Exp}_{G_1}^{\text{prg-0}}(B_1) \Rightarrow 1\,\right] \\
&\quad + \Pr\left[\,\mathbf{Exp}_{G_2}^{\text{prg-1}}(B_2) \Rightarrow 1\,\right] - \Pr\left[\,\mathbf{Exp}_{G_2}^{\text{prg-0}}(B_2) \Rightarrow 1\,\right] && (7)\\
&= \mathbf{Adv}_{G_1}^{\text{prg*}}(B_1) + \mathbf{Adv}_{G_2}^{\text{prg*}}(B_2) && (8)\\
&= \mathbf{Adv}_{G_1}^{\text{prg}}(B_1) + \mathbf{Adv}_{G_2}^{\text{prg}}(B_2) && (9)
\end{aligned}$$

Solution: Therefore, if G_1 and G_2 are secure PRGs, the quantity on the right would be small. Then, the quantity on the left would also be small, and G is a secure PRG as well.

We justify our derivation above. Equation (4) follows from our result in the first problem. Equation (5) follows from observations 1 and 2. Equation (6) is obtained by adding and substracting a single quantity. The four terms of Equation (7) follow from observations 3, 4, 5, and 6, in that order. Equation (8) follows from applying the advantage function definition to G_1, B_1 and G_2, B_2. Equation (9) follows from our result in the first problem.

Resource usage: B_1 and B_2 run A and additionally use only $O(1)$ amount of time since the operations are manipulations of strings of fixed sizes. They have no access to oracles.

4. Let n be a positive integer. Recall that $[\mathsf{KG}]$ denotes the set of all possible keys output by the algorithm KG. Let $\mathsf{MA} = (\mathsf{KG}, \mathsf{Tag}, \mathsf{Vf})$ be a MAC scheme secure under the SUF-CMA security notion, and let $\{0,1\}^n$ be the message space for MA. We define $\mathsf{MA}' = (\mathsf{KG}, \mathsf{Tag}', \mathsf{Vf}')$ where, for all $M \in \{0,1\}^{2n}$, for all $K \in [\mathsf{KG}]$,

$$\mathsf{Tag}'_K(M) \;=\; \mathsf{Tag}_K(M[1]) \| \mathsf{Tag}_K(M[2])$$

where $M = M[1]M[2]$ and $|M[1]| = |M[2]|$.

(a) Write a deterministic and stateless algorithm Vf' that would ensure that MA' satisfies the correctness condition.

Solution:
Algorithm $\mathsf{Vf}'_K(M, T)$:
  Parse M as $M1$ and $M2$ such that $|M1| = |M2|$
  Parse T as $T1$ and $T2$ such that $|T1| = |T2|$
  If $\mathsf{Vf}_K(M1, T1) = \mathsf{Vf}_K(M2, T2) = 1$ then return 1 else return 0

(b) Is MA' necessarily a secure MAC scheme? Prove your answer. Be sure to provide a complete proof. Specifically, if you answer yes, specify a reduction along with an analysis relating the advantages of relevant adversaries and their resource usage. If you answer no, specify a counterexample, an attack, and an analysis of the adversary's advantage and resource usage. As always, an adversary requiring a minimal amount of resources while achieving a high advantage value is better.

Solution: No, it isn't. We define the following adversary.
Adversary $A^{\mathtt{Tag},\mathtt{Vf}}$:
(1) $T \xleftarrow{\$} \mathtt{Tag}(0^n 1^n)$
(2) Parse T as $T1 \| T2$ such that $|T1| = |T2|$
(3) $\mathtt{Vf}(1^n 0^n, T2 \| T1)$

Let S be the set maintained by the challenger in $\mathbf{Exp}^{\text{suf-cma}}_{\mathsf{MA}'}(A)$. We argue that

1. S does not contain the forgery at the time the query to the verification oracle is made on line (3).

   Justification: After line (1) is executed, $S = \{(0^n 1^n, T)\}$. There are no further tagging queries thereafter. Thus, when line (3) is executed, S does not contain $(1^n 0^n, T2 \| T1)$.

2. $(1^n 0^n, T2 \| T1)$ is a winning query, i.e., $\mathsf{Vf}_K(1^n 0^n, T2 \| T1) = 1$.

   Justification: From lines (1) and (2) and the construction of the MAC scheme, we know that

$$T1 = \mathsf{Tag}_K(0^n) \qquad \text{and} \qquad T2 = \mathsf{Tag}_K(1^n)\ . \tag{10}$$

From our answer in part (a), the verification equation on the second line will return 1 since $\mathsf{Vf}'_K(1^n0^n, T2\|T1)$ results in $\mathsf{Vf}_K(1^n, T2)$ and $\mathsf{Vf}_K(0^n, T1)$, both of which return 1 due to Equation (10).

Thus, A's advantage is as follows:

$$\mathbf{Adv}^{\text{suf-cma}}_{\mathsf{MA}'}(A) = \Pr\left[\, \mathbf{Exp}^{\text{suf-cma}}_{\mathsf{MA}'}(A) \Rightarrow \mathbf{T} \,\right] = 1 \;.$$

Resource usage: A submits 1 tagging query of length $2n$ and 1 verification query of length $2n$ plus the length of the tag under MA'. The running time is the time it takes to execute the two oracle queries and the constant time for manipulating the strings.

5. Let n be a positive integer. Recall that $[\mathsf{KG}]$ denotes the set of all possible keys output by the algorithm KG. Let $\mathsf{MA} = (\mathsf{KG}, \mathsf{Tag}, \mathsf{Vf})$ be a MAC scheme secure under the SUF-CMA security notion, and let $\{0,1\}^n$ be the message space for MA. We define $\mathsf{MA}' = (\mathsf{KG}, \mathsf{Tag}', \mathsf{Vf}')$ where, for all $M \in \{0,1\}^n$, for all $K \in [\mathsf{KG}]$,

$$\mathsf{Tag}'_K(M) \;=\; \mathsf{Tag}_K(M)\|\mathsf{Tag}_K(M) \;.$$

(a) Write a deterministic and stateless algorithm Vf' that would ensure that MA' satisfies the correctness condition.

Solution:
Algorithm $\mathsf{Vf}'_K(M, T)$:
Parse T as $T1$ and $T2$ such that $|T1| = |T2|$
If $\mathsf{Vf}_K(M, T1) = \mathsf{Vf}_K(M, T2) = 1$ then return 1 else return 0

(b) Is MA' necessarily a secure MAC scheme? Prove your answer. Be sure to provide a complete proof. Specifically, if you answer yes, specify a reduction along with an analysis relating the advantages of relevant adversaries and their resource usage. If you answer no, specify a counterexample, an attack, and an analysis of the adversary's advantage and resource usage. As always, an adversary requiring a minimal amount of resources while achieving a high advantage value is better.

Solution: This scheme is secure. Given a forger A against MA', we construct a forger B against MA as follows.

Adversary $B^{\mathtt{Tag},\mathtt{Vf}}$:
Run A answering its queries as follows:
On a tagging query M:
$T1 \xleftarrow{\$} \mathtt{Tag}(M)$
$T2 \xleftarrow{\$} \mathtt{Tag}(M)$
Return $T1\|T2$ to A
On a verification query (M, T):
Parse T as $T1\|T2$ where $|T1| = |T2|$
If $\mathtt{Vf}(M, T1) = \mathtt{Vf}(M, T2) = 1$ then return 1 else return 0
Until A halts

Let S' be the set maintained by the challenger in $\mathbf{Exp}^{\text{suf-cma}}_{\mathsf{MA}'}(A)$, and let S be the set maintained by the challenger in $\mathbf{Exp}^{\text{suf-cma}}_{\mathsf{MA}}(B)$. Also, let $(M, T1\|T2)$ be A's winning query. Then, we know that

1. when A submits $(M, T1\|T2)$ to the verification oracle, S' does not contain $(M, T1\|T2)$, and
2. $\mathsf{Vf}'_K(M, T1\|T2)$ returns 1.

The first fact means that either $(M, T1)$ or $(M, T2)$ or both are not in S. The second fact means that $\mathsf{Vf}_K(M, T1) = \mathsf{Vf}_K(M, T2) = 1$ according to how the verification algorithm works as specified in part (a). Thus, both of B's verification queries corresponding to A's winning query are valid. Putting the two facts together, we have that either or both $(M, T1)$ and $(M, T2)$ are winning queries for B. Thus, B's advantage is

$$\mathbf{Adv}^{\text{suf-cma}}_{\mathsf{MA}}(B) \geq \mathbf{Adv}^{\text{suf-cma}}_{\mathsf{MA}'}(A) \ .$$

Thus, if MA is secure, the left quantity will be small, making the right quantity small. Therefore, MA' will be secure as well.

Resource usage: Suppose A submits q_s and q_v queries to its tagging and verification oracles totalling μ_s and μ_v bits, respectively. Then, B submits $2q_s$ and $2q_v$ tagging and verification queries, respectively. Furthermore, the number of bits for B's tagging and verification oracle queries are $2\mu_s$ and at most $\mu_v + q_v * m$ where m is the maximum message length among A's tagging queries, respectively. The running time of B is essentially that of A, plus the time it takes to execute the tagging and verification oracles and constant time for the string manipulation involved.

6. Let n be a positive integer. Recall that $[\mathsf{KG}]$ denotes the set of all possible keys output by the algorithm KG. Let $\mathsf{MA}_1 = (\mathsf{KG}, \mathsf{Tag}_1, \mathsf{Vf}_1)$ and $\mathsf{MA}_2 = (\mathsf{KG}, \mathsf{Tag}_2, \mathsf{Vf}_2)$ be MAC schemes secure under the SUF-CMA security notion, and let $\{0,1\}^n$ be the message space for MA_1 and MA_2. We define $\mathsf{MA}_3 = (\mathsf{KG}, \mathsf{Tag}_3, \mathsf{Vf}_3)$ where, for all $M \in \{0,1\}^n$, for all $K \in [\mathsf{KG}]$,

$$\mathsf{Tag}_3(K, M) \ = \ \mathsf{Tag}_1(K, M)\|\mathsf{Tag}_2(K, M) \ .$$

(Note that the notation here is slightly different from the previous question to avoid potential confusion regarding the algorithm name and the subscript K.)

(a) Write a deterministic and stateless algorithm Vf_3 that would ensure that MA_3 satisfies the correctness condition.

Let $t1$ and $t2$ be the lengths of the tags under the scheme MA_1 and MA_2, respectively.

Solution:

Algorithm $\mathsf{Vf}_3(K, M, T)$:

Parse T as $T1$ and $T2$ such that $|T1| = t1$ and $|T2| = t2$

If $\mathsf{Vf}_1(K, M, T1) = \mathsf{Vf}_2(K, M, T2) = 1$ then return 1 else return 0

(b) Is MA_3 necessarily a secure MAC scheme? Prove your answer. Be sure to provide a complete proof. Specifically, if you answer yes, specify a reduction along with an analysis relating the advantages of relevant adversaries and their resource usage. If you answer no, specify a counterexample, an attack, and an analysis of the adversary's advantage and resource usage. As always, an adversary requiring a minimal amount of resources while achieving a high advantage value is better.

Solution: This scheme is not secure. Consider, for example, schemes MA_1 and MA_2 such that Tag_1 and Tag_2 have the following relationship: for all $K \in [\mathsf{KG}]$ and $M \in \{0,1\}^n$,

$$\mathsf{Tag}_2(K, M) = \mathsf{Tag}_1(K, \overline{M}) \tag{11}$$

where $\overline{M}$ denotes $M \oplus 1^n$. We present the following adversary attacking MA_3.

Adversary $A^{\mathtt{Tag},\mathtt{Vf}}$:

(1) $T \stackrel{\$}{\leftarrow} \mathtt{Tag}(0^n)$

(2) Parse T as $T1 \| T2$ such that $|T1| = t1$ and $|T2| = t2$

(3) $\mathtt{Vf}(1^n, T2 \| T1)$

Let S be the set maintained by the challenger in $\mathbf{Exp}^{\text{suf-cma}}_{\mathsf{MA}_3}(A)$. We argue that

1. S does not contain the forgery $(1^n, T2\|T1)$ at the time the query to the verification oracle is made on line (3).

   Justification: After line (1) is executed, $S = \{(0^n, T)\}$. There are no further tagging queries thereafter. Thus, when line (3) is executed, S does not contain $(1^n, T2\|T1)$.

2. $(1^n, T2\|T1)$ is a winning query, i.e., $\mathsf{Vf}_3(K, 1^n, T2\|T1) = 1$.

   Justification: From lines (1) and (2), the construction of MA_3, and Equation (11), we know that

$$T1 = \mathsf{Tag}_1(K, 0^n) = \mathsf{Tag}_2(K, 1^n) \tag{12}$$
$$T2 = \mathsf{Tag}_2(K, 0^n) = \mathsf{Tag}_1(K, 1^n) \tag{13}$$
$$\mathsf{Vf}_1(K, 1^n, T2) = \mathsf{Vf}_2(K, 0^n, T2) = 1 \tag{14}$$
$$\mathsf{Vf}_2(K, 1^n, T1) = \mathsf{Vf}_1(K, 0^n, T1) = 1 \tag{15}$$

   where Equation (14) and Equation (15) follow from Equation (13) and Equation (12), respectively.

   From our answer in part (a), the verification equation on the second line will return 1 since $\mathsf{Vf}_3(K, 1^n, T2\|T1)$ results in $\mathsf{Vf}_1(K, 1^n, T2)$ and $\mathsf{Vf}_2(K, 1^n, T1)$, both of which return 1 due to Equation (14) and Equation (15), respectively.

Thus, A's advantage is as follows:

$$\mathbf{Adv}^{\text{suf-cma}}_{\mathsf{MA}_3}(A) = \Pr\left[\, \mathbf{Exp}^{\text{suf-cma}}_{\mathsf{MA}_3}(A) \Rightarrow \mathbf{T} \,\right] = 1\,.$$

Resource usage: A submits 1 tagging query of length n and 1 verification query of length $n + t1 + t2$. The running time is the time it takes to execute the two oracle queries and the constant time for manipulating the strings.