All-Or-Nothing-Transforms (AONTs)

Ruth Ng

June 6, 2017

Sections 1 to 6 were written in preparation for my presentation. Section 7 onwards are made up of comments and additions made after in-class discussions.

1 Syntax

An *All-Or-Nothing-Transform* AONT specifies two algorithms (AONT.Transform, AONT.Inverse), and a block length AONT.bl. Then, we can associate with AONT a domain and range, AONT.Dom, AONT.Rng $\subset$ $\{\{0,1\}^{\mathsf{AONT.bl}}\}^*$(the set of strings having length that is a multiple of AONT.bl). We call the domain the "message sequences" and the range the "pseudo-message sequences". Then, we have that AONT.Transform : AONT.Dom $\to$ AONT.Rng, a randomized algorithm, and AONT.Inverse : AONT.Rng $\to$ AONT.Dom, a deterministic algorithm.

2 Correctness

The correctness condition for AONT is

$$\Pr\left[\,\mathsf{AONT.Inverse}(\mathsf{AONT.Transform}((m_1, m_2 \ldots m_s)) = (m_1, m_2, \ldots m_s))\,\right] = 1$$

where the probability is taken over all possible message sequences $(m_1, m_2 \ldots m_s)$ and all possible randomness of the AONT.Transform function. We also assume that (assumed but not explicitly stated in all the papers):

$$\Pr\left[\,M, N \in \mathsf{AONT.Dom}, |M| = |N|, X \leftarrow_\$ \mathsf{AONT.Transform}(M), Y \leftarrow_\$ \mathsf{AONT.Transform}(N) : |X| = |Y|\,\right] = 1$$

3 Rivest (1997)

$$
\begin{array}{l}
\underline{\mathbf{G}^{\mathrm{ind}}_{\mathsf{AONT}}(A)} \\
\quad b \leftarrow_{\$} \{0,1\} \\
\quad b' \leftarrow_{\$} A^{\mathrm{LR}} \\
\quad \mathbf{return}\ (b = b') \\[4pt]
\underline{\mathrm{LR}(M, N, i)} \\
\quad \mathbf{if}\ |M| \neq |N|\ \mathbf{then} \\
\qquad |\quad \mathbf{return}\ \bot \\
\quad \mathbf{end} \\
\quad \mathbf{if}\ b = 0\ \mathbf{then} \\
\qquad |\quad (m_1, m_2, \ldots m_{s'}) \leftarrow_{\$} \mathsf{AONT.Transform}(M) \\
\quad \mathbf{else} \\
\qquad |\quad (m_1, m_2, \ldots m_{s'}) \leftarrow_{\$} \mathsf{AONT.Transform}(N) \\
\quad \mathbf{end} \\
\quad \mathbf{if}\ i > s'\ \mathbf{then} \\
\qquad |\quad \mathbf{return}\ \bot \\
\quad \mathbf{end} \\
\quad m_i \leftarrow \epsilon \\
\quad \mathbf{return}\ (m_1, m_2, \ldots m_{s'})
\end{array}
$$

Then we say that the indistinguishability adversary A has AONT-IND advantage:

$$
\mathbf{Adv}^{\mathrm{aont\text{-}ind}}_{\mathsf{AONT}}(A) = 2 \cdot \Pr\left[\mathbf{G}^{\mathrm{ind}}_{\mathsf{AONT}}(A)\right] - 1
$$

4 Boyko (1999)/ Canetti et. al (2000)

$$
\begin{array}{|l|}
\hline
\end{array}
$$

$\mathbf{G}^{\mathrm{leak}}_{\mathsf{AONT},l}(A)$

    $b \leftarrow_\$ \{0,1\}$

    $b' \leftarrow_\$ A^{\mathrm{LR}}$

    return $(b = b')$

$\mathrm{LR}(M, N, S)$

    if $|M| \neq |N|$ **then**

    | **return** $\perp$

    end

    if $b = 0$ **then**

    | $y \leftarrow_\$ \mathsf{AONT}.\mathsf{Transform}(M)$

    else

    | $y \leftarrow_\$ \mathsf{AONT}.\mathsf{Transform}(N)$

    end

    if $(|S| \neq |y|) \vee (\mathsf{Hamm}(S) > (|y| - l))$ **then**

    | **return** $\perp$

    end

    $y \leftarrow y \mathbin{\&} S$

    return y

Note that $|M|$ is the length of the string M in bits, $\&$ is a bitwise AND and $\mathsf{Hamm}(M)$ takes the hamming weight of M

Then we say that the leakage adversary A has l-AONT-LEAK advantage:

$$\mathbf{Adv}^{\mathrm{aont\text{-}leak}}_{\mathsf{AONT},l}(A) = 2 \cdot \Pr\left[\mathbf{G}^{\mathrm{leak}}_{\mathsf{AONT},l}(A)\right] - 1$$

5   Leakage Resilience Model

$$
\boxed{
\begin{array}{l}
\underline{\mathbf{G}^{\text{lr}}_{\text{AONT},m}(A)} \\
\quad b \leftarrow\!\!{}_\$ \{0,1\} \\
\quad b' \leftarrow\!\!{}_\$ A^{\text{LR}} \\
\quad \textbf{return } (b = b') \\
\underline{\text{LR}(M, N, C)} \\
\quad \textbf{if } |M| \neq |N| \textbf{ then} \\
\quad\quad | \quad \textbf{return } \bot \\
\quad \textbf{end} \\
\quad \textbf{if } b = 0 \textbf{ then} \\
\quad\quad | \quad y \leftarrow\!\!{}_\$ \text{AONT.Transform}(M) \\
\quad \textbf{else} \\
\quad\quad | \quad y \leftarrow\!\!{}_\$ \text{AONT.Transform}(N) \\
\quad \textbf{end} \\
\quad \textbf{if } (C \notin \mathcal{C}_{|y|,(|y|-m)}) \textbf{ then} \\
\quad\quad | \quad \textbf{return } \bot \\
\quad \textbf{end} \\
\quad \textbf{return } C(y)
\end{array}
}
$$

Note that $\mathcal{C}_{n,m}$ is the set of boolean circuits taking n inputs and m outputs, expressed in a string in some reasonable encoding. Then, for $C \in \mathcal{C}_{n,m}$, when we run $C(S)$ for some binary string S of length n, C will take as input the bits of S and return a m bit long string.

Then we say that the leakage resilience adversary A has m-AONT-LR advantage:

$$
\mathbf{Adv}^{\text{aont-lr}}_{\text{AONT},m}(A) = 2 \cdot \Pr\left[\mathbf{G}^{\text{lr}}_{\text{AONT},m}(A) \right] - 1
$$

6   Relationship between Notions

6.1   AONT.bl − AONT − LEAK $\implies$ AONT − IND

Theorem 6.1 *For any* AONT − IND *adversary* A, *we can construct* AONT.bl − AONT − LEAK *adversary* B, *running in the same time and making the same number of queries, such that*

$$
\boldsymbol{Adv}^{aont\text{-}ind}_{\text{AONT}}(A) \leq \boldsymbol{Adv}^{aont\text{-}leak}_{\text{AONT,AONT.bl}}(B)
$$

Here is the adversary:

$$
\boxed{
\begin{array}{l}
\underline{B^{\mathrm{LR}}} \\
\quad b \leftarrow_\$ A^{\mathrm{SIMLR}} \\
\quad \textbf{return } b \\
\underline{\mathrm{SIMLR}(M, N, i)} \\
\quad mask \leftarrow \epsilon \\
\quad s \leftarrow \lceil \frac{|M|}{\mathsf{AONT.bl}} \rceil \\
\quad \textbf{for } j = 1, 2, \ldots s \textbf{ do} \\
\qquad \textbf{if } j \neq i \textbf{ then} \\
\qquad\quad |\quad mask \leftarrow mask || 1^{\mathsf{AONT.bl}} \\
\qquad \textbf{else} \\
\qquad\quad |\quad mask \leftarrow mask || 0^{\mathsf{AONT.bl}} \\
\qquad \textbf{end} \\
\quad \textbf{end} \\
\quad \textbf{return } \mathrm{LR}(M, N, mask)
\end{array}
}
$$

6.2 $l-\mathsf{AONT}-\mathsf{LR} \implies l-\mathsf{AONT}-\mathsf{LEAK}$

Theorem 6.2 *For any* $l-\mathsf{AONT}-\mathsf{LEAK}$ *adversary* A, *we can construct* $l-\mathsf{AONT}-\mathsf{LR}$ *adversary* B, *running in the same time and making the same number of queries, such that*

$$
\boldsymbol{Adv}^{aont\text{-}leak}_{\mathsf{AONT},l}(A) \leq \boldsymbol{Adv}^{aont\text{-}lr}_{\mathsf{AONT},l}(B)
$$

Here is the adversary:

$$
\boxed{
\begin{array}{l}
\underline{B^{\mathrm{LR}}} \\
\quad b \leftarrow_\$ A^{\mathrm{SIMLR}} \\
\quad \textbf{return } b \\
\underline{\mathrm{SIMLR}(M, N, mask)} \\
\quad \textbf{return } \mathrm{LR}(M, N, C_{mask}) \\
\underline{C_{mask}(X)} \\
\quad \textbf{return } mask \ \& \ X
\end{array}
}
$$

6.3 $\mathsf{AONT} - \mathsf{IND} \nRightarrow \mathsf{AONT.bl} - \mathsf{AONT} - \mathsf{LEAK}$

Consider the following AONT scheme, $\mathsf{Checksum}$, which is defined for all choices of $\mathsf{Checksum.bl}$. It has $\mathsf{Checksum.Dom} = \{0,1\}^{\mathsf{Checksum.bl}}$ and $\mathsf{Checksum.Rng} = \{0,1\}^{\mathsf{Checksum.bl}^2}$:

$\mathsf{Checksum.Transform}(m)$	$\mathsf{Checksum.Inverse}(m'_1, m'_2 \ldots m'_{\mathsf{Checksum.bl}})$
$m'_{\mathsf{Checksum.bl}} \leftarrow m$ $\textbf{for } i = 1, 2, \ldots \mathsf{Checksum.bl} - 1 \textbf{ do}$ $\quad \| \ m'_i \leftarrow_\$ \{0,1\}^{\mathsf{Checksum.bl}}$ $\quad \| \ m'_{\mathsf{Checksum.bl}} \leftarrow m'_i \oplus m'_{\mathsf{Checksum.bl}}$ $\textbf{end}$ $\textbf{return } (m'_1, m'_2 \ldots m'_{\mathsf{Checksum.bl}})$	$m \leftarrow m'_{\mathsf{Checksum.bl}}$ $\textbf{for } i = 1, 2, \ldots \mathsf{Checksum.bl} - 1 \textbf{ do}$ $\quad \| \ m \leftarrow m'_i \oplus m$ $\textbf{end}$ $\textbf{return } m$

First, let's show that Checksum is $\mathsf{Checksum.bl - AONT - IND}$ secure. Since the pseudo-message blocks are such that $m = \oplus_{i=1}^{\mathsf{Checksum.bl}} m_i'$, and $\mathsf{Checksum.bl} - 1$ blocks were chosen at random, independent of m, the loss of any one block will render the distribution of the remaining blocks completely independent of m. Therefore, (much like in the one-time pad), $\mathbf{Adv}_{\mathsf{Checksum}}^{\text{aont-ind}}(A) = 0$ for all A.

Next, we can provide a $\mathsf{Checksum.bl - AONT - LEAK}$ adversary A.

$$
\boxed{
\begin{aligned}
&\underline{A^{\text{LR}}} \\
&\quad mask \leftarrow \epsilon \textbf{ for } i = 1, 2 \ldots \mathsf{Checksum.bl} \textbf{ do} \\
&\quad \quad \mid \quad mask \leftarrow mask || 1^{\mathsf{Checksum.bl}} 0 \\
&\quad \textbf{end} \\
&\quad (m_1', m_2' \ldots m_{\mathsf{Checksum.bl}}') \leftarrow \text{LR}(0^{\mathsf{Checksum.bl}}, 1^{\mathsf{Checksum.bl}}, mask) \\
&\quad m \leftarrow \oplus_{i=1}^{\mathsf{Checksum.bl}} (m_i') \\
&\quad m \leftarrow m \ \& \ 1^{\mathsf{Checksum.bl}} 0 \\
&\quad \textbf{if } m = 0^{\mathsf{Checksum.bl}} \textbf{ then} \\
&\quad \quad \mid \quad \textbf{return } 0 \\
&\quad \textbf{else} \\
&\quad \quad \mid \quad \textbf{return } 1 \\
&\quad \textbf{end}
\end{aligned}
}
$$

Then we have that $\mathbf{Adv}_{\mathsf{Checksum}, \mathsf{Checksum.bl}}^{\text{aont-leak}}(A) = 1$, since the adversary is able to retrieve any $\mathsf{Checksum.bl} - 1$ bits of the original message.

6.4 $\mathsf{AONT.bl - AONT - LEAK} \not\Rightarrow \mathsf{AONT.bl - AONT - LR}$

This is in the context of the RO model, specifically where a secure instance of OAEP is assumed.

Consider the package transform proposed by Rivest, denoted $\mathsf{Package}$. Note that $\mathsf{Package.Dom} = \mathsf{Package.Rng} = \{X \in \{0,1\}^* : |X| \text{ is a multiple of } \mathsf{AONT.bl}\}$

$$
\begin{aligned}
&\underline{\mathsf{Package.Transform}(m_1, m_2, \ldots m_s)} \\
&\quad K \leftarrow_{\$} \{0,1\}^{\mathsf{Package.bl}} \\
&\quad K' \leftarrow_{\$} \{0,1\}^{\mathsf{Package.bl}} \\
&\quad m_{s+1}' \leftarrow K' \\
&\quad \textbf{for } i = 1, 2 \ldots s \textbf{ do} \\
&\quad \quad \mid \ m_i' \leftarrow m_i \oplus E(K', \langle i \rangle) \\
&\quad \quad \mid \ h_i \leftarrow E(K, m_i' \oplus \langle i \rangle) \\
&\quad \quad \mid \ m_{s+1}' \leftarrow m_{s+1}' \oplus h_i \\
&\quad \textbf{end} \\
&\quad \textbf{return } (m_1', m_2' \ldots m_s', m_{s+1}', K)
\end{aligned}
\qquad
\begin{aligned}
&\underline{\mathsf{Package.Inverse}(m_1', m_2' \ldots m_{s'}')} \\
&\quad \textbf{if } (s' \leq 2) \textbf{ then} \\
&\quad \quad \mid \quad \textbf{return } \perp \\
&\quad \textbf{end} \\
&\quad K \leftarrow m_{s'}' \\
&\quad K' \leftarrow m_{s'-1}' \\
&\quad s \leftarrow s' - 2 \\
&\quad \textbf{for } (i = 1, 2, \ldots s) \textbf{ do} \\
&\quad \quad \mid \ h_i \leftarrow E(K, m_i' \oplus \langle i \rangle) \\
&\quad \quad \mid \ K' \leftarrow K' \oplus h_i \\
&\quad \textbf{end} \\
&\quad \textbf{for } (i = 1, 2, \ldots s) \textbf{ do} \\
&\quad \quad \mid \ m_i \leftarrow E(K', \langle i \rangle) \oplus m_i' \\
&\quad \textbf{end} \\
&\quad \textbf{return } (m_1, m_2 \ldots m_s)
\end{aligned}
$$

Then, from Boyko (1999) we know that when OAEP is used as E, we have that $\mathsf{Package}$ is

AONT.bl − AONT − IND − LEAK is as secure as OAEP. We now present a AONT.bl − AONT − IND − LR adversary A.

$$
\begin{array}{|l|}
\hline
\underline{A^{\mathrm{LR}}} \\
\quad X || 0^{\mathsf{AONT.bl}} \leftarrow_{\$} \mathrm{LR}(0^{\mathsf{AONT.bl}}, 1^{\mathsf{AONT.bl}}, C) \\
\quad \textbf{if } X = 1^{\mathsf{AONT.bl}} \textbf{ then} \\
\quad \quad | \quad \textbf{return } 1 \\
\quad \textbf{else} \\
\quad \quad | \quad \textbf{return } 0 \\
\quad \textbf{end} \\
\underline{C(X)} \\
\quad Y \leftarrow \mathsf{Package.Inverse}(X) \\
\quad \textbf{return}(X || 0^{\mathsf{AONT.bl}}) \\
\hline
\end{array}
$$

Then we have that $\mathbf{Adv}^{\text{aont-lr}}_{\mathsf{Package},\mathsf{Package.bl}}(A) = 1$, by the correctness condition of the AONT. One can note that the A runs in time proportional to the running time of Package.Inverse, which should be PT in any usable AONT. Note that the padding of $0^{\mathsf{AONT.bl}}$ in the circuit is for convenience since the output of Inverse is only 1 block long, while the input to it is 3 blocks.

From class, this argument assumes the existence of random oracles. Which, if we do, makes this definitions slightly more like the one in section 7 instead. Therefore, I'm not entirely sure how much this counterexample matters/ holds up

7   Trivial Attack on LR Model

As discussed in class, in a similar vein to the adversary in the previous subsection, we can construct an adversary for an arbitrary AONT which makes it not even $1 − \mathsf{AONT} − \mathsf{LR}$ secure (and by extension not $\mathsf{m} − \mathsf{AONT} − \mathsf{LR}$ secure for any m. Consider the following adversary A:

$$
\begin{array}{|l|}
\hline
\underline{A^{\mathrm{LR}}} \\
\quad X \leftarrow_{\$} \mathrm{LR}(0^{\mathsf{AONT.bl}}, 1^{\mathsf{AONT.bl}}, C) \\
\quad \textbf{return } X \\
\underline{C(X)} \\
\quad Y \leftarrow \mathsf{Package.Inverse}(X) \\
\quad \textbf{return}(Y \ \& \ 1) \\
\hline
\end{array}
$$

Then we have that

$$\mathbf{Adv}^{\text{aont-lr}}_{\text{Package,Package.bl}}(A)$$

$$= 2 \cdot \Pr\left[\mathbf{G}^{\text{lr}}_{\text{AONT},m}(A)\right] - 1$$

$$= \Pr\left[\mathbf{G}^{\text{lr}}_{\text{AONT},m}(A) \mid b = 1\right] - \Pr\left[\mathbf{G}^{\text{lr}}_{\text{AONT},m}(A) \mid b = 0\right]$$

$$= 1 - 0 = 1$$

(By the correctness condition)

8 Improvement to the LR Model

We discussed improving the model by allowing a random oracle to Transform and Inverse, which is not provided to the circuits. This will allow for Transform, Inverse algorithms that the circuits cannot simulate fully.

The intuition behind this is that the LR model considers a side-channel attack. The "circuits", in real life, will not be performing complex hashing or block ciphers when they leak information, given that most of the information comes in the form of power analysis and other similar side channels.