1 Syntax

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

$\underline{\mathbf{G}^{\mathrm{ind}}_{\mathsf{AONT}}(A)}$

$b \leftarrow\!\!\$\; \{0,1\}$
$b' \leftarrow\!\!\$\; A^{\mathrm{LR}}$
return $(b = b')$

$\underline{\mathrm{LR}(M, N, i)}$

if $|M| \neq |N|$ **then**
  | **return** $\perp$
end
if $b = 0$ **then**
  | $(m_1, m_2, \ldots m_{s'}) \leftarrow\!\!\$\; \mathsf{AONT.Transform}(M)$
else
  | $(m_1, m_2, \ldots m_{s'}) \leftarrow\!\!\$\; \mathsf{AONT.Transform}(N)$
end
if $i > s'$ **then**
  | **return** $\perp$
end
$m_i \leftarrow \epsilon$
return $(m_1, m_2, \ldots m_{s'})$

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

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

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

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

$b \leftarrow_{\$} \{0,1\}$
$b' \leftarrow_{\$} A^{\text{LR}}$
return $(b = b')$

$\underline{\text{LR}(M,N,S)}$

if $|M| \neq |N|$ **then**
  return $\perp$
end
if $b = 0$ **then**
  $y \leftarrow_{\$} \mathsf{AONT.Transform}(M)$
else
  $y \leftarrow_{\$} \mathsf{AONT.Transform}(N)$
end
if $(|S| \neq |y|) \vee (\mathsf{Hamm}(S) > (|y| - l))$ **then**
  return $\perp$
end
$y \leftarrow y \;\&\; S$
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:

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

5 Leakage Resilience Model

$\underline{\mathbf{G}^{\mathrm{lr}}_{\mathsf{AONT},m}(A)}$

$b \leftarrow_\$ \{0,1\}$
$b' \leftarrow_\$ A^{\mathrm{LR}}$
return $(b = b')$

$\underline{\mathrm{LR}(M, N, C)}$

if $|M| \neq |N|$ **then**
| **return** $\perp$
end
if $b = 0$ **then**
| $y \leftarrow_\$ \mathsf{AONT.Transform}(M)$
else
| $y \leftarrow_\$ \mathsf{AONT.Transform}(N)$
end
if $(C \notin \mathcal{C}_{|y|,(|y|-m)})$ **then**
| **return** $\perp$
end
return $C(y)$

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:

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

6 Relationship between Notions

6.1 $\mathsf{AONT.bl} - \mathsf{AONT} - \mathsf{LEAK} \implies \mathsf{AONT} - \mathsf{IND}$

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

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

Here is the adversary:

```
B^LR
    b <-$ A^SimLR
    return b
SimLR(M, N, i)
    mask <- ε
    s <- ⌈|M| / AONT.bl⌉
    for j = 1, 2, ... s do
        if j ≠ i then
            mask <- mask||1^AONT.bl
        else
            mask <- mask||0^AONT.bl
        end
    end
    return LR(M, N, mask)
```

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*

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

Here is the adversary:

```
B^LR
    b <-$ A^SimLR
    return b
SimLR(M, N, mask)
    return LR(M, N, C_mask)
C_mask(X)
    return mask & X
```

6.3 $\mathsf{AONT}-\mathsf{IND} \not\Rightarrow \mathsf{AONT.bl}-\mathsf{AONT}-\mathsf{LEAK}$

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

Checksum.Transform(m)	Checksum.Inverse($m'_1, m'_2 \dots m'_{\mathsf{Checksum.bl}}$)

$m'_{\mathsf{Checksum.bl}} \leftarrow m$
for $i = 1, 2, \dots \mathsf{Checksum.bl} - 1$ **do**
    $m'_i \leftarrow\$ \{0,1\}^{\mathsf{Checksum.bl}}$
    $m'_{\mathsf{Checksum.bl}} \leftarrow m'_i \oplus m'_{\mathsf{Checksum.bl}}$
end
return $(m'_1, m'_2 \dots m'_{\mathsf{Checksum.bl}})$
	$m \leftarrow m'_{\mathsf{Checksum.bl}}$
	for $i = 1, 2, \dots \mathsf{Checksum.bl} - 1$ **do**
	    $m \leftarrow m'_i \oplus m$
	end
	return m

First, let's show that $\mathsf{Checksum}$ is $\mathsf{Checksum.bl} - \mathsf{AONT} - \mathsf{IND}$ 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), $\mathbf{Adv}^{\text{aont-ind}}_{\mathsf{Checksum}}(A) = 0$ for all A.

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

```
A^LR
  mask ← ε for i = 1, 2 ... Checksum.bl do
  |  mask ← mask||1^Checksum.bl 0
  end
  (m'_1, m'_2 ... m'_Checksum.bl) ← LR(0^Checksum.bl, 1^Checksum.bl, mask)
  m ← ⊕_{i=1}^{Checksum.bl} (m'_i)
  m ← m & 1^Checksum.bl 0
  if m = 0^Checksum.bl then
  |  return 0
  else
  |  return 1
  end
```

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

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

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 $\mathsf{Package.Dom} = \mathsf{Package.Rng} = \{X \in \{0,1\}^* : |X| \text{ is a multiple of } \mathsf{AONT.bl}\}$

```
Package.Transform(m_1, m_2, ... m_s)
  K ←$ {0,1}^Package.bl
  K' ←$ {0,1}^Package.bl
  m'_{s+1} ← K'
  for i = 1, 2 ... s do
  |  m'_i ← m_i ⊕ E(K', ⟨i⟩)
  |  h_i ← E(K, m'_i ⊕ ⟨i⟩)
  |  m'_{s+1} ← m'_{s+1} ⊕ h_i
  end
  return (m'_1, m'_2 ... m'_s, m'_{s+1}, K)
```

```
Package.Inverse(m'_1, m'_2 ... m'_{s'})
  if (s' ≤ 2) then
  |  return ⊥
  end
  K ← m'_{s'}
  K' ← m'_{s'-1}
  s ← s' − 2
  for (i = 1, 2, ... s) do
  |  h_i ← E(K, m'_i ⊕ ⟨i⟩)
  |  K' ← K' ⊕ h_i
  end
  for (i = 1, 2, ... s) do
  |  m_i ← E(K', ⟨i⟩) ⊕ m'_i
  end
  return (m_1, m_2 ... m_s)
```

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

$\mathsf{AONT.bl} - \mathsf{AONT} - \mathsf{IND} - \mathsf{LEAK}$ is as secure as OAEP. We now present a $\mathsf{AONT.bl} - \mathsf{AONT} - \mathsf{IND} - \mathsf{LR}$ adversary A.

$\underline{A^{\mathrm{LR}}}$

$X||0^{\mathsf{AONT.bl}} \leftarrow\!\!\$\ \mathrm{LR}(0^{\mathsf{AONT.bl}}, 1^{\mathsf{AONT.bl}}, C)$

if $X = 1^{\mathsf{AONT.bl}}$ **then**

  return 1

else

  return 0

end

$\underline{\mathrm{C(X)}}$

$Y \leftarrow \mathsf{Package.Inverse}(X)$

return$(X||0^{\mathsf{AONT.bl}})$

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 $\mathsf{Package.Inverse}$, which should be PT in any usable AONT.