1 Rivest (1997)

1.1 Original Definition

The original definition of Rivest looks like this:

Consider a function f that takes the message sequence $m_1, m_2 \dots m_s$ and returns pseudo-message sequence $m'_1, m'_2 \dots m'_{s'}$. We call f an AONT when the following are satisfied:

- The transformation f is reversible: Given the pseudo-message sequence, one can obtain the original message sequence.

- Both f and its inverse are efficiently computable (PT)

- It is computationally infeasible to compute any function of any message block if any one of the pseudo-message blocks is unknown.

1.2 Formalized Definition

This is my attempt to formalize this definition with concrete security games.

Note that because we are talking in concrete security, I make no comment on efficiency of any of the algorithms

An *All-Or-Nothing-Transform* AONT specifies two algorithms ($\mathsf{AONT.Transform}$, $\mathsf{AONT.Inverse}$), and a block length $\mathsf{AONT.bl}$. We have that $\mathsf{AONT.Transform} : \{\{0,1\}^{\mathsf{AONT.bl}}\}^* \to \{\{0,1\}^{\mathsf{AONT.bl}}\}^*$. We call the domain of this function "message sequences" and the range "pseudo-message sequences". Then $\mathsf{AONT.Inverse}$ is the inverse of this function, meaning that $\mathsf{AONT.Inverse} : \{\{0,1\}^{\mathsf{AONT.bl}}\}^* \to \{\{0,1\}^{\mathsf{AONT.bl}}\}^*$, a mapping that only needs to be defined on pseudo-message sequences that can be generated by $\mathsf{AONT.Transform}$. $\mathsf{AONT.Transform}$ can (and should) be randomized, while $\mathsf{AONT.Inverse}$ is not randomized.

The correctness condition for AONT is

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

where the probability is taken over all possible message sequences $(m_1, m_2 \dots m_s)$ and all possible randomness of the $\mathsf{AONT.Transform}$ function.

Now we can define the following security game:

$\underline{\mathbf{G}^{\text{aont}}_{\mathsf{AONT}}(A)}$

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

$\underline{\text{LR}(m^*, n^*, i, j)}$

if
$(|m^*| \neq \mathsf{AONT.bl}) \vee (|n^*| \neq \mathsf{AONT.bl}) \vee (i > j)$
then
return $\perp$
end
for $(x = 1, 2, \ldots j)$ **do**
if $(x \neq i)$ **then**
$m_x \leftarrow_{\$} \{0,1\}^{\mathsf{AONT.bl}}$
$n_x \leftarrow_{\$} \{0,1\}^{\mathsf{AONT.bl}}$
else
$m_x \leftarrow m^*$
$n_x \leftarrow n^*$
end
end
if $b = 0$ **then**
$y \leftarrow_{\$} \mathsf{AONT.Transform}(m_1, m_2 \ldots m_s)$
else
$y \leftarrow_{\$} \mathsf{AONT.Transform}(n_1, n_2 \ldots n_s)$
end
return y

Then we say that the AONT indistinguishability advantage of an A is given by:

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

1.3 Package Transform

Rivest provides a construction of an AONT proceeding as follows. It makes use of an arbitrary block cipher, with key of length $\mathsf{AONT.bl}$, $E : \{0,1\}^{\mathsf{AONT.bl}} \times \{0,1\}^{\mathsf{AONT.bl}} \to \mathsf{AONT.bl}$. I edited this a bit for clarity (with respect to the keys).

$\mathsf{AONT.Transform}(m_1, m_2, \ldots m_s)$	$\mathsf{AONT.Inverse}(m'_1, m'_2 \ldots m'_{s'})$

if $\exists i. \|m_i\| \neq \mathsf{AONT.bl}$ **then**
    return $\perp$
end
$K \leftarrow\!\!\$\ \{0,1\}^{\mathsf{AONT.bl}}$
$K' \leftarrow\!\!\$\ \{0,1\}^{\mathsf{AONT.bl}}$
$m'_{s+1} \leftarrow K'$
for $i = 1, 2 \ldots s$ **do**
    $m'_i \leftarrow m_i \oplus E(K', \langle i \rangle_{\mathsf{AONT.bl}})$
    $h_i \leftarrow E(K, m'_i \oplus \langle i \rangle_{\mathsf{AONT.bl}})$
    $m'_{s+1} \leftarrow m'_{s+1} \oplus h_i$
end
return $(m'_1, m'_2 \ldots m'_s, m'_{s+1}, K)$
	if $(\exists i. \|m'_i\| \neq \mathsf{AONT.bl}) \vee (s' \leq 2)$ **then**
	    return $\perp$
	end
	$K \leftarrow m'_{s'}$
	$K' \leftarrow m'_{s'-1}$
	$s \leftarrow s' - 2$
	for $(i = 1, 2, \ldots s)$ **do**
	    $h_i \leftarrow E(K, m'_i \oplus i)$
	    $K' \leftarrow K' \oplus h_i$
	end
	for $(i = 1, 2, \ldots s)$ **do**
	    $m_i \leftarrow E(K', i) \oplus m'_i$
	end
	return $(m_1, m_2 \ldots m_s)$

No proof is provided as to the security of this scheme. The phrasing of the "informal" proof also seems problematic because it talks about an adversary that is attempting to "compute any message block" instead of any function of any message block, which is not the same as was suggested by the definition.