1 Syntax

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

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.

2 Rivest (1997)

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

4 To dos

- Prove that the package transform with OAEP/ OWFs work (explicitly) for the Rivest definition/ strong-Rivest definition
- What is AONT used for and what kind of security do we need for that
- What is the application I was thinking of and what kind of security do we need for that?