Reading Guide For:Random Oracles Methodology , Revisited

Joann, Matt, Scott, Christine, Sujata

October 3, 2004

Summary

In this paper the authors consider the security of schemes in the random oracle model and the security of the resulting implementation . The result of the paper is negative:

"**There exist signature and encryption schemes which are secure in the Random Oracle Model, but for which any implementation of the random oracle results in insecure schemes**."

The authors illustrate the random oracle model and how to implement such a system . Then they present the necessary notation for the proofs and discuss some important desired properties of the implementation of the Random Oracle Model. Finally they present a formal proof for the negative result stated above.

Overview of Section 1

Section 1 gives an overview of the entire paper. They explain a popular methodology for designing cryptographic protocols pointing out that it is unclear how to put this methodology on firm ground. Rather than considering just a single function to replace the random oracle they propose replacing it with a function ensemble would still maintain the security of the scheme. Note that a function ensemble is a sequence of function families. That is it consists of several function families.

They suggest identifying useful properties of a random oracle and determine a function or a class of functions with these properties. They specifically consider a property called **correlation intractability** (Correlation intractability is explained briefly in section 1.2.1 and in detail in Section 3.) They conclude that a provably secure scheme in the Random Oracle Model can and does have structural flaws.

In **Section 1.1** they describe the random oracle model environment and explain how to implement it with a single function and with a function ensemble. Note that "implementing" an ideal system means that function f which will be evaluated each time the ideal system queries the oracle.

When using a single function, an ideal system Π is transformed to an interactive system in the real world by replacing the oracle with function f. They have a simple example where they show ideal system is secure but its implementation using f is not secure.

When using an ensemble $\mathcal{F}$ the random oracle is replaced by a function randomly selected from a collection of functions. A formal definition of a function ensemble is found in section 2. The example used for the single function proof cannot be applied for a function ensemble. They give a detailed proof for the function ensemble in **Section 3**

In **Section 1.2** they introduce correlation intractability, restricted correlation intractability, and present some informal theorems relating to the failure of the random oracle methodology. To understand correlation intractability, it is important to understand evasive relations. Basically a relation R which has access to a random oracle $\mathcal{O}$ is called evasive if given an input string x the pair $(x, O(x))$ is not in R.

Now correlation intractability of a function ensemble is a property in which given some function $f_s \in F_k$, it is hard to find some pre-images x such that $(x, f_s(x))$ satisfy some evasive relation R where satisfy means the inputs and outputs are in R.

They also talk about the notion of restricted correlation intractability where they show (in **Section 5**) that their negative result can be applied even in the case where the inputs to function f_s have a restricted length. **Note that it is not necessary for you to read Section 5 of this paper.**

Overview of Section 2

In this section they introduce some notation which will be frequently used in later proofs. Some important terms and notations that must be noted are

1. Polynomial Bound: A function $f(n)$ is said to be polynomially bounded if $f(n) = O(n^k)$.
2. l_{out}: This a length function which is one of the parameters which determines the set of possible functions. l_{out} is superlogarithmic and polynomially bounded. A possible example of l_{out} could be $l_{out}(k) = 3k$, which is polynomially bounded.
3. $l_{out} - ensemble$: This is denoted by $\mathcal{F}$. This consists of function families

$$\mathcal{F}_k = \{f_s : \{0,1\}^* \rightarrow \{0,1\}^{l_{out}(k)}\}_{s \epsilon \{0,1\}^k} \tag{1}$$

   In Equation 1

   (a) $\mathcal{F}_k$ denotes a function family

   (b) f_s is a particular function which maps from binary strings of any length to binary strings of length $l_{out}(k)$ where k is a security parameter.

   (c) Note that $\mapsto$ in this paper denotes the function mapping from the domain to the range (same as $\rightarrow$), this is NOT the standard.

   (d) Now to select a function to replace the oracle, you uniformly select s from $\{0,1\}^k$ and then choose the corresponding function, f_s. Note that here $|f_s(x)|$ = $l_out(k)$ implies the length of the output of function f_s on input x.

In addition to formally defining a function ensemble they have made important remarks regarding this definition. Remark 1 emphasizes that even though $f_s(\cdot)$ has been formally defined over all inputs, in cryptographic applications it is used with certain length restricted inputs. In Remark 2 they give an example of the statement made in Remark 1.

Overview of Section 3

Section 3 discusses the correlation intractability property and why the authors use this property. The reason for correlation intractability is that the authors need to describe a function ensemble but also show that it behaves like a random oracle. (Since the function definition is known, it cannot be a random function - input/output behavior can be determined.) To start their definition of correlation intractability, they begin with collision-intractability, which means it is difficult to find two preimages that the function maps to the same resultant image, i.e. two inputs that map to the same output — this is the same as collision

resistant. They go on to generalize this to correlation intractable — given any relation R it is infeasible to find a sequence of preimages and their images that satisfy R (the inputs and outputs are in R). So for a given relation say $y = 2x$, y = x^2 is not an evasive relation due to $< 2, 4 >$. In **Section 3.1** the authors present two definitions: evasive relations and correlation intractability. A relation R is an evasive relation if the probability of finding a preimage and image in R is negligible (the probability is extremely low - specifically $1/p(n)$ where p is a polynomial function). A function ensemble is correlation intractable if for every polynomial time machine M (a machine who outputs in $O(n^k)$ time)) and every evasive relation there is a negligible probability of finding an image and preimage in that relation. Finally in Section 3.2 the authors refute the existence of correlation-intractable ensembles by considering a machine that computes the identity function.

Overview of Section 4

In section 4 the authors construct signature and encryption schemes to prove that any implementation of the Random Oracle Model will be insecure. The hardest part of this section are a few complexity terms (like super polynomial) and the use of CS (Computationally Sound) proofs. We will explain these below.

The first part of section 4 gives an outline of how they structure their proofs. The basic steps they prove are

1. For any function ensemble $\mathcal{F}$ there exists a signature scheme which is secure in Random Oracle Model but is not secure when implemented using $\mathcal{F}$

2. There exists a signature scheme which is secure in Random Oracle Model but for which any implementation (not just $\mathcal{F}$ above) results in an insecure scheme. (To prove this the authors have used an extra time requirement which they remove in the next step.)

3. They prove the above can be done without the extra time requirement.

In section 4.1 First Step: they modify a generic signature scheme and show that it is indistinguishable from the original scheme when the relation R is evasive. **They conclude that while the above signature scheme is secure in Random Oracle Model when implemented with $\mathcal{F}$ the scheme is totally breakable under adaptive chosen message attack and existentially forgeable under key only attack.** They prove this using the intuition that when $\mathcal{F}$ is used to implement the scheme the seed s becomes a part of the public verification key and is known to the adversary.

In section 4.2 Second Step: To present a proof they have to consider an enumeration of all polynomial time algorithms. The running time of all the algorithms cannot be bounded by a single polynomial, so they introduce a super polynomial function to bound the running time. Basically the required super polynomial is a function whose running time is greater than the running time of each polynomial function in the enumeration.

The proof is similar to the one in Section 4.1 but instead of a single function ensemble they have an universal function ensemble which represents all possible function ensembles. The proof has the similar intuition as in 4.1 but concludes that any function ensemble implementation will be insecure even if its secure in the random oracle model.

The drawback of this proof is that it requires that signature and verification algorithms run in super polynomial time. This is because each algorithm has to compute the universal function ensemble which takes at most super polynomial time.

In section 4.3 Third Step: they eliminate the super polynomial time by introducing CS proofs. The CS proofs are defined in the appendix. A CS proof consists of two polynomial time oracle machines, a prover PRV, and a verifier VER. A prover computes a proof π and verifier either accepts or rejects π using the machines. The important part is that a CS proof provides the following:

1. Efficiency conditions: The Prover can solve things in polynomial time at worst.
2. Perfect completeness: The verifier always accepts the prover's proof with a probability of 1
3. Computational soundness: the chance of a verifier accepting a bad proof (not from the Prover) is minuscule $\leq poly(k + |w|)/2^k$

Going on to the proof: While the previous proofs checked for a relation, in this proof the signer and verifier receive a CS-proof of the claim that $(msg, \mathcal{O}(msg)) \in R^{\mathcal{U}}$ and check the validity of the CS-proof with the verifier VER. From the efficiency condition above the signing and verifying algorithms are now bounded by polynomial time.

In their definition of $S_u'^{\mathcal{O}}$ in the first step do note that $|(x, y)|$ denotes the length of $x + y$, x being the input and y the output.

Thus, the authors conclude from the above steps, that while the signature scheme is secure in the Random Oracle Model, any implementation results in signature scheme which can be forged, that is an insecure signature scheme.

In section 4.4 they show how public key encryption schemes which are proved to be secure in the Random Oracle Model, can yield insecure implementations. Note that the structure of Theorem 9 differs from Theorem 8 mainly in assuming that there exists a semantically secure encryption scheme in the Random Oracle Model. This proof is along the same lines as the proof in section 4.3.