انجمن رمز ایران
Iranian Society of Cryptology

دانشگاه شهید بهشتی

# On the Indifferentiability of Hash Functions

The Random Oracle model, introduced by Bellare and Rogaway [2], also well studied by Canetti *et al.* [3], has been one of the important tools used to prove the security of a cryptosystem based on hash functions. A common approach in using this model is as follows: Let $H^f$ represents a hash function based on the building block $f$ and $R$ denotes an ideal hash function or a random oracle. We denote the instantiations of any real cryptosystem $C(.)$ with $H^f$ and $R$ by $C(H^f)$ and $C(R)$ respectively. To prove the security of $C(H^f)$, first $C(R)$ will be proved secure. Next, it will be shown that the security of $C(R)$ will not be affected if $R$ is replaced with $H^f$. This is done using the notion of indistinguishability where the attacker interacts with $H^f$ directly but not $f$. $H^f$ and $R$ are said to be indistinguishable if no (efficient) distinguisher algorithm $D(.)$, which when connected to either $H^f$ or $R$, is able to decide whether it is interacting with $H^f$ or $R$.

On the other hand, almost all hash functions, e.g., MD4 [10], MD5 [9], SHA-0 [7], RIPEMD family [8] and SHA family [4] use a fixed length input compression function to process an arbitrary length message using MD structure. These compression functions play the role of $f$ in $H^f$. In reality, the distinguisher can access $f$ as a public parameter. To allow this ability to the distinguisher, the notion of indifferentiability has been introduced by Maurer *et al.* [6]. This notion is an extension of indistinguishability where the distinguisher can query both $H^f$ and $f$. In the notion of indifferentiability, if the component $H^f$ is indifferentiable from $R$, then the security of any cryptosystem $C(R)$ is not affected if one replaces $R$ with $H^f$ [6, p.3]. In [1], we observed that in the seminal work on indifferentiability analysis of iterated hash functions by Coron *et al.* and in subsequent works, the initial value ($IV$ ) of hash functions is *fixed*. In addition, these indifferentiability results do not depend on the *Merkle-Damgard˚ (MD) strengthening* in the padding functionality of the hash functions. We proposed a generic $n$-bit iterated hash function framework based on an $n$-bit compression function called Suffixfree-Prefix-free (SFPF) which works for *arbitrary IV* s and does not possess *MD strengthening* and formally proved that SFPF is indifferentiable from a random oracle (RO) when the compression function is viewed as a fixed input-length random oracle (FIL-RO). On the other hand, Preneel, Govaerts and Vandewalle (PGV) analysed the security of single-block-length block cipher based compression functions assuming that the underlying block cipher has no weaknesses. They showed that twelve out of sixtyfour possible compression functions are collision and ( second ) preimage resistant. Black, Rogaway and Shrimpton formally proved this result in the ideal cipher model. However, in the indifferentiability security framework introduced by Maurer, Renner and Holenstein, all these twelve schemes are easily differentiable from a fixed input-length random oracle (FIL-RO) even when their underlying block cipher is ideal. In [5], We addressed the problem of building indifferentiable compression functions from the PGV compression functions. We considered a general form of sixty-four PGV compression functions and replace the linear feed-forward operation in this generic PGV compression function with an ideal block cipher independent of the one used in the generic PGV construction. This modified construction is called a generic modified PGV (MPGV). We analysed indifferentiability of the generic MPGV construction in the ideal cipher model and show that twelve out of sixty-four MPGV compression functions in this framework are indifferentiable from a FIL-RO. To our knowledge, this was the first result showing that two independent block ciphers are sufficient to design indifferentiable single-blocklength compression functions.