VERSITA

TATRA MOUNTAINS
Mathematical Publications

DOI: 10.2478/v10127-012-0045-5
Tatra Mt. Math. Publ. **53** (2012), 155–187

ON PSEUDO-RANDOM ORACLES

Michal Rjaško

ABSTRACT. Many cryptographic systems which involve hash functions have proof of their security in a so called random oracle model. Behavior of hash functions used in such cryptographic systems should be as close as possible to the behavior of a random function. There are several properties of hash functions dealing with a random behavior. A hash function is pseudo-random oracle if it is indifferentiable from a random oracle. However, it is well known that hash functions based on the popular Merkle-Damgård domain extension transform do not satisfy the pseudo-random oracle property. On the other hand no attack is known for many concrete applications utilizing Merkle-Damgård hash functions. Hence, a weakened notion called public-use pseudo random oracle was introduced. The property can be met by the Merkle-Damgård construction and is sufficient for several important applications. A hash function is public use pseudo-random oracle if it is indifferentiable from a random oracle with public messages (i.e., all messages hashed so far are available to all parties). This is the case of most hash based signature schemes.

In this paper we analyze relationship between the property pseudo-random oracle and its variant public image pseudo-random oracle. Roughly, a hash function is public image pseudo-random oracle if it is indifferentiable from a random oracle with public images (i.e., all images of messages hashed so far are available to all parties, messages are kept secret). We prove that the properties are equivalent.

1. Introduction

The primary security property of cryptographic hash functions has historically been collision resistance. A hash function is collision resistant, if it is hard to find two different messages which hash to the same image. However, collision resistance alone is insufficient for arguing security of many important applications. For some of the applications (e.g., Fiat-Shamir signatures, RSA-FDH) a hash function must have "random behaviour", which is hard (or even impossible) to define in the standard model.

2010 Mathematics Subject Classification: 94A60.
Keywords: random oracle, cryptographic hash function, pseudo-random oracle.
This research was supported by the Comenius University Grant No. UK/426/2012.

Hence, Bellare and Rogaway [4] introduced a so called random oracle model, which models a hash function as a publicly available random function (random oracle). Using this framework, one can prove security of many important schemes. A proof in the random oracle model does not guarantee security when we replace the random oracle with a real hash function [6]. However, such a proof is believed to ensure that there are no structural flaws in the scheme and thus one can heuristically hope that the scheme remains flawless when the random oracle is replaced with a "well designed" hash function.

Cryptographic hash functions are often built using some domain extension transform (e.g., Merkle-Damgård construction [8], [15]) from a smaller primitive called compression function. On the other hand, in the random oracle model hash functions are modeled as monolithic random function without any subcomponents. Hence, Coron et al. [7] introduced a pseudo-random oracle property, which models a hash function as a function constructed by a domain extension transform from a fixed input length random oracle. A domain extension transform is pseudo-random oracle (pro) if it securely extends domain of an ideal compression function (i.e., fixed-input length random oracle) to a variable input length random oracle. The property is based on the indifferentiability framework [14]. Similarly to the random oracle model, the pseudo-random oracle property does not guarantee security when we replace ideal compression function with a real one. On the other hand, one can heuristically hope that if the real compression function is well designed, then also the resulting hash function is "good".

As it is shown in [7], the (strengthened) Merkle-Damgård construction [8], [15] is not pseudo-random oracle. However, still many applications utilize hash functions based on the Merkle-Damgård construction, but no practical attacks against these applications have been found. This leads to a disconnection between theory and practice. Dodis, Ristenpart and Shrimpton [9] presented a weaker security notion than the pseudo-random oracle called public use pseudo-random oracle (pub-pro). The property pub-pro is sufficient for arguing security of important applications (e.g., hash based digital signature schemes) and yet is met by the Merkle-Damgård transformation.

The property pub-pro guarantees security (in the random oracle model) of applications, which never evaluate hashes on secret inputs. That is, all messages (and thus corresponding images) evaluated so far are public and accessible to all adversaries. For example, this is the case of most of hash-based digital signature schemes. On the contrary, the pro property keeps all evaluated messages secret.

It is clear that the pro property is stronger than pub-pro, i.e., if a domain extension transform is pro, then it is pub-pro. The opposite direction does not hold, for example the Merkle-Damgård domain extension transform is pub-pro, but not pro [9].

Our contributions. In this paper we analyze a property somewhere between pub-pro and pro. In particular, we introduce a property called public image pseudo-random oracle (img-pro), for which only images of messages hashed so far are public. The messages are secret. Again, it is clear that pub-pro implies img-pro and img-pro implies pro.

The main goal of this paper is to prove that pro is equivalent to img-pro. Hence, it does not matter whether adversaries are able to see all images returned by a pseudo-random oracle. This corresponds to an intuition arising from the fact that output of the pseudo-random oracle should not reveal any information about the evaluated message.

Organization. In Section 2 we introduce some useful notations and definitions. In Section 3 we present formal definitions of the properties pro, pub-pro and img-pro. In Section 4 we show that pro is equivalent to img-pro.

2. Preliminaries

We write $M \xleftarrow{\$} \mathcal{S}$ for the uniform random selection of M from the set $\mathcal{S}$. Concatenation of finite strings M_1 and M_2 is denoted by $M_1||M_2$ or simply M_1M_2, $\overline{M}$ denotes bitwise complement of the string M. The ith bit of a string M is $M[i]$, thus $M = M[1]||\cdots||M[|M|]$. By $M_1, \ldots, M_l \xleftarrow{d} M$, where M is a string, we denote the following semantics:

(1) Pad M with the suffix pad $:= 1||0^{d-((|M|+1) \bmod d)}$.

(2) Parse the string $M||\text{pad}$ into $M_1, M_2, \ldots M_l$, where $|M_i| = d$ for $1 \leq i \leq l$. It must hold that $M_1||M_2||\ldots||M_l = M||\text{pad}$.

Let $\text{Func}(D, R)$ represent the set of all function $\rho : D \to R$ and let $RF_{D,R}$ be a function chosen randomly from the set $\text{Func}(D, R)$ (i.e., $RF_{D,R} \xleftarrow{\$} \text{Func}(D, R)$). We sometimes write $RF_{d,r}$ or $\text{Func}(d, r)$ when $D = \{0,1\}^d$ and $R = \{0,1\}^r$. Similarly, we write $RF_{*,r}$ or $\text{Func}(*, r)$ when $D = \{0,1\}^*$ and $R = \{0,1\}^r$. If i is an integer, then $\langle i \rangle_r$ is the r-bit string representation of i. If r is omitted, then $\langle i \rangle$ is the shortest string representation of i (e.g., if $i = 3$, then $\langle i \rangle = 11$).

Negligible function. A function f is negligible if for every polynomial $p(\cdot)$ there exists N such that for every $n > N$ it holds that $f(n) < \frac{1}{p(n)}$. Negligible functions are denoted as $\text{negl}(\cdot)$.

2.1. Interactive boolean circuits

A boolean circuit (definition is from [11]) is a directed acyclic graph without isolated vertices. There are three types of vertices: sources, sinks and internal vertices.

- Internal vertices are vertices having incoming and outgoing edges (i.e., they have in-degree and out-degree at least 1). Internal vertices are called *gates*. Each gate is labeled by a Boolean operation, typically are considered operations $\wedge, \vee$ and $\neg$.
- Sources are vertices with in-degree 0. In the context of boolean circuits, sources are called input terminals. Each input terminal is labeled by a natural number, which represents index of the circuit's input bit. If the circuit has m input terminals, then they are labeled as $1, 2, \ldots, m$, i.e., we disallow different input terminals to be labeled by the same number.
- Sinks have out-degree 0. Sinks are called output terminals. Each output terminal is labeled with a natural number, which represents index of the circuit's output bit. If the circuit has y output terminals, then they are labeled as $1, 2, \ldots, y$, i.e., we disallow different output terminals to be labeled by the same number.

An interactive boolean circuit (IBC) C is a boolean circuit with special oracle gates. The IBC C can communicate with other IBCs $C_1, \ldots, C_n$ via the oracle gates. Each oracle gate is labeled by a name of an IBC (i.e., if C has access to IBCs $C_1, \ldots, C_n$, then the gates's label is one of $C_1, \ldots, C_n$). If an oracle gate has label C_i, we say that the oracle gate is of type C_i. Each input edge to an oracle gate is labeled with a natural number $1, \ldots, m$, where m is in-degree of the oracle gate. We disallow the same labels for different input edges. An input edge to an oracle gate of type C_i with label j represents jth input bit to the IBC C_i. Output edges of an oracle gate are also labeled with a natural number $1, 2, \ldots, y$. Same labels for different output edges are allowed. An output edge from the gate of type C_i with label j represents jth output bit of the IBC C_i.

Whenever computation reaches an oracle gate G of type C_i, the IBC C_i is invoked on the input of G and output of C_i is passed to the output of G. We call such an operation a query to the oracle C_i. By $C^{C_1,\ldots,C_n}$ we denote that C contains oracle gates of type $C_1, \ldots, C_n$.

Each IBC can implement various interfaces $(t_1, t_2, \ldots)$. An interface specifies what needs to be given on the input to an oracle gate to invoke particular functionality of the IBC. We write $C = (t_1, t_2, \ldots)$ meaning that C implements interfaces $t_1, t_2, \ldots$ (For example, an interface can specify that if the first input bit is 0, then a functionality t_1 is invoked. If the first bit is 1 then a functionality t_2 is invoked.)

We sometimes distinguish between private and public interfaces of an IBC C. In this case we write $C = \big((t_1, t_2, \ldots), (t'_1, t'_2, \ldots)\big)$, where $t_1, t_2, \ldots$ are private interfaces and $t'_1, t'_2, \ldots$ are public. We write $P^{C_{\text{pub}}}$ to denote that an IBC P has oracle access only to public interfaces of an IBC C. Similarly, by $P^{C_{\text{priv}}}$ we denote the fact that P has access only to private interfaces of C.

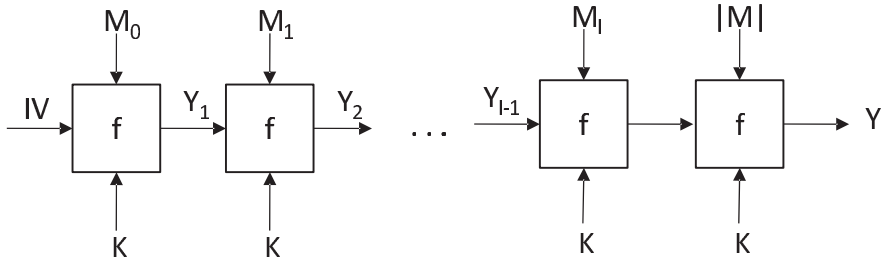

FIGURE 1. Merkle-Damgård domain extension transform.

Evaluation order. Note that since an IBC C can contain oracle gates, the output of C can depend on the order of evaluation of some oracle gates (which are not connected by a path). By C_E we denote an evaluation of C with the specified order E. In most cases we omit to specify the evaluation order of some IBC C. It means that the corresponding discussion holds for any evaluation order of the IBC C.

Domain extension transform. Let $n \in \mathbb{N}$ be a security parameter. A domain extension transform (DET) H is an IBC, the size of which is polynomial in n with oracle access to a function $f : \{0,1\}^{y+d} \to \{0,1\}^y$, where $d, y \in \mathbb{N}$ are polynomially related to the security parameter n (i.e., $d = p_1(n)$ and $y = p_2(n)$ for some polynomials p_1, p_2). The function f is called compression function.

Distinguisher. A distinguisher D is an IBC with one output bit. Besides standard input, D can contain several random input bits, which are initialized with value uniformly chosen from $\{0,1\}$.

2.2. Merkle-Damgård construction

Let $Y_0 = IV$ be some constant initialization vector from the set $\{0,1\}^y$. The strengthened Merkle-Damgård (SMD) domain extension transform operates in the following way (see Figure 1).

Algorithm SMD$^f(M)$
the algorithm has oracle access to $f : \{0,1\}^y \times \{0,1\}^d \to \{0,1\}^y$.
(1) $(M_1, \ldots, M_l) \xleftarrow{d} M$,
(2) $M_{l+1} \leftarrow \langle |M| \rangle_d$,
(3) $Y_0 \leftarrow IV$,
(4) **for** $i = 1$ **to** $l + 1$ **do**,
(5) $Y_i \leftarrow f(Y_{i-1}, m_i)$,
(6) **return** $Y_{l'}$.

By SMD^f we denote the hash function family created by the SMD domain extension transform from the compression function $f\colon \{0,1\}^d \times \{0,1\}^y \to \{0,1\}^y$.

The key security feature of the MD construction is that it preserves collision resistance. If the compression function f is collision resistant, then so is the resulting hash function SMD^f [8], [15].

3. Pseudo-random oracles

3.1. Pseudo-random oracle

Pseudo-random oracle (pro) [2], [3], [7] is a property of domain extension transforms for cryptographic hash functions based on the indifferentiability framework [14]. A hash function $H^f\colon \{0,1\}^* \to \{0,1\}^y$ based on an ideal compression function f (i.e., fixed input length (FIL) random oracle) is pseudo-random oracle if it is indifferentiable from a variable input length (VIL) random oracle.

Let H be a domain extension transform, D be a distinguisher and S a simulator. We define the following pro advantage:

$$\begin{aligned}\mathbf{Adv}^{\mathrm{pro}}_{H,S}(D) := \Big| &\Pr\left[f \leftarrow \mathrm{RF}_{y+d,y}; D^{H^f,f} \to 1\right] \\ &- \Pr\left[F \leftarrow \mathrm{RF}_{*,y}; D^{F,S^F} \to 1\right]\Big|.\end{aligned}$$

We say that the domain extension transform H is pro if there exists a polynomial simulator S such that for any polynomial distinguisher D there is a negligible function negl such that

$$\mathbf{Adv}^{\mathrm{pro}}_{H,S}(D) \leq \mathrm{negl}(n).$$

The pseudo-random oracle property is meaningful only in the random-oracle model. Since H is based on an "uncertain" random compression function f, the pro is rather a property of domain extension transforms. Thus H securely extends the domain of the fixed-input length random oracle f to the variable-input length pseudo-random oracle.

Strong vs. weak indifferentiability. Note that there are two different definitions of pseudo-random oracle. M a u r e r et al. [14] used a different quantifier ordering. Their definition said that for all efficient distinguishers D there exists an efficient simulator S such that $\mathbf{Adv}^{\mathrm{pro}}_{H,S}(D)$ is negligible. We adopt the labeling from [16] where they refer to the notion from [14] as weak indifferentiability and from [7] (which is used in this paper) as strong. It is clear that strong indifferentiability implies weak. In this paper, we restrict our discussion to the strong version of the pseudo-random oracle.

3.2. Public use random oracle

Many applications compute hashes only from public messages, i.e., messages that are available to any party and thus any party is able to compute hash of the messages. The security of these applications is not affected when all messages

are revealed to adversaries. Important applications of such public use of hash functions are digital signature schemes or even some encryption schemes.

Public use pseudo-random oracle security notion [9] captures such applications. The public use random oracle is an ideal primitive with two interfaces. The first interface is available to all parties and performs the usual evaluation of the random oracle—given a message M it outputs its image Y. The second interface is available only to adversaries (and simulators) and when queried it reveals all evaluated messages and their corresponding images made so far to the first interface.

Let $\rho = RF_{*,y}$. Formally, the public use random oracle is an ideal cryptosystem $F = \big((F_{\text{eval}}), (F_{\text{eval}}, F_{\text{reveal}})\big)$, where F_{eval} implements the random function ρ and F_{reveal} reveals all queries (messages with their corresponding images) asked to the public interface F_{eval} (see Figure 2). If ρ is a FIL random function, we say F is a FIL public use random oracle. FIL public use ROs are denoted in lower case, i.e., $f := \big((f_{\text{eval}}), (f_{\text{eval}}, f_{\text{reveal}})\big)$. Let $RF^{\text{pub}}_{m,y}$ denote the public use random oracle implementing a random function $\rho : \{0,1\}^m \to \{0,1\}^y$.

$F_{\text{eval}}(M)$	$F_{\text{reveal}}()$
$\mathcal{Q} \leftarrow Q \cup \big(M, \rho(M)\big)$ **Ret** $\rho(M)$	**Ret** $\mathcal{Q}$

FIGURE 2. Public use random oracle interfaces.

D o d i s et al. [9] defined a so called public use pseudo-random oracle property. A hash function $H^f : \{0,1\}^* \to \{0,1\}^y$ based on a FIL public use RO f is public use pseudo-random oracle (pub-pro) if it is indifferentiable from a VIL public use random oracle.

Let H be a domain extension transform, D be an adversary and S a simulator implementing two interfaces S_{eval} and S_{reveal}. We define the following pub-pro advantage

$$\mathbf{Adv}^{\text{pub}-\text{pro}}_{H,S}(D) := \Big| \Pr\Big[f \leftarrow \text{RF}^{\text{pub}}_{y+d,y}; \mathcal{D}^{H^f_{\text{eval}}, f} \to 1\Big] - \Pr\Big[F \leftarrow \text{RF}^{\text{pub}}_{*,y}; D^{F_{\text{eval}}, S^F} \to 1\Big]\Big|.$$

We say that the domain extension transform H is pub-pro if there exists a polynomial simulator S such that for any polynomial distinguisher D there is a negligible function negl such that

$$\mathbf{Adv}^{\text{pub}-\text{pro}}_{H,S}(D) \leq \text{negl}(n).$$

3.3. Public image pseudo-random oracle

The property pub-pro enables simulators to see complete queries (queries and corresponding answers) asked by a distinguisher D to it's first oracle. Due to this fact, the property pub-pro is weaker than pro [9]. When proving indifferentiability of two cryptosystems, to simplify the proof we would like to consider as strongest simulators as possible. Thus, an interesting question is what happens if we allow simulators to see only responses of the D's first oracle.

A public image random oracle is a cryptosystem $F = \big((F_{\text{eval}}), (F_{\text{eval}}, F_{\text{ireveal}})\big)$, where F_{eval} implements a random function ρ and F_{ireveal} reveals answers to all queries asked to the public interface F_{eval}. Let $RF^{\text{img}}_{m,y}$ denote the public image random oracle implementing a random function $\rho\colon \{0,1\}^m \to \{0,1\}^y$.

Definition 1 (img-pro)**.** Let H be a domain extension transform, D be a distinguisher and S a simulator implementing only one interface. We define the following img-pro advantage

$$\mathbf{Adv}^{\text{img}-\text{pro}}_{H,S}(D) := \Big| \Pr\Big[f \leftarrow \text{RF}_{y+d,y}; D^{H^f,f} \to 1\Big] - \Pr\Big[F \leftarrow \text{RF}^{\text{img}}_{*,y}; D^{F_{\text{eval}},S^F} \to 1\Big]\Big|.$$

We say that the domain extension transform H is img-pro, if there exists a polynomial simulator S such that for all polynomial distinguishers D there is a negligible function negl such that

$$\mathbf{Adv}^{\text{img}-\text{pro}}_{H,S}(D) \leq \text{negl}(n).$$

Remark 1. Note that it has no meaning to define img-pro property with f being a FIL public image random oracle, i.e., $f = \big((f_{\text{eval}}), (f_{\text{eval}}, f_{\text{ireveal}})\big)$. Since a simulator (which would implement interfaces $\big((S_{\text{eval}}), (S_{\text{eval}}, S_{\text{ireveal}})\big)$) cannot see queries asked by D to it's first oracle (only their responses), the simulator is unable to implement S_{ireveal} consistently.

In Section 4 we show that img-pro is equivalent to pro, i.e., ability to see the list of images, which F_{eval} responds, does not make simulators stronger.

3.4. Real vs. random world

The definitions of pro, pub-pro and img-pro are comparing two scenarios. In the first scenario, "real world", the distinguisher has access to a "real" hash function constructed via a domain extension transform from an ideal compression function and to the compression function. In the second scenario, "random world", the distinguisher has access to a (public use/image) random oracle and to a simulator. To shorten our presentation we often write

$$\Pr_{H,f,D}\Big[D^{H^f,f} \to 1\Big],$$

what means that the probability goes over random selection of the FIL random oracle f and random coins of D. Similarly, we write

$$\Pr_{F,S,D}\left[D^{F,S^F} \to 1\right],$$

what means that the probability goes over random selection of the VIL random oracle F and random coins of D and S.

The pro, pub-pro and img-pro distinguisher D can contain two types of oracle gates. We refer to the gates corresponding to the D's first oracle (H^f in the real world, F in the random world) as to F-gates. Similarly, by f-gates we denote gates corresponding to the D's second oracle (f in the real world and S^F in the random world).

To denote a particular F-gate we use upper case letters, i.e., G. When considering an f-gate we use lower case, i.e., g.

To differentiate between different types of simulators we use the following notation. A pro simulator is a simulator with access to an oracle, which implements a VIL random oracle. Similarly, a pub-pro simulator has oracle access to VIL public use random oracle and img-pro simulator's oracle is VIL public image random oracle.

4. Pro is equivalent to img-pro

It is easy to see that if a domain extension transform H is pro, then it is img-pro. If there exists a pro simulator S_{pro} such that for all pro distinguishers D is

$$\mathbf{Adv}^{\text{pro}}_{H,S_{\text{pro}}}(D) \geq \varepsilon(n),$$

then there exists an img-pro simulator S_{ipro} such that for all img-pro distinguishers D' is

$$\mathbf{Adv}^{\text{img-pro}}_{H,S_{\text{ipro}}}(D') \geq \varepsilon(n).$$

The simulator S_{ipro} does the same as S_{pro} (it's possible, since S_{ipro} has access to all oracles to which S_{pro} has access). Hence, we can state the following theorem.

Theorem 1. *Let H be a domain extension transform which is pro. Then H is img-pro.*

The opposite direction is more involved, because of the following problem. Let S_{ipro} be an img-pro simulator and D be a distinguisher for S_{ipro}. Since S_{ipro} can see answers of queries asked by D to it's first oracle, D can pass some information to S_{ipro}, which no pro simulator S_{pro} can see. For example, consider the extension attack, which proves that the Merkle-Damgård domain extension transform [8], [15] is not pro. Let D be the following distinguisher (see Figure 3(a))

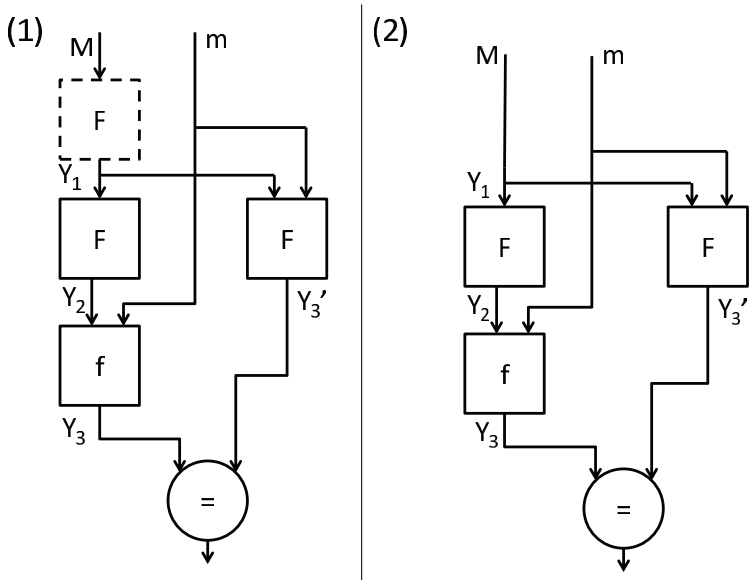

FIGURE 3. Pro distinguishers for the Merkle-Damgård domain extension transform. The first distinguisher (left) reveals input to the other F-gates for some img-pro simulator via the dashed F-gate. On the other hand, the second distinguisher (right) has non-negligible advantage against the Merkle-Damgård domain extension transform in the img-pro sense.

Distinguisher $D^{F,f}$

(1) Choose $M \xleftarrow{\$} \{0,1\}^y$.
(2) Compute $Y_1 \leftarrow F(M)$.
(3) Compute $Y_2 \rightarrow F(Y_1)$.
(4) Choose $m \xleftarrow{\$} \{0,1\}^d$, compute $Y_3 \leftarrow f(Y_2, m)$.
(5) Compute $Y_3' \leftarrow F(Y_1||m)$.
(6) Output 1 if $Y_3 = Y_3'$ and 0 otherwise.

Let H be the Merkle-Damgård domain extension transform. For all pro simulators S_{pro}, the probability $\Pr_{F,S,D}[D^{F,S^F} \rightarrow 1]$ is negligible, since S_{pro} is unable to guess Y_1. On the other hand $\Pr_{H,f,D}[D^{H^f,f} \rightarrow 1] = 1$. However, an img-pro simulator can see Y_1, hence, it can query $Y_1||m$. Note that a distinguisher D with step (2) replaced by $Y_1 := M$ would fool also img-pro simulators (see Figure 3(b)), since no img-pro simulator is able to guess the message M.

In the rest of this section we prove that if a domain extension transform is img-pro, then it is also pro.

To simplify our proof, we restrict the set of domain extension transforms. We focus only on those transforms, whose output is equal to the output of one of the containing oracle gates. This restriction avoids problems with "partially instantiated" transforms [10], which instantiate several oracle gates with some real compression function.

Definition 2 (Standard DET)**.** Let H be a domain extension transform. We say H is standard if value of $H^f(M)$ is equal to the output of one of the containing f-gates g. We call such a gate g final.

Note that most of the popular domain extension transforms (e.g., MD [8], [15], HMAC [1], EMD [2], [3]) are standard.

Definition 3 (Oracle-oracle output bit)**.** Let D be a distinguisher in pro sense. Let G be an oracle gate in D. We say that ith output bit of the gate G is oracle-oracle output bit if there exists a path in D starting at the ith output bit of the gate G and ending at some oracle gate G'.

In the following definition we formally define the term "minimal" distinguisher (in the pro sense). The minimal distinguisher represents the minimal algorithm (i.e., an algorithm without any "unnecessary" computation), which is able to distinguish between real and random worlds. The minimal distinguisher is defined for a domain extension transform H and a pro simulator S.

Definition 4 (Minimal distinguisher)**.** Let H be a domain extension transform and let S be a pro simulator for H. Let $\mathcal{D}_1$ denote the set of all polynomial distinguishers with non-negligible advantage in pro sense against S (i.e., for all $D \in \mathcal{D}_1$ holds that $\mathbf{Adv}^{\mathrm{pro}}_{H,S}(D)$ is non-negligible). Let $\mathcal{D}_2 \subseteq \mathcal{D}_1$ be the set which contains distinguishers with minimal number of F-gates. Let $\mathcal{D}_3 \subseteq \mathcal{D}_2$ be the set containing only distinguishers with minimal number of f-gates. Let $\mathcal{D}_4 \subseteq \mathcal{D}_3$ be the set containing only distinguishers with minimal number of oracle-oracle output bits. Finally, let $\mathcal{D}_5 \subseteq \mathcal{D}_4$ be the set containing distinguishers of minimal average depth (average over length of all input-output paths).
We say that a distinguisher D is minimal against S and H, if $D \in \mathcal{D}_5$.

Note that a minimal distinguisher against some simulator S and a domain extension transform H has non-negligible advantage in pro sense.

In the following two simple lemmas we show that a minimal distinguisher D cannot contain gates which have the same input only in the real world. We show that if two gates have the same input in the real world then almost certainly they have the same input in the random world also. Moreover, in the Lemma 2 we show that a minimal distinguisher gains non-negligible advantage only when all gates have different input.

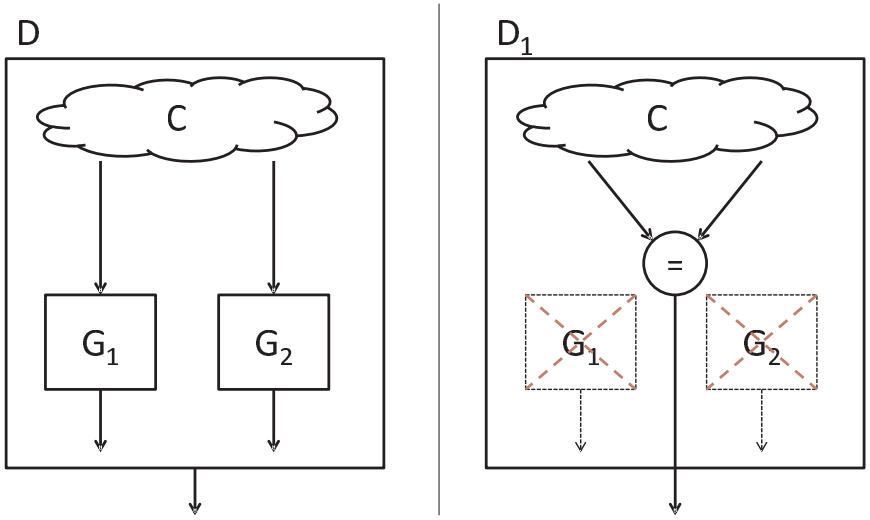

FIGURE 4. Construction of the distinguisher D_1 (right) from the distinguisher D (left). The distinguisher D_1 compares inputs to the gates G_1 and G_2 and outputs 1 if and only if they are equal. The gates G_1 and G_2 are removed.

Let G_1 and G_2 be two oracle gates in a distinguisher D. By $\mathrm{EqI}(G_1, G_2)$ we denote the event that during evaluation of the distinguisher D the gates G_1 and G_2 have the same input. Similarly, by $\mathrm{EqO}(G_1, G_2)$ we denote the event that the gates G_1 and G_2 have the same output.

LEMMA 1. *Let H be a domain extension transform, S be a pro simulator for H and D be a minimal distinguisher against S. Let G_1 and G_2 be two different oracle gates in D of the same type. Then there exist negligible functions* negl_1 *and* negl_2 *such that*

$$\left| \Pr_{H,f,D}\left[EqI(G_1, G_2)\right] - \Pr_{F,S,D}\left[EqI(G_1, G_2)\right] \right| \leq \mathrm{negl}_1(n),$$

$$\left| \Pr_{H,f,D}\left[EqO(G_1, G_2)\right] - \Pr_{F,S,D}\left[EqO(G_1, G_2)\right] \right| \leq \mathrm{negl}_2(n).$$

P r o o f. Consider a distinguisher D_1 (Figure 4) which does the same as D until its computation reaches the gates G_1 and G_2. Then it compares inputs to G_1 and G_2 and outputs 1 if and only if they are equal. Clearly, D_1 is smaller than D. Since D is minimal, the advantage of D_1 must be negligible. Hence, there exists

a negligible function negl_1 such that

$$\left| \Pr_{H,f,D_1}\left[{D_1}^{H^f,f} \to 1\right] - \Pr_{F,S,D_1}\left[{D_1}^{F,S^F} \to 1\right] \right| \leq \mathrm{negl}_1(n).$$

However, $\mathrm{EqI}(G_1, G_2)$ is true if and only if D_1 returns 1. Hence,

$$\left| \Pr_{H,f,D}\big[\mathrm{EqI}(G_1, G_2)\big] - \Pr_{F,S,D}\big[\mathrm{EqI}(G_1, G_2)\big] \right| \leq \mathrm{negl}_1(n). \tag{1}$$

This completes the first part of the proof.

If the gates G_1 and G_2 are F-gates, then in the random world the gates represent an uniform VIL random function F. Thus there exists a negligible function negl_2 such that

$$\left| \Pr_{F,S,D}\big[\mathrm{EqI}(G_1, G_2)\big] - \Pr_{F,S,D}\big[\mathrm{EqO}(G_1, G_2)\big] \right| \leq \mathrm{negl}_2(n). \tag{2}$$

Hence, by equations (1) and (2) we have

$$\left| \Pr_{H,f,D}\big[\mathrm{EqO}(G_1, G_2)\big] - \Pr_{F,S,D}\big[\mathrm{EqO}(G_1, G_2)\big] \right| \leq \mathrm{negl}_2(n) + \mathrm{negl}_1(n).$$

If the gates G_1 and G_2 are f-gates, then in the real world they represent FIL random function f. Hence, there exits a negligible function negl_3 such that

$$\left| \Pr_{H,f,D}\big[\mathrm{EqI}(G_1, G_2)\big] - \Pr_{H,f,D}\big[\mathrm{EqO}(G_1, G_2)\big] \right| \leq \mathrm{negl}_3(n). \tag{3}$$

By combining equations (1) and (3) we have

$$\left| \Pr_{H,f,D}\big[\mathrm{EqO}(G_1, G_2)\big] - \Pr_{F,S,D}\big[\mathrm{EqO}(G_1, G_2)\big] \right| \leq \mathrm{negl}_3(n) + \mathrm{negl}_1(n).$$

This completes the second part of the proof. □

Lemma 2. *Let H be a domain extension transform, S be a pro simulator for H and D be a minimal distinguisher against S. Then for all oracle gates G_1 and G_2 of the same type in the distinguisher D there exists a negligible function* negl_1 *such that*

$$\left| \Pr_{H,f,D}\left[D^{H^f,f} \to 1 \wedge \mathit{EqI}(G_1, G_2)\right] - \Pr_{F,S,D}\left[D^{F,S^F} \to 1 \wedge \mathit{EqI}(G_1, G_2)\right] \right| \leq \mathrm{negl}_1(n).$$

Moreover, there exists a negligible function negl_2 *such that*

$$\left| \Pr_{H,f,D}\left[D^{H^f,f} \to 1 \wedge \mathit{EqO}(G_1, G_2)\right] - \Pr_{F,S,D}\left[D^{F,S^F} \to 1 \wedge \mathit{EqO}(G_1, G_2)\right] \right| \leq \mathrm{negl}_2(n).$$

Proof. Let G_1 and G_2 be two oracle gates in D. Without loss of generality we can assume that there does not exist a path from G_1 to G_2 (since a path between G_1 and G_2 can exist only in one direction). Consider a distinguisher D_2 (Figure 5) which is the same as D but all edges starting at the gate G_1 are redirected such that they start at the gate G_2 and the gate G_1 is removed. Moreover, the distinguisher D_2 compares input to the removed gate G_1 with

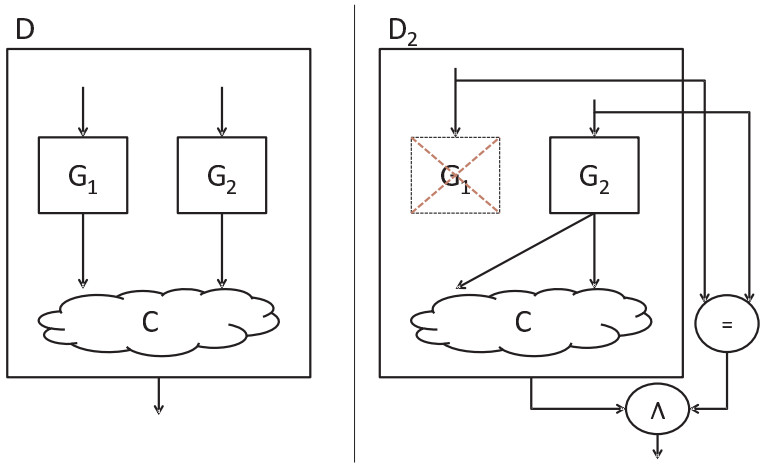

FIGURE 5. Construction of the distinguisher D_2 (right) from the distinguisher D (left). All edges starting at the gate G_1 are redirected such that they start at the gate G_2. Inputs to the gates G_1 and G_2 are compared and D_2 outputs 1 only if they are equal. The gate G_1 is removed.

the input to the gate G_2 and outputs 1 only if the inputs are equal. It is clear that D_2 outputs the same as D if the gates G_1 and G_2 have the same output. On the other hand, if the gates G_1 and G_2 have different input, then D_2 always outputs 0. Hence,

$$\Pr_{H,f,D_2}\left[{D_2}^{H^f,f} \to 1\right] = \Pr_{H,f,D}\left[D^{H^f,f} \to 1 \wedge \mathrm{EqI}(G_1,G_2)\right],$$
$$\Pr_{F,S,D_2}\left[{D_2}^{F,S^F} \to 1\right] = \Pr_{F,S,D}\left[D^{F,S^F} \to 1 \wedge \mathrm{EqI}(G_1,G_2)\right].$$

Since D_2 is smaller than D, from the assumption that D is minimal there exists a negligible function negl_1 such that

$$\begin{aligned}\mathrm{negl}_1(n) &\geq \mathbf{Adv}^{\mathrm{pro}}_{S,H}(D_2)\\ &= \left|\Pr_{H,f,D_2}\left[{D_2}^{H^f,f} \to 1\right] - \Pr_{F,S,D_2}\left[{D_2}^{F,S^F} \to 1\right]\right|\\ &= \Big|\Pr_{H,f,D}\left[D^{H^f,f} \to 1 \wedge \mathrm{EqI}(G_1,G_2)\right]\\ &\quad - \Pr_{F,S,D}\left[D^{F,S^F} \to 1 \wedge \mathrm{EqI}(G_1,G_2)\right]\Big|.\end{aligned}$$

This completes the first part of the proof. The second part of the proof comes from the fact that if the gates G_1 and G_2 are of the same type and they have the same output, then except the negligible probability they have the same input

also (see also discussion in the previous lemma). Hence, there exists a negligible function negl_2 such that

$$\Big| \Pr_{H,f,D}\Big[D^{H^f,f}\to 1 \wedge \mathrm{EqI}(G_1,G_2)\Big] - \Pr_{H,f,D}\Big[D^{H^f,f}\to 1 \wedge \mathrm{EqO}(G_1,G_2)\Big]\Big| \le \mathrm{negl}_2(n)$$

and

$$\Big| \Pr_{F,S,D}\Big[D^{F,S^F}\to 1 \wedge \mathrm{EqI}(G_1,G_2)\Big] - \Pr_{F,S,D}\Big[D^{F,S^F}\to 1 \wedge \mathrm{EqO}(G_1,G_2)\Big]\Big| \le \mathrm{negl}_2(n).$$

Thus,

$$\Big| \Pr_{H,f,D}\Big[D^{H^f,f}\to 1 \wedge \mathrm{EqO}(G_1,G_2)\Big] - \Pr_{F,S,D}\Big[D^{F,S^F}\to 1 \wedge \mathrm{EqO}(G_1,G_2)\Big]\Big| \le \mathrm{negl}_1(n) + \mathrm{negl}_2(n).$$

□

A gate G in a distinguisher D is end gate if all paths from G to the output bit do not contain oracle gates. Let $L(D) = \{G_1, \ldots, G_l\}$ denote the set of all end gates in D. In the following lemmas we prove an intuitive fact, that if a domain extension transform H is pub-pro then the only chance how a distinguisher D can distinguish between real and random worlds is to compute the hash of some message M in two different ways. Moreover, a simulator must be unable to obtain the message M.

We start by two lemmas, which state that there is at least one f-gate and one F-gate in the set of end gates in a minimal distinguisher D.

Lemma 3. *Let H be a standard domain extension transform which is not pro and S be a simulator which output is indistinguishable from a random function. Let D be a minimal distinguisher against S and H. Then there exists an F-gate $G \in L(D)$.*

Proof. Assume the contrary that all gates in $L(D)$ are f-gates. Let DI denote the event that during evaluation of D all gates in $L(D)$ have distinct input.

If DI does not hold for some evaluation of D, then there exist at least two gates with the same input—we can apply the construction of a smaller distinguisher D_2 from Lemma 2. Hence, there exists a negligible function negl_1 such that

$$\Big| \Pr_{H,f,D}\Big[D^{H^f,f}\to 1 \wedge \neg\mathrm{DI}\Big] - \Pr_{F,S,D}\Big[D^{F,S^F}\to 1 \wedge \neg\mathrm{DI}\Big]\Big| \le \mathrm{negl}_1(n).$$

Similarly, we can apply the construction D_1 from Lemma 1. Thus there exists a negligible function negl_2 such that

$$\left| \Pr_{H,f,D}[\mathrm{DI}] - \Pr_{F,S,D}[\mathrm{DI}] \right| = \left| \Pr_{H,f,D}[\neg\mathrm{DI}] - \Pr_{F,S,D}[\neg\mathrm{DI}] \right| \leq \mathrm{negl}_2(n).$$

Let $\alpha := \Pr_{H,f,D}[\neg\mathrm{DI}]$, then

$$\alpha - \mathrm{negl}_2(n) \leq \Pr_{F,S,D}[\neg\mathrm{DI}] \leq \alpha + \mathrm{negl}_2(n).$$

Now consider D's advantage against the simulator S and the domain extension transform H

$$\begin{aligned}
&\mathbf{Adv}^{\mathrm{pro}}_{S,H}(D) \\
&= \Big| \Pr_{H,f,D}[D^{H^f,f} \to 1|\mathrm{DI}] \cdot \Pr_{H,f,D}[\mathrm{DI}] - \Pr_{F,S,D}\left[D^{F,S^F} \to 1|\mathrm{DI}\right] \cdot \Pr_{F,S,D}[\mathrm{DI}] \\
&\quad + \Pr_{H,f,D}[D^{H^f,f} \to 1 \wedge \neg\mathrm{DI}] - \Pr_{F,S,D}\left[D^{F,S^F} \to 1 \wedge \neg\mathrm{DI}\right] \Big| \\
&\leq \left| (1-\alpha) \Pr_{H,f,D}\left[D^{H^f,f} \to 1|\mathrm{DI}\right] - (1-\alpha) \cdot \Pr_{F,S,D}\left[D^{F,S^F} \to 1|\mathrm{DI}\right] \right| \\
&\quad + \mathrm{negl}_1(n) + \mathrm{negl}_2(n) \\
&\leq (1-\alpha) \cdot \left| \Pr_{H,f,D}\left[D^{H^f,f} \to 1|\mathrm{DI}\right] - \Pr_{F,S,D}\left[D^{F,S^F} \to 1|\mathrm{DI}\right] \right| \\
&\quad + \mathrm{negl}_1(n) + \mathrm{negl}_2(n).
\end{aligned}$$

If all gates in $L(D)$ have different input, the distribution of output of gates in $L(D)$ is the same in both worlds. In the real world, all gates in $L(D)$ correspond to the FIL random function f. In the random world, gates from $L(D)$ corresponds to the simulator, which output is indistinguishable from a random function. Hence, there exists a negligible function negl_3 such that

$$\left| \Pr_{H,f,D}\left[D^{H^f,f} \to 1|\mathrm{DI}\right] - \Pr_{F,S,D}\left[D^{F,S^F} \to 1|\mathrm{DI}\right] \right| \leq \mathrm{negl}_3(n).$$

Thus,

$$\mathbf{Adv}^{\mathrm{pro}}_{S,H}(D) \leq \mathrm{negl}_1(n) + \mathrm{negl}_2(n) + \mathrm{negl}_3(n).$$

This contradicts the minimality of D. □

Lemma 4. *Let H be a standard domain extension transform which is pub-pro, S be a pro simulator for H and D be a minimal distinguisher against S. Then there exists an f-gate $g \in L(D)$.*

Proof. Assume the contrary that all gates in $L(D)$ are F-gates. Let DI denote the event that during some evaluation of D all gates in $L(D)$ have different input.

By Lemmas 1 and 2 there exist negligible functions negl_1 and negl_2 such that (see also proof of the previous lemma for more detailed discussion)

$$\left| \Pr_{H,f,D}[\mathrm{DI}] - \Pr_{F,S,D}[\mathrm{DI}] \right| = \left| \Pr_{H,f,D}[\neg\mathrm{DI}] - \Pr_{F,S,D}[\neg\mathrm{DI}] \right| \leq \mathrm{negl}_1(n), \tag{4}$$

$$\left| \Pr_{H,f,D}\left[D^{H^f,f} \to 1 \wedge \neg\mathrm{DI}\right] - \Pr_{F,S,D}\left[D^{F,S^F} \to 1 \wedge \neg\mathrm{DI}\right] \right| \leq \mathrm{negl}_2(n). \tag{5}$$

Let $\alpha := \Pr_{H,f,D}[\neg\mathrm{DI}]$, then

$$\alpha - \mathrm{negl}_1(n) \leq \Pr_{F,S,D}[\neg\mathrm{DI}] \leq \alpha + \mathrm{negl}_1(n).$$

Hence,

$$\begin{aligned}
&\mathbf{Adv}^{\mathrm{pro}}_{S,H}(D) \\
&= \left| \Pr_{H,f,D}\left[D^{H^f,f} \to 1|\mathrm{DI}\right] \cdot \Pr_{H,f,D}[\mathrm{DI}] - \Pr_{F,S,D}\left[D^{F,S^F} \to 1|\mathrm{DI}\right] \cdot \Pr_{F,S,D}[\mathrm{DI}] \right. \\
&\quad \left. + \Pr_{H,f,D}\left[D^{H^f,f} \to 1 \wedge \neg\mathrm{DI}\right] - \Pr_{F,S,D}\left[D^{F,S^F} \to 1 \wedge \neg\mathrm{DI}\right] \right| \\
&\leq \left| (1-\alpha) \Pr_{H,f,D}\left[D^{H^f,f} \to 1|\mathrm{DI}\right] - (1-\alpha) \cdot \Pr_{F,S,D}\left[D^{F,S^F} \to 1|\mathrm{DI}\right] \right| \\
&\quad + \mathrm{negl}_2(n) + \mathrm{negl}_1(n) \\
&\leq \left| \Pr_{H,f,D}\left[D^{H^f,f} \to 1|\mathrm{DI}\right] - \Pr_{F,S,D}\left[D^{F,S^F} \to 1|\mathrm{DI}\right] \right| \\
&\quad + \mathrm{negl}_2(n) + \mathrm{negl}_1(n).
\end{aligned} \tag{6}$$

Note that for the domain extension transform H, the simulator S, and the minimal distinguisher D, the probabilities

$$\Pr_{F,S,D}\left[D^{F,S^F} \to 1|\mathrm{DI}\right] \quad \text{and} \quad \Pr_{H,f,D}\left[D^{H^f,f} \to 1|\mathrm{DI}\right]$$

are fixed. Hence, there exists the following distinguisher D' in pub-pro sense:

Distinguisher D'^{O_1,o_2}

(1) Simulate $D^{O_1,o_2} \to b$.
(2) Output b if all gates in $L(D)$ have different input during the simulation of D in the first step. Otherwise
   - if $\Pr_{F,S,D}\left[D^{F,S^F} \to 1|\mathrm{DI}\right] > \Pr_{H,f,D}\left[D^{H^f,f} \to 1|\mathrm{DI}\right]$ output 0,
   - otherwise output 1.

Let S' be some pub-pro simulator and consider the advantage of D' against the simulator S'. To shorten our presentation, let

$$\beta := \Pr_{H,f,D'}\left[\neg\mathrm{DI}\right] \quad \text{and} \quad \gamma := \Pr_{F,S',D'}\left[\neg\mathrm{DI}\right].$$

We have

$$\begin{aligned}
&\mathbf{Adv}^{\mathrm{pub-pro}}_{S',H}(D') \\
&= \Big| \Pr_{H,f,D'}\Big[{D'}^{H^{f},f} \to 1\Big] - \Pr_{F,S',D'}\Big[{D'}^{F_{\mathrm{eval}},{S'}^{F}} \to 1\Big]\Big| \\
&= \Big|(1-\beta)\cdot \Pr_{H,f,D'}\Big[{D'}^{H^{f},f} \to 1|\mathrm{DI}\Big] - (1-\gamma)\cdot \Pr_{F,S',D'}\Big[{D'}^{F_{\mathrm{eval}},{S'}^{F}} \to 1|\mathrm{DI}\Big] \\
&\quad + \beta\cdot \Pr_{H,f,D'}\Big[{D'}^{H^{f},f} \to 1|\neg\mathrm{DI}\Big] - \gamma\cdot \Pr_{F,S',D'}\Big[{D'}^{F_{\mathrm{eval}},{S'}^{F}} \to 1|\neg\mathrm{DI}\Big]\Big| \\
&= \Bigg| \Pr_{H,f,D'}\Big[{D'}^{H^{f},f} \to 1|DI\Big] - \Pr_{F,S',D'}\Big[{D'}^{F_{\mathrm{eval}},{S'}^{F}} \to 1|\mathrm{DI}\Big] \\
&\quad + \beta\cdot\left(\Pr_{H,f,D'}\Big[{D'}^{H^{f},f} \to 1|\neg\mathrm{DI}\Big] - \Pr_{H,f,D'}\Big[{D'}^{H^{f},f} \to 1|\mathrm{DI}\Big]\right) \\
&\quad + \gamma\cdot\left(\Pr_{F,S',D'}\Big[{D'}^{F,{S'}^{F}} \to 1|\neg\mathrm{DI}\Big] - \Pr_{F,S',D'}\Big[{D'}^{F_{\mathrm{eval}},{S'}^{F}} \to 1|\mathrm{DI}\Big]\right)\Bigg|
\end{aligned}$$

From the definition of the distinguisher D' we have that D and D' outputs the same if gates in $L(D)$ have different input. Moreover, if all gates in $L(D)$ in evaluation of ${D'}^{F_{\mathrm{eval}},{S'}^{F}}$ have different input, then the distribution of their output is the same as in evaluation of D^{F,S^F}. Hence,

$$\begin{aligned}
&\mathbf{Adv}^{\mathrm{pub-pro}}_{S',H}(D') \\
&\leq \Bigg| \Pr_{H,f,D}\Big[D^{H^{f},f} \to 1|DI\Big] - \Pr_{F,S,D}\Big[D^{F,S^F} \to 1|\mathrm{DI}\Big] \\
&\quad + \beta\cdot\left(\Pr_{H,f,D'}\Big[{D'}^{H^{f},f} \to 1|\neg\mathrm{DI}\Big] - \Pr_{H,f,D}\Big[D^{H^{f},f} \to 1|\mathrm{DI}\Big]\right) \\
&\quad + \gamma\cdot\left(\Pr_{F,S',D'}\Big[{D'}^{F_{\mathrm{eval}},{S'}^{F}} \to 1|\neg\mathrm{DI}\Big] - \Pr_{F,S,D}\Big[D^{F,S^F} \to 1|\mathrm{DI}\Big]\right)\Bigg|
\end{aligned}$$

Note that from the definition of the distinguisher D' we have that the probabilities $\Pr_{H,f,D'}\big[{D'}^{H^{f},f} \to 1|\neg\mathrm{DI}\big]$ and $\Pr_{F,S',D'}\big[{D'}^{F_{\mathrm{eval}},{S'}^{F}} \to 1|\neg\mathrm{DI}\big]$ are either 1 or 0.

- If $\Pr_{F,S,D}\big[D^{F,S^F} \to 1|\mathrm{DI}\big] > \Pr_{H,f,D}\big[D^{H^{f},f} \to 1|\mathrm{DI}\big]$, their value is 0. Thus we have

$$\begin{aligned}
&\mathbf{Adv}^{\mathrm{pub-pro}}_{S',H}(D') \\
&= \Bigg|\left(\Pr_{H,f,D}\Big[D^{H^{f},f} \to 1|DI\Big] - \Pr_{F,S,D}[D^{F,S^F} \to 1|\mathrm{DI}]\right) \\
&\quad + \beta\cdot\left(0 - \Pr_{H,f,D}[D^{H^{f},f} \to 1|\mathrm{DI}]\right) + \gamma\cdot\left(0 - \Pr_{F,S,D}\Big[D^{F,S^F} \to 1|\mathrm{DI}\Big]\right)\Bigg|.
\end{aligned}$$

All three expressions in brackets in the equation above are negative. Hence,

$$\mathbf{Adv}^{\text{pub}-\text{pro}}_{S',H}(D') \geq \Big| \Pr_{H,f,D}\Big[D^{H^f,f} \to 1|\text{DI}\Big] - \Pr_{F,S,D}\Big[D^{F,S^F} \to 1|\text{DI}\Big]\Big|. \quad (7)$$

- If $\Pr_{F,S,D}\big[D^{F,S^F} \to 1\big] \leq \Pr_{H,f,D}\big[D^{H^f,f} \to 1\big]$, we have

$$\begin{aligned}
&\mathbf{Adv}^{\text{pub-pro}}_{S',H}(D') \\
&= \Bigg| \left(\Pr_{H,f,D}[D^{H^f,f} \to 1|DI] - \Pr_{F,S,D}[D^{F,S^F} \to 1|\text{DI}] \right) \\
&\quad + \beta \cdot \left(1 - \Pr_{H,f,D}\Big[D^{H^f,f} \to 1|\text{DI}\Big] \right) + \gamma \cdot \left(1 - \Pr_{F,S,D}\Big[D^{F,S^F} \to 1|\text{DI}\Big] \right) \Bigg|.
\end{aligned}$$

In this case, all three expressions in brackets are non-negative, thus

$$\mathbf{Adv}^{\text{pub-pro}}_{S',H}(D') \geq \Big| \Pr_{H,f,D}\Big[D^{H^f,f} \to 1|\text{DI}\Big] - \Pr_{F,S,D}\Big[D^{F,S^F} \to 1|\text{DI}\Big]\Big|. \quad (8)$$

By combining inequalities (6), (7) and (8) we have

$$\mathbf{Adv}^{\text{pub}-\text{pro}}_{S',H}(D') + \text{negl}_2(n) + \text{negl}_1(n) \geq \mathbf{Adv}^{\text{pro}}_{S,H}(D).$$

This contradicts the assumption that H is pub-pro. □

The following key lemma states that the set $L(D)$ of end gates of any minimal distinguisher for some "reasonable" simulator must contain exactly one f-gate and one F-gate. Moreover, the simulator is unable to query a message on input to the F-gate.

Let S be some pro simulator, D be a distinguisher and let G_1 and G_2 be two oracle gates (of arbitrary type) in the distinguisher D. By $\text{EqO}(G_1, G_2)$ we denote an event that the gates G_1 and G_2 have the same output in some computation of D.

Fix some evaluation order of D. Let G be an F-gate in D and g be an f-gate. Let $g_1, \ldots, g_l$ be f-gates evaluated before the gate g. By $\text{Que}_G(g)$ we denote the event that during some computation of D in the random world, the simulator S during evaluation of gates $g_1, \ldots, g_l, g$ asks it is oracle the same query as is the input to the gate G.

Lemma 5. *Let H be a standard domain extension transform which is pub-pro, S be a pro simulator for H given by lemma 3 and D be a minimal distinguisher against S. Then there exists an F-gate $G \in L(D)$, an f-gate $g \in L(D)$ and a negligible function* negl *such that*

$$\mathbf{Adv}^{\text{pro}}_{H,S}(D) \leq \Big| \Pr_{H,f,D}\big[EqO(G,g)\big] - \Pr_{F,S,D}\big[\text{Que}_G(g)\big] \Big| + \text{negl}(n).$$

Moreover, $L(D) = \{G, g\}$.

P r o o f. By Lemmas 3 and 4 there exist at least one F-gate $G \in L(D)$ and at least one f-gate $g \in L(D)$. Consider the gate g.

Since D is minimal, there must exist a negligible function negl_1 such that for all D's F-gates G' different from G holds

$$\left| \Pr_{H,f,D}\left[\mathrm{EqO}(G', g)\right] - \Pr_{F,S,D}\left[\mathrm{EqO}(G', g)\right] \right| \leq \mathrm{negl}_1(n). \tag{9}$$

Otherwise we could construct a smaller distinguisher D_3 (see Figure 6) which would check the equality of outputs of G' and g (D_3 would be without the gate G).

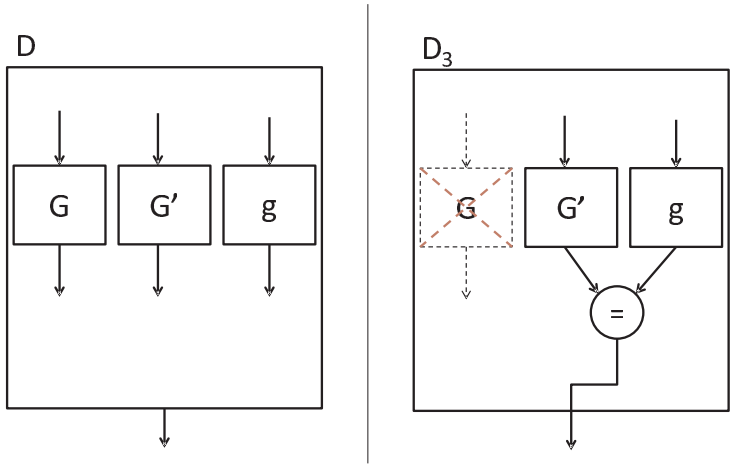

FIGURE 6. Construction of the distinguisher D_3 (right) from the distinguisher D (left). The distinguisher D_3 compares outputs of the gates G' and g and outputs 1 if and only if they are equal. The gate G is removed.

Moreover, there must exist a negligible function negl_2 such that for all oracle gates G' holds

$$\begin{aligned} &\left| \Pr_{H,f,D}\left[D^{H^f,f} \to 1 | \mathrm{EqO}(G', g)\right]\right. \\ &\left. - \Pr_{F,S,D}\left[D^{F,S^F} \to 1 | \mathrm{EqO}(G', g)\right]\right| \leq \mathrm{negl}_2(n). \end{aligned} \tag{10}$$

Otherwise we could construct a smaller distinguisher D_4 (see Figure 7) which would replace output of g with output of G'. The gate g is then removed from the distinguisher D_4. Note that since $g \in L(D)$ there cannot be a path from the gate g to the gate G'.

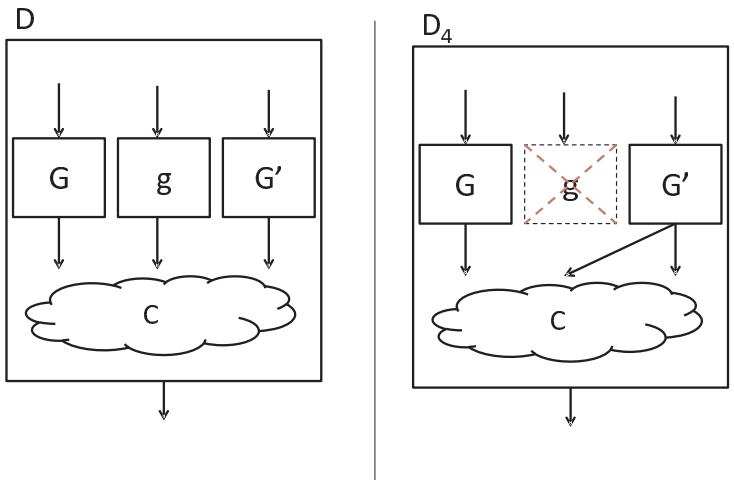

FIGURE 7. Construction of the distinguisher D_4 (right) from the distinguisher D (left): the output of the gate g is replaced by the output of the gate G'. The gate g is then removed.

Let $G_1, \ldots, G_k$ be all oracle gates in D except the gate g, where G_k denotes the gate G. We have

$$
\begin{aligned}
&\mathbf{Adv}^{\mathrm{pro}}_{S,H}(D) \\
&= \left| \Pr_{H,f,D}\left[D^{H^f,f} \to 1\right] - \Pr_{F,S,D}\left[D^{F,S^F} \to 1\right] \right| \\
&\le \Bigg| \Pr_{H,f,D}\left[D^{H^f,f} \to 1 \wedge \mathrm{EqO}(G_k, g)\right] \\
&\quad - \Pr_{F,S,D}\left[D^{F,S^F} \to 1 \wedge \mathrm{EqO}(G_k, g)\right] \\
&\quad + \Pr_{H,f,D}\left[D^{H^f,f} \to 1 \wedge \bigwedge_{i=1}^{k} \neg\mathrm{EqO}(G_i, g)\right] \\
&\quad - \Pr_{F,S,D}\left[D^{F,S^F} \to 1 \wedge \bigwedge_{i=1}^{k} \neg\mathrm{EqO}(G_i, g)\right] \Bigg| \\
&\quad + \Bigg| \Pr_{H,f,D}\left[D^{H^f,f} \to 1 \wedge \neg\mathrm{EqO}(G_k, g) \wedge \neg\left(\bigwedge_{i=1}^{k-1} \neg\mathrm{EqO}(G_i, g)\right)\right] \\
&\quad - \Pr_{F,S,D}\left[D^{F,S^F} \to 1 \wedge \neg\mathrm{EqO}(G_k, g) \wedge \neg\left(\bigwedge_{i=1}^{k-1} \neg\mathrm{EqO}(G_i, g)\right)\right] \Bigg|. \qquad (11)
\end{aligned}
$$

Now we prove the following three statements.

- By inequalities (9) and (10), the last absolute value in inequality (11) must be negligible. That is, there exists a negligible function negl_3 such that

$$\left| \Pr_{H,f,D}\left[D^{H^f,f} \to 1 \wedge \neg\mathrm{EqO}(G_k,g) \wedge \neg\left(\bigwedge_{i=1}^{k-1} \neg\mathrm{EqO}(G_i,g)\right)\right] \right. \tag{12}$$
$$\left. - \Pr_{F,S,D}\left[D^{F,S^F} \to 1 \wedge \neg\mathrm{EqO}(G_k,g) \wedge \neg\left(\bigwedge_{i=1}^{k-1} \neg\mathrm{EqO}(G_i,g)\right)\right] \right| \leq \mathrm{negl}_3(n).$$

- By equation (10) we have

$$\left| \Pr_{H,f,D}\left[D^{H^f,f} \to 1 | \mathrm{EqO}(G_k,g)\right] - \Pr_{F,S,D}\left[D^{F,S^F} \to 1 | \mathrm{EqO}(G_k,g)\right] \right| \leq \mathrm{negl}_2(n).$$

  Let α denote the probability $\alpha := \Pr_{H,f,D}\left[D^{H^f,f} \to 1 | \mathrm{EqO}(G_k,g)\right]$. Hence,

$$\alpha - \mathrm{negl}_2(n) \leq \Pr_{F,S,D}\left[D^{F,S^F} \to 1 | \mathrm{EqO}(G_k,g)\right] \leq \alpha + \mathrm{negl}_2(n). \tag{13}$$

- Since D is minimal, there must exist a negligible function negl_4 such that

$$\left| \Pr_{H,f,D}\left[D^{H^f,f} \to 1 | \bigwedge_{i=1}^{k} \neg\mathrm{EqO}(G_i,g)\right] \right.$$
$$\left. - \Pr_{F,S,D}\left[D^{F,S^F} \to 1 | \bigwedge_{i=1}^{k} \neg\mathrm{EqO}(G_i,g)\right] \right| \leq \mathrm{negl}_4(n).$$

  Otherwise we could construct a smaller distinguisher D' such that g would be replaced by a random string. Since $g \in L(D)$ and H is standard, if output of the gate g is different from all of the other gates, then it is distribution cannot be distinguished from a distribution of a random string. This holds in both real and random worlds (otherwise we could construct another smaller distinguisher D'' which would check the distribution of g and would be without the gate G_k). Let β temporarily denote the probability $\beta := \Pr_{H,f,D}\left[D^{H^f,f} \to 1 | \bigwedge_{i=1}^{k} \neg\mathrm{EqO}(G_i,g)\right]$. We can rewrite the above inequality as

$$\beta - \mathrm{negl}_4(n) \leq \Pr_{F,S,D}\left[D^{F,S^F} \to 1 | \bigwedge_{i=1}^{k} \neg\mathrm{EqO}(G_i,g)\right] \leq \beta + \mathrm{negl}_4(n). \tag{14}$$

Using inequalities (12), (13) and (14), the inequality (11) can be rewritten as follows

$$\mathbf{Adv}^{\mathrm{pro}}_{S,H}(D) \leq \Bigg| \alpha \cdot \Pr_{H,f,D}\left[\mathrm{EqO}(G_k, g)\right] - \alpha \cdot \Pr_{F,S,D}\left[\mathrm{EqO}(G_k, g)\right] + \beta \cdot \Pr_{H,f,D}\left[\bigwedge_{i=1}^{k} \neg \mathrm{EqO}(G_i, g)\right] - \beta \Pr_{F,S,D}\left[\bigwedge_{i=1}^{k} \neg \mathrm{EqO}(G_i, g)\right] \Bigg| + \mathrm{negl}_2(n) + \mathrm{negl}_4(n) + \mathrm{negl}_3(n). \tag{15}$$

Consider the probability

$$\begin{aligned} &\Pr_{H,f,D}\left[\bigwedge_{i=1}^{k} \neg \mathrm{EqO}(G_i, g)\right] \\ &= 1 - \Pr_{H,f,D}\left[\neg\left(\bigwedge_{i=1}^{k} \neg \mathrm{EqO}(G_i, g)\right)\right] \\ &= 1 - \Pr_{H,f,D}\left[\mathrm{EqO}(G_k, g)\right] \\ &\quad - \Pr_{H,f,D}\left[\neg \mathrm{EqO}(G_k, g) \wedge \neg\left(\bigwedge_{i=1}^{k-1} \neg \mathrm{EqO}(G_i, g)\right)\right]. \end{aligned}$$

Similar equality holds for the random world, i.e.,

$$\Pr_{F,S,D}\left[\bigwedge_{i=1}^{k} \neg \mathrm{EqO}(G_i, g)\right] = 1 - \Pr_{H,f,D}\left[\mathrm{EqO}(G_k, g)\right] - \Pr_{H,f,D}\left[\neg \mathrm{EqO}(G_k, g) \wedge \neg\left(\bigwedge_{i=1}^{k-1} \neg \mathrm{EqO}(G_i, g)\right)\right]$$

By similar construction as the construction D_3 in the equation (9) we have:

$$\Bigg| \Pr_{H,f,D}\left[\neg \mathrm{EqO}(G_k, g) \wedge \neg\left(\bigwedge_{i=1}^{k-1} \neg \mathrm{EqO}(G_i, g)\right)\right] - \Pr_{F,S,D}\left[\neg \mathrm{EqO}(G_k, g) \wedge \neg\left(\bigwedge_{i=1}^{k-1} \neg \mathrm{EqO}(G_i, g)\right)\right] \Bigg| \leq \mathrm{negl}_1(n).$$

Thus we can rewrite the inequality (15)

$$\begin{aligned}\mathbf{Adv}^{\mathrm{pro}}_{S,H}(D) \leq \Bigg| & \alpha \cdot \left(\Pr_{H,f,D}\big[\mathrm{EqO}(G_k, g)\big] - \Pr_{F,S,D}\big[\mathrm{EqO}(G_k, g)\big] \right) \\ & + \beta \cdot \left(\Pr_{F,S,D}\big[\mathrm{EqO}(G_k, g)\big] - \Pr_{H,f,D}\big[\mathrm{EqO}(G_k, g)\big] \right) \Bigg| \\ & + \mathrm{negl}_2(n) + \mathrm{negl}_4(n) + \mathrm{negl}_3(n) + \mathrm{negl}_1(n) \\ \leq \Big| & \Pr_{H,f,D}\big[\mathrm{EqO}(G_k, g)\big] - \Pr_{F,S,D}\big[\mathrm{EqO}(G_k, g)\big] \Big| \\ & + \mathrm{negl}_2(n) + \mathrm{negl}_4(n) + \mathrm{negl}_3(n) + \mathrm{negl}_1(n).\end{aligned}$$

Now consider the random-world scenario. The simulator S can guess output of the gate G_k with non-negligible probability only if it asks its oracle the same query as input to G_k. Hence, there exists a negligible function negl such that

$$\mathbf{Adv}^{\mathrm{pro}}_{S,H}(D) \leq \Big| \Pr_{H,f,D}\big[\mathrm{EqO}(G_k, g)\big] - \Pr_{F,S,D}\big[\mathrm{Que}_{G_k}(g)\big] \Big| + \mathrm{negl}(n).$$

Since D is minimal, then G and g are the only gates in $L(D)$. Otherwise we could construct a smaller distinguisher D_5 which would check the equality of outputs of the gates G and g (D_5 would be without the other gates in $L(D)$). □

Theorem 2. *Let H be a standard domain extension transform which is img-pro, then H is pro.*

Proof. Since the pub-pro simulators have all the information as img-pro simulators (and possibly more), it is clear that if H is not pub-pro, then it is not img-pro. Hence, in the rest of this proof we assume that H is pub-pro.

Assume the contrary that H is img-pro and not pro. Thus, there exists an img-pro simulator S_{ipro} and a negligible function negl_1 such that for all distinguishers D holds

$$\mathbf{Adv}^{\mathrm{img-pro}}_{H,S_{\mathrm{ipro}}}(D) \leq \mathrm{negl}_1(n).$$

From the assumption that H is not pro, we have that for all pro simulators S_{pro} there exists a distinguisher D and a non-negligible function $\varepsilon_1(n)$ such that

$$\mathbf{Adv}^{\mathrm{pro}}_{H,S_{\mathrm{pro}}}(D) \geq \varepsilon_1(n).$$

Assume without loss of generality that S_{ipro} does not ask the same query twice. Moreover, assume that S_{ipro} makes queries to F_{eval} as soon as possible. That is, let S be some img-pro simulator, D be a distinguisher and E some evaluation order of D. Let $g_1, \ldots, g_l$ be all f-gates in D such that g_{i+1} is evaluated after g_i. Consider that during some computation C of $D^{F_{\mathrm{eval}}, S^F}$, the simulator S in evaluation of the gate g_i asks it's oracle F_{eval} queries $Q_i := (M_{i,1}, \ldots, M_{i,q_i})$, which are the same as input to some F-gate in D. Queries made by S which

does not correspond to some F-gate in D are not in the list Q_i. Let r_D, r_S, r_F be lists of random coins used by D, S and F in the computation C and consider that $\sum_{i=1}^{l} q_i > 0$. Let

$$\delta_{D,r_D,r_S,r_F}(S) := \frac{\sum_{i=1}^{l}(i \cdot q_i)}{l \cdot \sum_{i=1}^{l} q_i}.$$

If $\sum_{i=1}^{l} q_i = 0$, then

$$\delta_{D,r_D,r_S,r_F}(S) := 0.$$

Let $\delta(S)$ be an average of $\delta_{D,r_D,r_S,r_F}(S)$ over all possible distinguishers D and random coins used during computation of D^{F_{eval},S^F}. Assume that S_{ipro} is an img-pro simulator with minimal $\delta(S_{\text{ipro}})$. That is, all other img-pro simulators S with negligible advantage against all distinguishers have $\delta(S)$ greater or equal to $\delta(S_{\text{ipro}})$.

Let S_1 and S_2 be some simulators. Note that the number $\delta(S_1)$ is smaller than $\delta(S_2)$ if S_1 asks queries corresponding to a distinguisher's F-gates sooner than S_2.

Let S_{pro} be the following simulator, which simulates S_{ipro}:

Simulator $S^F_{\text{pro}}(w)$
The simulator maintains the list L, which contains all answers of the oracle F to queries asked by S_{pro}.

- Simulate $S_{\text{ipro}}(w) \to y$.
  - When S_{ipro} asks an F_{eval} query M, then S_{pro} queries $Y := F(M)$, stores Y to the list L and returns Y to S_{ipro}.
  - When S_{ipro} asks an F_{ireveal} query, then S_{pro} returns the list L to the simulator S_{ipro}.
- Output y.

Let D be a minimal distinguisher against the simulator S_{pro} and H. Let $G, g \in L(D)$ be two oracle gates given by Lemma 5, where

$$\varepsilon_1(n) \leq \left| \Pr_{H,f,D}\left[\text{EqO}(G,g)\right] - \Pr_{F,S_{\text{pro}},D}\left[\text{Que}_G(g)\right] \right|.$$

Note that the output of the simulator S_{ipro} must be indistinguishable from a random function (otherwise there exists a distinguisher with non-negligible advantage against S_{ipro}). Hence, also S_{pro} has output indistinguishable from a random function. Thus it's possible to apply Lemma 5.

Consider an evaluation order E of the distinguisher D such that the F-gate G is evaluated after the f-gate g.

If the gate G is the only F-gate in D, then S_{pro} and S_{ipro} have the same advantage against the distinguisher D, i.e.,

$$\mathbf{Adv}^{\text{pro}}_{H,S_{\text{pro}}}(D) = \mathbf{Adv}^{\text{img-pro}}_{H,S_{\text{ipro}}}(D).$$

In the img-pro simulation $\mathbf{Adv}^{\mathrm{img-pro}}_{H,S_{\mathrm{ipro}}}(D)$ all F_{ireveal} queries, which S_{ipro} asks, return an empty list. The same holds for the pro simulation $\mathbf{Adv}^{\mathrm{pro}}_{H,S_{\mathrm{pro}}}(D)$. Hence, in this case S_{pro} correctly simulates S_{ipro}. However, this contradicts the assumption that H is img-pro.

Hence, besides the gate G there must exist another F-gate in the distinguisher D. Let $G_1, \ldots, G_k$ be all F-gates in D such that for all $i = 1, \ldots, k$ the gate G_{i+1} is evaluated after the gate G_i in the evaluation order E (note that the gates G and G_k are the same). From the assumption that H is img-pro we have

$$\begin{aligned}
\varepsilon_1(n) &\leq \Big| \Pr_{H,f,D}\big[\mathrm{EqO}(G,g)\big] - \Pr_{F,S_{\mathrm{pro}},D}\big[\mathrm{Que}_G(g)\big] \Big| \\
&= \Big| \Pr_{H,f,D}\big[\mathrm{EqO}(G,g)\big] - \Pr_{F,S_{\mathrm{ipro}},D}\big[\mathrm{Que}_G(g)\big] \\
&\quad + \Pr_{F,S_{\mathrm{ipro}},D}\big[\mathrm{Que}_G(g)\big] - \Pr_{F,S_{\mathrm{pro}},D}\big[\mathrm{Que}_G(g)\big] \Big| \\
&\leq \Big| \Pr_{F,S_{\mathrm{ipro}},D}\big[\mathrm{Que}_G(g)\big] - \Pr_{F,S_{\mathrm{pro}},D}\big[\mathrm{Que}_G(g)\big] \Big| \\
&\quad + \mathrm{negl}_1(n).
\end{aligned}$$

Let DO denote the event that all gates $G_1, \ldots, G_k$ have distinct output. Let AQ denote the event that all gates $G_1, \ldots, G_{k-1}$ were queried by a simulator, i.e., $\mathrm{AQ} \Leftrightarrow \bigwedge_{i=1}^{k-1} \mathrm{Que}_{G_i}(g)$. We have

$$\begin{aligned}
\varepsilon_1(n) \leq{}& \Big| \Pr_{F,S_{\mathrm{ipro}},D}\big[\mathrm{Que}_G(g) \wedge \neg\mathrm{DO}\big] - \Pr_{F,S_{\mathrm{pro}},D}\big[\mathrm{Que}_G(g) \wedge \neg\mathrm{DO}\big] \Big| \\
&+ \Big| \Pr_{F,S_{\mathrm{ipro}},D}\big[\mathrm{Que}_G(g) \wedge \mathrm{DO} \wedge \mathrm{AQ}\big] - \Pr_{F,S_{\mathrm{pro}},D}\big[\mathrm{Que}_G(g) \wedge \mathrm{DO} \wedge \mathrm{AQ}\big] \Big| \\
&+ \Big| \Pr_{F,S_{\mathrm{ipro}},D}\big[\mathrm{Que}_G(g) \wedge \mathrm{DO} \wedge \neg\mathrm{AQ}\big] - \Pr_{F,S_{\mathrm{pro}},D}\big[\mathrm{Que}_G(g) \wedge \mathrm{DO} \wedge \neg\mathrm{AQ}\big] \Big| \\
&+ \mathrm{negl}_1(n). \qquad (16)
\end{aligned}$$

We now show that all three absolute values in the inequality above are negligible.

1. If DO is not true then there exist gates G_i, G_j, where $i < j$ with the same output. Consider a similar construction to one in Lemma 2. Let D_1 (Figure 8) be a distinguisher which is the same as D but all edges starting at the gate G_j are redirected such that they start at the gate G_i. The gate G_j is removed. The distinguisher D_1 outputs 1 if and only if $\mathrm{EqO}(G,g)$ and $\mathrm{EqI}(G_i, G_j)$. From the definition of the distinguisher D_1 we have

$$\Pr_{H,f,D_1}\Big[{D_1}^{H^f,f} \to 1\Big] = \Pr_{H,f,D}\big[\mathrm{EqO}(G,g) \wedge \mathrm{EqI}(G_1,G_2)\big],$$

$$\Pr_{F,S,D_1}\Big[{D_1}^{F,S^F} \to 1\Big] = \Pr_{F,S,D}\big[\mathrm{EqO}(G,g) \wedge \mathrm{EqI}(G_1,G_2)\big].$$

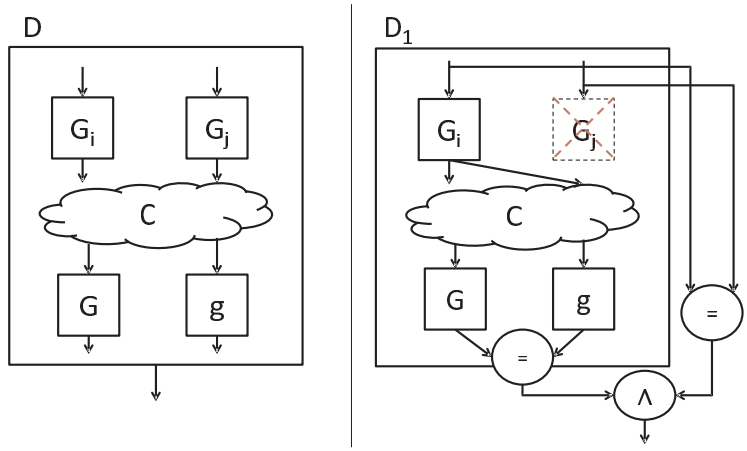

FIGURE 8. Construction of the distinguisher D_1 used in the proof of Theorem 2. The distinguisher D_1 (right) is the same as D but all edges starting at the gate G_j are redirected such that they start at the gate G_i. The gate G_j is removed. The distinguisher D_1 outputs 1 if and only if $\mathrm{EqO}(G, g)$ and $\mathrm{EqI}(G_i, G_j)$.

Since D_1 is smaller than D, from the assumption that D is minimal there exists a negligible function negl_3 such that

$$\left| \Pr_{H,f,D}\left[{D_1}^{H^f,f} \to 1\right] - \Pr_{F,S,D_1}\left[{D_1}^{F,S^F} \to 1\right] \right| \leq \mathrm{negl}_3(n).$$

Hence,

$$\left| \Pr_{H,f,D}\left[\mathrm{EqO}(G,g) \wedge \mathrm{EqI}(G_i,G_j)\right] - \Pr_{F,S_{\mathrm{pro}},D}\left[\mathrm{EqO}(G,g) \wedge \mathrm{EqI}(G_i,G_j)\right] \right| \leq \mathrm{negl}_3(n).$$

If the gates G_i and G_j have the same output in the random world, then they have the same input also (except some negligible probability). Thus,

$$\left| \Pr_{H,f,D}\left[\mathrm{EqO}(G,g) \wedge \mathrm{EqO}(G_i,G_j)\right] - \Pr_{F,S_{\mathrm{pro}},D}\left[\mathrm{EqO}(G,g) \wedge \mathrm{EqO}(G_i,G_j)\right] \right| \leq \mathrm{negl}_3(n).$$

The simulator S_{pro} is unable to output the same string as the output from the gate G unless it asks it's oracle the same query as the input to the gate G. Thus,

$$\left| \Pr_{H,f,D}\left[\mathrm{EqO}(G,g) \wedge \mathrm{EqO}(G_i,G_j)\right] - \Pr_{F,S_{\mathrm{pro}},D}\left[\mathrm{Que}_G(g) \wedge \mathrm{EqO}(G_i,G_j)\right] \right| \leq \mathrm{negl}_3(n).$$

From the assumption that H is img-pro we have

$$\left| \Pr_{H,f,D}\left[\mathrm{EqO}(G,g) \wedge \mathrm{EqO}(G_i,G_j)\right] - \Pr_{F,S_{\mathrm{ipro}},D}\left[\mathrm{Que}_G(g) \wedge \mathrm{EqO}(G_i,G_j)\right] \right| \leq \mathrm{negl}_1(n).$$

Hence,

$$\Big| \Pr_{F,S_{\text{ipro}},D}\big[\text{Que}_G(g) \wedge \text{EqO}(G_i, G_j)\big] - \Pr_{F,S_{\text{pro}},D}\big[\text{Que}_G(g) \wedge \text{EqO}(G_i, G_j)\big] \Big| \leq \text{negl}_1(n) + \text{negl}_3(n).$$

However, DO is not true if and only if there exist gates G_i, G_j such that $\text{EqO}(G_i, G_j)$ is true. Thus

$$\Big| \Pr_{F,S_{\text{ipro}},D}[\text{Que}_G(g) \wedge \neg\text{DO}] - \Pr_{F,S_{\text{pro}},D}[\text{Que}_G(g) \wedge \neg\text{DO}] \Big| \leq \text{negl}_1(n) + \text{negl}_3(n). \tag{17}$$

2. Consider the gate G_1 in the distinguisher D. Evaluation of the gate G_1 does not depend on any other F-gate (since it is the first evaluated F-gate in D). Thus, since the $\delta(S_{\text{ipro}})$ is minimal, if in the gate g' the simulator S_{ipro} asks the same query as the input to the gate G_1, then so does S_{pro} (except a negligible probability negl_2). Otherwise we could construct a simulator S'_{ipro} with smaller $\delta(S'_{\text{ipro}})$. Thus, for all f-gates g' in D holds

$$\Big| \Pr_{F,S_{\text{ipro}},D}\big[\text{Que}_{G_1}(g')\big] - \Pr_{F,S_{\text{pro}},D}\big[\text{Que}_{G_1}(g')\big] \Big| \leq \text{negl}_2(n).$$

Similarly, consider that the simulator S_{ipro} have already asked queries M_i, where $M_i = \text{Input}(G_i)$, $i < j < k$. Then the query M_j, which is the same as input to the gate G_j, can be computed without using the F_{ireveal} oracle. Hence, from the definition of the S_{pro} we have that for all f-gates g' and all $j = 1, \ldots, k$ holds

$$\left| \Pr_{F,S_{\text{ipro}},D}\left[\text{Que}_{G_j}(g') \middle| \bigwedge_{i=1}^{j-1} \text{Que}_{G_i}(g')\right] - \Pr_{F,S_{\text{pro}},D}\left[\text{Que}_{G_j}(g') \middle| \bigwedge_{i=1}^{j-1} \text{Que}_{G_i}(g')\right] \right| \leq \text{negl}_2(n).$$

Thus, if AQ is true then the simulator S_{pro} has the same view as the simulator S_{ipro}, except the negligible probability $\text{negl}_2(n)$. Therefore,

$$\Big| \Pr_{F,S_{\text{ipro}},D}\big[\text{Que}_G(g) \wedge \text{DO} \wedge \text{AQ}\big] - \Pr_{F,S_{\text{pro}},D}\big[\text{Que}_G(g) \wedge \text{DO} \wedge \text{AQ}\big] \Big| \leq \text{negl}_2(n). \tag{18}$$

3. If $\neg\text{AQ} \wedge \text{DO}$ is true, then there exist a gate G_i of which input was not queried by both: the simulator S_{ipro} (S_{pro}) and the distinguisher D. Note that in the random world scenario of the D's advantage against S_{pro}, the output of such a gate G_i cannot be distinguished from a random string. Let D_2 (Figure 9)

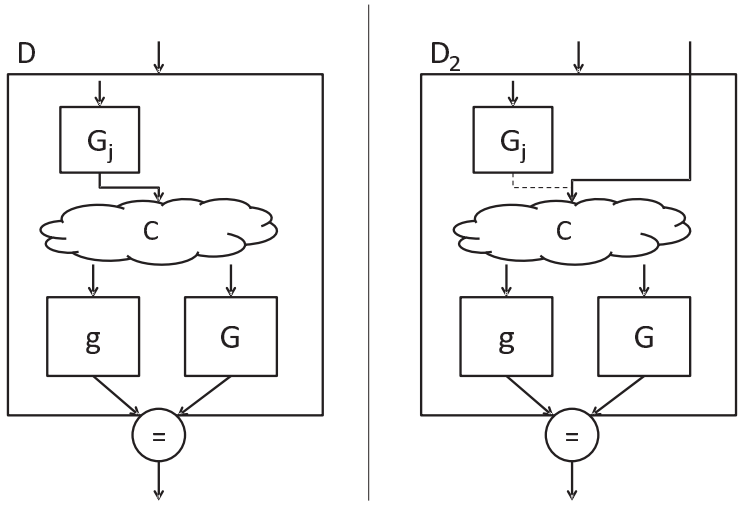

FIGURE 9. Construction of the distinguisher D_2 used in the proof of Theorem 2. The distinguisher D_2 (right) does the same as the distinguisher D (left) but the output of the gate G_j is replaced by a fresh random string.

be the same distinguisher as D, but the output of the gate G_i is replaced by a random string. Hence,

$$\Pr_{F,S_{\text{pro}},D}\left[\text{Que}_G(g)|\text{DO} \wedge \neg\text{Que}_{G_i}(g)\right] = \Pr_{F,S_{\text{pro}},D_2}\left[\text{Que}_G(g)|\text{DO} \wedge \neg\text{Que}_{G_i}(g)\right]. \tag{19}$$

Let D_3 (Figure 10) be a distinguisher, which is the same as D but the input to the gate G_i is replaced by a random string. In the random world scenario of the D_3's advantage against S_{pro}, the output of such a gate cannot be distinguisher from a random string too, hence,

$$\Pr_{F,S_{\text{pro}},D_3}\left[\text{Que}_G(g)|\text{DO} \wedge \neg\text{Que}_{G_i}(g)\right] = \Pr_{F,S_{\text{pro}},D_2}\left[\text{Que}_G(g)|\text{DO} \wedge \neg\text{Que}_{G_i}(g)\right]. \tag{20}$$

Consider the img-pro case. If the gate G_i was not queried by the simulator S_{ipro} and there is no gate in the distinguisher D with same output, then the view of S_{ipro} is the same in the case of D_3 as in the case of D. Hence,

$$\Pr_{F,S_{\text{ipro}},D_3}\left[\text{Que}_G(g)|\text{DO} \wedge \neg\text{Que}_{G_i}(g)\right] = \Pr_{F,S_{\text{ipro}},D}\left[\text{Que}_G(g)|\text{DO} \wedge \neg\text{Que}_{G_i}(g)\right]. \tag{21}$$

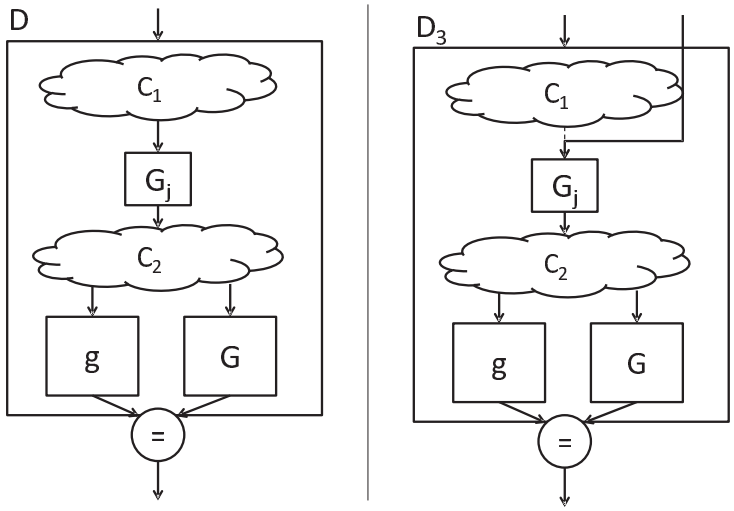

FIGURE 10. Construction of the distinguisher D_3 used in the proof of Theorem 2. The distinguisher D_3 (right) does the same as the distinguisher D (left) but the input to the gate G_j is replaced by a fresh random string.

From the equations (19), (20) and (21) we have

$$\begin{aligned}&\Pr_{F,S_{\text{ipro}},D}\big[\text{Que}_G(g)|\text{DO}\wedge\neg\text{Que}_{G_i}(g)\big]\\&=\Pr_{F,S_{\text{pro}},D}\big[\text{Que}_G(g)|\text{DO}\wedge\neg\text{Que}_{G_i}(g)\big].\end{aligned}$$

If $\neg$DO is true, then there exist two gates with the same output. By Lemma 1 there exists a negligible function negl_4 such that

$$\Big|\Pr_{H,f,D}\big[\neg\text{DO}\big]-\Pr_{F,S_{\text{pro}},D}\big[\neg\text{DO}\big]\Big|\leq\text{negl}_4(n).$$

From the assumption that H is img-pro we have:

$$\Big|\Pr_{H,f,D}\big[\neg\text{DO}\big]-\Pr_{F,S_{\text{ipro}},D}\big[\neg\text{DO}\big]\Big|\leq\text{negl}_1(n).$$

Hence,

$$\begin{aligned}\text{negl}_1(n)+\text{negl}_4(n)&\geq\Big|\Pr_{F,S_{\text{ipro}},D}\big[\neg\text{DO}\big]-\Pr_{F,S_{\text{pro}},D}\big[\neg\text{DO}\big]\Big|\\&=\Big|\Pr_{F,S_{\text{ipro}},D}\big[\text{DO}\big]-\Pr_{F,S_{\text{pro}},D}\big[\text{DO}\big]\Big|.\end{aligned}\tag{22}$$

Let $\alpha := \Pr_{F,S_{\mathrm{pro}},D}[\mathrm{DO}]$, we have

$$\begin{aligned}
&\Big| \Pr_{F,S_{\mathrm{ipro}},D}\big[\mathrm{DO}\big] - \Pr_{F,S_{\mathrm{pro}},D}\big[\mathrm{DO}\big] \Big| \\
&= \Big| \Pr_{F,S_{\mathrm{ipro}},D}\big[\mathrm{DO} \wedge \neg \mathrm{Que}_{G_i}(g)\big] - \Pr_{F,S_{\mathrm{pro}},D}\big[\mathrm{DO} \wedge \neg \mathrm{Que}_{G_i}(g)\big] \\
&\quad + \Pr_{F,S_{\mathrm{ipro}},D}\big[\mathrm{Que}_{G_i}(g)|\mathrm{DO}\big] \cdot \Pr_{F,S_{\mathrm{ipro}},D}[\mathrm{DO}] \\
&\quad - \Pr_{F,S_{\mathrm{pro}},D}\big[\mathrm{Que}_{G_i}(g)|\mathrm{DO}\big] \cdot \Pr_{F,S_{\mathrm{pro}},D}[\mathrm{DO}] \Big| \\
&\geq \Big| \Pr_{F,S_{\mathrm{ipro}},D}\big[\mathrm{DO} \wedge \neg \mathrm{Que}_{G_i}(g)\big] - \Pr_{F,S_{\mathrm{pro}},D}\big[\mathrm{DO} \wedge \neg \mathrm{Que}_{G_i}(g)\big] \\
&\quad + \alpha \cdot \left(\Pr_{F,S_{\mathrm{ipro}},D}\big[\mathrm{Que}_{G_i}(g)|\mathrm{DO}\big] - \Pr_{F,S_{\mathrm{pro}},D}\big[\mathrm{Que}_{G_i}(g)|\mathrm{DO}\big] \right) \Big| \\
&\quad - \mathrm{negl}_4(n) - \mathrm{negl}_1(n).
\end{aligned}$$

The probability that S_{pro} asks the same query as the input to the gate G_i is always smaller or equal to the probability that the same does S_{ipro}, i.e.,

$$\Pr_{F,S_{\mathrm{ipro}},D}\big[\mathrm{Que}_{G_i}(g)|\mathrm{DO}\big] \geq \Pr_{F,S_{\mathrm{pro}},D}\big[\mathrm{Que}_{G_i}(g)|\mathrm{DO}\big].$$

Hence,

$$\begin{aligned}
&\Big| \Pr_{F,S_{\mathrm{ipro}},D}\big[\mathrm{DO}\big] - \Pr_{F,S_{\mathrm{pro}},D}\big[\mathrm{DO}\big] \Big| \\
&\geq \Big| \Pr_{F,S_{\mathrm{ipro}},D}\big[\mathrm{DO} \wedge \neg \mathrm{Que}_{G_i}(g)\big] - \Pr_{F,S_{\mathrm{pro}},D}\big[\mathrm{DO} \wedge \neg \mathrm{Que}_{G_i}(g)\big] \Big| \\
&\quad - \mathrm{negl}_4(n) - \mathrm{negl}_1(n). \qquad (23)
\end{aligned}$$

Thus, by combining the inequalities (22) and (23) we have

$$\begin{aligned}
&\Big| \Pr_{F,S_{\mathrm{ipro}},D}\big[\mathrm{Que}_G(g) \wedge \mathrm{DO} \wedge \neg \mathrm{AQ}\big] - \Pr_{F,S_{\mathrm{pro}},D}\big[\mathrm{Que}_G(g) \wedge \mathrm{DO} \wedge \neg \mathrm{AQ}\big] \Big| \\
&\leq 2 \cdot \big(\mathrm{negl}_1(n) + \mathrm{negl}_4(n)\big). \qquad (24)
\end{aligned}$$

By using inequalities (16), (17), (18) and (24) we have

$$\varepsilon_1(n) \leq 3 \cdot \mathrm{negl}_1(n) + \mathrm{negl}_2(n) + \mathrm{negl}_3(n) + 2 \cdot \mathrm{negl}_4(n).$$

This contradicts the assumption that H is not pro. □

In view of Theorems 1 and 2, we can state the following corollary.

Corollary 1. *Let H be a standard domain extension transform. Then H is pro if and only if H is img-pro.*

4.1. Remarks to the proof

Note that we proved the equivalence between the properties pro and img-pro under the following two restrictions:

- We considered only standard domain extension transforms. This restriction avoids problems with partially instantiated domain extension transforms, which for example replace their final f-gates with some one-way function based on a standard-model assumption. We note that most of constructions of domain extension transforms designed so far are standard [2], [3], [5], [8], [9], [12], [13], [15].
- We used the strong indifferentiability instead of the weak. In the weak indifferentiability, any successful distinguisher must be universal, i.e., it has to distinguish real world and random world for all simulators. This fact makes analysis of the distinguisher harder.

Thus, it remains an open problem to analyze relationship between the properties pro and img-pro in the weak indifferentiability settings.

REFERENCES

[1] BELLARE, M.—CANNETI, R.—KRAWCZYK, H.: *Keying hash functions for message authentication*, in: Advances in Cryptology—Crypto '96 (N. Koblitz, ed.), Santa Barbara, California, USA, 1996, Lecture Notes in Comput. Sci., Vol. 1109, Springer-Verlag, Berlin, 1996, pp. 1–15.

[2] BELLARE, M.—RISTENPART, T.: *Hash functions in the dedicated-key setting: design choices and MPP transforms*, in: Internat. Colloq. on Automata, Languages, and Progamming, Lecture Notes in Comput. Sci., Vol. 4596, Springer-Verlag, Berlin, 2006, pp. 399–410.

[3] BELLARE, M.—RISTENPART, T.: *Multi-property-preserving hash domain extension and the EMD transform*, in: Advances in Cryptology—ASIACRYPT '06 (X. Lai et al., eds.), Shanghai, China, 2006, Lecture Notes in Comput. Sci., Vol. 4284, Springer-Verlag, Berlin, 2006, pp. 299–314.

[4] BELLARE, M.—ROGAWAY, P.: *Random oracles are practical: a paradigm for designing efficient protocols*, in: 1st ACM Conf. on Comput. and Commun. Security—CCCS '93 (D. Denning et al., eds.), Fairfax, VA, USA, 1993, ACM, New York, 1993, pp. 62–73.

[5] BIHAM, E.—DUNKELMAN, O.: *A framework for iterative hash functions: Haifa,* in: Proc. of 2nd NIST Cryptographic Hash Workshop, Santa Barbara, CA, USA, 2006.

[6] CANETTI, R.—GOLDREICH, O.—HALEVI, S.: *The random oracle methodology, revisited*, J. ACM **51** (2004), 557–594.

[7] CORON, J. S.—DODIS, Y.—MALINAUD, C.—PUNIYA, P.: *Merkle-Damgård revisited: How to construct a hash function*, in: Advances in Cryptology—CRYPTO '05, Lecture Notes in Comput. Sci., Vol. 3621, Springer-Verlag, 2005, pp. 430–448.

[8] DAMGÅRD, I.: *A design principle for hash functions*, in: Advances in Cryptology––CRYPTO '89 (G. Brassard, ed.), Santa Barbara, CA, USA, 1989, Lecture Notes in Comput. Sci., Vol. 435, Springer-Verlag, Berlin, 1989, pp. 416–427.

[9] DODIS, Y.—RISTENPART, T.—SHRIMPTON, T.: *Salvaging Merkle-Damgård for practical applications*, in: Advances in Cryptology—EUROCRYPT '09 (J.-J. Quisquater, J. Vandewalle, eds.), Houthalen, Belgium, Lecture Notes in Comput. Sci., Vol. 5479, Springer-Verlag, Berlin, 2009, pp. 371–388.

[10] FLEISCHMANN, E.—GORSKI, M.—LUCKS, S.: *Some observations on indifferentiability*, in: Proc. of the 15th Austral. Conf. on Inform. Security and Privacy—ACISP '10 (R. Steinfeld, P. Hawkes, eds.), Sydney, Australia, Lecture Notes in Comput. Sci., Vol. 6168, Springer-Verlag, Berlin, 2010, pp. 117–134.

[11] GOLDREICH, O.: *Computational Complexity—a Conceptual Perspective*. Cambridge University Press, 2008.

[12] LISKOV, M.: *Constructing an ideal hash function from weak ideal compression functions,* in: Proc. of the 13th Internat. Conf. on Selected Areas in Cryptography—SAC '06 (E. Biham et al., eds.), Montreal, Canada, 2006, Lecture Notes in Comput. Sci., Vol. 4356, Springer-Verlag, Berlin, 2007, pp. 358–375.

[13] LUCKS, S.: *A failure-friendly design principle for hash functions,* in: Proc. of the 11th Internat. Conf. on Theory and Appl. of Cryptology and Inform. Security—ASIACRYPT '05 (R. Bimal, ed.), Chennai, India,, 2005, Lecture Notes in Comput. Sci., Vol. 3788, Springer--Verlag, Berlin, 2005, pp. 474–494.

[14] MAURER, U.—RENNER, R.—HOLENSTEIN, C.: *Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology*, in: Theory of Cryptography, 1st Theory of Cryptography Conf.—TCC '04 (M. Naor, ed.), Cambridge, MA, USA, Lecture Notes in Comput. Sci., Vol. 2951, Springer-Verlag, Berlin, 2004, pp. 21–39.

[15] MERKLE, R.: *One way hash functions and DES*, in: Advances in Cryptology––CRYPTO '89 (G. Brassard, ed.), Santa Barbara, CA, USA, 1989, Lecture Notes in Comput. Sci., Vol. 435, Springer-Verlag, Berlin, 1989, pp. 428–446.

[16] RISTENPART, T.—SHACHAM, H.—SHRIMPTON, T.: *Careful with composition: Limitations of the indifferentiability framework,* in: Advances in Cryptology—EUROCRYPT '11 (K. G. Paterson, ed.) Tallinn, Estonia, 2011, Lecture Notes in Comput. Sci., Vol. 6632, Springer-Verlag, Berlin, 2011, pp. 487–506.

Received October 18, 2012

Department of Computer Science
Faculty of Mathematics, Physics and Informatics
Comenius University
Mlynská dolina
SK–842-48 Bratislava
SLOVAKIA

E-mail: rjasko@dcs.fmph.uniba.sk