Digital Signatures with Minimal Overhead from Indifferentiable Random Invertible Functions

Eike Kiltz*[1], Krzysztof Pietrzak†[2], and Mario Szegedy[3]

[1]Horst-Görtz Institute for IT Security, Ruhr-Universität Bochum, Germany,
eike.kiltz@rub.de
[2]Institute of Science and Technology, Austria, pietrzak@ist.ac.at
[3]Rutgers University, USA, szegedy@dragon.rutgers.edu

Abstract

In a digital signature scheme with message recovery, rather than transmitting the message m and its signature σ, a single enhanced signature τ is transmitted. The verifier is able to recover m from τ and at the same time verify its authenticity. The two most important parameters of such a scheme are its security and overhead $|\tau| - |m|$. A simple argument shows that for any scheme with "n bits security" $|\tau| - |m| \geq n$, i.e., the overhead is lower bounded by the security parameter n. Currently, the best known constructions in the random oracle model are far from this lower bound requiring an overhead of $n + \log q_h$, where q_h is the number of queries to the random oracle. In this paper we give a construction which basically matches the n bit lower bound. We propose a simple digital signature scheme with $n + o(\log q_h)$ bits overhead, where q_h denotes the number of random oracle queries.

Our construction works in two steps. First, we propose a signature scheme with message recovery having optimal overhead in a new ideal model, the random invertible function model. Second, we show that a four-round Feistel network with random oracles as round functions is tightly "public-indifferentiable" from a random invertible function. At the core of our indifferentiability proof is an almost tight upper bound for the expected number of edges of the densest "small" subgraph of a random Cayley graph, which may be of independent interest.

Keywords: digital signatures, indifferentiability, Feistel, Additive combinatorics, Cayley graph.

1 Introduction

When transmitting a message m over an unauthenticated public channel, one usually appends a string σ to the message that can be used to verify (relative to a public key) the authenticity of the message. This string σ is called a *digital signature of m*. More generally, one transforms the message m into an *enhanced signature* τ such that (i) the original message m can be recovered from τ; (ii) the authenticity of m can be verified from τ. This is called a digital signature scheme with

*Funded by a Sofja Kovalevskaja Award of the Alexander von Humboldt Foundation and the German Federal Ministry for Education and Research.

†Supported by the European Research Council under the European Unions Seventh Framework Programme (FP7/2007-2013) / ERC Starting Grant (259668-PSPC)

message recovery (MR) and is used to save on bandwidth, i.e., to minimize the *signature overhead* informally defined as $\mathsf{O} = |\tau| - |m|$ (signature length minus message length). Standard bodies for signature schemes (e.g. ISO/IEC 9796 and IEEE P1363a) contain several schemes with MR. In this paper we ask the natural question: *what is the minimal overhead required to achieve a desired security level?*

1.1 Bounds on the Overhead

A Trivial Lower Bound for Every Scheme. Following [3], we say that a signature scheme has "n-bit security" if all adversaries A attacking the scheme have success ratio $\mathsf{SR}(\mathsf{A})$ at most 2^{-n}, where $\mathsf{SR}(\mathsf{A}) := \mathsf{success}(\mathsf{A})/\mathsf{time}(\mathsf{A})$. A natural lower bound for the overhead of a signature scheme (with or without message recovery) for n-bit security is $\mathsf{O} \geq n$ bits. This is since for a signature scheme with O bits of overhead any random bit string τ constitutes a valid enhanced signature with probability $2^{-\mathsf{O}}$. Hence an adversary A guessing a single random authenticated message τ has success ratio $\mathsf{SR}(\mathsf{A}) = 2^{-\mathsf{O}}$ which implies $\mathsf{O} \geq n$.

Overhead of Schemes without MR. In standard digital signature schemes (without message recovery) such as RSA full domain hash [5], the probabilistic signature scheme PSS [5], or (pairing-based) BLS signatures [6] the overhead equals the size of a signature. Since classical signatures contain (at least) one group element (e.g., $\mathbb{Z}_N^*$ or an elliptic curve group) whose representation requires at least $2n$ bits (for n bits security, due to generic square-root attacks) we cannot hope to obtain an overhead smaller than $2n$ bits. The above lower bounds do not apply for schemes without such a group structure, in particular schemes based on lattices or codes, but for other reasons these schemes tend to have a very large overhead and/or prohibitively large public parameters.

Overhead of Schemes with MR in the RO model. Computing the overhead for a given signature scheme turns out to be a bit subtle and depends on the security reduction. We exemplify such a calculation for the RSA-based probabilistic signature scheme with message recovery $\mathsf{PSS\text{-}MR}[n_0, n_1]$ [5], which can be seen as a two-round Feistel construction. $\mathsf{PSS\text{-}MR}[n_0, n_1]$ has an overhead of $n_0 + n_1$ bits, where parameter n_0 controls the randomness and n_1 the amount of added redundancy used during signing. The minimal size of n_0 and n_1 providing a given security level can be computed from the security reduction. The security reduction from [5] in the random oracle model [4] transforms an adversary against $\mathsf{PSS\text{-}MR}[n_0, n_1]$ making q_s (online) signing and q_h (offline) hash queries with success probability $\varepsilon_{\mathsf{PSS\text{-}MR}}$ into an adversary against RSA with success probability $\varepsilon_{\mathsf{RSA}}$ such that $\varepsilon_{\mathsf{PSS\text{-}MR}} = \varepsilon_{\mathsf{RSA}} + \varepsilon_{sim}$, where $\varepsilon_{sim} = (q_s + q_h)^2(2^{-n_0} + 2^{-n_1})$. An easy computation shows that this implies $\mathsf{O}_{\mathsf{PSS\text{-}MR}} = n_0 + n_1 \geq 2n + 2\log_2(q_h)$ bits of overhead for n bits security.[1] An improved security reduction by Coron gives $\mathsf{O}_{\mathsf{PSS\text{-}MR}} \geq 2n + \log_2(q_h) + \log_2(q_s)$. Recently, an alternative security reduction for PSS-MR was proposed in [15] demonstrating a tight security reduction for $\mathsf{PSS\text{-}MR}[n_0 = 0, n_1]$ with zero-padding from the (stronger) phi-hiding assumption [7]. However, the required overhead is still $\mathsf{O}_{\mathsf{PSS\text{-}MR}} = n + \log_2(q_h)$ bits, stemming from an additive term $\varepsilon_{sim} = q_h^2/2^{n_1}$ in the security reduction.

The Random Invertible Permutation Model. Besides the popular random-oracle model, signature schemes have also been analyzed in other idealized models. In particular, [16, 8] propose

[1] For n-bit security of $\mathsf{PSS\text{-}MR}[n_0, n_1]$ we require $\mathsf{SR}(\mathsf{A}) \leq 2^{-n+1}$ which is implied by $\varepsilon_{\mathsf{RSA}}/\mathsf{time}(\mathsf{A}) \leq 2^{-n}$ and $\varepsilon_{sim}/\mathsf{time}(\mathsf{A}) \leq 2^{-n}$. With $\mathsf{time}(\mathsf{A}) \geq q_s + q_h$ we obtain $n_0 \geq n + \log_2(q_h)$ and $n_1 \geq n + \log_2(q_h)$ and consequently the overhead is $\mathsf{O} = n_0 + n_1 \geq 2n + 2\log_2(q_h)$.

Type	Required overhead O for n bits security			Security
	asymptotic	$n = 80$	$q_h \leq 2^{60}, q_s \leq 2^{40}$	reduction
2-round Feistel	$2n + 2(\log q_h)$	320	280	Bellare-Rogaway [5]
2-round Feistel	$2n + \log(q_h) + \log(q_s)$	320	240	Coron [9]
2-round Feistel	$n + \log(q_h)$	160	140	Kakvi-Kiltz [15]
6-round Feistel	$n + \log(q_h)$	160	140	[16, 8]+[17]
4-round Feistel	$n + o(\log q_h)$	97	93	this work (1024-bit RSA)
4-round Feistel	$n + o(\log q_h)$	92	90	this work (2048-bit RSA)

Table 1: Overhead of RSA-based signature schemes with message recovery in the random oracle model for n bits security assuming the adversary makes at most q_h hash and q_s signing queries. The table shows the overhead required for $n = 80$ (and only the trivial upper bound $q_h + q_s \leq 2^{80}$) and when we additionally assume that the number of random-oracle/signature queries are upper bounded by $q_h \leq 2^{60}$ and $q_s \leq 2^{40}$, respectively. As the $o(\log q_h)$ term in our bound depends on the domain, we give the bounds for 1024 and 2048 bits RSA.

a digital signature scheme with message recovery, together with optimal security reduction in the ideal *random invertible permutation model.* Unfortunately, unlike for random oracles, there is no standard cryptographic object which could be used to directly instantiate random invertible permutations over a large domain.[2] In order to get a construction in the random oracle model, one can replace the random invertible permutation $\mathcal{P}$ with some construction $\mathsf{C}^{\mathcal{H}}$ (based on a random oracle $\mathcal{H}$) that is *indifferentiable* [18, 10] from $\mathcal{P}$. In the context of signature schemes, already a weaker notion called "public-indifferentiability" [22, 11, 17] is sufficient. In [17] it is proven that a six-round Feistel network with random round functions is public-indifferentiable from a random invertible permutation. (For full indifferentiability more rounds are needed [14].) Unfortunately, the reduction from [17] is not tight in the oracle query complexity (i.e., the number of queries made by the simulator is quadratic in the number of the queries made by the distinguisher), and as a consequence the required overhead is $\log(q_h)$ bits larger than in the ideal permutation model.

Table 1 summarizes the signature overhead and gives concrete parameters for a typical security parameter of $n = 80$ bits and using 1024/2048-bit RSA. (Parameters for $n \in \{128, 192, 256\}$ can be computed accordingly.) We remark that the table is only valid for sufficiently large messages, i.e., if $|M| \geq 1024 - \mathsf{O}$. For smaller messages standard signatures such as BLS naturally outperform any RSA-based signature scheme with MR.

1.2 Our contribution

Our main contribution is to revisit and affirmatively answer the question whether there exist signature schemes with minimal overhead in the random oracle model. In a first step we show that such a scheme exists in a new ideal model which we call *random invertible function* model, provided that the ideal functions' image is sufficiently sparse. Next, we show that a Feistel network with four rounds and random oracles as round functions is public-indifferentiable from a random invertible function *with an almost tight reduction.* Combining the two steps, we obtain a new signature

[2] For fixed small domain, one might use a block-cipher with a fixed key. Though, the heuristic to replace a random permutation with a block-cipher like AES with fixed known keys is not as well analyzed as replacing a random oracle with a strong cryptographic hash function.

scheme with message recovery with almost minimal overhead in the random oracle model.

Signature scheme with MR from random invertible functions. Given a trapdoor permutation $\mathsf{TDP} = (\mathsf{f}, \mathsf{f}^{-1})$ over $\{0,1\}^k$ and an injective function $\mathcal{F} : \{0,1\}^m \rightarrow \{0,1\}^k$ $(k > m)$ that can be queried in both directions, we can define a signature scheme with message recovery $\mathsf{SIG\text{-}MR}^{\mathcal{F}}$ as follows. The enhanced signature τ on a message m is defined as $\tau = \mathsf{f}^{-1}(\mathcal{F}(m))$. Signature recovery first evaluates the trapdoor permutation on τ and checks if the result has a valid pre-image or not, i.e., $\{m, \perp\} = \mathcal{F}^{-1}(\mathsf{f}(\tau))$. If the result is not $\perp$, it returns message m. The overhead of $\mathsf{SIG\text{-}MR}^{\mathcal{F}}$ is $\mathsf{O} = k - m$ bits. It is a straight-forward generalization of [16, 15], to prove that the resulting signature scheme $\mathsf{SIG\text{-}MR}^{\mathcal{F}}$ is tightly secure (losing an additive factor $q_{\mathcal{F}}/2^{k-m}$, where $q_{\mathcal{F}}$ is the number of queries to $\mathcal{F}$) if $\mathcal{F}$ is chosen at random. (The above scheme can only be proved secure assuming TDP is lossy [21]. Using a trick of [16] we can also prove a slightly modified scheme tightly secure assuming TDP is one-way.)

Instantiating Invertible Random Functions with Random Oracles. To instantiate the above scheme in the random oracle model, we must replace the random invertible function $\mathcal{F} : \{0,1\}^m \rightarrow \{0,1\}^k$ with a construction $\mathsf{C}^{\mathcal{H}}$ that is public-indifferentiable from $\mathcal{F}$.

It is easy to construct a random invertible function $\mathcal{F} : \{0,1\}^m \rightarrow \{0,1\}^k$ from a random invertible permutation $\mathcal{P} : \{0,1\}^k \rightarrow \{0,1\}^k$ (by setting $\mathcal{F}(x) = \mathcal{P}(x \| 0^{k-m})$) with a tight reduction. But as discussed above, we do not know how to instantiate $\mathcal{P}$ in the random oracle model without losing at least a quadratic factor in the oracle query complexity [17]. Furthermore, it is well known that a five (or less) round Feistel network cannot be pub-indifferentiable from a random invertible permutation [17].

A formal definition of pub-indifferentiability is given in Definition 2.1. The important parameters are the error ε_{sim} and the number of queries q_{S} made by the simulator S, which are both functions in the number of queries q_{D} made by the distinguisher D. In order to get a reduction with optimal overhead, i.e., where the security (in bits) is not much smaller than the overhead $\mathsf{O} = k - m$, we need $q_{\mathsf{S}} \approx q_{\mathsf{D}}$ and $\varepsilon_{sim} \approx q_{\mathsf{D}}/2^{k-m}$.

Two Feistel rounds. As a simple warmup example we show that a two-round Feistel network (with random oracles as round functions) is pub-indifferentiable from $\mathcal{F}$ with

$$\varepsilon_{sim} = q_{\mathsf{D}}^2/2^{k-m} \qquad \text{and} \qquad q_{\mathsf{S}} = q_{\mathsf{D}}.$$

The resulting signature scheme (as explained above) requires an overhead of $\mathsf{O} = n + \log_2(q_h)$ to achieve n bits security. This essentially reproves the overhead of $\mathsf{PSS\text{-}MR}$ obtained in [15].

Four Feistel rounds. As the main technical result of this paper we give a construction $\mathsf{C}_{4F}^{\mathcal{H}}$ based on a four round Feistel network and prove it pub-indifferentiable from $\mathcal{F}$ with

$$\varepsilon_{sim} \leq q_{\mathsf{D}}^{1+o(1)}/2^{k-m} \qquad \text{and} \qquad q_{\mathsf{S}} = \tilde{O}(q_{\mathsf{D}}). \tag{1}$$

Hence the resulting signature scheme has an overhead of $\mathsf{O} = n + o(\log q_h)$ bits, cf. Table 1. The $o(1)$ term can be computed explicitly and leads to 95 bits overhead for $n = 80$ bits security if the domain of the TDP is at least 1024 bits. (More concretely, the $o(1)$ term goes to 0 as the ratio of the security we want to achieve, divided by the domain size of the TDP, decreases.)

In the proof of (1), the variable $Q(\mu, q) = \max_{\mathcal{X},\mathcal{Z}} |\{(x,z) \mid x \in \mathcal{X}, z \in \mathcal{Z}, z - x \in \mathcal{B}\}|$ (where $\mathcal{B}, \mathcal{X}, \mathcal{Z}$ are q element subsets of $\mathbb{Z}_\mu$ and $\mathcal{B}$ is sampled uniformly at random) will play a central role.

This variable has a natural interpretation in graph theoretical terms as we'll explain in Section 5.1. We prove by a compression argument (Corollary 6.3) an upper bound

$$\text{for each } 0 < a < 1/4 : Q(\mu, \mu^a) \leq \mu^{a+2a^2} \quad \text{(with probability extremely close to 1).} \tag{2}$$

We believe that this bound may be of independent interest. It complements a result of Alon et al. [2, Th. 4] which states that $Q(\mu, \mu^a) \approx \mu^{3a-1}$ for $2/3 < a \leq 1$, i.e. their bound applies to large subgraphs of size $\geq \mu^{2/3}$.

We prove (Theorem 4.2) that the four round Feistel network $\mathsf{C}_{4F}^{\mathcal{H}}$ is pub-indifferentiable form a random invertible function with an error with a simulator making $q_{\mathsf{S}} = \tilde{O}(q_{\mathsf{D}})$ queries and failing with probability $\varepsilon_{sim} = O(\mathsf{E}[Q(\mu, q_{\mathsf{D}})]/2^{k-m})$ and a queries. Setting $q_{\mathsf{D}} = \mu^a$ in (2) this gives the claimed bound (1) on the pub-indifferentiability of $\mathsf{C}_{4F}^{\mathcal{H}}$.

Very informally, the term $Q(\mu, q_{\mathsf{D}})$ show up in the proof as follows. Consider a q_{D} query adversary who queries an unbalanced three round Feistel nework as illustrated in Figure 2. Assume she queried the third oracle $\mathcal{H}_3$ on inputs $\mathcal{Z}$ and the second on inputs $\mathcal{Y}$ (receiving outputs $\mathcal{B}$). Next, she chooses some set $\mathcal{X}$ and queries the network on inputs $(x, 0)$. If for some $x \in \mathcal{X}$ we have $\mathcal{H}_1(x) \in \mathcal{Y}$ and $x + \mathcal{H}_2(\mathcal{H}_1(x)) \in \mathcal{Z}$, then the input to $\mathcal{H}_3$ on this query has been already fixed, and the simulator can't program it.[3] Any tuple $(x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ where $x + \mathcal{H}_2(y) = z$ can lead to such a failure with probability $\mathbb{Z}_\rho^{-1} \approx 2^{m-k}$ (namely, if $\mathcal{H}_1(x) = y$). The expected number of such tuples (for an optimal choice of $\mathcal{X}, \mathcal{Z}$ after seeing the random $\mathcal{B}$) is $Q(\mu, q_{\mathsf{D}})$.

We leave it is an interesting open problem whether our techniques can be used to prove better bounds for constructions of *permutations* from random oracles. As mentioned above, currently all such constructions suffer from a quadratic increase in the oracle query complexity. Another interesting question is, whether random invertible functions can be used to build chosen-ciphertext secure encryption with optimal overhead. Interestingly, the construction from [1] also uses a four round Feistel network, but the proven security suffers from a quadratic loss in running time.

2 Preliminaries

For $n \in \mathbb{N}$, we write 1^n for the string of n ones, and $[n]$ for $\{1, \ldots, n\}$. $|x|$ denotes the length of a bitstring x, while $|S|$ denotes the size of a set S. $s \leftarrow S$ denotes sampling an element s uniformly at random from the set S. For an algorithm A, we write $z \leftarrow \mathsf{A}(x, y, \ldots)$ to indicate that A is a (probabilistic) algorithm that outputs z on input $(x, y, \ldots)$. In the following we will introduce some basic cryptographic objects that (for simplicity) are defined over bit-strings (rather than arbitrary domains).

2.1 Ideal primitives and indifferentiability

Throughout, we use the letter $\mathcal{H}$ to denote a random oracle [4], $\mathcal{P}$ for a random invertible permutation and $\mathcal{F}$ for a random invertible function.

A random oracle $\mathcal{H} : \mathcal{D} \to \mathcal{R}$ with input domain $\mathcal{D} \subset \{0,1\}^*$ and range $\mathcal{R} \subset \{0,1\}^*$ is a function chosen uniformly at random from all functions $\mathcal{D} \to \mathcal{R}$. A random invertible function

[3] Our actual construction has one more round, so we also program the right half of the output of the network. Moreover the right half contains not just the redundance $0 \in \mathbb{Z}_\mu$, but another element $\mathbb{Z}_\mu$ which is part of the message, this way the right half is large enough so we can ignore other terms which come up in the proof and depend on the collision probability of random elements over this domain.

$\mathcal{F} : \mathcal{D} \to \mathcal{R}$ is a function chosen uniformly at random from all injective functions (i.e., all functions where $x \neq x' \Rightarrow \mathcal{F}(x) \neq \mathcal{F}(x')$). A random invertible permutation $\mathcal{P}$ is a random injective function where $\mathcal{D} \equiv \mathcal{R}$.

Unlike for $\mathcal{H}$, which can only be queried in forward direction, whenever we consider algorithms with oracle access to $\mathcal{F}$ (or $\mathcal{P}$), it is always understood that $\mathcal{F}$ can be queried also in inverse direction. Technically, we can think of $\mathcal{F}$ as being given by two oracles $\mathcal{F}$ and $\mathcal{F}^{-1}$, where $\mathcal{F}^{-1}(\mathcal{F}(x)) = x$ and $\mathcal{F}^{-1}(y) = \perp$ if y is not in the range of $\mathcal{F}$.

Below we define a pub-indifferentiable [11, 22] construction of $\mathcal{F}$ from $\mathcal{H}$. The public indifferentiability notion differs from the standard indifferentiability notion [18, 10] by the fact that in the public notion the simulator S gets to see all queries made by D.

Definition 2.1 (pub-indifferentiability) *A $(q_{\mathsf{D}}, q_{\mathsf{S}}, \varepsilon_{sim}, t_{sim})$-public-indifferentiable construction of a random invertible function $\mathcal{F}$ from a random oracle $\mathcal{H}$ is a stateless oracle circuit C and a (stateful, probabilistic) simulator S such that for any distinguisher D making at most q_{D} oracle queries, S makes at most q_{S} oracle queries, runs in time at most t_{sim} and the following holds:*

$$|\Pr[\mathsf{D}^{\mathcal{F},\mathsf{S}^{\mathcal{F}}}(1^n) = 1] - \Pr[\mathsf{D}^{\mathsf{C}^{\mathcal{H}},\mathcal{H}}(1^n) = 1]| \leq \varepsilon_{sim},$$

here the second oracle $\mathsf{S}^{\mathcal{F}}$ gets to see also the queries made by $\mathsf{D}^{\mathcal{F},\mathsf{S}^{\mathcal{F}}}$ to the first oracle $\mathcal{F}$.

2.2 Digital signatures with message recovery

A digital signature scheme with message recovery $\mathsf{SIG\text{-}MR} = (\mathsf{G}_{\mathsf{SIG\text{-}MR}}, \mathsf{Sign}, \mathsf{Recover})$ consists of three algorithms and two function families $m(n), k(n)$ describing message space $\{0,1\}^{m(n)}$ and signature space $\{0,1\}^{k(n)}$. Key generation $\mathsf{G}_{\mathsf{SIG\text{-}MR}}$ generates a keypair $(pk, sk) \leftarrow \mathsf{G}(1^n)$ for a secret signing key sk and a public verification key pk. The signing algorithm Sign on input a message $M \in \{0,1\}^{m(n)}$ and the secret signing key, and returns an enhanced signature $\tau \leftarrow \mathsf{Sign}_{sk}(M) \in \{0,1\}^{k(n)}$ of the message. The recovery algorithm $\mathsf{Recover}$ takes a verification key pk and an enhanced signature τ as input and returns $M \leftarrow \mathsf{Recover}_{pk}(\tau)$, where $M \in \{0,1\}^{m(n)} \cup \{\perp\}$. We require that $\Pr[\mathsf{Recover}_{pk}(\mathsf{Sign}_{sk}(M)) = M] = 1$.

The security of the signature scheme can be analyzed in a model where an idealized primitive exists, for example a random oracle or a random invertible function. In that case the adversary and the scheme get access to the idealized primitive $\mathcal{O}$ by making oracle calls.

Security. Let us recall the *existential unforgeability against chosen message attacks* (EUF-CMA) security game [12] relative to the ideal primitive $\mathcal{O}$, played between a challenger and a forger A.

1. The challenger runs $\mathsf{G}_{\mathsf{SIG\text{-}MR}}(1^n)$ to generate a keypair (pk, sk). Forger A receives pk as input.
2. Forger A may ask the challenger to sign a number of messages and evaluate the ideal object $\mathcal{O}$. To query the i-th signature, A submits a message $M_i \in \{0,1\}^{m(n)}$ to the challenger. The challenger returns an enhanced signature τ_i under sk for this message. For the j-th query to $\mathcal{O}$, A submits a query x_j to the challenger who returns the values $\mathcal{O}(x_j)$.
3. Forger A outputs an enhanced signature τ^*.

Let $M^* \leftarrow \mathsf{Recover}(pk, \tau^*)$ be the recovered message of A's forgery. The game outputs 1 (meaning forger A wins the game) if $M^* \neq \perp$ (i.e., τ^* is a valid enhanced signature) and $M^* \neq M_i$ for all i. The success probability of A is the probability that the game outputs 1.

Definition 2.2 (Security and Overhead of $\mathsf{SIG\text{-}MR}$) *Let $\mathcal{O}$ be an ideal primitive and let $\mathsf{SIG\text{-}MR}^{\mathcal{O}}$ be a signature scheme with message recovery, where $\{0,1\}^{m(n)}$ is the message and $\{0,1\}^{k(n)}$ is the signature space. Let $t_{sig}, q_s, q_o, \varepsilon_{sig}$ be functions of a security parameter n.*

Security: $\mathsf{SIG\text{-}MR}^{\mathcal{O}}$ *is $(t_{sig}, q_s, q_o, \varepsilon_{sig})$-secure relative to $\mathcal{O}$, if all adversaries A running in time at most t_{sig} making at most q_s signing queries and q_o queries to $\mathcal{O}$ (this includes direct queries to $\mathcal{O}$, but also the queries to $\mathcal{O}$ done during evaluation of the signature queries), have success probability at most ε_{sig}. If $\mathcal{O}$ is a random oracle (random invertible function), then we say that $\mathsf{SIG\text{-}MR}^{\mathcal{O}}$ is secure in the random oracle (random invertible function) model.*

n-bit security: *We say $\mathsf{SIG\text{-}MR}^{\mathcal{O}}$ has n bits of security against q_s, q_o queries if it is $(t_{sig}, q_s, q_o, \varepsilon_{sig})$-secure for all $t_{sig}, \varepsilon_{sig}$ satisfying $\varepsilon_{sig}/t_{sig} \leq 2^{-n}$. We simply say it has n bits security if it has n bits security for any q_s, q_o (we can always assume the trivial upper bound $q_s + q_o \leq t_{sig} \leq 2^n$.[4])*

Overhead: *The overhead of $\mathsf{SIG\text{-}MR}^{\mathcal{O}}$ is defined as $k(n) - m(n)$. $\mathsf{O}_{\mathsf{SIG\text{-}MR}^{\mathcal{O}}}(n, q_s, q_o)$ denotes the overhead required in the construction $\mathsf{SIG\text{-}MR}^{\mathcal{O}}$ to reach n bits security against q_s and q_o queries. $\mathsf{O}_{\mathsf{SIG\text{-}MR}^{\mathcal{O}}}(n)$ is short for $\mathsf{O}_{\mathsf{SIG\text{-}MR}^{\mathcal{O}}}(n, 2^n, 2^n)$.*

In the following we will propose a scheme with finite message space. To obtain a scheme for any larger message space, one can apply the domain extension given in Appendix B.

Using a composition theorem [18], we can express the security of a signature scheme proven secure in the invertible function model when we replace the invertible random function $\mathcal{F}$ with an pub-indifferentiable constructions $\mathsf{C}^{\mathcal{H}}$ as follows.

Theorem 2.3 *If $\mathsf{SIG\text{-}MR}^{\mathcal{F}}$ is $(t_{sig}, q_s, q_h, \varepsilon_{sig})$-secure in the random invertible function model, and C is a $(q_{\mathsf{D}} = q_h, q_{\mathsf{S}}, \varepsilon_{sim}, t_{sim})$-pub-indifferentiable construction of $\mathcal{F}$ from $\mathcal{H}$ (cf. Def.2.1), then $\mathsf{SIG\text{-}MR}^{\mathsf{C}^{\mathcal{H}}}$ is $(t_{sig} - t_{sim}, q_s, q_{\mathsf{S}}, \varepsilon_{sig} + \varepsilon_{sim})$-secure in the random oracle model.*

2.3 Trapdoor Permutations

A trapdoor permutation $\mathsf{TDP} = (\mathsf{G}_{\mathsf{TDP}}, \mathsf{f}, \mathsf{f}^{-1})$ over domain $\mathcal{D}(n) = \{0,1\}^{k(n)}$ consists of three ppt algorithms. Key generation $\mathsf{G}_{\mathsf{TDP}}$ generates a keypair $(ek, td) \leftarrow \mathsf{G}_{\mathsf{TDP}}(1^n)$ of evaluation key and trapdoor. For every (ek, td) in the domain of $\mathsf{G}_{\mathsf{TDP}}(1^n)$, $\mathsf{f}(ek, \cdot)$ and $\mathsf{f}^{-1}(td, \cdot)$ compute permutations $\mathsf{f}_{ek}(\cdot), \mathsf{f}_{td}^{-1}(\cdot)$ on $\{0,1\}^{k(n)}$ s.t. for all $x \in \{0,1\}^{k(n)}$: $\mathsf{f}_{td}^{-1}(\mathsf{f}_{ek}(x)) = x$. We say TDP is homomorphic if $(\mathcal{D}(n), \circ)$ is a group and for all $x_1, x_2 \in \mathcal{D}(n)$, $\mathsf{f}_{ek}(x_1) \circ \mathsf{f}_{ek}(x_2) = \mathsf{f}_{ek}(x_1 \circ x_2)$.

We now recall the security properties of one-wayness and regular lossiness [15, 21].

Definition 2.4 (Security of TDP) *Let $t = t(n)$ and $\varepsilon_{one-way} = \varepsilon_{one-way}(n)$ be functions of a security parameter n. TDP is $(\varepsilon_{one-way}, t)$-one-way if for all adversaries A running in time at most t, $\Pr[\mathsf{A}(ek, \mathsf{f}_{ek}(x)) = x] \leq \varepsilon_{one-way}$, where $(ek, td) \leftarrow \mathsf{G}_{\mathsf{TDP}}(1^n)$, $x \leftarrow \{0,1\}^{k(n)}$.*

Definition 2.5 (Lossy TDP) *Let $t_{lossy} = t_{lossy}(n)$, $\ell = \ell(n)$ and $\varepsilon_{lossy} = \varepsilon_{lossy}(n)$ be functions of a security parameter n. A trapdoor permutation TDP over domain $\{0,1\}^{k(n)}$ is regular $(\varepsilon_{lossy}, t_{lossy}, \ell)$-lossy if there exists a ppt algorithm G_{lossy} (the lossy key generator) that on input 1^n outputs ek' such that*

[4] As $\varepsilon \leq 1$, $\varepsilon_{sig}/t_{sig} \leq 2^{-n}$ for every $t_{sig} \geq 2^n$, so we only have to look at the case $t_{sig} \leq 2^n$.

1. *(indistinguishability of real and lossy keys) for all adversaries* A *running in time at most* t_{lossy}*,* $\Pr[\mathsf{A}(ek) = 1] - \Pr[\mathsf{A}(ek') = 1] \leq \varepsilon_{lossy}$*, where* $(ek, td) \leftarrow \mathsf{G}_{\mathsf{TDP}}(1^n)$ *and* $ek' \leftarrow \mathsf{G}_{lossy}(1^n)$*;*

2. *(lossiness)* $\mathsf{f}_{ek'}(\cdot)$ *is* ℓ*-to-*1*, i.e.* $\forall x \in \{0,1\}^{k(n)} : |\{z \; : \; f_{ek'}(z) = f_{ek'}(x)\}| = \ell$*.*

For any $\ell \geq 1$, a lossy trapdoor permutation is collision-resistant when instantiated in lossy mode [21]. The most important example of a trapdoor permutation is RSA with domain $\mathbb{Z}_N^*$, defined as $\mathsf{f}_{N,e}(x) = x^e \bmod N$. It is homomorphic with respect to modular multiplication. It is one-way under the RSA assumption; for any $e < N^{1/4}$ it is furthermore regular e-lossy under the phi-hiding assumption [15], where e is the public RSA exponent. Another example of a (homomorphic and regular lossy) trapdoor function is Paillier [20].

3 Signatures with MR from random invertible functions

Let $k = k(n)$ and $m = m(n)$ be functions with $k(n) \geq m(n)$. Let TDP be a trapdoor permutation over domain $\{0,1\}^k$ and $\mathcal{F} : \{0,1\}^m \to \{0,1\}^k$ be a random invertible function. We build a signature scheme with message recovery $\mathsf{SIG\text{-}MR}^{\mathcal{F}} = (\mathsf{G}_{\mathsf{SIG\text{-}MR}}, \mathsf{Sign}, \mathsf{Recover})$ with message space $\mathcal{M}(n) = \{0,1\}^m$ and signature space $\mathcal{S}(n) = \{0,1\}^k$. $\mathsf{G}_{\mathsf{SIG\text{-}MR}}(1^n)$ runs $(ek, td) \leftarrow \mathsf{G}_{\mathsf{TDP}}(1^n)$. It returns $pk = ek$ and $sk = td$.

```
Algorithm $\mathsf{Sign}_{sk}(M \in \{0,1\}^m)$
    $y := \mathcal{F}(M) \in \{0,1\}^k$
    Return $\tau = \mathsf{f}_{td}^{-1}(y) \in \{0,1\}^k$
```

```
Algorithm $\mathsf{Recover}_{pk}(\tau \in \{0,1\}^k)$
    $y = \mathsf{f}_{ek}(\tau)$
    If $\mathcal{F}^{-1}(y) = \perp$ then return $\perp$
    Else return $M = \mathcal{F}^{-1}(y)$
```

Note that $\mathsf{SIG\text{-}MR}$ has $n_1 = k - m$ bits of redundancy and correctness follows since TDP is a permutation.

The following theorem proves security provided TDP is regular lossy. Its proof is similar to the one of FDH in [15] and postponed to Appendix C.

Theorem 3.1 *Suppose* TDP *is regular* $(\ell, t_{lossy}, \varepsilon_{lossy})$*-lossy (i.e., lossy by* $\log_2(\ell)$ *bits) and* $\mathcal{F}$ *is a random invertible function from* $\{0,1\}^m$ *to* $\{0,1\}^k$*. Then* $\mathsf{SIG\text{-}MR}^{\mathcal{F}}$ *is* $(t_{sig}, q_s, q_f, \varepsilon_{sig})$ *secure with*

$$t_{sig} \approx t_{lossy}, \quad \varepsilon_{sig} = (2\ell - 1)/\ell \cdot \varepsilon_{lossy} + \frac{q_f}{2^{k-m}}.$$

In case TDP only satisfies the weaker security property of $(t, \varepsilon_{one-way})$-one-wayness, we only can obtain a non-tight security reduction [9] with respect to $\varepsilon_{one-way}$. As we will show now, a tight security reduction from one-wayness can be obtained by padding M with one random bit b, using a reduction technique by Katz and Wang [16]. We now define an alternative signature scheme $\mathsf{SIG\text{-}MR}_{\mathrm{ow}}^{\mathcal{F}}$ with message space $\mathcal{M}(n) = \{0,1\}^{m-1}$ which can be proved tightly secure from one-wayness of TDP.

```
Algorithm $\mathsf{Sign}_{sk}(M \in \{0,1\}^{m-1})$
    $b(M) \leftarrow \{0,1\}$
    $y := \mathcal{F}(b \| M) \in \{0,1\}^k$
    Return $\tau = \mathsf{f}_{td}^{-1}(y) \in \{0,1\}^k$
```

```
Algorithm $\mathsf{Recover}_{pk}(\tau \in \{0,1\}^k)$
    $y = \mathsf{f}_{ek}(z)$
    If $\mathcal{F}^{-1}(y) = \perp$ then return $\perp$
    Else compute $b \| M = \mathcal{F}^{-1}(y)$
    Return $M$
```

It is furthermore enforced that Sign always uses the same random bit $b = b(M)$ for message M. (E.g., by defining $b = \mathsf{PRF}_K(M)$.) Note that $\mathsf{SIG\text{-}MR}^{\mathcal{F}}_{\mathrm{ow}}$ has $k - m + 1$ bits redundancy.

The proof of the following theorem is postponed to Appendix C.

Theorem 3.2 *Suppose* TDP *is homomorphic and* $(t, \varepsilon_{one-way})$*-one-way and* $\mathcal{F}$ *is a random injective function from* $\{0,1\}^m$ *to* $\{0,1\}^k$*. Then* $\mathsf{SIG\text{-}MR}^{\mathcal{F}}_{ow}$ *is* $(t, q_s, q_f, 2\varepsilon_{one-way} + \frac{q_f}{2^{k-m}})$ *secure.*

4 Pub-Indifferentiable Constructions based on Feistel Networks

4.1 The two round Feistel network

Consider the two-round construction $\mathsf{C}^{\mathcal{H}}_{2f} : \mathbb{Z}_\mu \to \mathbb{Z}_\mu \times \mathbb{Z}_\rho$ Figure 1 (left) which is derived from an unbalanced two-round Feistel network ϕ_{2f} instantiated with random oracles $\mathcal{H}_1 : \mathbb{Z}_\mu \to \mathbb{Z}_\rho, \mathcal{H}_2 : \mathbb{Z}_\rho \to \mathbb{Z}_\mu$

$$\phi_{2f}(x, v) = (x + \mathcal{H}_2(\mathcal{H}_1(x) + v), \mathcal{H}_1(x) + v) \qquad \phi^{-1}_{2f}(w, y) = (w - \mathcal{H}_2(y), y - \mathcal{H}_1(w - \mathcal{H}_2(y))$$

$$\text{as} \qquad \mathsf{C}^{\mathcal{H}}_{2f}(x) = \phi_{2f}(x, 0) \qquad {\mathsf{C}^{\mathcal{H}}_{2f}}^{-1}(w, y) = \begin{cases} x & \text{if } \phi^{-1}_{2f}(w, y) = (x, 0) \\ \perp & \text{otherwise} \end{cases}$$

This will serve as an example of a simple indifferentiability proof and to prepare for our four round Feistel network in the next section

Theorem 4.1 (pub-indifferentiability of C_{2f}, implicit in [5]) $\mathsf{C}^{\mathcal{H}}_{2f}$ *as illustrated in Figure 1 (left) is* $(q_{\mathsf{D}}, q_{\mathsf{S}}, \varepsilon_{sim}, t_{sim})$*-pub-indifferentiable from* $\mathcal{F}$ *(cf. Def. 2.1) where*

$$q_{\mathsf{S}} = q_{\mathsf{D}} \qquad t_{sim} = q_{\mathsf{D}} \cdot polylog(\mu) \qquad \varepsilon_{sim} = q^2_{\mathsf{D}}/\rho,$$

More precisely, we can set $t_{sim} = O(q_{\mathsf{D}} \log(q_{\mathsf{D}}) \log(\mu))$ *using that the cost per (find or insert) operation on a sorted list with* $\leq q_{\mathsf{D}}$ *elements of size* $\log(\mu)$ *bits is* $O(\log(q_{\mathsf{D}}) \log(\mu))$*.*

The proof of Theorem 4.1 is postponed to Appendix A. There we also formally show that a combination with Theorems 3.1/3.2 and Theorem 2.3 leads to the overhead of $\mathsf{O}(n, q_h, q_s) = n + \log(q_h)$ bits for the two schemes $\mathsf{SIG\text{-}MR}^{\mathsf{C}^{\mathcal{H}}_{2f}}[\mathsf{RSA}]$ and $\mathsf{SIG\text{-}MR}^{\mathsf{C}^{\mathcal{H}}_{2f}}_{\mathrm{ow}}[\mathsf{RSA}]$ in the random oracle model.

4.2 The four round Feistel network

We will prove the following theorem which bounds the pub-indifferentiability of our main construction $\mathsf{C}^{\mathcal{H}}_{4F}$ as illustrated in Figure 1 (right) in terms of the variable $Q(\mu, q)$ (which we discussed in the introduction, and will define formally in Section 5.1).

Theorem 4.2 (pub-indifferentiability of $\mathsf{C}^{\mathcal{H}}_{4F}$) $\mathsf{C}^{\mathcal{H}}_{4F}$ *as illustrated in Figure 1 (right) is* $(q_{\mathsf{D}}, q_{\mathsf{S}}, \varepsilon_{sim}, t_{sim})$*-pub-indifferentiable from* $\mathcal{F}$ *(cf. Def. 2.1) where*

$$q_{\mathsf{S}} \leq q_{\mathsf{D}} \log(\rho) \qquad t_{sim} = q_{\mathsf{S}} \cdot polylog(\mu) \qquad \varepsilon_{sim} = \frac{2\mathsf{E}[Q(\mu, q_{\mathsf{D}})]}{\rho} + \frac{2q^4_{\mathsf{D}}}{\mu} + \frac{2q^2_{\mathsf{D}}}{\rho^2} \cdot \left(\frac{\log(\rho)}{\log(\rho/q_{\mathsf{D}})}\right)^2. \quad (3)$$

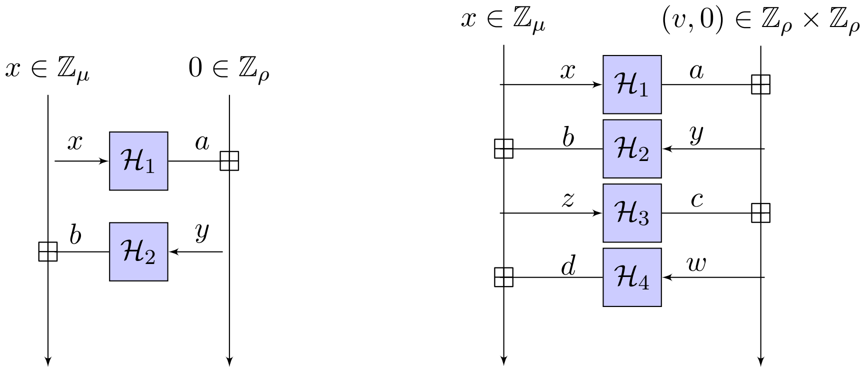

Figure 1: **(left)** Two round Feistel network $\phi_{2f} : \mathbb{Z}_\mu \times \mathbb{Z}_\rho \to \mathbb{Z}_\mu \times \mathbb{Z}_\rho$, the construction $\mathsf{C}_{2f}^{\mathcal{H}} : \mathbb{Z}_\mu \to \mathbb{Z}_\mu \times \mathbb{Z}_\rho$ of a random invertible function $\mathcal{F}$ from a random oracle $\mathcal{H}$ is derived from ϕ_{2f} by setting the right part to 0, i.e. $\mathsf{C}_{2f}^{\mathcal{H}}(x) = \phi_{2f}(x, 0)$. $\boxplus$ denotes component-wise addition in the respective domains. **(right)** Four round Feistel network ϕ_{4F}, our main construction is derived from it as $\mathsf{C}_{4F}^{\mathcal{H}}(x, v) = \phi_{4F}(x, v, 0)$.

Given Theorem 4.2 we will now compute the concrete overhead of $\mathsf{SIG\text{-}MR}^{\mathsf{C}_{4F}^{\mathcal{H}}}[\mathsf{RSA}]$ and $\mathsf{SIG\text{-}MR}_{\mathrm{ow}}^{\mathsf{C}_{4F}^{\mathcal{H}}}[\mathsf{RSA}]$. Let $N = pq$ be the RSA modulus with $k = \log N$ and recall that $\log \mu = k - \log \rho$, where $\log \rho$ is the redundancy of the scheme. It is easy to verify that for all practically relevant values, the first term in ε_{sim} in (3) is the dominating one. With the same argument as in the case of two rounds, by Theorems 3.1/3.2 and Theorem 2.3 the overhead for n-bit security can (up to a small additive constant) be computed as

$$\mathsf{O}(n, q_h, q_s) = n + \log \mathsf{E}[Q(\mu, q_h)] - \log q_h. \tag{4}$$

In order to bound $\mathsf{E}[Q(\mu, q_h)]$ we assume $n \leq \log \rho \leq 1.25n$ and hence $\log \mu = \log N - 2 \log \rho \geq \log N - 2(1.25n)$. The following table summarizes the overhead $\mathsf{O}(n, q_s, q_h)$ for $n = 80$ bits security using (4) and the bounds on $\Pr[Q(\mu, q_h) \geq q_h 2^s]$ from Theorem 6.2 in Section 6. We use $\log N \in \{1024, 2048\}$ as bit-length of RSA and $\log q_h \in \{60, 80\}$ as upper bound on the random oracle queries.

$\log N$	$\log q_h$	$t = \log \mu$	s	$\Pr[Q(\mu, q_h) \geq q_h 2^s]$	$\mathsf{O}(n, q_h, q_s)$
1024	80	824	17	2^{-427} $(l = 8)$	≈ 97
1024	60	824	13	2^{-430} $(l = 10)$	≈ 93
2048	80	1848	12	2^{-230} $(l = 16)$	≈ 92
2048	60	1848	10	2^{-92} $(l = 18)$	≈ 90

5 Indifferentiability Proof for Four Round Feistel

The proof of Theorem 4.2 is organized as follows. In Section 5.1 we formally define $Q(q, \mu)$. In Section 5.2 we prove a technical lemma (Lemma 5.2) which informally bounds the advantage of any q query adversary in making a fresh query x to $\mathcal{H}_1$ (variables as in Figure 2, right) such that for some v, in the evaluation of $\phi_{4F}(x, v, 0)$ the input z to $\mathcal{H}_3$ has already been queried. Next, in

Lemma 5.3 we prove the pub-indifferentiability of $\mathsf{C}^{\mathcal{H}}_{4F}(\cdot) = \phi_{4F}(\cdot,\cdot,0)$ as claimed in Theorem 4.2. Finally, in Section 6 we prove a $q^{1+o(1)}$ upper bound on $Q(\mu, q)$.

5.1 Density of Subgraphs of Random Cayley Graphs

For $\mu, q \in \mathbb{N}$ let $\mathcal{B}$ be a subset of $\mathbb{Z}_\mu$ of size q, we define the value

$$Q(\mu, q, \mathcal{B}) = \max_{\mathcal{X},\mathcal{Z}\subset\mathbb{Z}_\mu, |\mathcal{X}|=|\mathcal{Z}|=q} |\{(x, z) \mid x \in \mathcal{X}, z \in \mathcal{Z}, z - x \in \mathcal{B}\}| \tag{5}$$

We will be interested in the random variable $Q(\mu, q, \mathcal{B})$ where $\mathcal{B}$ is a randomly chosen q element subset of $\mathbb{Z}_\mu$, we denote this variable by $Q(\mu, q)$.

It will be convenient to think of $Q(\mu, q)$ in terms of random Cayley graphs as we will explain now. For $\mathcal{B} \subset \mathbb{Z}_\mu, |\mathcal{B}| = q$ we denote with $\mathcal{C}(\mu, q, \mathcal{B})$ the bipartite graph with μ vertices on each side which we identify with the elements of $\mathbb{Z}_\mu$. The edge set is $e(\mathcal{C}(\mu, q, \mathcal{B})) = \{(x, z) \ : \ z - x \in \mathcal{B}\}$, that is, (x, z) is an edge if $x + b = z$ for some $b \in \mathcal{B}$. With $\mathcal{C}(\mu, q)$ we denote the random graph $\mathcal{C}(\mu, q, \mathcal{B})$ for a random $\mathcal{B} \subset \mathbb{Z}_\mu, |\mathcal{B}| = q$.

With this notion $Q(\mu, q, \mathcal{B})$ is the maximum number of edges in any subgraph of $\mathcal{C}(\mu, q, \mathcal{B})$ with q vertices on each side. Trivial lower and upper bounds on $Q(\mu, q, \mathcal{B})$ are

$$\forall \mathcal{B} \subset \mathbb{Z}_q, |\mathcal{B}| = q \ : \ 2q - 1 \le Q(\mu, q, \mathcal{B}) \le 2q^2.$$

In the proposition below we observe that known results on the edge density of graphs without 4-cycles already give us an $q^{1.5}$ upper bound on the expected value $\mathsf{E}[Q(\mu, q)]$. In Section 6 we will prove an upper bound of $q^{1+o(1)}$, thus almost matching the lower bound.

Proposition 5.1 *If* $\mu \ge q^5$ *then* $\mathsf{E}[Q(\mu, q)] \le q^{1.5} + 3q$.

Proof. We first observe that $\mathcal{C}(\mu, q, \mathcal{B}) \leftarrow \mathcal{C}(\mu, q)$ has a 4-cycle with probability at most

$$\Pr[G \leftarrow \mathcal{C}(\mu, q) \ : \ G \text{ has a 4-cycle}] \le q^4/\mu.$$

The proposition now follows from a result by Naor and Verstraëte [19] who show that a bipartite graph with q vertices on each side that does not contain a 4-cycle has at most $q^{1.5} + 2q$ edges. With probability at most $q^4/\mu \le 1/q$ we have a cycle, in which case we use the trivial q^2 upper bound which adds another $q = q^2/q$ to the expected value. ■

5.2 A Game on Three Round Feistel

In this section we describe a game, where an attacker A can query three randomly chosen functions $\mathcal{H}_1, \mathcal{H}_3 : \mathbb{Z}_\mu \to \mathbb{Z}_\rho, \mathcal{H}_2 : \mathbb{Z}_\rho \to \mathbb{Z}_\mu$, which we think of as round functions of a Feistel network ϕ_{3f} as illustrated in Figure 2 (left). Informally, the adversary wins if she makes a fresh query x to $\mathcal{H}_1$, such that the input z to $\mathcal{H}_3$ in the evaluation of $\phi_{3f}(x, 0)$ has already been queried. We will call this game the z-collision game and prove (in Lemma 5.2 below) an $\mathsf{E}[Q(\mu, q)]/\rho$ upper bound for any q-query adversary for the z-collision game.

Next, we will show that the same bound on the winning advantage holds for a similar game on the Feistel-network ϕ_{3F} as illustrated in Figure 2 (right), where we have an extra $\mathbb{Z}_\rho$ domain on the right side. Here we say the adversary wins if she makes a query $x \in \mathbb{Z}_\mu$ such that there exists a $v \in \mathbb{Z}_\rho$ such that in the evaluation of $\phi_{3F}(x, v, 0)$ the input z is not fresh.

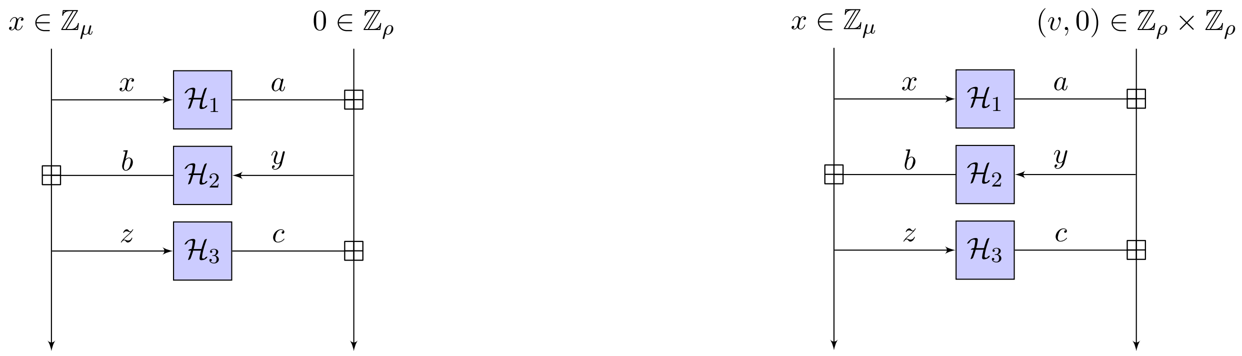

Figure 2: **(left)** An unbalanced three-round Feistel network ϕ_{3f} over $\mathbb{Z}_\mu \times \mathbb{Z}_\rho$. **(right)** The three-round Feistel network ϕ_{3F} with an extra $\mathbb{Z}_\rho$ domain on the right side. This permutations define constructions of invertible functions $\mathsf{C}^{\mathcal{H}}_{3f}(x) = \phi_{3f}(x, 0)$ and $\mathsf{C}^{\mathcal{H}}_{3F}(x, v) = \phi_{3F}(x, v, 0)$ by fixing the rightmost $\mathbb{Z}_\rho$ part of the input to 0.

We will later use the bound on the winning advantage for the z-collision game on ϕ_{3F} as key technical lemma to prove the pub-indifferentiability of $\mathsf{C}^{\mathcal{H}}_{4F}$. As in the pub-indifferentiability proof the random functions are defined via lazy sampling (done by the simulator), we will already use lazy sampling in the proof of our upper bound for the z-collision game. More precisely, we will consider functions $\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2, \hat{\mathcal{H}}_3$ which initially are undefined on all inputs. The sets $\mathcal{X}, \mathcal{Z} \subset \mathbb{Z}_\mu, \mathcal{Y} \subset \mathbb{Z}_\rho$ denote the inputs to $\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_3$ and $\hat{\mathcal{H}}_2$ on which the outputs have been defined, initially $\mathcal{X}, \mathcal{Y}, \mathcal{Z} = \emptyset$. Moreover we initialize a variable $\mathsf{FAIL} := 0$, the adversary wins the game if at the end of the game $\mathsf{FAIL} > 0$. We will assume that $\hat{\mathcal{H}}_2$ is a random *injective* function as this will make the proofs a bit cleaner. One cannot distinguish a random from a random injective function with range $\mathbb{Z}_\mu$ making q queries with advantage better than q^2/μ. As we will later set $\mu \geq \rho^3$, this term will be dominated by other terms $\Omega(q/\rho)$ and thus we will simply ignore it.

The z-collision game on ϕ_{3f}. Consider an adversary A who can make queries to the three functions (at most most q to each) which are answered as follows:

$\hat{\mathcal{H}}_2$ **query** $y \in \mathbb{Z}_\rho$: If $y \notin \mathcal{Y}$ sample $b \leftarrow \mathbb{Z}_\mu \setminus \mathcal{B}$ and set $\hat{\mathcal{H}}_2(y) := b$.[5] Output $\hat{\mathcal{H}}_2(y)$.

$\hat{\mathcal{H}}_3$ **query** $z \in \mathbb{Z}_\mu$: If $z \notin \mathcal{Z}$ sample $c \leftarrow \mathbb{Z}_\rho$ and set $\hat{\mathcal{H}}_3(z) := c$. Output $\hat{\mathcal{H}}_3(z)$.

$\hat{\mathcal{H}}_1$ **query** $x \in \mathbb{Z}_\mu$: If $x \notin \mathcal{X}$ then

1. sample $a \leftarrow \mathbb{Z}_\rho$ and set $\hat{\mathcal{H}}_1(x) := a$.
2. If $x + \hat{\mathcal{H}}_2(a) \in \mathcal{Z}$ then $\mathsf{FAIL} := \mathsf{FAIL} + 1$.

Output $\hat{\mathcal{H}}_1(x)$.

We will now upper bound the success probability of any adversary making at most q queries to each of the three function to win the z-collision game (i.e. achieve $\mathsf{FAIL} > 0$).

Trivial lower and upper bounds on the winning advantage of the z-collision game are q/ρ and q^2/ρ. We will now give an upper bound on the advantage in terms of $Q(\mu, q)$ introduced in Section 5.1.

[5] Note that we sample the output from $\mathbb{Z}_\mu \setminus \mathcal{B}$ because we want $\hat{\mathcal{H}}_2$ to behave like a random *injective* function.

Lemma 5.2 *The advantage of any q-query adversary A in winning the z-collision game on ϕ_{3f} (i.e. force $\mathsf{FAIL} > 0$) is at most*

$$\Pr[\mathsf{FAIL} > 0] \leq \mathsf{E}[Q(\mu, q)]/\rho$$

Proof. We will only sketch the proof, as this lemma is used to get an intuition for Lemma 5.3 below.

Consider the queries $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ made by A, and recall that $\mathcal{B} = \hat{\mathcal{H}}_2(\mathcal{Y})$. Consider the subgraph with bipartition $(\mathcal{X}, \mathcal{Y})$ of the bipartite Cayley graph $\mathcal{C}(\mu, q, \mathcal{B})$. Now for any $x \in \mathcal{X}$, let $\mathcal{B}_x = \{b \in \mathcal{B} \,:\, x + b \in \mathcal{Z}\}$ be the number of edges from x to $\mathcal{Z}$. The x query will increase FAIL iff the $a \leftarrow \mathbb{Z}_\rho$ we sampled is a preimage of some $b \in \mathcal{B}_x$ (i.e. $\hat{\mathcal{H}}_2(a) = b \in \mathcal{B}_x$), this probability is $|\mathcal{B}_x|/\rho$. Summing over all $x \in \mathcal{X}$ we get $\mathsf{E}[\mathsf{FAIL}] = \sum_{x \in \mathcal{X}} |\mathcal{B}_x|/\rho \leq Q(\mu, q, \mathcal{B})/\rho$. This is the expectation after $\mathcal{B}$ has been fixed, as $\mathcal{B}$ is a random subset

$$\mathsf{E}[\mathsf{FAIL}] \leq \mathsf{E}[Q(\mu, q)]/\rho. \tag{6}$$

Finally, note that $\mathsf{E}[\mathsf{FAIL}] \geq \Pr[\mathsf{FAIL} = 1]$ as FAIL is an integer ≥ 0.

Let us mention that eq.(6) is tight and equality is achieved by the following attack strategy. First make any q queries to $\hat{\mathcal{H}}_2$ which gives us a random set $\mathcal{B}$ (as $\hat{\mathcal{H}}_2$ is injective, $|\mathcal{B}| = |\mathcal{Y}| = q$). Next, identify the sets $\mathcal{X}, \mathcal{Z}$ of size q s.t. the $(\mathcal{X}, \mathcal{Z})$ subgraph of $\mathcal{C}(\mu, q, \mathcal{B})$ has the most edges, by definition this number is $Q(\mu, q, \mathcal{B})$. Make all $\mathcal{Z}$ queries to $\hat{\mathcal{H}}_3$, followed by all $\mathcal{X}$ queries to $\hat{\mathcal{H}}_1$. ■

Ultimately, our goal is to prove pub-indifferentiability from $\mathcal{F}$. The above lemma is a good start as it tells us that for the evaluation function $\mathsf{C}^{\mathcal{H}}_{3f}(x) = \phi_{3f}(x, 0)$ with probability $1 - \mathsf{E}[Q(\mu, q)]/\rho$ the following holds: whenever a q-query adversary makes an x query to $\hat{\mathcal{H}}_1$, the resulting z input to $\hat{\mathcal{H}}_3$ will be "fresh", and thus we will be able to program the output $c := \hat{\mathcal{H}}_3(z)$ such that it is consistent with $\mathcal{F}(x)$. By adding one more round to the Feistel network we will be able to program also the left $\mathbb{Z}_\mu$ part of the input. Unfortunately, this will only work as long as the inputs to this fourth function are fresh. As its inputs are over $\mathbb{Z}_\rho$, there is a $\Theta(q^2/\rho)$ chance we have a collision on these inputs and will not be able to program after all. Summing up, we are no better than the q^2/ρ bound we already got for the two round Feistel in Theorem 4.1. To overcome this problem, we will simply increase the domain on the right side of the Feistel to $\mathbb{Z}_\rho \times \mathbb{Z}_\rho$, but in order to not increase the redundancy space, this extra $\mathbb{Z}_\rho$ space is used for the message, not redundancy. We will now show that the z-collision game on this new $\mathsf{C}^{\mathcal{H}}_{3F}(x, v) = \phi_{3F}(x, v, 0)$ padding scheme is still hard.

The z-collision game on ϕ_{3F}.

$\hat{\mathcal{H}}_2$ **query** $y \in \mathbb{Z}_\rho \times \mathbb{Z}_\rho$: If $y \notin \mathcal{Y}$ sample $b \leftarrow \mathbb{Z}_\mu \setminus \mathcal{B}$ and set $\hat{\mathcal{H}}_2(y) := b$. Output $\hat{\mathcal{H}}_2(y)$.

$\hat{\mathcal{H}}_3$ **query** $z \in \mathbb{Z}_\mu$: If $z \notin \mathcal{Z}$ sample $c \leftarrow \mathbb{Z}_\rho \times \mathbb{Z}_\rho$ and set $\hat{\mathcal{H}}_3(z) := c$. Output $\hat{\mathcal{H}}_3(z)$.

$\hat{\mathcal{H}}_1$ **query** $x \in \mathbb{Z}_\mu$: If $x \notin \mathcal{X}$ then

1. sample $(a_0, a_1) \leftarrow \mathbb{Z}_\rho \times \mathbb{Z}_\rho$ and set $\hat{\mathcal{H}}_1(x) := (a_0, a_1)$.
2. For all $(y_0, y_1) \in \mathcal{Y}$ where $a_1 = y_1$ and $x + \hat{\mathcal{H}}_2(y_0, y_1) \in \mathcal{Z}$ set $\mathsf{FAIL} := \mathsf{FAIL} + 1$.

Output $\hat{\mathcal{H}}_1(x)$.

Lemma 5.3 *The advantage of any q-query adversary* A *in winning the z-collision game on ϕ_{3F} (i.e. force* $\mathsf{FAIL} > 0$*) is at most*

$$\Pr[\mathsf{FAIL} > 0] \leq \mathsf{E}[Q(\mu, q)]/\rho$$

Proof. Consider the queries $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ made by A, and recall that $\mathcal{B} = \hat{\mathcal{H}}_2(\mathcal{Y})$. For any query $x \in \mathcal{X}$, let $\mathcal{B}_x = \{b \in \mathcal{B} \ : \ x + b \in \mathcal{Z}\}$. Note that $|\mathcal{B}_x|$ is the number of edges from x to $\mathcal{Z}$ in the Cayley graph $\mathcal{C}(\mu, q, \mathcal{B})$. The expected value by which this x query will increase FAIL is

$$|\mathcal{B}_x|/\rho.$$

To see this, note that for any $b \in \mathcal{B}$, the probability that FAIL will increase because of this b is 0 if $x + b \notin \mathcal{Z}$ (equivalently $b \notin \mathcal{B}_x$), and $1/\rho$ otherwise. More precisely, FAIL will increase if the (a_0, a_1) we sample and the preimage (y_0, y_1) of b (i.e. $\hat{\mathcal{H}}_2(y_0, y_1) = b$) satisfy $a_1 = y_1$, and as a_1 is uniform, $\Pr[a_1 = y_1] = 1/\rho$. By definition $\sum_{x \in \mathcal{X}} |\mathcal{B}_x|$ is the number of edges of the $(\mathcal{X}, \mathcal{Z})$ subgraph of $\mathcal{C}(\mu, q, \mathcal{B})$, we have for a fixed $\mathcal{B}$

$$\mathsf{E}[\mathsf{FAIL}] \leq Q(\mu, q, \mathcal{B})/\rho$$

and as $\mathcal{B}$ is sampled uniformly at random

$$\mathsf{E}[\mathsf{FAIL}] = \mathsf{E}[Q(\mu, q)]/\rho.$$

5.3 Proof of Theorem 4.2

Let $q_{\mathsf{D}} = q_{\mathcal{F}} + q_{\mathcal{H}}$, where $q_{\mathcal{F}}$ and $q_{\mathcal{H}}$ denotes the number of queries D makes to its first and second oracle, respectively. As in the proof of Theorem 4.1 for the two-round Feistel our simulator $\mathsf{S}^{\mathcal{F}}$ (given access to a random function $\mathcal{F} : \mathbb{Z}_\mu \times \mathbb{Z}_\rho \to \mathbb{Z}_\mu \times \mathbb{Z}_\rho \times \mathbb{Z}_\rho$) will define fake random oracles $\hat{\mathcal{H}}_i, i = 1, \ldots, 4$ by lazy sampling. $\hat{\mathcal{H}}_1(x) = \Diamond$ denotes $\hat{\mathcal{H}}_1$ is undefined on input x. The sets $\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{W}$ define the inputs on which $\hat{\mathcal{H}}_1, \ldots, \hat{\mathcal{H}}_4$ have already been defined. The sets $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$ are the corresponding outputs, e.g. $\mathcal{A} = \hat{\mathcal{H}}_1(\mathcal{X})$. The simulator also initializes variables $\mathsf{FAIL}_i := 0$ for $i \in \{0, 1, 2, 3, 4\}$ which will only be used in the proof. Informally, whenever the simulator cannot define the $\hat{\mathcal{H}}_i$'s consistently, it sets $\mathsf{FAIL}_j := 1$ (for which j depends on the reason why it fails) and aborts, by this we mean it just stops giving any more outputs.

Below we define how $\mathsf{S}^{\mathcal{F}}$ answers the $q_{\mathcal{H}}$ queries to $\hat{\mathcal{H}}_i, i \in \{1, 2, 3, 4\}$ and updates its state on the $q_{\mathcal{F}}$ queries to $\mathcal{F}$, but let us first give some intuition.

One important property of $\mathsf{S}^{\mathcal{F}}$ is the fact that whenever it assign a value $\hat{\mathcal{H}}_i(\alpha) := \beta$, then β is uniformly random and independent of the variables we've seen so far (in some cases $\mathsf{S}^{\mathcal{F}}$ samples β at random itself, sometimes the randomness comes from $\mathcal{F}$.) This property is required so we can later argue that distinguishing oracles $\mathcal{F}, \mathsf{S}^{\mathcal{F}}$ from $\mathsf{C}_{4F}^{\mathcal{H}}, \mathcal{H}$ is upper bounded by the probability that $\mathsf{S}^{\mathcal{F}}$ fails.

Another important invariant is that whenever $\mathsf{C}_{4F}^{\hat{\mathcal{H}}}(x, v)$ is defined (i.e. the $\hat{\mathcal{H}}_i$ are defined on all inputs required to evaluate $\phi_{4F}(x, v, 0)$ with round functions $\hat{\mathcal{H}}_i$) and the simulator did not yet fail, then the output is consistent with $\mathcal{F}$, i.e. $\mathcal{F}(x, v) = \mathsf{C}_{4F}^{\hat{\mathcal{H}}}(x, v)$. That this invariant holds can be seen as follows. Assume the evaluation of $\mathsf{C}_{4F}^{\hat{\mathcal{H}}}(x, v)$ is defined and consider the inputs x, y, z to the first three round functions. By definition of the simulator, if the last of these inputs to be defined was x (i.e. we assigned a value to $\hat{\mathcal{H}}_1(x)$ at a point where $\hat{\mathcal{H}}_2(y), \hat{\mathcal{H}}_3(z)$ were already defined) or y, then the simulator failed (we will have set $\mathsf{FAIL}_0 = 1$ if the query was y or $\mathsf{FAIL}_1 = 1$ if it was x). If the last value to be defined was $\hat{\mathcal{H}}_3(z)$, then by definition of the simulator also $\hat{\mathcal{H}}_4$ will be programmed such that the entire evaluation of $\mathsf{C}_{4F}^{\hat{\mathcal{H}}}(x, v)$ is defined and outputs $\mathcal{F}(x, v)$. The variable FAIL_2 will be set to 1 if we failed to program $\hat{\mathcal{H}}_4$ because it was already defined on the required input. The variable FAIL_3 will be set to 1 if there exist two different query pairs $(x, y), (x', y')$ that are consistent with z, as here we won't be able to program $\hat{\mathcal{H}}_3(z)$ to be consistent with both.

If D makes a $\mathcal{F}(x, v)$ query, our simulator makes sure that $\mathsf{C}_{4F}^{\hat{\mathcal{H}}}(x, v)$ is defined and equals $\mathcal{F}(x, v)$ by doing the same it would do if the adversary queries $\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2, \hat{\mathcal{H}}_3$ (in this order) on the inputs required to evaluate $\mathsf{C}_{4F}^{\hat{\mathcal{H}}}(x, v)$.

We now define how $\mathsf{S}^{\mathcal{F}}$ answers the $\leq q_{\mathcal{H}}$ queries to $\hat{\mathcal{H}}_i, i \in \{1, 2, 3, 4\}$ and updates its state on the $\leq q_{\mathcal{F}}$ queries to $\mathcal{F}, \mathcal{F}^{-1}$ as follows:

- **$\hat{\mathcal{H}}_2$ query** $y \in \mathbb{Z}_\rho \times \mathbb{Z}_\rho$**:** If $y \notin \mathcal{Y}$ sample $b \leftarrow \mathbb{Z}_\mu$ and set $\hat{\mathcal{H}}_2(y) := b$. Output $\hat{\mathcal{H}}_2(y)$.
  If $x + b = z$ for some $x \in \mathcal{X}, z \in \mathcal{Z}$ set $\mathsf{FAIL}_0 := 1$ and abort.
- **$\hat{\mathcal{H}}_4$ query** $w \in \mathbb{Z}_\rho \times \mathbb{Z}_\rho$**:** If $w \notin \mathcal{W}$ sample $d \leftarrow \mathbb{Z}_\mu$ and set $\hat{\mathcal{H}}_4(w) := d$. Output $\hat{\mathcal{H}}_4(w)$.
- **$\hat{\mathcal{H}}_3$ query** $z \in \mathbb{Z}_\mu$**:** If $z \notin \mathcal{Z}$ and there exist *two* distinct pairs of messages $x, (y_0, y_1)$ and $x', (y_0', y_1')$ in $\mathcal{X} \times \mathcal{Y}$ s.t. with $(a_0, a_1) = \hat{\mathcal{H}}_1(x), (a_0', a_1') = \hat{\mathcal{H}}_1(x')$

  1. $a_1 = y_1$ and $a_1' = y_1'$
  2. $z = \hat{\mathcal{H}}_2(y_0, y_1) + x = \hat{\mathcal{H}}_2(y_0', y_1') + x'$

  then set $\mathsf{FAIL}_3 := 1$ and abort. (Here we fail as z appears in two queries $(x, y_0 - a_0, 0)$ and $(x', y_0' - a_0', 0)$ to $\mathsf{C}_{4F}^{\hat{\mathcal{H}}}$, and we won't be able to program $\hat{\mathcal{H}}_3(z)$ to be consistent with both).
  Otherwise, if $z \notin \mathcal{Z}$ and there exists *exactly one* pair $x, (y_0, y_1) \in \mathcal{X} \times \mathcal{Y}$ s.t. $a_1 = y_1$ and $z = \hat{\mathcal{H}}_2(y_0, y_1) + x$, try to program $\hat{\mathcal{H}}_3, \hat{\mathcal{H}}_4$ s.t. $\mathsf{C}_{4F}^{\hat{\mathcal{H}}}(x, y_0 - a_0, 0) = \mathcal{F}(x, y_0 - a_0)$ as follows:
  1. Query $(f_0, f_1, f_2) \leftarrow \mathcal{F}(x, y_0 - a_0)$
  2. $\hat{\mathcal{H}}_3(z) := (f_1 - y_0, f_2 - y_1)$ (program $\hat{\mathcal{H}}_3(z)$)
  3. If $(f_1, f_2) \in \mathcal{W}$ set $\mathsf{FAIL}_2 := 1$ and abort (fail due to collision on w value)
  4. $\hat{\mathcal{H}}_4(f_1, f_2) := f_0 - z$ (program $\hat{\mathcal{H}}_4(w)$)

  Otherwise, if $z \notin \mathcal{Z}$ and no such pair exists, sample $c \leftarrow \mathbb{Z}_\rho \times \mathbb{Z}_\rho$ and set $\hat{\mathcal{H}}_3(z) := c$.
  Output $\hat{\mathcal{H}}_3(z)$.
- **$\hat{\mathcal{H}}_1$ query** $x \in \mathbb{Z}_\mu$**:** If $x \in \mathcal{X}$ output $\hat{\mathcal{H}}_1(x)$.
  Otherwise, if $x \notin \mathcal{X}$ sample $(a_0, a_1) \leftarrow \mathbb{Z}_\rho \times \mathbb{Z}_\rho$, set $\hat{\mathcal{H}}_1(x) := (a_0, a_1)$, output $\hat{\mathcal{H}}_1(x)$.
  Then for all $(y_0, y_1) \in \mathcal{Y}$ where $a_1 = y_1$ try to program $\hat{\mathcal{H}}_3, \hat{\mathcal{H}}_4$ s.t. $\mathsf{C}_{4F}^{\hat{\mathcal{H}}}(x, y_0 - a_0, 0) = \mathcal{F}(x, y_0 - a_0)$ as follows:
  1. Query $(f_0, f_1, f_2) \leftarrow \mathcal{F}(x, y_0 - a_0)$
  2. If $\hat{\mathcal{H}}_2(y_0, y_1) + x \in \mathcal{Z}$ set $\mathsf{FAIL}_1 := 1$ and abort (fail due to collision on z value)
  3. Now make a z query to $\hat{\mathcal{H}}_3(z)$ as described above.[6]
- **$\mathcal{F}$ query** $(x, v) \in \mathbb{Z}_\mu \times \mathbb{Z}_\rho$**:** Here the simulator doesn't have to output anything, but it updates its state trying to program the $\hat{\mathcal{H}}_i$ s.t. $\mathsf{C}_{4F}^{\hat{\mathcal{H}}}(x, v, 0) = \mathcal{F}(x, v)$ as follows: query $(a_0, a_1) \leftarrow \hat{\mathcal{H}}_1(x)$ (as described above), then query $b \leftarrow \hat{\mathcal{H}}_2(v + a_0, a_1)$ and finally $(c_0, c_1) \leftarrow \hat{\mathcal{H}}_3(x + b)$ (note that we don't have to query $\hat{\mathcal{H}}_4$ explicitly as the $\hat{\mathcal{H}}_3$ query already programs $\hat{\mathcal{H}}_4$.)
- **$\mathcal{F}^{-1}$ query** $u \in \mathbb{Z}_\mu \times \mathbb{Z}_\rho \times \mathbb{Z}_\rho$**:** query $(x, v) \leftarrow \mathcal{F}^{-1}(u)$, if $(x, v) \neq \perp$ and $\hat{\mathcal{H}}_1(x) \neq \Diamond$ (i.e., the adversary inverted $\mathcal{F}$ on "fresh" value) set $\mathsf{FAIL}_4 := 1$.

[6] If we get into the "exactly one case" in processing the $\hat{\mathcal{H}}_3(z)$ query, then $\hat{\mathcal{H}}_3, \hat{\mathcal{H}}_4$ will be programmed such that $\mathsf{C}_{4F}^{\hat{\mathcal{H}}}(x, y_0 - a_0, 0) = \mathcal{F}(x, y_0 - a_0)$. The only other option is the "two" case, where we'll fail with $\mathsf{FAIL}_3 = 1$.

We will bound the probability that $\mathsf{FAIL}_i = 1$ for $i \in \{0,1,2,3,4\}$, and then can use standard arguments to show that the advantage of distinguishing $\mathcal{F}, \mathsf{S}^{\mathcal{F}}$ from $\mathsf{C}^{\mathcal{H}}, \mathcal{H}$ is upper bounded by the sum of these probabilities.

We first consider FAIL_4, which is the probability that D by manages to make a query $u \in \mathbb{Z}_\mu \times \mathbb{Z}_\rho \times \mathbb{Z}_\rho$ which hits one of the $\mathbb{Z}_\mu \times \mathbb{Z}_\rho$ elements in the (uniformly random) range of $\mathcal{F}$. The probability of u falling in this set is at most $1/\rho$, taking the union bound we get

$$\Pr[\mathsf{FAIL}_4 = 1] \leq q_{\mathsf{D}}/\rho$$

We will now bound the sizes of $\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{W}$ which will give the same upper bounds on $\mathcal{A}, \mathcal{B}, \mathcal{C}$ and $\mathcal{D}$, respectively. As $\mathcal{X}$ and $\mathcal{Y}$ can only increase by at most 1 on a $\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2$ or $\mathcal{F}$ query, we have

$$|\mathcal{X}| \ , \ |\mathcal{Y}| \leq q_{\mathsf{D}} \tag{7}$$

Bounding $|\mathcal{Z}|$ and $|\mathcal{W}|$ is less trivial, as an $\hat{\mathcal{H}}_1$ query can increase $|\mathcal{Z}|$ and $|\mathcal{W}|$ by as much as $|\mathcal{Y}|$, and thus we only get a q_{D}^2 upper bound. Fortunately the *expected* increase of $|\mathcal{Z}|, |\mathcal{W}|$ on a query x to $\hat{\mathcal{H}}_1$ is only $|\mathcal{Y}|/\rho \leq q_{\mathsf{D}}/\rho \leq 1$ as $(y_0, y_1) \in \mathcal{Y}$ will increase $|\mathcal{Z}|, |\mathcal{W}|$ if $y_1 = a_1$ for the randomly chosen $a_1 \in \mathbb{Z}_\rho$. Moreover a $\hat{\mathcal{H}}_3$ or $\hat{\mathcal{H}}_4$ query can increase $|\mathcal{Z}|$ and $|\mathcal{W}|$ by at most 1. We get a bound on the expected size of $\mathcal{Z}, \mathcal{W}$ of

$$\mathsf{E}[|\mathcal{Z}|] \ , \ \mathsf{E}[|\mathcal{W}|] \leq 2q_{\mathsf{D}} \tag{8}$$

Below, we will also need an upper bound on $\mathsf{E}[|\mathcal{W}|^2]$. As mentioned, the expected increase of $|\mathcal{W}|$ with every x query is $|\mathcal{Y}|/\rho$, the case which maximizes $\mathsf{E}[|\mathcal{W}|^2]$ is when this increase is either $|\mathcal{Y}|$ (with probability ρ^{-1}) or 0, as this maximizes the variance of $|\mathcal{W}|$, and thus also the expectation $\mathsf{E}[|\mathcal{W}|^2]$. The probability that we increase by $|\mathcal{Y}|$ more than t times is at most $(q_{\mathsf{D}}/\rho)^t$. Setting $t := -\log\rho/\log(q_{\mathsf{D}}/\rho) = \log\rho/\log(\rho/q_{\mathsf{D}})$ this is $(q_{\mathsf{D}}/\rho)^t = 2^{-\log\rho} = 1/\rho$ we get $\Pr\left[|\mathcal{W}| \geq q_{\mathsf{D}} \cdot \frac{\log\rho}{\log(\rho/q_{\mathsf{D}})}\right] \leq 1/\rho$. The same bound holds for $|\mathcal{Z}|$, and from now on we'll assume

$$|\mathcal{Z}| \ , \ |\mathcal{W}| \leq q_{\mathsf{D}} \cdot \log\rho/\log(\rho/q_{\mathsf{D}}) \tag{9}$$

We can safely ignore the tiny $1/\rho$ probability that this fails to hold.

We will now bound the probability that $\mathsf{FAIL}_0 := 1$ will be set in any of the queries to $\hat{\mathcal{H}}_2$. There are at most $|\mathcal{X}||\mathcal{Z}|$ possible $b \in \mathbb{Z}_\mu$ s.t. $x + b = z$ for some $x \in \mathcal{X}, z \in \mathcal{Z}$, thus a random b will fall in this set with probability at most $|\mathcal{X}||\mathcal{Z}|/\mu$. Taking the union bound over all $\leq q_{\mathsf{D}}$ queries to $\hat{\mathcal{H}}_2$

$$\Pr[\mathsf{FAIL}_0 = 1] \leq \frac{q_{\mathsf{D}}\mathsf{E}[|\mathcal{X}||\mathcal{Z}|]}{\mu} \leq \frac{q_{\mathsf{D}}^2\mathsf{E}[|\mathcal{Z}|]}{\mu} \leq \frac{2q_{\mathsf{D}}^3}{\mu} \tag{10}$$

Next, we will bound the probability that $\mathsf{FAIL}_2 := 1$ will be set while processing a z query to $\hat{\mathcal{H}}_3$. We set $\mathsf{FAIL}_2 := 1$ if the uniformly random (f_1, f_2) falls into the set $\mathcal{W}$, which happens with probability $|\mathcal{W}|/(\rho^2)$. Taking the union bound over the at most $|\mathcal{W}|$ times we increase $\mathcal{W}$ and (9) in the second step

$$\Pr[\mathsf{FAIL}_2 = 1] \leq \frac{\mathsf{E}[|\mathcal{W}|^2]}{\rho^2} \leq \frac{q_{\mathsf{D}}^2}{\rho^2} \cdot \left(\frac{\log\rho}{\log(\rho/q_{\mathsf{D}})}\right)^2 \tag{11}$$

To bound the probability that $\mathsf{FAIL}_1 := 1$ will be set, we observe that from an adversary D who manages to set $\mathsf{FAIL} := 1$ when interacting with $\mathcal{F}, \mathsf{S}^{\mathcal{F}}$, we can get an adversary A which wins the z-collision game on ϕ_{3F} with at least the same probability. Using this observation with the bound from Lemma 5.3 and the upper bound on $|\mathcal{Z}|$ as given (9) would give $\Pr[\mathsf{FAIL}_1 = 1] \leq \mathsf{E}[Q(\mu, q_\mathsf{D} \log \rho / \log(\rho/q_\mathsf{D}))]/\rho$. We can get a better bound using $\mathsf{E}[|\mathcal{Z}|] \leq 2q_\mathsf{D}$ and the fact that for any $c \geq 1$, $Q(\mu, q)$ increases by at most a factor c if we allow the $\mathcal{Z}$ in (5) to have size cq instead of q (this follows from the maximality of $\mathcal{Z}$.) In particular this gives

$$\Pr[\mathsf{FAIL}_1 = 1] \leq 2\mathsf{E}[Q(\mu, q_\mathsf{D})]/\rho \tag{12}$$

We will now bound $\Pr[\mathsf{FAIL}_3 = 1]$. To have $\mathsf{FAIL}_3 = 1$ it must be the case that at some point the adversary makes a fresh query $y \notin \mathcal{Y}$ s.t. the random output $\hat{\mathcal{H}}_3(y) = b$ satisfies $x + b = x' + b' = z$ for some $(x, x', b') \in \mathcal{X} \times \mathcal{X} \times \mathcal{B}$ and $z \notin \mathcal{Z}$ (it's not possible the x query was made after the y query as by our handling of $\hat{\mathcal{H}}_1$ queries in this case we'd have $z \in \mathcal{Z}$). As there are at most $|\mathcal{X}|^2|\mathcal{Y}| \leq q_\mathsf{D}^3$ triples, (x', b', x), each giving rise to one possible "target" $b = x' + b' - x$, the probability to hit any of them in the at most q_D queries is at most

$$\Pr[\mathsf{FAIL}_3 = 1] \leq \frac{q_\mathsf{D}^4}{\mu}$$

Finally, we can bound

$$\begin{aligned} & |\Pr[\mathsf{D}^{\mathsf{C}_{4F}^{\mathcal{H}},\mathcal{H}}(1^n) = 1] - \Pr[\mathsf{D}^{\mathcal{F},\mathsf{S}^{\mathcal{F}}}(1^n) = 1]| \\ \leq \quad & \sum_{i=0}^{3} \Pr[\mathsf{FAIL}_i = 1] \\ \leq \quad & \frac{2\mathsf{E}[Q(\mu, q_\mathsf{D})]}{\rho} + \frac{2q_\mathsf{D}^4}{\mu} + \frac{q_\mathsf{D}^2}{\rho^2} \cdot \left(\frac{\log \rho}{\log(\rho/q_\mathsf{D})} \right)^2 \end{aligned} \tag{13}$$

using standard arguments like in the proof of Theorem 4.1. ■

6 A Bit of Additive Combinatorics

Additive combinatorics deals with questions of the sort that given an Abelian group A, find subsets $\mathcal{Z}, \mathcal{X}$ of given size that minimizes the size of

$$\mathcal{Z} - \mathcal{X} = \{z - x | z \in \mathcal{Z}, \ \ x \in \mathcal{X}\}$$

Often we also want to find out the structure of such optimal (or nearly optimal) $\mathcal{Z}, \mathcal{X}$ pairs. Such pairs are of course special, and we do not have too many of them. Analogous questions are also raised when the '$-$' is replaced with '+'.

Here we investigate a variant, where we also have a third set $\mathcal{B} \subseteq A$ with the same size as $\mathcal{Z}$ and $\mathcal{X}$ with the property that $z - x \in \mathcal{B}$ for an unusually large number (say, $|\mathcal{B}|^{3/2}$) of (x, z) pairs with $z \in \mathcal{Z}$ and $x \in \mathcal{X}$. We show that for an adequately small random $\mathcal{B}$ it is very unlikely that we can find *any* $\mathcal{Z}, \mathcal{X}$ such that $\mathcal{Z}, \mathcal{X}$ and $\mathcal{B}$ form a triplet as above. We may interpret our result as a property of the random Cayley graph generated by $\mathcal{B}$.

Remark 6.1 *Although our setting is natural and undoubtedly useful for the application at hand, the problem we raise does not seem to have been studied before. An often-cited work of B. J. Green [13] computes the maximum clique size of (dense) random Cayley graphs of cyclic groups and of*

$\mathbb{Z}_2^n$. Other authors e.g. Christofides and N. Alon have also investigated random Cayley graphs, but with focus on Hamiltonicity, chromatic number, etc. The size of the generator set, unlike in our case, in most studies are either very small (poly($\log |A|$)) or very large ($\Omega(|A|)$). Since spectra of random Cayley graphs have been studied, it is conceivable that there is a shorter analytic proof to our statement. We use simple combinatorics to prove our theorem.

We (non-crucially) set the Abelian group A to be the cyclic group $\mathbb{Z}_\mu$, where μ is a prime. Let $1 \le q \le \mu$ arbitrary, but we will think of it as a small constant power of μ, for instance $q = \mu^{0.1}$. For a set $\mathcal{B} \subset \mathbb{Z}_\mu, |\mathcal{B}| = q$ define

$$Q(\mu, q, \mathcal{B}) = \max_{\mathcal{X},\mathcal{Z} \subset \mathbb{Z}_\mu, |\mathcal{X}|=|\mathcal{Z}|=q} |\{(x, z) \mid z \in \mathcal{Z},\ x \in \mathcal{X},\ z - x \in \mathcal{B}\}| \tag{14}$$

Expression (14) becomes a random variable $Q(\mu, q, .)$ as $\mathcal{B}$ ranges over all uniformly random $\mathcal{B} \subseteq \mathbb{Z}_\mu$ of size q. This random variable was also defined in Section 5.1. The minimum value of this random variable is at least q, because for any $\mathcal{B}$ one can choose $\mathcal{Z} = \mathcal{B}$ and $0 \in \mathcal{X}$. We show that if q is a small power of μ, the probability of the event that this random variable much exceeds q is small. To obtain practical expressions in the theorem and simpler formulas in the proof, we introduce $\mu = 2^t$ and $q = 2^r$.

Theorem 6.2 *For $0 < r < t/4$, and for every $s, l > 0$, $2^s \ge l^2$ it holds that*

$$\Pr[Q(2^t, 2^r) \ge 2^{r+s}] \le 2^{-DB+t}$$

where $D = \lceil 2^{s-\frac{r}{l}}/(2l + 2)\rceil$ and $B = t - l(r + 1)$.

Corollary 6.3 *Let $q = \mu^a$, where $a \le 1/4$. If q is large enough (while parameter a is fixed), then*

$$\Pr[Q(\mu, q) \ge q^{1+2a}] \le 2^{-q^a/2}$$

We defer the proof of the corollary to after that of the theorem.

Proof. (of Theorem 6.2) Let $\mu = 2^t$ denote the size of the group, which we assume to be $\mathbb{Z}_\mu$, but this is not essential. We prove Theorem 6.2 by an information compression argument. What we show is that a set $\mathcal{B}$ satisfying $|\mathcal{B}| = 2^r$, $Q(2^t, 2^r, \mathcal{B}) \ge 2^{r+s}$ has a lot of constant size linear relations between its elements, which allows us to describe it with significantly less than $\log \binom{2^t}{2^r}$ bits.

In order to encode a $\mathcal{B} \subseteq \mathbb{Z}_\mu$ for which $|\mathcal{B}| = 2^r$, $Q(2^t, 2^r, \mathcal{B}) \ge 2^{r+s}$ efficiently, we show that any such $\mathcal{B}$ has a decomposition $\mathcal{B} = \mathcal{D} \cup \overline{\mathcal{D}}$, where $|\mathcal{D}| = D$ as in the theorem, $\overline{\mathcal{D}} = \mathcal{B} \setminus \mathcal{D}$, and there exist fixed $x, z \in \mathbb{Z}_\mu$ that the elements b of $\mathcal{D}$ can be ordered suitably and be expressed as

$$b = \epsilon(z - x) - \epsilon_1 b_1 - \ldots - \epsilon_{l-1} b_{l-1}, \tag{15}$$

where $b_1, \ldots, b_{l-1}$ are either from $\overline{\mathcal{D}}$ or from elements of $\mathcal{D}$ that are expressed earlier. The numbers $\epsilon, \epsilon_1, \ldots, \epsilon_{l-1}$ are all in $\{-1, 1\}$. The saving per every item in $\mathcal{D}$ is the difference measured in bits between its description length via (15) versus their default information cost per item. The latter is:

$$\frac{\log \binom{2^t}{2^r} - \log \binom{2^t}{2^r - D}}{D} \sim t - r$$

Since the sequence $\epsilon_1, b_1, \ldots, \epsilon_{l-1}, b_{l-1}$ together with ϵ can be described with $(l-1)(r+1)+1$ bits (each b_i is element of $\mathcal{B}$ which is already on our list, so has an r-bit description), our saving per item is

$$B = t - r - (l-1)(r+1) - 1 = t - l(r+1)$$

bits. Our total saving is then $DB - t$, since we also need t bits to describe $z - x$ (once for the entire $\mathcal{D}$). The upper bound on the probability of the event $Q(2^t, 2^r, \mathcal{B}) \geq 2^{r+s}$ is then 2^{-DB+t}.

We are left to construct the $(\mathcal{D}, \overline{\mathcal{D}})$ decomposition and to calculate D. Consider a $\mathcal{B}$ that satisfies $Q(2^t, 2^r, \mathcal{B}) \geq 2^{r+s}$. Then there are $\mathcal{X}, \mathcal{Z} \subseteq \mathbb{Z}_\mu$, $|\mathcal{X}| = |\mathcal{Z}| = 2^r$ such that $|\{(x,z) \mid x \in \mathcal{X}, z \in \mathcal{Z}, z - x \in \mathcal{B}\}| \geq 2^{r+s}$. We fix such an $\mathcal{X}, \mathcal{Z}$ pair. Let G be the bipartite graph with bipartition $(\mathcal{X}, \mathcal{Z})$ and edge set

$$e(G) = \{(x,z) \mid \ x \in \mathcal{X}, z \in \mathcal{Z}, z - x \in \mathcal{B}\}.$$

By our assumption $|e(G)| \geq 2^{r+s}$. If we iteratively remove the minimum degree vertex from G until all degrees of the resulting graph are at least $2^s/2$ (i.e. the average degree of G divided by two), it is easy to show that this process ends up with a non-empty graph G' with minimum degree at least $2^s/2$. Fix a vertex $x \in \mathcal{X} \cap V(G')$. Our proof hinges upon the following construction:

Definition 6.4 *Let P_i for $i = 1, 2, \ldots$ be the set of all those (not necessarily simple) paths π of length i in G' (the length is the number of edges) that satisfy:*

1. *π starts at x*
2. *No two edges edges of π have identical labels, where a label of an edge (v, w) ($v \in \mathcal{X}$, $w \in \mathcal{Z}$) is by definition $w - v$.*

Let π be a path in P_i and let $d = d(\pi)$ denote the degree of its end point z. All edges incident to z have distinct labels, so the number of those edges incident to z whose label do not coincide with any labels we already have in π is at least $d - i$. Thus π has $d - i \geq \frac{2^s}{2} - i$ continuations in P_{i+1}. Therefore, by induction, for $i \geq 1$:

$$|P_i| \geq \prod_{j=0}^{i-1} \left(\frac{2^s}{2} - j\right) > \frac{1}{e}\frac{2^{is}}{2^i}.$$

Consider the set P_l. Notice that if l is odd, then every path in P_l end in $\mathcal{Z}$, otherwise they all end in $\mathcal{X}$. Since the nodes of G' are from $\mathcal{X} \cup \mathcal{Z}$ and $|\mathcal{X}|, |\mathcal{Z}| = 2^r$, there must be a $z \in \mathcal{X}$ (if l is even) or $z \in \mathcal{Z}$ (if l is odd) such that at least $\frac{|P_l|}{2^r} \geq \frac{1}{e}\frac{2^{ls-r}}{2^l}$ paths from P_l end in z.

Let T be the set of the paths in P_l that end in this z. We will use the paths in T to find a lot of small linear relations among the elements of $\mathcal{B}$. For a path π let $\ell(\pi)$ denote the set of labels that occur on its edges, and define $\mathcal{D}_0 = \cup_{\pi \in T} \ell(\pi)$, which is just the collection of all labels that ever occur in those paths of P_l that end in z. Of course, $\mathcal{D}_0 \subseteq \mathcal{B}$, because all labels along the edges of G' are in $\mathcal{B}$. In order to estimate $|\mathcal{D}_0|$ we view a path $\pi \in P_l$ as an ordered sequence of labels. Each $\pi \in P_l$ uniquely corresponds to such a sequence of length l (although not necessarily every element of $\mathcal{D}_0^l$ is an element of P_l). Since from an alphabet of size $|\mathcal{D}_0|$ we can create at most $|\mathcal{D}_0|^l$ different sequences of length l, we have that

$$|\mathcal{D}_0| \geq |T|^{1/l} \geq \left(\frac{1}{e}\frac{2^{ls-r}}{2^l}\right)^{1/l} \geq 2^{s-r/l}/(2+2/l).$$

We are now ready to define the decomposition $\mathcal{B} = \mathcal{D} \cup \overline{\mathcal{D}}$ as promised in the beginning. The role of x and z in expression (15) will be played by the common starting- and end-point of all paths in T. For any path $\pi \in T$ we have that

$$z - x = b_1 - b_2 + b_3 - \ldots + b_l \quad \text{(if } l \text{ is odd, otherwise the last sign is a minus)}$$

It is a trivial matter to transform the above equation into (15), where b is one of the b_is (our choice which one). What remains is to show is that starting from a subset of T we can to generate all remaining elements by (15) such, that the number of generated elements is no less than the bound we require. A combinatorial lemma will help us in this.

Definition 6.5 *We say that a set $\{h_1, \ldots, h_{l-1}\}$ of nodes in an undirected hyper-graph $\mathcal{H}$ generates node h, if $\{h_1, \ldots, h_{l-1}, h\}$ is a hyper-edge. A generator set for $\mathcal{H}$ is a subset of nodes from which we can iteratively generate the entire vertex set of $\mathcal{H}$.*

Lemma 6.6 *Let $\mathcal{H}$ be an hyper-graph on m nodes such that every edge is a set of size at most l, and every node is contained in at least one hyper-edge. Then $\mathcal{H}$ has a generator of size at most $\frac{(l-1)m}{l}$.*

Proof. The proof is by induction on l. The claim is trivial for $l = 1$. Take a minimal generator set X for $\mathcal{H}$. If it does not satisfy our condition, then $|X| > \frac{(l-1)m}{l}$. Consider the hyper-graph $\mathcal{H}'$ we get from $\mathcal{H}$ by restricting all of its nodes and edges to X. Since a minimal generator set in $\mathcal{H}$ cannot properly contain any hyper-edge, every hyper-edge in $\mathcal{H}'$ has size at most $l - 1$. Thus by induction $\mathcal{H}'$ has a generator set Y of size at least $\frac{(l-2)|X|}{l-1}$. But $Y \cup \overline{X}$ generates $\mathcal{H}$, and it has size at most $\frac{l-2}{l-1}|X| + m - |X| \leq \frac{(l-1)m}{l}$.

We now apply this lemma for the hyper-graph, which has vertex set $\mathcal{D}_0$ and edge set $\{\ell(\pi) \mid \pi \in T\}$. We get a generator set of size $(l-1)|\mathcal{D}_0|/l$. We put the elements of this generator set into $\overline{\mathcal{D}}$, as well as the elements of $\mathcal{B}$ that are not in $\mathcal{D}_0$. We can generate the remaining elements of $\mathcal{D}_0$ out of these via (15), and we let these form the set $\mathcal{D}$. The size of $\mathcal{D}$ is $|\mathcal{D}_0|/l = \frac{2^{s-r/l}}{(2l+2)}$.

Proof. (of Corollary 6.3) In Theorem 6.2 we set $a = r/t$, $s = 2r^2/t$, $l = \frac{3t}{4(r+1)}$. This gives $B = t/4$ and

$$D = \frac{\exp_2\left(2\frac{r^2}{t} - \frac{4r(r+1)}{3t}\right)}{2l+2} = q^{2\frac{r}{t} - \frac{4(r+1)}{3t}}/(2l+2) \geq q^{a/2}$$

if q is large enough (above $\exp_2(z)$ is by definition 2^z). Thus $2^{-DB+t} \leq 2^{-q^a/2}$ when q is sufficiently large.

Acknowledgements

We thank Mihir Bellare for suggesting the open problem to us. Furthermore, we are grateful to Yannick Seurin for pointing out a gap in the definition of the simulator in a previous version of this paper.

References

[1] M. Abe, E. Kiltz, and T. Okamoto. Chosen ciphertext security with optimal ciphertext overhead. In J. Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 355–371. Springer, Dec. 2008. 5

[2] N. Alon, T. Kaufman, M. Krivelevich, and D. Ron. Testing triangle-freeness in general graphs. *SIAM J. Discrete Math.*, 22(2):786–819, 2008. 5

[3] M. Bellare and T. Ristenpart. Simulation without the artificial abort: Simplified proof and improved concrete security for Waters' IBE scheme. In A. Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 407–424. Springer, Apr. 2009. 2

[4] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73. ACM Press, Nov. 1993. 2, 5

[5] M. Bellare and P. Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. In U. M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 399–416. Springer, May 1996. 2, 3, 9

[6] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In C. Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 514–532. Springer, Dec. 2001. 2

[7] C. Cachin, S. Micali, and M. Stadler. Computationally private information retrieval with polylogarithmic communication. In J. Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 402–414. Springer, May 1999. 2

[8] B. Chevallier-Mames, D. H. Phan, and D. Pointcheval. Optimal asymmetric encryption and signature paddings. In J. Ioannidis, A. Keromytis, and M. Yung, editors, *ACNS 05*, volume 3531 of *LNCS*, pages 254–268. Springer, June 2005. 2, 3

[9] J.-S. Coron. On the exact security of full domain hash. In M. Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 229–235. Springer, Aug. 2000. 3, 8

[10] J.-S. Coron, Y. Dodis, C. Malinaud, and P. Puniya. Merkle-Damgård revisited: How to construct a hash function. In V. Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 430–448. Springer, Aug. 2005. 3, 6

[11] Y. Dodis, T. Ristenpart, and T. Shrimpton. Salvaging Merkle-Damgård for practical applications. In A. Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 371–388. Springer, Apr. 2009. 3, 6

[12] S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, Apr. 1988. 6

[13] B. Green. Counting sets with small sumset, and the clique number of random cayley graphs. *Combinatorica*, pages 307–326, 2005. 17

[14] T. Holenstein, R. Künzler, and S. Tessaro. The equivalence of the random oracle model and the ideal cipher model, revisited. In L. Fortnow and S. P. Vadhan, editors, *43rd ACM STOC*, pages 89–98. ACM Press, June 2011. 3

[15] S. A. Kakvi and E. Kiltz. Optimal security proofs for full domain hash, revisited. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 537–553. Springer, Apr. 2012. 2, 3, 4, 7, 8, 24

[16] J. Katz and N. Wang. Efficiency improvements for signature schemes with tight security reductions. In S. Jajodia, V. Atluri, and T. Jaeger, editors, *ACM CCS 03*, pages 155–164. ACM Press, Oct. 2003. 2, 3, 4, 8

[17] A. Mandal, J. Patarin, and Y. Seurin. On the public indifferentiability and correlation intractability of the 6-round Feistel construction. In R. Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 285–302. Springer, Mar. 2012. 3, 4

[18] U. M. Maurer, R. Renner, and C. Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In M. Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 21–39. Springer, Feb. 2004. 3, 6, 7

[19] A. Naor and J. Verstraëte. A note on bipartite graphs without 2k-cycles. *Comb. Probab. Comput.*, 14(5-6):845–849, Nov. 2005. 11

[20] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In J. Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 223–238. Springer, May 1999. 8

[21] C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In R. E. Ladner and C. Dwork, editors, *40th ACM STOC*, pages 187–196. ACM Press, May 2008. 4, 7, 8

[22] K. Yoneyama, S. Miyagawa, and K. Ohta. Leaky random oracle. *IEICE Transactions*, 92-A(8):1795–1807, 2009. 3, 6

A Indifferentiability of the two round Feistel network

A.1 Proof of Theorem 4.1

Proof. Let $q_{\mathsf{D}} = q_{\mathcal{F}} + q_{\mathcal{H}}$, where $q_{\mathcal{F}}$ and $q_{\mathcal{H}}$ denote the number of queries D makes to its first and second oracle, respectively. We have to specify a simulator S such that for any q_{D}-query distinguisher D and a random invertible function $\mathcal{F} : \mathbb{Z}_\mu \to \mathbb{Z}_\mu \times \mathbb{Z}_\rho$

$$|\Pr[\mathsf{D}^{\mathsf{C}^{\mathcal{H}}_{2f},\mathcal{H}}(1^n) = 1] - \Pr[\mathsf{D}^{\mathcal{F},\mathsf{S}^{\mathcal{F}}}(1^n) = 1]| \leq q_{\mathsf{D}}^2/\rho, \tag{16}$$

where the probability is over the choice of $\mathcal{F}, \mathcal{H}$ and the randomness used by S and D. The simulator $\mathsf{S}^{\mathcal{F}}$ will internally define fake random oracles $\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2$ by lazy sampling which initially are undefined on all inputs. $\hat{\mathcal{H}}_1(x) = \Diamond$ denotes $\hat{\mathcal{H}}_1$ is undefined on input x. The set $\mathcal{X} \subset \mathbb{Z}_\mu$ will denote the inputs on which $\hat{\mathcal{H}}_1$ has already been defined, and $\mathcal{A} = \hat{\mathcal{H}}_1(\mathcal{X})$ are the corresponding outputs. Similarly $\mathcal{Y}, \mathcal{B} = \hat{\mathcal{H}}_2(\mathcal{Y})$ denote the inputs and the corresponding outputs on which $\hat{\mathcal{H}}_2$ has been defined. The simulator also initializes a variable $\mathsf{FAIL} := 0$ which will only be used in the proof below. Informally, if at the end of the experiment $\mathsf{FAIL} = 1$ then this indicates that $\mathsf{S}^{\mathcal{F}}$ failed to define the $\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2$ such that $\mathsf{C}^{\hat{\mathcal{H}}}_{2f}(\cdot)$ looks consistent with $\mathcal{F}(\cdot)$ given all queries made so far. If this happens, the simulator aborts which means it refuses to answer any more queries. We now define how $\mathsf{S}^{\mathcal{F}}$ answers queries to $\hat{\mathcal{H}}_1$ and $\hat{\mathcal{H}}_2$ and how it updates its state if $\mathsf{D}^{\mathcal{F},\mathsf{S}^{\mathcal{F}}}$ makes an $\mathcal{F}$ or $\mathcal{F}^{-1}$ query.

$\hat{\mathcal{H}}_2$ **query** $y \in \mathbb{Z}_\rho$ **:** If $y \notin \mathcal{Y}$ (equivalently $\hat{\mathcal{H}}_2(y) = \Diamond$) sample $b \leftarrow \mathbb{Z}_\mu$ and set $\hat{\mathcal{H}}_2(y) := b$. Output $\hat{\mathcal{H}}_2(y)$.

$\hat{\mathcal{H}}_1$ **or** $\mathcal{F}$ **query** $x \in \mathbb{Z}_\mu$ **:** If $x \notin \mathcal{X}$ try to program the $\hat{\mathcal{H}}_i$ s.t. $\mathsf{C}_{2f}^{\hat{\mathcal{H}}}(x, 0) = \mathcal{F}(x)$ as follows

1. query $(f_0, f_1) \leftarrow \mathcal{F}(x)$ and set $\hat{\mathcal{H}}_1(x) := f_1$.
2. if $f_1 \in \mathcal{Y}$ set $\mathsf{FAIL} := 1$ and abort.
3. set $\hat{\mathcal{H}}_2(f_1) := x - f_0$.

If this is an $\hat{\mathcal{H}}_1$ (and not a $\mathcal{F}$) query output $\hat{\mathcal{H}}_1(x)$.

$\mathcal{F}^{-1}$ **query** $u \in \mathbb{Z}_\mu \times \mathbb{Z}_\rho$**:** query $x \leftarrow \mathcal{F}^{-1}(u)$, if $x \neq \perp$ and $\hat{\mathcal{H}}_1(x) \neq \Diamond$ set $\mathsf{FAIL} := 1$ and abort.

Considering the efficiency of our simulator, note that $\mathsf{S}^{\mathcal{F}}$ does exactly one oracle query for every $\mathcal{F}, \mathcal{F}^{-1}$ and $\mathcal{H}_1$ query of $\mathsf{D}^{\mathcal{F},\mathcal{H}}$ (and no query for an $\mathcal{H}_2$ ar query), so $q_{\mathsf{S}} \leq q_{\mathcal{H}} + q_{\mathcal{F}} = q_{\mathsf{D}}$.

To prove eq.(16) we will first bound the probability that $\mathsf{FAIL} = 1$ in the above experiment. Note that $|\mathcal{Y}|$ (i.e., the number of inputs on which $\hat{\mathcal{H}}_2$ is defined) increases by at most 1 on every $\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2$ and $\mathcal{F}$ query, thus

$$|\mathcal{Y}| \leq q_{\mathcal{H}} + q_{\mathcal{F}}.$$

Further, FAIL can only be set to 1 on a $\hat{\mathcal{H}}_1$ or $\mathcal{F}$ query, and this happens if the uniformly random f_1 is in $\mathcal{Y}$. For every query, this happens with probability $\leq |\mathcal{Y}|/\rho$. Taking the union bound over all $\hat{\mathcal{H}}_1, \mathcal{F}$ queries we get

$$\Pr[\mathsf{FAIL} = 1] \leq (q_{\mathcal{H}} + q_{\mathcal{F}})^2/\rho.$$

Next, well argue that

$$\left|\Pr[\mathsf{D}^{\mathsf{C}_{2f}^{\mathcal{H}},\mathcal{H}}(1^n) = 1] - \Pr[\mathsf{D}^{\mathcal{F},\mathsf{S}^{\mathcal{F}}}(1^n) = 1]\right| \leq \Pr[\mathsf{FAIL} = 1]. \tag{17}$$

Note that the two equations above imply eq.(16).

Let $\langle \mathsf{D}^{\mathcal{F},\mathsf{S}^{\mathcal{F}}} \rangle$ denote the transcript containing all queries and corresponding answers of the oracle queries made by D. For any possible transcript τ

$$\forall \tau \ : \ \Pr[\tau = \langle \mathsf{D}^{\mathcal{F},\mathsf{S}^{\mathcal{F}}} \rangle \wedge \mathsf{FAIL} = 0] \leq \Pr[\tau = \langle \mathsf{D}^{\mathsf{C}_{2f}^{\mathcal{H}},\mathcal{H}} \rangle]. \tag{18}$$

To see this, note that S assigns uniformly random values to the $\hat{\mathcal{H}}_i$ (the randomness is either sampled directly or comes from the random function $\mathcal{F}$) which are independent of anything that happend so far. (18) implies (17) by standard arguments like the fundamental lemma of game-playing. ■

A.2 The overhead of the two round Feistel construction

By combining Theorem 4.1 with Theorems 3.1/3.2 and Theorem 2.3 we obtain the following corollary.

Corollary A.1 (Security and minimal overhead for $\mathsf{SIG\text{-}MR}_{2f}^{\mathcal{H}}$) *Let* TDP *be a* $(t_{lossy}, \varepsilon_{lossy}, \ell)$*-lossy trapdoor permutation. Then* $\mathsf{SIG\text{-}MR}_{2f}^{\mathcal{H}} = \mathsf{SIG\text{-}MR}^{\mathsf{C}_{2f}^{\mathcal{H}}}[\mathsf{TDP}]$ *is a* $(t_{sig}, q_s, q_h, \varepsilon_{sig})$*-secure signature scheme with*

$$t_{sig} = t_{lossy} - (q_h + q_s) \cdot poly(n), \quad \varepsilon_{sig} = \frac{2(q_s + q_h)^2}{\rho} + \frac{2\ell - 1}{\ell}\varepsilon_{lossy}. \tag{19}$$

Assuming $2\varepsilon_{lossy}/t_{sig} \leq 2^{-n-1}$, the overhead required to get n bits security with (q_s, q_h) queries is

$$\mathsf{O}(n, q_s, q_h) \geq n + 2 + \log(q_h + q_s) \quad \text{or} \quad \mathsf{O}(n) \geq 2n + 2$$

assuming only the trivial $q_h + q_s \leq 2^n$ bound on the number of queries.

Proof. We compute the overhead according to Definition 2.2. Using $2\varepsilon_{lossy}/t_{sig} \leq 2^{-n-1}$ and $t_{sig} \geq q_s + q_h$ we obtain by (19)

$$\frac{\varepsilon_{sig}}{t_{sig}} \leq \frac{2(q_s + q_h)^2}{\rho t_{sig}} + 2^{-n-1} \leq \frac{2(q_s + q_h)}{\rho} + 2^{-n-1}$$

To get n bits of security we must set the overhead ρ such that $\varepsilon_{sig}/t_{sig} \leq 2^{-n}$, which holds for $\log \rho := n + 2 + \log(q_s + q_h)$. ■

Note that $\mathsf{SIG\text{-}MR}_{2f}$ is the same as $\mathsf{PSS\text{-}MR}$ with modular addition instead of xor. Hence Corollary A.1 essentially reproves a theorem of [15] about the security of $\mathsf{PSS\text{-}MR}$, expressed in our general framework. (We use modular addition since we use $\mathsf{SIG\text{-}MR}_{2f}$ only as an introductory example. The same bounds can be proved for xor.)

The following lemma says that one cannot avoid the additional additive factor q_h^2/ρ in the security reduction, hence the overhead of $\mathsf{O}_{\mathsf{SIG\text{-}MR}_{2f}} = n + \log(q_h + q_s)$ bits is optimal for $\mathsf{SIG\text{-}MR}_{2f}$.

Lemma A.2 *If there exists a one-way (lossy) TDP, then there exists a one-way (lossy) TDP' such that for all q, $\mathsf{SIG\text{-}MR}^{\mathsf{C}_{2f}^{\mathcal{H}}}[\mathsf{TDP}]$ is not $(t_{sig} = O(q), q_h = q, q_s = 0, \varepsilon_{sig} = q^2/\rho)$-secure.*

Proof. We define the evaluation function of TDP' as $\mathsf{f}'(z, y) := (z, \mathsf{f}(y)) \in \mathbb{Z}_\mu \times \mathbb{Z}_\rho$. Clearly, one-wayness and lossiness are inherited. The attack on TDP' is as follows. First, A picks uniform $x_1, \ldots, x_q$ and computes $y_i := \mathsf{f}(x_i)$. Next, it makes q queries arbitrary distinct $m_1, \ldots, m_q$ to the $\mathcal{H}_1$ oracle. The probability that there exists indices $i, j \in \{1, \ldots, q\}$ such that $\mathcal{H}_1(m_i) = y_j$ is bounded by q^2/ρ. In case they exist, then $\tau := (m_i + \mathcal{H}_2(y_j), x_j)$ is a valid signature, i.e., $\mathsf{Recover}'(\tau) = m_i$. ■

B Domain Extension

Let $\mathcal{H}$ be a random oracle and let $\mathsf{SIG\text{-}MR}^{\mathcal{H}}$ be a signature scheme with message recovery with message space $\{0,1\}^m$. We will now describe a simple (almost) generic way to turn $\mathsf{SIG\text{-}MR}^{\mathcal{H}}$ into a scheme $\mathsf{SIG\text{-}MR}_*^{\mathcal{H}}$ with arbitrary larger message space $\{0,1\}^m \times \{0,1\}^*$ without increasing the redundancy and where $\mathsf{SIG\text{-}MR}_*^{\mathcal{H}}$ comes with almost the same security guarantee (in the random oracle model) as $\mathsf{SIG\text{-}MR}^{\mathcal{H}}$.

The scheme $\mathsf{SIG\text{-}MR}_*^{\mathcal{H}}$ has the same key-space as $\mathsf{SIG\text{-}MR}^{\mathcal{H}}$, and signs a message $(M_0, M_1) \in \{0,1\}^m \times \{0,1\}^*$ by computing $\tau \leftarrow \mathsf{SIG\text{-}MR}^{\mathcal{H}}(M_0)$, but where each hash function call $\mathcal{H}(x)$ is replaced with $\mathcal{H}(M_1, x)$.[7] The signature is (M_1, τ). To verify (M_1, τ) one simply verifies τ as in

[7] Here $\mathcal{H}(a, b)$ means we invoke $\mathcal{H}$ on some efficiently uniquely decodable encoding of the message pair (a, b). Such an encoding is, for example, given by $0^{l_a} \| L_a \| a \| b$ where L_a is the length of a in binary, and l_a is the length of L_a.

$\mathsf{SIG\text{-}MR}^{\mathcal{H}}$, but using the hash function $\mathcal{H}(M_1, \cdot)$. If this verification accepts and outputs a message M_0, the verification for $\mathsf{SIG\text{-}MR}^{\mathcal{H}}_*$ accepts and outputs (M_0, M_1).[8]

We claim that $\mathsf{SIG\text{-}MR}^{\mathcal{H}}_*$ is secure if $\mathsf{SIG\text{-}MR}^{\mathcal{H}}$ is. To see this, first assume the adversary against $\mathsf{SIG\text{-}MR}^{\mathcal{H}}_*$ only makes signature/hash queries for the same fixed M_1 (i.e., signature queries (M_0, M_1) for any M_0 and hash queries (M_1, x) for any x.) Then the security of $\mathsf{SIG\text{-}MR}^{\mathcal{H}}_*$ can be proven exactly as for $\mathsf{SIG\text{-}MR}^{\mathcal{H}}$, except that throughout the security experiment we use the random oracle $\mathcal{H}(M_1, \cdot)$ instead of $\mathcal{H}(\cdot)$.

Let us now consider the general case where the adversary makes signature/hash queries for different M_1. In the security proof for $\mathsf{SIG\text{-}MR}^{\mathcal{H}}$ we run a simulator S to program the random oracle. In the proof for $\mathsf{SIG\text{-}MR}^{\mathcal{H}}_*$ we now simply start a new simulator S_{M_1} whenever the adversary makes a query with a fresh M_1. We can think of this proof as programming many independent random oracles $\mathcal{H}(M_1, \cdot)$ for different M_1. The simulation fails, if any of the simulators S_{M_1} fails. If our scheme is proven to be $(q, \varepsilon(q))$-pub-indifferentiable where $\varepsilon(q)$ is convex, in particular, if for any $q_1, \ldots, q_z, \sum_{i=1}^{z} q_i = q$ it satisfies $\sum_{i=1}^{z} \varepsilon(q_i) \leq \varepsilon(q)$. Then we also get $(q, \varepsilon(q))$-pub-indifferentiable for the potentially many different simulations. The bounds on pub-indifferentiability we prove are of the form $\varepsilon(q) = q^{1+c}/d$ (for some constant $c > 0$ and a term d that does not depend on q), and thus satisfy this convexity condition.

C Omitted proofs of Section 3

C.1 Proof of Theorem 3.1

Proof. Let A be an adversary against the signature scheme that runs in time t_{sig}, makes at most q_s queries to the signing oracle, q_f queries to the random injective function $\mathcal{F}$ and its inverse $\mathcal{F}^{-1}$, and outputs a forgery with probability ε_{sig}. To prove the bound on ε_{sig} we will proceed by defining a number of games G_0-G_4.

Game G_0. This is the UF-CMA game and hence $\Pr[\mathsf{G}_0 = 1] = \varepsilon_{sig}$.

Game G_1. We simulate the random injective injective function $\mathcal{F} : \{0,1\}^m \to \{0,1\}^k, \mathcal{F}^{-1} : \{0,1\}^k \to \{0,1\}^m \cup \{\perp\}$ using lazy sampling. The idea is that we simulate $\mathcal{F}$ by internally storing a random invertible permutation Π over $\{0,1\}^k$ and defining $\pi(M) := \Pi(M||0^{k-m})$. In the simulation all function values of Π, Π^{-1} are initialized to $\Diamond$.

```
Algorithm $\mathcal{F}(M \in \{0,1\}^m)$
    $x := M||0^{k-m} \in \{0,1\}^k$
    If $\Pi(x) \neq \Diamond$ then return $\mathcal{F}(M) = \Pi(x)$
    Repeat $y \leftarrow \{0,1\}^k$ until $\Pi(y) = \Diamond$
    $\Pi^{-1}(y) := x; \Pi(x) := y$
    Return $\mathcal{F}(M) = y$
```

```
Algorithm $\mathcal{F}^{-1}(y \in \{0,1\}^k)$
    If $\Pi^{-1}(y) \neq \Diamond$ then $x = \Pi^{-1}(y)$
        Else Repeat $x \leftarrow \{0,1\}^k$ until $\Pi(x) \neq \Diamond$
        $\boxed{\text{Abort if } x = M\|0^{k-m} \text{ //only in } \mathsf{G}_2}$
    $\Pi^{-1}(y) := x; \Pi(x) := y$
    If $x = M||0^{k-m}$ then return $\mathcal{F}^{-1}(y) = M$
        Else return $\mathcal{F}^{-1}(y) = \perp$
```

[8] A more efficient solution (whenever the padding queries $\mathcal{H}$ more than once) is to prepend $\mathcal{G}(M_1)$ instead of M_1 for some collision resistant hash function $\mathcal{G}$ (e.g., a random oracle). Alternatively, if $\mathcal{H}$ is an iterated hash function, one must hash the prefix M_1 only once, and can then evaluate $\mathcal{H}(M_1, x)$ at basically the cost of hashing only x. The complexity added by the two solutions outlined above is just the cost of hashing M_1 once.

Note that this simulation is efficient as long as $q_f < 2^{k-1}$. (For $q_f > 2^{k-1}$ there also exists efficient simulations.) Furthermore, the simulation is perfect and hence $|\Pr[\mathsf{G}_0 = 1] = \Pr[\mathsf{G}_1 = 1]|$.

Game G_2. We change the simulation of $\mathcal{F}^{-1}$ by adding an abort condition. $|\Pr[\mathsf{G}_1 = 1] = \Pr[\mathsf{G}_2 = 1]| \leq \frac{q_f}{2^{k-m}}$.

Game G_3. We now change the way we define $\Pi(M||0^{k-m})$ in the simulation of $\mathcal{F}$ such that we can simulate signing without knowing the secret-key.

```
Algorithm $\mathcal{F}(M)$
    $x := M||0^{k-m}$
    If $\Pi(x) \neq \Diamond$ then return $\mathcal{F}(M) = \Pi(x)$
    Repeat $\tau(M) \leftarrow \{0,1\}^k$;  $y = \mathsf{f}_{ek}(\tau(M))$
        until $\Pi(y) = \Diamond$
    $\Pi^{-1}(y) := x; \Pi(x) := y$
    Return $\mathcal{F}(M) = y$
```

```
Algorithm $\mathsf{Sign}(M)$
    If $\mathcal{F}(M) = \Diamond$ then call $\mathcal{F}(M)$
    Return $\tau(M)$        // $\tau(M)$ is always defined
```

Since f_{ek} defines a permutation, this does not change the distribution of $\mathcal{F}$. Consequently, $\Pr[\mathsf{G}_2 = 1] - \Pr[\mathsf{G}_3 = 1]$.

Game G_4. Switch ek of TDP to lossy. More formally, game G_4 is like G_3 with the difference that ek from pk is now generated using the lossy trapdoor generation algorithm $\mathsf{G}_{lossy}(1^n)$. Clearly, since from G_3 on signing does not use trapdoor td any more,

$$\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_2 = 1] \leq \varepsilon_{lossy}.$$

By the regular lossyness of f_{ek}, the value $\tau(M^*)$ is information-theoretically hidden amongst the ℓ possible pre-images of $\mathsf{f}_{ek}(\tau(M^*))$ and with probability $\frac{\ell-1}{\ell}$ we have $\mathsf{f}_{ek}(\tau(M^*)) = \mathsf{f}_{ek}(\tau^*)$ with $\tau(M^*) \neq \tau^*$. In the latter case we have a collision which contradicts again lossyness. More formally we can show that

$$\Pr[\mathsf{G}_4 = 1] \leq \frac{\ell - 1}{\ell} \cdot \varepsilon_{lossy}.$$

Summing up, we get $\varepsilon_{sig} \leq \Pr[\mathsf{G}_0 = 1] - \Pr[\mathsf{G}_4 = 1] \leq \varepsilon_{lossy} + \frac{\ell-1}{\ell}\varepsilon_{lossy} + \frac{q_f}{2^{k-m}}$ as claimed.

C.2 Proof of Theorem 3.2

Proof. Let A be an adversary against the signature scheme that runs in time t_{sig}, makes at most q_s queries to the signing oracle, q_f queries to $\mathcal{F}$, and outputs a forgery with probability ε_{sig}. Games G_0 until G_2 are the same as in the proof of Theorem 3.1, with the obvious adoptions due to bit b. We have $\varepsilon_{sig} = \Pr[\mathsf{G}_0 = 1]$ and $\Pr[\mathsf{G}_0 = 1] - \Pr[\mathsf{G}_2 = 1] \leq \frac{q_f}{2^{k-m}}$.

Game G_3. Define this game as G_3 in the proof of Theorem 3.1 with a different simulation of $\mathcal{F}$ and Sign.

```
Algorithm $\mathcal{F}(b\|M)$
    $x := b\|M||0^{k-m}$
    If $\Pi(x) \neq \Diamond$ then return $\mathcal{F}(M) = \Pi(x)$
    If $b(m) = \perp$ then $b(m) \leftarrow \{0,1\}$
    if $b(m) = b$ then
    Repeat $\tau(M) \leftarrow \{0,1\}^k$;  $y = \mathsf{f}_{ek}(\tau(M))$
        until $\Pi(y) = \Diamond$
    Else Repeat $y \leftarrow \{0,1\}^k$ until $\Pi(y) = \Diamond$    (*)
    $\Pi^{-1}(y) := x; \Pi(x) := y$
    Return $\mathcal{F}(M) = y$
```

```
Algorithm $\mathsf{Sign}(M \in \{0,1\}^{m-1})$
    If $b(M) = \perp$ then $b(M) \leftarrow \{0,1\}$
    If $\mathcal{F}(b(M)\|M) = \perp$ then call $\mathcal{F}(b(M)\|M)$
    Return $\tau(M)$
```

With the same argument as in the proof of Theorem 3.1 we get

$$\Pr[\mathsf{G}_3 = 1] = \Pr[\mathsf{G}_2 = 2].$$

Let τ^* be the forgery and let $b^*||M^*$ be the bit and the recovered message. Note that the value $b(M^*)$ is information-theoretically hidden from the adversary's view and with probability 1/2 we have $b(M^*) \neq b^*$. In the latter case the adversary has managed to find a pre-image under f_{ek} on an uniformly distributed value coming from the outside of the experiment. More formally we claim that

$$\Pr[\mathsf{G}_3 = 1] \leq \frac{1}{2} \cdot \varepsilon_{one-way}.$$

We sketch a proof of the claim. For the simulation, we need TDP to be homomorphic to be able to embed one single challenge from the one-wayness experiment into all extended signatures τ which contain $b||M$ such that $b \neq b(M)$. Concretely, the adversary A against one-wayness inputs ek and $y = \mathsf{f}_{ek}(x)$. It simulates the oracles $\mathcal{F}$ and Sign as above where in the case $(*)$ $(b \neq b(M))$ of the $\mathcal{F}$-simulation he defines $\mathcal{F}(b||M) := \mathsf{f}_{ek}(x(M)) \circ y$, for $x(M) \leftarrow \{0,1\}^k$. Finally, when A outputs his forgery τ^*, A recovers $b^*||M^*$ and aborts if $b(M^*) = b^*$. This happens with probability 1/2. Otherwise, we have $\tau^* = \mathsf{f}^{-1}(\mathcal{F}(b^*||M^*)) = x(M^*) \circ \mathsf{f}_{ek}^{-1}(y)$, from which the pre-image of y can be computed.

Summing up, we get $\varepsilon_{sig} \leq \Pr[\mathsf{G}_0 = 1] - \Pr[\mathsf{G}_3 = 1] \leq \frac{1}{2}\varepsilon_{one-way} + \frac{q_f}{2^{k-m}}$ as claimed.