Correlated-Input Secure Hash Functions

Vipul Goyal
Microsoft Research, India
Email: vipul@microsoft.com

Adam O'Neill*
University of Texas at Austin
Email: adamo@cs.utexas.edu

Vanishree Rao*
University of California, Los Angeles
Email: vanishri@cs.ucla.edu

Abstract

We undertake a general study of hash functions secure under *correlated inputs*, meaning that security should be maintained when the adversary sees hash values of many related high-entropy inputs. Such a property is satisfied by a random oracle, and its importance is illustrated by study of the "avalanche effect," a well-known heuristic in cryptographic hash function design. One can interpret "security" in different ways: e.g., asking for one-wayness or that the hash values look uniformly and independently random; the latter case can be seen as a generalization of correlation-robustness introduced by Ishai et al. (CRYPTO 2003). We give specific applications of these notions to password-based login and efficient search on encrypted data. Our main construction achieves them (without random oracles) for inputs related by *polynomials* over the input space (namely $\mathbb{Z}_p$ for a prime number p), based on corresponding variants of the q-Diffie Hellman Inversion assumption. Additionally, we show relations between correlated-input secure hash functions and cryptographic primitives secure under related-key attacks. Using our techniques, we are also able to obtain a host of new results for such related-key attack secure cryptographic primitives.

1 Introduction

In practice, it is often useful to view a cryptographic hash function like a random oracle, as formalized in the random oracle model [BR93]. However, as random oracles do not exist in reality (and indeed, in general, the random oracle model may lead to insecure schemes [CGH04]), an important line of research suggested by [CGH04] seeks to formalize various useful properties satisfied by a random oracle and construct hash functions meeting them under standard assumptions. In this paper, we do so for what we call *correlated-input security*, meaning that (various notions of) security should be maintained when the adversary sees hash values of many related high-entropy inputs.

The importance of correlated-input security in practice is illustrated by the so-called *avalanche effect*, a well-known heuristic in cryptographic hash function design. (The name "avalanche effect" was coined by Feistel [Fei73], although the idea goes back to Shannon's notion of diffusion [Sha49].) Roughly, the avalanche effect states that making any change to an input should result in a drastically different hash value. Clearly, such a hash function should satisfy a notion of correlated-input security. Our results help to shed light on whether or not this is feasible from a theoretical perspective.

1.1 Notions of Correlated-Input Security

We get different specific notions of correlated-input security depending on how we interpret "security." We first discuss the different interpretations we consider and and how we formalize the resulting notions.

*Work done in part while visiting Microsoft Research, India.

Three Notions The first and most basic interpretation we consider is "one-wayness." To formalize one-wayness under correlated-inputs, we consider a hash function H and circuits $C_1, \ldots, C_n$, where each C_i takes as input some random coins and outputs a point in the domain of H. The adversary is given hash values $H(x_1), \ldots, H(x_n)$, where each x_i is the output of $C_i(r)$ for random coins r. Note that each C_i is run on the *same* random coins. Therefore, the x_i are correlated.[1] The adversary's goal is to output an x' such that $H(x') = H(x_i)$ for some i. Informally, we say that H is one-way under correlated inputs for a class of circuits $\{C\}$ if, for any n and any choice of $C_1, \ldots, C_n$ from $\{C\}$, any efficient adversary succeeds with small (negligible) probability.

The next interpretation we consider is "unpredictability." To formalize unpredictabililty under correlated inputs, we consider a hash function H and circuits $C_1, \ldots, C_{n+1}$, where each C_i is as before. Now the adversary is given hash values $H(x_1), \ldots, H(x_n)$ and tries to output $H(x_{n+1})$, where each x_i is $C_i(r)$, the value output by the circuit, as explained before. The notion is defined for a class of circuits $\{C\}$ analogously to the one-wayness case. It mainly serves as a stepping-stone to our final notion, described next.

Finally, the last interpretation we consider is "pseudorandomness." To formalize pseudorandomness under correlated-inputs, we consider a hash function H and circuits $C_1, \ldots, C_{n+1}$, each C_i is as before. Now the adversary is given hash values $H(x_1), \ldots, H(x_n)$ as well as a "challenge" value that is either $H(x_{n+1})$ or a random string of appropriate length, where each x_i is the output of $C_i(r)$ as before. (This of course requires the circuits to have distinct outputs.) Again, the notion is also defined with respect to a class of circuits $\{C\}$ analogously.

Discussion We make a few observations about these notions. One is that they are only achievable for a class of circuits $\{C\}$ such that $C(r)$ for random r has sufficient min-entropy for any C in the class. In fact, it is not hard to show that a random oracle satisfies our notions for the class of all such circuits. However, in the standard model they are in general only achievable by a *keyed* hash function H. To see this, fix an unkeyed hash function H and consider circuits C_1, C_2 where $C_1(r)$ outputs r and $C_2(r)$ outputs $H(r)$. Clearly, no fixed H is even one-way under correlated-inputs for these circuits. By a similar argument, the circuits must not depend on the choice of the hash key. We stress that in our notions the hash function key is *public*. Similar counter-examples show that in general pseudorandom generators and functions do not necessarily meet our notions (note that the latter has a *secret* key) (see Appendix C for more details).

Another important point is that when considering non-uniform inputs, these notions are non-standard even in the case of a single input. This case has been a subject of prior work, however. For example, it is known that a one-way function that is sufficiently hard to invert on uniform inputs is also hard to invert on non-uniform inputs with enough min-entropy, and [DP08, RTTV08] show a similar (but more technically challenging) result for pseudorandom generators. Additionally, we note that, in our setting of correlated inputs, for any *a priori bounded* number of circuits, pseudorandomness can be met statistically (i.e., against unbounded adversaries). This follows from a generalization of the classical Leftover Hash Lemma [HILL99] to a bounded number of correlated sources as in [O'N10, KPSY09]. However, this approach is basically impractical as both the key-size of the hash function as well as the min-entropy requirement on each individual input depend on the bound. We emphasize that the focus of our work is therefore on an *unbounded* number of correlated inputs in the *computational* setting. Indeed, some of our results concern the natural case where each (correlated) input is individually uniform, which isolates the issue of correlations from the orthogonal one of non-uniformity.

1.2 Applications

We now discuss some specific practical applications for our new notions.

[1] For example, the x_i's might agree in most bit positions but vary in the others. It may even be the case that a single input x_i completely determines the rest.

Password-based login An application of our notion of one-wayness under correlated-inputs is password-based login. For example, UNIX maintains a "password" file that, for each user in the system, stores a hash of his password. Then, when someone claiming to be some user supplies an input for the user's password, the hash value of the supplied input is compared against the stored hash value. If they do not match, he is not allowed to login. Here, the goal is to prevent an adversary with access to the password file from gaining the ability to impersonate a user. Informally, it is often said that the property of the hash function needed to ensure this is one-wayness. But the standard notion of one-wayness is obviously insufficient in such scenarios. Passwords, while they should contain entropy, are certainly not uniformly random. Moreover, passwords are typically *correlated*, both across different users, and across the same user on different systems (and the adversary may recover the password file for multiple systems).

This issue seems to be largely ignored in prior work. A paper (which we already mentioned) that considers the relevance of one-way functions for high entropy inputs to this application is [WG00]; however, they do not consider multiple related inputs and relations among them. Our notion of one-wayness under correlated input seems to be an appropriate security notion for this application.[2]

Efficient search on encrypted data An application of our notion of pseudorandomness under correlated-inputs is efficient search on encrypted data. It is becoming increasingly common for companies to store large amounts of data remotely on servers maintained by an untrusted third party. To provide privacy for the client, the data should be encrypted. However, we still want to allow search on the data without retrieving and decrypting the entire database. Techniques like public-key encryption with keyword search [BCOP04] make search possible, but it takes linear time in the database size. On the other hand, practitioners require search time to be comparable to that for unencrypted data.

This problem was first studied from a cryptographic perspective by Bellare et al. [BBO07], who introduced deterministic encryption and the more general concept of efficiently searchable encryption (ESE) as a solution. The basic idea is to attach a hash of each keyword to an encrypted file. Keywords are obviously not uncorrelated, and thus our notions are natural to apply.

1.3 Our Construction and its Security

Next we turn to whether our security definitions can be achieved and under what cryptographic assumptions.

Our Construction We propose the following construction: Letting G be a group of prime order p, the hash key is a random generator $g \in G$ and random $c \in \mathbb{Z}_p$, and the evaluation of the hash on input $x \in \mathbb{Z}_p$ is $g^{1/(x+c)}$, where $1/(x+c)$ denotes the inverse of $x+c$ modulo p.

Security of the Above Construction We show that this construction is secure under each of our three notions of security assuming (appropriate variants of) the *q-Diffie-Hellman inversion assumption* (q-DHI). Roughly speaking, this assumption says that given $g^{\alpha}, g^{\alpha^2}, \ldots, g^{\alpha^q}$, it is hard to compute $g^{1/\alpha}$. An assumption of this form was first introduced by Boneh and Boyen [BB04a], who considered it in groups with a *a bilinear map* (pairing) e and asked that it be hard to compute (or distinguish from random) $e(g,g)^{1/x}$ instead of $g^{1/x}$. However, since our hash function is deterministic and thus automatically publically verifiable, we do not need bilinear maps here.

The class of circuits we consider in our proofs are the ones that are (efficiently) representable by a polynomial over $\mathbb{Z}_p$ (the input space of the hash function). In other words, each $x_i = f_i(x)$, where f_i is a polynomial over $\mathbb{Z}_p$ and x is randomly chosen but fixed for all i. This is quite a broad class; for instance, taking f_1 to be the identity polynomial covers the well-known attacks on RSA [CFPR96].

[2]For simplicity, this ignores "salting" the passwords, which can be viewed as considering a randomized hash function instead of a deterministic one. However, this circumvents the core issue that in practice a deterministic cryptographic hash function is assumed to satisfy correlated-input security; indeed, salting passwords is only meant to prevent dictionary or "rainbow" attacks.

First, we show that our inversion hash is one-way under correlated inputs for this class of circuits assuming the q-strong discrete logarithm (q-SDL) assumption (which is weaker than q-DHI in that it need only be hard to compute α itself).[3] For unpredictability and psuedorandomness, we require an additional assumption that each input to the hash function is individually uniform.[4] However we stress that given other inputs, an input may even be fully computable. For this class of polynomials, we show the inversion hash is unpredictable under correlated inputs assuming q-DHI. Using standard hardcore bit techniques this already gives a construction with small output length achieving pseudorandomness under correlated inputs with the same assumption. However, we directly show pseudorandomness under correlated inputs of our construction for the same class of polynomials assuming the decisional version of q-DHI.

Discussion Our construction is inspired by the Boneh-Boyen short signature scheme [BB04b] and Dodis and Yampolskiy (verifiable) pseudorandom function [DY05]. In particular, as previously explained by Bellare and Cash [BC10], one can show the latter achieves security under related-key attacks (for certain relation classes), and as we observe below security under related-key attacks turns out to in some sense be a "dual" of correlated-input pseudorandomness (so similarity of the constructions is ultimately not surprising). On the other hand, our results also concern other security notions like one-wayness, pertain to different relation classes than suggested by [BC10], and use more standard assumptions without bilinear maps.

We build on the proof techniques introduced in [BB04b] and incorporate some new ideas. A direct adaptation of the proof techniques from [BB04b] is not sufficient for our purposes. In more detail, in [BB04b], the public value (the message) changes with each evaluation and is chosen by the adversary while in our scenario, the public value (the constant c) is chosen by the challenger and remains the same throughout. The secret value in [BB04b] (the secret key of the signature scheme) remains the same throughout while in our scenario, the secret value (the message x) changes with each evaluation and different messages may be related with each other. Consequently, while our construction is inspired by [BB04b], the actual proof is quite different and the security is proven under a different assumption (i.e., decisional q-DHI as opposed to q-SDH).

We also note that our security proofs are under a notion of "selective" security where the circuits that sample the inputs do not depend on the public hash key. As we mentioned, in general this restriction is inherent. However, for restricted classes of circuits (such as arithmetic circuits we consider) which are not able to efficiently compute the hash function in question, it may be possible to achieve an adaptive notion of security even by using an *unkeyed* hash function. We discuss a positive result for this case below.

Finally, we note that our construction as defined is not compressing. However, once we obtain a construction meeting any of our notions, it is easy to obtain one which is also compressing. In the case of one-wayness we can apply a collision-resistant hash to the output, and in the case of pseudorandomness we can truncate the output. It can be shown that the resulting (compressing) hash function retains correlated-input security.

1.4 Relations to Related-Key Attacks and Additional Results

Security under related-key attacks (RKA), first formalized by [BK03] in the context of pseudorandom functions/permutations, is a well-established notion that, like correlated-input security, asks for security to be maintained under related values of a "secret" input. We explore relations between hash functions for which outputs satisfy pseudorandomness under correlated-inputs (which we simply call CI-secure hash functions from hereon) and RKA-security of various cryptographic primitives.

[3] In fact, for this result the c component of the hash key can be any fixed element in $\mathbb{Z}_p$ (for instance, set $c = 0$).

[4] This translates to the requirement that the polynomials individually have uniform output on uniform input (e.g., this is the case for permutation polynomials). By making non-standard assumptions, it may be possible to drop this restriction. However, considering individually uniform but correlated inputs to the hash function is natural, and we focus on results under standard assumptions.

Equivalence to One-Input RKA-wPRF We first observe that a CI-secure hash is in some sense equivalent to what we call a "one-input RKA-secure weak pseudorandom function (RKA-wPRF)." To define such a wPRF F, the adversary is given an input-output pair $(x, F(K, x))$ for random x and may query for for other outputs of the form $F(\phi(K), x)$ for relations ϕ of its choosing. (Note that the same x is re-used each time.) Following [HLO10], we note that as compared to a CI-secure hash, the role of the key and the input are simply "switched."[5] Note that a one-input RKA-wPRF is implied by an RKA-PRF; in fact, if we start with an RKA-PRF (rather than wPRF), the resulting CI-secure hash *does not need a public key.* The latter is significant because the recent breakthrough work of Bellare and Cash [BC10] gives RKA-secure PRFs under the decisional Diffie-Hellman assumption for adaptively chosen, group-induced relations (i.e., multiplication by a constant). We thus obtain an unkeyed adaptively-secure CI-secure hash for the corresponding class of circuits.

A General Transformation from CI-hash to other RKA-secure cryptographic primitives Additionally, we propose a transformation to "bootstrap" any cryptographic primitive to one that is RKA-secure: simply hash the coins used to generate the secret key for the former. (This can be seen as replacing a random oracle (RO) in this transformation with a CI-secure hash.) Note, however, that in the case the CI-secure hash has a public key, an authentic version of the latter is then needed by any algorithm that uses the secret key (e.g., the signing algorithm for a signature scheme), which may not always be practical.

Adaptively secure RKA-secure weak PRF and symmetric key encryption Our main construction of CI-secure hash is selectively secure, leading to selective security under RKA (i.e., the adversary chooses the relations before seeing the public parameters). However, by using our techniques in a non-blackbox way, we can sometimes achieve adaptive security instead and without any public key. In particular, in Appendix 6 we show how to do this for "multi-input" RKA-wPRFs (where a fresh random input is selected for each query). Building on that construction, we further show how to construct a RKA-secure symmetric encryption, a primitive introduced concurrently to our work in [AHI10];[6]. For these primitives we are able to handle the entire class of (non-zero) polynomials over $\mathbb{Z}_p$. Note that the construction of RKA-secure symmetric encryption provided in [AHI10] could only handle the classes of correlations represented by additions in Z_p.

1.5 Other Related Work

Our work is related to several other lines of research, as we now discuss.

Realizing Random Oracles As we mentioned earlier, our work can be seen as extending a research agenda proposed by Canetti, Halevi, and Goldriech [CGH04], in which one identifies and realizes useful properties of a RO. Indeed, by using the techniques of [BBO07, Theorem 5.1] one can show that a RO meets all the security definitions we consider (which is why we have sought realizations under standard cryptographic assumptions). Other useful properties of a RO that have undergone a similar treatment include perfect one-wayness [Can97, CMR98] and non-malleability [BCFW09]. We note that none of these works address security under multiple correlated inputs.

In fact, a significant prior work of Ishai et al. [IKNP03, Definition 1] considered a notion of "correlation-robustness" for pseudorandom hash functions, with the motivation of instantiatng the RO in their oblivious transfer protocols. Their notion is more restrictive than our notion of pseudorandomness under correlated-input, as the former is defined only for hash functions that output a single bit and considers inputs obtained by computing the exclusive-or's of a "master" random input s with public random values.

[5] Note, however, in correlated-input security it may be the case that none of the considered circuits output the identity, whereas in related-key security the identity relation is always considered.

[6] We obtained this result after seeing [AHI10], but the rest of our work was concurrent.

Finally, we note that while realizing the "avalanche effect" satisfied by a RO forms a major motivation for considering correlated-input security, it is not the only way the latter could be formalized. In particular, it talks only about the change in the output behavior relative to any change to an *unknown* input. The notion of "(multiple input) correlation intractability" due to [CGH04] is a possible formalization the effect without this restriction. On the other hand, the latter notion seems harder to work with and more difficult to achieve.

Deterministic Encryption Our security notions can be viewed as relaxations or variants of the notions of privacy proposed for deterministic encryption (DE) in [BBO07, BFO08, BFOR08, O'N10] (that seek to hide partial information) in the case of hash functions rather than encryption schemes. Indeed, the results of of [BBO07, BFO08, BFOR08, O'N10] show that in some sense the "hard part" of realizing DE without random oracles is dealing with correlations among the inputs. Our work studies this issue at a more basic level, asking whether it is feasible even without supporting decryption and for weaker security notions like one-wayness.

Related Security Notions Recently, Rosen and Segev [RS09] introduced the notion of *correlated-product secure* trapdoor functions (TDFs). Correlated-product security was later considered for hash functions in [HLO10]. Correlated-product security differs from our notions in that the former refers to security when related inputs are evaluated under *independent instances* of the function; in other words, there does not exist a single function which is evaluated on related inputs (as is the case for correlated-input security). Indeed, our techniques are quite different and unrelated to those in [RS09, HLO10].

The recent work of Goldenberg and Liskov [GL10] also considers a form of correlated-input security (which they call "related-secret" security) for various primitives. While their work has some similarities to ours (for example, they consider related-secret one-way functions), there are some important differences. Namely, they focus on hardcore bits and pseudorandom functions rather than hash functions, and they follow the definitional framework for related-key attacks introduced in [BK03] (indeed the latter are the main motivation for their work). Additionally, their results are mainly negative in nature.

Independent Work Independently of our work, Bellare, Cash, and Miller [BCM11] undertake a more general study of RKA security for various cryptographic primitives including wPRFs, PRFs, signatures, and encryption (symmetric, public-key, and identity-based). They study relations between these primitives, what classes of relations are achievable for them in principle, and also show applications to security under key-dependent messages. The overlap to our work is that they also consider RKA security for wPRFs and signatures, and to achieve RKA-secure signatures they use a construction similar to our "coin hashing" one (but instead of a hash function they use a form of RKA-secure PRG).

2 Preliminaries

Notations. For a finite set X, $|X|$ denotes the size of the set X. $|x|$ denotes the length of a binary string x. $x \xleftarrow{\$} X$ denotes the operation of selecting a random element x from X. $x \leftarrow y$ denotes assignment of a value y to x. Let $\mathsf{FF}(X, Y, Z)$ be the set of all families of functions $F : X \times Y \longrightarrow Z$. For sets X, Y, let $\mathsf{Fun}(X, Y)$ be the set of all functions mapping X to Y. For brevity, we say that an algorithm outputs a set/function as a shorthand to mean that it outputs their descriptions. Let λ denote the security parameter.

Complexity Assumptions. We first state our complexity assumptions, namely q-DHI [BB04a, BB04b], which is weaker than q-BDHI [BB04a] as well as q-SDH [BB04b], and we introduce what we call the q-Strong Discrete Logarithm (q-SDL) assumption which is weaker than all of q-DHI, q-BDHI, and q-SDH assumptions.

Let GrpGen be a PPT algorithm that takes as input the security parameter 1^λ and outputs parameters for some cyclic multiplicative group $\mathbb{G}$, including the group order p which is a $poly(\lambda)$-bit integer, a

generator g, and an efficient algorithm (e.g., circuit) for multiplication (and thus also exponentiation). We denote it as $(\mathbb{G}, p, g) \leftarrow \mathsf{GrpGen}(1^\lambda)$.

q-Strong Discrete Logarithm (q-SDL) Problem The q-SDL problem in $\mathbb{G}$ is defined as follows: given a $(q+1)$-tuple $(g, g^x, g^{x^2}, \ldots, g^{x^q}) \in (\mathbb{G}^\star)^{q+1}$ for some unknown $x \in \mathbb{Z}_p^\star$, output x.
An algorithm $\mathcal{A}$ solves the q-SDL problem in the group $\mathbb{G}$ with advantage ϵ if

$$\mathsf{SDL\text{-}Adv}_{\mathcal{A},q} := \Pr[\mathcal{A}(g, g^x, g^{x^2}, \ldots, g^{x^q}) = x] \geq \epsilon$$

where the probability is over the random choice of generator $g \in \mathbb{G}^\star$, the random choice of $x \in \mathbb{Z}_p^\star$, and the random bits consumed by $\mathcal{A}$.

Definition 1 *We say that the (q, t, ϵ)-SDL assumption holds in $\mathbb{G}$ (or GrpGen satisfies the (q, t, ϵ)-SDL assumption) if no probabilistic t-time algorithm has advantage at least ϵ in solving the q-SDL problem in $\mathbb{G}$.*

It is easy to see that the 1-SDL assumption is equivalent to the standard Discrete Logarithm assumption.

q-Diffie-Hellman Inversion (q-DHI) Problem The q-DHI problem [BB04a, BB04b] in $\mathbb{G}$ is defined as follows: given a $(q+1)$-tuple $(g, g^x, g^{x^2}, \ldots, g^{x^q}) \in (\mathbb{G}^\star)^{q+1}$ for some unknown $x \in \mathbb{Z}_p^\star$, output $g^{\frac{1}{x}} \in \mathbb{G}$.
An algorithm $\mathcal{A}$ solves the q-DHI problem in the group $\mathbb{G}$ with advantage ϵ if

$$\mathsf{DHI\text{-}Adv}_{\mathcal{A},q} := \Pr[\mathcal{A}(g, g^x, g^{x^2}, \ldots, g^{x^q}) = g^{\frac{1}{x}}] \geq \epsilon$$

where the probability is over the random choice of generator $g \in \mathbb{G}^\star$, the random choice of $x \in \mathbb{Z}_p^\star$, and the random bits consumed by $\mathcal{A}$.

Definition 2 *We say that the (q, t, ϵ)-DHI assumption holds in $\mathbb{G}$ (or GrpGen satisfies the (q, t, ϵ)-DHI assumption) if no probabilistic t-time algorithm has advantage at least ϵ in solving the q-DHI problem in $\mathbb{G}$.*

q-DHI Problem can be equivalently stated as follows [BB04b]: given a $(q+2)$-tuple $(g, g^x, g^{x^2}, \ldots, g^{x^q}, c) \in (\mathbb{G}^\star)^{q+1} \times \mathbb{Z}_p \backslash \{-x\}$ for some unknown $x \in \mathbb{Z}_p^\star$, output $g^{\frac{1}{x+c}} \in \mathbb{G}$. This definition also clearly points out the distinction between the q-DHI problem and the q-SDH problem: in case of the q-DHI problem, the value of c is prescribed in the problem instance itself, whereas in case of the q-SDH problem, the solver is free to choose $c \in \mathbb{Z}_p$ and output a pair, $(c, g^{\frac{1}{x+c}})$. Obviously, the q-DHI assumption is weaker than the q-SDH assumption.

Decisional q-Diffie-Hellman Inversion (Decisional q-DHI) Problem The decisional q-DHI problem in $\mathbb{G}$ is defined as follows: given a $(q+1)$-tuple $(g, g^x, g^{x^2}, \ldots, g^{x^q}) \in (\mathbb{G}^\star)^{q+1}$ for some unknown $x \in \mathbb{Z}_p^\star$, distinguish between $g^{\frac{1}{x}}$ and a random element $R \xleftarrow{\$} \mathbb{G}$.
An algorithm $\mathcal{A}$ solves the decisional q-DHI problem in $\mathbb{G}$ with advantage ϵ if

$$\begin{aligned}\mathsf{DDHI\text{-}Adv}_{\mathcal{A},q} :=& |\Pr[\mathcal{A}(g, g^x, g^{x^2}, \ldots, g^{x^q}, g^{\frac{1}{x}}) = 1] \\ & - \Pr[\mathcal{A}(g, g^x, g^{x^2}, \ldots, g^{x^q}, R) = 1]| \\ \geq& \epsilon\end{aligned}$$

where the probability is over the random choice of generator $g \in \mathbb{G}^\star$, the random choice of $x \in \mathbb{Z}_p^\star$, the random choice of $R \in \mathbb{G}^\star$, and the random bits consumed by $\mathcal{A}$. The distribution on the left is referred to as $\mathcal{P}_{\mathcal{DDHI}}$ and the distribution on the right as $\mathcal{R}_{\mathcal{DDHI}}$.

Definition 3 *We say that the decisional (q, t, ϵ)-DHI assumption holds in $\mathbb{G}$ (or GrpGen satisfies the decisional (q, t, ϵ)-DHI assumption) if no probabilistic t-time algorithm has advantage at ϵ in solving the decisional q-DHI problem in $\mathbb{G}$.*

3 Our Model: Correlated Input Security

In this section we define our new notion of security for cryptographic hash functions. We would be interested in preserving various properties of hash functions (like one-wayness and pseudo-randomness) when the function maybe evaluated on a tuple of inputs which maybe be *correlated* in an arbitrary way. Standard notions of security do not provide any guarantee in such a setting. In Appendix C, we discuss examples of functions which are secure in the standard sense but may be completely insecure when evaluated on multiple inputs which are correlated.

Before we go further, we first discuss how we represent correlations among a tuple of inputs $(m_1, \ldots, m_n)$. In general, such an input tuple maybe generated by a polynomial-size sampler circuit $Samp$. In other words, $Samp$ takes a random tape r (of appropriate length) as input and outputs $(m_1, \ldots, m_n) \leftarrow Samp(r)$. Note such a sampler circuit can generate the input tuple for any type of polynomial-time computable correlations. Equivalently, one can generate the (correlated) tuple of inputs using a *tuple* of polynomial-size circuits $(C_1, \ldots, C_n)$ when initialized on the same random tape. In other words, fix a random string r and set $m_i \leftarrow C_i(r)$. It is easy to see that both these sampling procedures are equivalent. For the rest of the paper, we shall stick to the latter mode of using a tuple of circuits.

Also, it will be understood that the range of every circuit considered is a subset of the input-space (or keyspace) in question. We refer to an adversary as $\{C\}$-restricted, if every circuit it queries belongs to the class $\{C\}$.

We first define the syntax for a general function family (or a hash function family if the functions are compressing). We will then move on to formalize the various security properties such a function family might satisfy.

Definition 4 (Function Family) *A family of deterministic functions $\mathcal{H}$ is specified by a PPT algorithm Gen. The algorithm Gen, given input 1^λ, outputs a parameter set $\mathcal{I}_h$, domain $\mathcal{D}_h$, and range $\mathcal{R}_h$, and outputs $c \in \mathcal{I}_h$ as a description of a function $h_c : \mathcal{D}_h \longrightarrow \mathcal{R}_h$. The sizes of the domain and range sets are each exponential in the security parameter.*

Now, we shall discuss our first notion of security called *correlated-input one-wayness*. Informally, we consider a function $h(\cdot)$ such that given $(h(m_1), \ldots, h(m_n))$, where inputs $(m_1, \ldots, m_n)$ maybe correlated, it is hard for any PPT adversary to output any valid preimage m_i. This can be viewed as a generalization of the standard notion of one-way functions. We allow the adversary to specify the correlations to the challenger by giving a tuple of circuits $(C_1, \ldots, C_n)$, where each circuit is from a class of correlated-input circuits $\{C\}$. Note that, for this definition to be satisfiable, each circuit $C_i, \in \{C\}$ individually should have high min-entropy output distribution for uniform random input distribution. [7] Thus, we quantify only over such circuits in our definition. We discuss it under both selective and adaptive security notions. More details follow.

In the following we shall only formalize the selective notion. The definitions for adaptive notions are given in Appendix A.

The Selective Correlated-Input Inverting experiment $\mathbf{Exp}^{sCI-inv}_{\mathcal{A},\mathcal{H},\{C\}}$. For a family of deterministic functions $\mathcal{H}$, an adversary $\mathcal{A}$, and a family of efficiently-computable correlated-input circuits $\{C\}$, we define the following game between a challenger and the adversary $\mathcal{A}$.

[7] This requirement is similar to one in the standard notion of one-way functions. If the input does not have sufficient min-entropy, it is easy to see that an adversary can guess a preimage and succeed with noticeable probability.

- **Setup Phase 1**. Challenger runs the Gen algorithm of $\mathcal{H}$ for a security parameter input 1^λ and gets $h_c : \mathcal{D}_h \longrightarrow \mathcal{R}_h$. Challenger gives $\mathcal{D}_h$ to $\mathcal{A}$.
- **Query Phase**. $\mathcal{A}$ chooses a positive integer n $(= poly(\lambda))$, and gives to the challenger n circuits $\{C_i\}_{i\in[n]} \subset \{C\}$.
- **Setup Phase 2**. Challenger gives $h_c(\cdot)$ to $\mathcal{A}$ and chooses r, a uniform random string of appropriate length.
- **Response Phase**. $\forall i \in [n]$, challenger responds via $h_c(C_i(r))$.
- **Invert Phase**. $\mathcal{A}$ outputs $(\hat{k}, \hat{y})$ for $\hat{k} \in [n]$ and $\hat{y} \in \mathcal{D}_h$.

The output of the experiment is defined to be 1 if $h_c(\hat{y}) = h_c(C_{\hat{k}}(r))$ and 0 otherwise.
We define the advantage of an adversary $\mathcal{A}$ in the above game as:

$$\mathsf{Adv}^{sCI-inv}_{\mathcal{A},\mathcal{H},\{C\}}(\lambda) = \Pr[\mathsf{Exp}^{sCI-inv}_{\mathcal{A},\mathcal{H},\{C\}} = 1]$$

The probability is over the random bits used by the challenger and the adversary.

Definition 5 *A family of functions $\mathcal{H}$ is said to be selective correlated-input one-way with respect to a family of correlated-input circuits $\{C\}$, if for all $\mathcal{A} \in PPT$ there exists a negligible function negl, such that:*

$$Adv^{sCI-inv}_{\mathcal{A},\mathcal{H},\{C\}}(\lambda) \leq negl(\lambda)$$

We now consider two more correlated-input security notions where we talk about unpredictability of the *output* as opposed to that of the input. Informally, we consider a function $h_c : \mathcal{D}_h \longrightarrow \mathcal{R}_h$ with the following properties. Consider a tuple of correlated inputs $(m_1, \ldots, m_{n+1})$. The adversary is given the function outputs $(h_c(m_1), \ldots, h_c(m_n))$ and it tries to compute $h_c(m_{n+1})$. In the first security notion called *correlated-input unpredictability* (CI-unpredictability), we require that it should be hard for the adversary to output $h_c(m_{n+1})$. In the next notion called *correlated-input pseudorandomness* (CI-pseudorandomness), we require that the adversary should not be able to distinguish $h_c(m_{n+1})$ from a random element in $\mathcal{R}_h$, given $(h_c(m_1), \ldots, h_c(m_n))$. It is easy to show that this notion of CI-pseudorandomness is equivalent to a notion where an adversary gets either $(h_c(m_1), \ldots, h_c(m_{n+1}))$ or $n+1$ independent random elements in $\mathcal{R}_h$ and is required to distinguish the two cases. Note that, for any of these notions to be satisfiable, besides the requirement that each circuit $C_i \in \{C\}$ individually should have high min-entropy output distribution for uniform random input distribution, we also require that, for every two distinct circuits C_i and C_j in $\{C\}$, and for a uniform random input r of appropriate length, $C_i(r) = C_j(r)$ happens only with negligible probability over the choice of r.

In trying to give more power to the adversary (thus making our definition stronger), we allow the adversary to specify the correlation by giving a tuple of circuits $(C_1, \ldots, C_{n+1})$, where $C_i \in \{C\}$, to the challenger, as before. In addition, for the selective case, for a tuple of inputs $(m_1, \ldots, m_{n+1})$, we allow the adversary to get outputs on n *adaptively chosen* input indices before trying to predict the remaining output.[8] More details follow.

[8] By incurring a security loss of a factor of n, this definition can actually be shown to be equivalent to a weaker definition where the adversary is required to predict the output specifically on input m_{n+1} fixed after it presents its queries but before it reveives the responses. However, working directly with this definition might lead to better concrete security guarantees in the real world.

The Selective Correlated-Input Predicting experiment $\mathsf{Exp}^{sCI-pred}_{\mathcal{A},\mathcal{H},\{C\}}$. For a family of deterministic functions $\mathcal{H}$, an adversary $\mathcal{A}$, and a family of efficiently-computable correlated-input circuits $\{C\}$, we define the following game between a challenger and the adversary $\mathcal{A}$.

- **Setup Phase 1**. Challenger runs the Gen algorithm of $\mathcal{H}$ for a security parameter input 1^λ and gets $h_c : \mathcal{D}_h \longrightarrow \mathcal{R}_h$. Challenger gives $\mathcal{D}_h$ to $\mathcal{A}$.
- **Query Phase**. $\mathcal{A}$ chooses a positive integer n $(= poly(\lambda))$, and gives to the challenger $n+1$ distinct circuits $\{C_i\}_{i\in[n+1]} \subset \{C\}$.
- **Setup Phase 2**. Challenger gives $h_c(\cdot)$ to $\mathcal{A}$ and chooses r, a uniform random string of appropriate length.
- **Partially Adaptive Query-Response Phase**. $\mathcal{A}$ presents n queries, where an ith query is an index $k_i \in [n+1]$. Challenger responds to it via $h_c(C_{k_i}(r))$.
- **Predict Phase**. The adversary outputs $\hat{y} \in \mathcal{R}_h$.

Let the index corresponding to the unqueried circuit be k_{n+1} (i.e., $k_{n+1} \in [n+1]$ be such that $k_{n+1} \neq k_i$ $\forall\ i \in [n]$). The output of the experiment is defined to be 1 if $\hat{y} = h_c(C_{k_{n+1}}(r))$ and 0 otherwise.
We define the advantage of an adversary $\mathcal{A}$ in the above game as:

$$\mathsf{Adv}^{sCI-pred}_{\mathcal{A},\mathcal{H},\{C\}}(\lambda) = \Pr[\mathsf{Exp}^{sCI-pred}_{\mathcal{A},\mathcal{H},\{C\}} = 1]$$

The probability is over the random bits used by the challenger and the adversary.

Definition 6 *A family of functions $\mathcal{H}$ is said to be selective correlated-input unpredictable with respect to a family of correlated-input circuits $\{C\}$, if for all $\mathcal{A} \in \mathsf{PPT}$ there exists a negligible function negl, such that:*

$$\mathsf{Adv}^{sCI-pred}_{\mathcal{A},\mathcal{H},\{C\}}(\lambda) \leq negl(\lambda)$$

The Selective Correlated-Input Distinguishing experiment $\mathsf{Exp}^{sCI-dist}_{\mathcal{A},\mathcal{H},\{C\}}(b)$. For a family of deterministic functions $\mathcal{H}$, an adversary $\mathcal{A}$, and a family of efficiently-computable correlated-input circuits $\{C\}$, and a random bit b, we define the following game between a challenger and the adversary $\mathcal{A}$.

- **Setup Phase 1**. Challenger runs the Gen algorithm of $\mathcal{H}$ for a security parameter input 1^λ and gets $h_c : \mathcal{D}_h \longrightarrow \mathcal{R}_h$. Challenger gives $\mathcal{D}_h$ to $\mathcal{A}$.
- **Query Phase**. $\mathcal{A}$ chooses a positive integer n $(= poly(\lambda))$, and gives to the challenger $n+1$ distinct circuits $\{C_i\}_{i\in[n+1]} \subset \{C\}$.
- **Setup Phase 2**. Challenger gives $h_c(\cdot)$ to $\mathcal{A}$ and chooses r, a uniform random string of appropriate length.
- **Partially Adaptive Query-Response Phase**. $\mathcal{A}$ presents n queries, where an ith query is an index $k_i \in [n+1]$. Challenger responds to it via $h_c(C_{k_i}(r))$.
- **Challenge Phase**. Let $k_{n+1} \in [n+1]$ be the index of the unqueried circuit. Let $z_0 \xleftarrow{\$} \mathcal{R}_h$ and $z_1 := h_c(C_{k_{n+1}}(r))$. Challenger gives z_b to $\mathcal{A}$.
- **Guess Phase**. The adversary outputs a guess $\hat{b}$ of b.

$\hat{b}$ is defined to be the output of the experiment.
We define the advantage of an adversary $\mathcal{A}$ in the above game as:

$$\mathsf{Adv}^{sCI-dist}_{\mathcal{A},\mathcal{H},\{C\}}(\lambda) = |\Pr[\mathsf{Exp}^{sCI-dist}_{\mathcal{A},\mathcal{H},\{C\}}(1) \longrightarrow 1] - \Pr[\mathsf{Exp}^{sCI-dist}_{\mathcal{A},\mathcal{H},\{C\}}(0) \longrightarrow 1]|$$

The probability is over the random bits used by the challenger and the adversary.

Definition 7 *A family of functions $\mathcal{H}$ is said to be selective correlated-input pseudorandom with respect to a family of correlated-input circuits $\{C\}$, if for all $\mathcal{A} \in PPT$ there exists a negligible function negl, such that:*

$$Adv^{sCI-dist}_{\mathcal{A},\mathcal{H},\{C\}}(\lambda) \leq negl(\lambda)$$

In general, we sometimes refer CI-pseudorandom functions as CI-secure functions.

4 Proposed Construction

In the sequel we give the construction of our function and prove that it is correlated-input secure for a class of polynomials over $\mathbb{Z}_p$ (where p is a prime number) in the sense of each of the three selective security models defined above.

Our construction is given in Figure 1.

$\mathsf{Gen}(1^\lambda)$. Run GrpGen: $(\mathbb{G}, p, g) \leftarrow \mathsf{GrpGen}(1^\lambda)$, where p is a prime number. Gen uniformly samples a random element c from $\mathbb{Z}_p$ and a random generator g of group $\mathbb{G}$. It outputs g, c and a function $h : \mathbb{Z}_p \longrightarrow \mathbb{G}$ defined by,

$$h(m) := g^{\frac{1}{m+c}}$$

for any $m \in \mathbb{Z}_p$ (where $\frac{1}{m+c}$ is computed $\mod p$).

Figure 1: Our Construction

Our proposed function is extremely simple and efficient to compute. The cost of computation is dominated by a single exponentiation operation. The construction can be seen as similar to a short signature scheme by Boneh and Boyen [BB04b]. Our main novelty can be seen in the proofs of security. Indeed, interestingly, our proofs show that the original signature scheme of Boneh and Boyen is secure even if an adversary is allowed to obtain messages signed by various *correlated secret signing keys*, where the correlations are from a set of polynomials over $\mathbb{Z}_p$. A detailed description of this implication is given in Appendix D.

4.1 Analysis of the above Construction.

We prove that our construction is selectively secure with respect to a class of correlations computable by polynomials over $\mathbb{Z}_p$. In what follows, we introduce some more notations that will be used in the rest of the paper and then discuss the three security games with correlated-input circuits computing polynomials over $\mathbb{Z}_p$.

Notations Let $deg[f(X)]$ denote the degree of a polynomial $f(X)$ over $\mathbb{Z}_p$. If the output distribution of a polynomial is uniform in $\mathbb{Z}_p$ (or, in other words, if the range of the polynomial is $\mathbb{Z}_p$ itself), then we refer to the polynomial as uniform-output polynomial. On the other hand, if the output distribution of

a polynomial has high min-entropy in $\mathbb{Z}_p$, then we refer to the polynomial as high-min-entropy-output polynomial (every non-zero polynomial of degree polynomial in the λ has a range of size exponential in the λ). We shall only consider polynomials of degree at least 1 and polynomial in λ.

Now we go in more detail. In the selective Correlated-Input Inverting experiment, we let the adversary choose any polynomially many non-zero polynomials over $\mathbb{Z}_p$, $\{f_i(X)\}_i \subset \mathbb{Z}_p[X]$. Let n denote the total number of queries (where, we emphasize, the value of n is adversarially chosen as opposed to the case in any n-wise independent functions where the maximum number of queries is fixed as n for the function). Once the challenger receives these n polynomials, it chooses $x \xleftarrow{\$} \mathbb{Z}_p$, and computes the inputs as $m_i := f_i(x), \forall i \in [n]$. And the rest of the experiment remains the same.

Similarly, in the selective Correlated-Input Predicting and distinguishing experiments, the adversary chooses $n+1$ distinct uniform-output polynomials $\{f_i(X)\}_{i\in[n+1]}$ (again, for an adversarially chosen value of n). The challenger receives these $n+1$ polynomials from the adversary, chooses $x \xleftarrow{\$} \mathbb{Z}_p$ and computes the inputs as $m_i := f_i(x), \forall i \in [n+1]$. And the rest of the experiment remains the same.

We now propose a simple lemma that would be used later in our proofs.

Lemma 1 *Let p be a prime number of size λ and $f_1(X)$ and $f_2(X)$ be two distinct univariate polynomials over $\mathbb{Z}_p$ such that $deg[f_1(X)]$ and $deg[f_2(X)]$ are each polynomial in λ. Then, for $r \xleftarrow{\$} \mathbb{Z}_p$, the probability that $f_1(r) = f_2(r)$ is negligible in λ.*

PROOF: The values of $x \in \mathbb{Z}_p$ for which $f_1(x) = f_2(x)$ are precisely the roots of the polynomial $f_1(X) - f_2(X)$ in $\mathbb{Z}_p$, and $f_1(X) - f_2(X)$ has at most $deg[f_1(X) - f_2(X)]$ roots in $\mathbb{Z}_p$. As $deg[f_1(X) - f_2(X)] \leq \max(deg[f_1(X)], deg[f_2(X)])$,

$$\Pr[r \xleftarrow{\$} \mathbb{Z}_p : f_1(r) = f_2(r)] \leq \frac{\max(deg[f_1(X)], deg[f_2(X)])}{p}$$

which is negligible in λ. ■

4.2 Selective Correlated-Input one-wayness

Theorem 2 *Suppose that (q, t', ϵ)-SDL assumption holds in $\mathbb{G}$. Let $\{C\}$ be a set of non-zero polynomials over $\mathbb{Z}_p$. Then, for $\mathcal{H}$ as in Figure 1, there exists no probabilistic t-time adversary $\mathcal{A}$ for which $\mathsf{Adv}^{sCI-inv}_{\mathcal{A},\mathcal{H},\{C\}}(\lambda)$ is at least ϵ provided that*

$$d \leq q \text{ and } t \leq t' - \Theta(nq\tau)$$

where $d = d(\lambda)$ upper bounds the sum of the degrees of the polynomials that $\mathcal{A}$ queries upon and τ is the maximum time for an exponentiation in $\mathbb{G}$ and $\mathbb{Z}_p$.

PROOF: Let there exist a polynomial-time adversary $\mathcal{A}$ with an advantage ϵ in the selective Correlated-Input Inverting game with respect to the class of non-zero polynomials over $\mathbb{Z}_p$. We build an algorithm $\mathcal{B}$ that interacts with $\mathcal{A}$ and solves a given random instance of q-SDL problem, with the same advantage ϵ, as follows.

Algorithm $\mathcal{B}$ is given a random instance $(g', (g')^x, (g')^{x^2}, \ldots, (g')^{x^q}) \in (\mathbb{G}^\star)^{q+1}$ of the q-SDL problem in $\mathbb{G}$, for some unknown $x \in \mathbb{Z}_p^\star$. The objective of $\mathcal{B}$ is to output x. $\mathcal{B}$ invokes $\mathcal{A}$ and interacts with it as follows.

Let us denote $(g')^{x^i}$ by g_i for $i \in [q]$, and g' by g_0.

Setup Phase 1: $\mathcal{B}$ gives p to $\mathcal{A}$.

Query Phase: $\mathcal{A}$ chooses a positive integer n ($= poly(\lambda)$), and gives to the challenger, n non-zero polynomials over $\mathbb{Z}_p$, $\{f_i(X)\}_{i\in[n]}$.

Setup Phase 2: $\mathcal{B}$ uniformly chooses a random element $c \xleftarrow{\$} \mathbb{Z}_p$.

Let us define a polynomial $f(X) := \prod_{i=1}^{n}(f_i(X) + c)$. Recall that $\sum_{i=1}^{n} deg[f_i(X)] \leq d$. So $f(X)$ can be expanded as $f(X) = \sum_{j=0}^{d} \alpha_j X^j$ where $\alpha_j \in \mathbb{Z}_p$ are the co-efficients of the polynomial $f(X)$. $\mathcal{B}$ computes $\prod_{j=0}^{d}(g_j)^{\alpha_j}$ and sets:

$$g \longleftarrow \prod_{j=0}^{d}(g_j)^{\alpha_j}.$$

Thus $g = (g')^{f(x)}$. g has correct distribution provided that $f(x) \neq 0$ (in other words, g is a random generator of $\mathbb{G}$). If, however, $f(x) = 0$ (or, equivalently, if x is a root of $f(X)$), then $\mathcal{B}$ can easily recover x by computing the roots of $f(X)$ in $\mathbb{Z}_p$ with any of the known probabilistic polynomial time methods [BO81].[9] Thus, $\mathcal{B}$ can recover the secret x and hence solve the given instance of the q-SDL problem with no further help from the adversary $\mathcal{A}$.

If $f(x) \neq 0$, then the algorithm $\mathcal{B}$ proceeds to interact with $\mathcal{A}$ and gives g and c to $\mathcal{A}$ as a description of the function $h(\cdot)$, where, for any $m \in \mathbb{Z}_p$, $h(m) := g^{\frac{1}{m+c}}$.

Response Phase: Let us define polynomials $f_i'(X) = f(X)/(f_i(X)+c)$, $\forall\, i \in [n]$. As before, we expand them as $f_i'(X) = \sum_{j=0}^{deg[f_i'(X)]} \beta_{ij} X^j$, where $\beta_{ij} \in \mathbb{Z}_p$. $\forall i \in [n]$, $\mathcal{B}$ computes

$$a_i \longleftarrow \prod_{j=0}^{deg[f_i'(X)]} (g_j)^{\beta_{ij}}.$$

and gives a_i to $\mathcal{A}$.

Note that a_i is the valid image of $f_i(x)$ under the specified function $h(\cdot)$, since $a_i = (g')^{f_i'(x)} = (g')^{\frac{f(x)}{f_i(x)+c}} = g^{\frac{1}{f_i(x)+c}} = h(f_i(x))$.

Invert Phase: $\mathcal{A}$ returns (k, y), where $k \in [n]$ and $y \in \mathbb{Z}_p$, such that $y = f_k(x)$ with probability ϵ.

In the event that $y = f_k(x)$, (or, in other words, x is a root of the polynomial $f_k(X) - y$), then $\mathcal{B}$ can compute the roots of $f_k(X) - y$ in $\mathbb{Z}_p$, choose that root s for which the equality $(g')^{s^i} = g_i$ holds $\forall$ $i \in [q]$, and conclude its own game by outputting s.

Thus, if $\mathcal{A}$ outputs $f_k(x)$ with probability ϵ, $\mathcal{B}$ outputs the solution to the given instance of q-SDL problem with probability at least ϵ.

The claimed bounds are evident by the construction of the reduction. This completes the proof of Theorem 2. ■

We note that our proof works even if, in the construction, we set c to any constant value in $\mathbb{Z}_p$ (say, for eg., $c = 0$ so that $h(m) := g^{\frac{1}{m}}$).

4.3 Selective Correlated-Input Unpredictability

Theorem 3 *Suppose that (q, t', ϵ')-DHI assumption holds in $\mathbb{G}$. Let $\{C\}$ be a set of uniform-output polynomials over $\mathbb{Z}_p$. Then, for $\mathcal{H}$ as in Figure 1, there exists no probabilistic t-time adversary $\mathcal{A}$ for which $\mathsf{Adv}^{sCI-pred}_{\mathcal{A},\mathcal{H},\{C\}}$ is at least ϵ provided that*

$$d \leq q + 1,\ \epsilon \geq 2(n+1)\epsilon' \text{ and } t \leq t' - \Theta(nq\tau)$$

where $d = d(\lambda)$ upper bounds the sum of the degrees the polynomials that $\mathcal{A}$ queries upon and τ is the maximum time for an exponentiation in $\mathbb{G}$ and $\mathbb{Z}_p$.

PROOF: Let there exist a polynomial-time adversary $\mathcal{A}$ with an advantage ϵ in the selective Correlated-Input Predicting game with respect to the class of uniform-output polynomials over $\mathbb{Z}_p$. We build an algorithm $\mathcal{B}$ that

[9] For a d-degree polynomial, Ben-Or's algorithm uses $O((\log p)d^{2+e})$ expected operations in $\mathbb{Z}_p$.

interacts with $\mathcal{A}$ and solves a given random instance of q-DHI problem with the advantage at least ϵ', as follows.

Algorithm $\mathcal{B}$ is given a random instance $(g', (g')^s, (g')^{s^2}, \ldots, (g')^{s^q}) \in (\mathbb{G}^\star)^{q+1}$ of the q-DHI problem in $\mathbb{G}$, for some unknown $s \in \mathbb{Z}_p^\star$. The objective of $\mathcal{B}$ is to output $(g')^{\frac{1}{s}}$. $\mathcal{B}$ invokes $\mathcal{A}$ and interacts with it as follows.

Setup Phase 1: $\mathcal{B}$ gives p to $\mathcal{A}$.

Query Phase: $\mathcal{A}$ chooses a positive integer n $(= poly(\lambda))$, and gives to the challenger, $n+1$ distinct uniform-output polynomials over $\mathbb{Z}_p$, $\{f_i(X)\}_{i\in[n+1]}$.

Setup Phase 2: $\mathcal{B}$ uniformly chooses a random element $r \xleftarrow{\$} \mathbb{Z}_p$. Let $x := s + r$. $\mathcal{B}$ computes $(g')^{x^i}$ from the given values, $(g')^{s^i}$, $i \in [q]$.

Let us denote $(g')^{x^i}$ by g_i for $i \in [q]$, and g' by g_0.

$\mathcal{B}$ then uniformly samples a random element $k \xleftarrow{\$} [n+1]$ and sets $c \longleftarrow -f_k(r)$. (Note that, since each polynomial $f_i(X)$, and in particular, $f_k(X)$, evaluates to a uniform random element in $\mathbb{Z}_p$ for a uniform random assignment of its variable in $\mathbb{Z}_p$, the resulting distribution of c is uniformly random in $\mathbb{Z}_p$, as required.)

Let us define a polynomial $l_k(X) := (f_k(X)+c)/(X-r)$ (Observe here that $f_k(X)+c = f_k(X)-f_k(r)$ has a factor $(X-r)$). Also define another polynomial,

$$f(X) := \prod_{i\in[n+1]-\{k\}} (f_i(X) + c) \cdot l_k(X). \tag{1}$$

Let $f(X)$ be expanded as $f(X) = \sum_{j=0}^{d-1} \alpha_j X^j$ where $\alpha_j \in \mathbb{Z}_p$ (as $\sum_{i=1}^{n+1} deg[f_i(X)] \leq d$ and $deg[l_k(X)] = deg[f_k(X)] - 1$). $\mathcal{B}$ computes $\prod_{j=0}^{d-1}(g_j)^{\alpha_j}$ and sets:

$$g \longleftarrow \prod_{j=0}^{d-1}(g_j)^{\alpha_j}.$$

Thus $g = (g')^{f(x)}$. g has correct distribution provided that $f(x) \neq 0$ (in other words, g is a random generator of $\mathbb{G}$). If, however, $f(x) = 0$ (or, equivalently, if x is a root of $f(X)$), then $\mathcal{B}$ can easily recover x by computing the roots and hence solve the given instance of the q-DHI problem with no further help from the adversary $\mathcal{A}$.

If $f(x) \neq 0$, then the algorithm $\mathcal{B}$ proceeds to interact with $\mathcal{A}$ and gives g and c to $\mathcal{A}$ as a description of the function $h(\cdot)$, where, for any $m \in \mathbb{Z}_p$, $h(m) := g^{\frac{1}{m+c}}$.

Partially Adaptive Query-Response Phase: Now let us define polynomials $f_i'(X) := f(X)/(f_i(X)+c)$, $\forall\ i \in [n+1] - \{k\}$. Let them be expanded as $f_i'(X) = \sum_{j=0}^{deg[f_i'(X)]} \beta_{ij} X^j$, where $\beta_{ij} \in \mathbb{Z}_p$. When $\mathcal{A}$ presents an ith query as an index $k_i \in [n+1]$ for $i \in [n+1]$, $\mathcal{B}$ responds as follows.

If $k_i = k$, then $\mathcal{B}$ terminates the simulation, and concludes its own game by outputting a random element $r^\star$ in $\mathbb{G}$. If $k_i \neq k$, then $\mathcal{B}$ computes

$$a_{k_i} \longleftarrow \prod_{j=0}^{deg[f_{k_i}'(X)]} (g_j)^{\beta_{k_i j}}.$$

and gives a_{k_i} to $\mathcal{A}$.

Note that a_{k_i} is the valid image of $f_{k_i}(x)$ under the specified function $h(\cdot)$, since $a_{k_i} = (g')^{f_{k_i}'(x)} = (g')^{\frac{f(x)}{f_{k_i}(x)+c}} = g^{\frac{1}{f_{k_i}(x)+c}} = h(f_{k_i}(x))$.

Predict Phase: Let $k \in [n+1]$ such that $k \neq k_i\ \forall\ i \in [n]$. $\mathcal{A}$ returns $y \in \mathbb{G}$, such that $y = h(f_k(x))$ with probability ϵ.

Let $f'(X) := f(X)/l_k(X)$ (Observe through Equation (1) that $l_k(X)$ completely divides $f(X)$). It follows that

$$\begin{aligned} h(f_k(x)) &= g^{\frac{1}{f_k(x)+c}} \\ &= (g')^{\frac{f(x)}{f_k(x)+c}} \\ &= (g')^{\frac{f'(x)\cdot l_k(x)}{f_k(x)+c}} \\ &= (g')^{\frac{f'(x)\cdot l_k(x)}{(x-r)\cdot l_k(x)}} \\ &= (g')^{\frac{f'(x)}{x-r}}. \end{aligned}$$

Let $f'(X)$ be rewritten as $f'(X) = l'_k(X)(X-r) + c^\star$, where $l'_k(X) \in \mathbb{Z}_p[X]$ and $c^\star \in \mathbb{Z}_p$. Let $l'_k(X)$ be expanded as $l'_k(X) = \sum_{j=0}^{deg[l'_k(X)]} \mu_j X^j$, where $\mu_j \in \mathbb{Z}_p$.

In the ensuing discussion, we proceed with the assumption that $c^\star \neq 0$, and at the end of the proof show that $c^\star \neq 0$ with all but negligible probability.

$\mathcal{B}$ first computes

$$(g')^{l'_k(x)} = \prod_{j=0}^{deg[l'_k(X)]} (g_j)^{\mu_j}$$

and then sets

$$g^\star \longleftarrow (y \cdot (g')^{-l'_k(x)})^{\frac{1}{c^\star}}$$

where $\frac{1}{c^\star}$ is computed $\mod p$. $\mathcal{B}$ concludes its own game by outputting $g^\star$.

If $\mathcal{A}$ correctly predicts the image of $f_k(x)$ under $h(\cdot)$, i.e., if $y = h(f_k(x))$, then,

$$\begin{aligned} g^\star &= ((g')^{\frac{f'(x)}{x-r}} \cdot (g')^{-l'_k(x)})^{\frac{1}{c^\star}} \\ &= ((g')^{l'_k(x)+\frac{c^\star}{x-r}} \cdot (g')^{-l'_k(x)})^{\frac{1}{c^\star}} \\ &= (g')^{\frac{1}{x-r}} \\ &= (g')^{\frac{1}{s}} \end{aligned}$$

which is the solution for the given instance of q-DHI problem.

Now we compute the probability with which $\mathcal{B}$ outputs this solution.

If $\mathcal{B}$ aborts the simulation, then $\mathcal{B}$ outputs the valid solution with at most negligible probability $(\frac{1}{p})$. If $\mathcal{B}$ does not abort, then the probability that $\mathcal{B}$ outputs the solution is ϵ. Let δ denote the probability that $\mathcal{B}$ does not abort. Then, the actual probability that $\mathcal{B}$ outputs the right solution to the problem instance is at least $\delta \cdot \epsilon$.

It is easy to observe that the view of $\mathcal{A}$ for $\mathcal{B}$'s choice k (in the **Setup Phase 2**) is identically distributed to the view of $\mathcal{A}$ for $\mathcal{B}$'s choice $k' \neq k$. This immediately implies that the probability that $k_i \neq k$ in an ith query is at least $\frac{1}{2(n+1)}$; i.e., $\delta \geq \frac{1}{2(n+1)}$. Hence, $\mathcal{B}$ outputs the solution $(g')^{\frac{1}{s}}$, to the given q-DHI problem instance with probability at least $\frac{1}{2(n+1)} \cdot \epsilon = \epsilon'$.

Now we return to the proof of the claim that $c^\star \neq 0$ with all but negligible probability. Let P denote the probability that $c^\star = 0$. If $c^\star = 0$, then $(X-r)$ completely divides $f'(X)$; in other words, (as the ring of polynomials over a finite field is a unique factorization domain), $f'(X) = \prod_{i\in[n+1]-\{k\}}(f_i(X)+c)$ has a factor $(X-r)$, and there exists $j \in [n+1]-\{k\}$, such that $f_j(X)+c$ has a factor $(X-r)$; i.e., r is a root of $f_j(X)+c$ or $f_j(r)+c = 0$. Recall that $c = -f_k(r)$. Thus, in effect, $c^\star = 0$ implies $f_j(r) = f_k(r)$.

Now recall that r was sampled uniformly at random in $\mathbb{Z}_p$. Since the polynomials $f_j(X)$ and $f_k(X)$ are distinct, from Lemma 1, we conclude that the probability that $f_j(r) = f_k(r)$, and in turn $c^\star = 0$, is at most $\frac{\max(deg[f_j(X)],deg[f_k(X)])}{p}$, which is negligible in λ.

The claimed bounds are evident by the construction of the reduction. This completes the proof of Theorem 3. ■

4.4 Selective Correlated-Input pseudorandomness

Theorem 4 *Suppose that decisional (q, t', ϵ')-DHI assumption holds in $\mathbb{G}$. Let $\{C\}$ be a set of uniform-output polynomials over $\mathbb{Z}_p$. Then, for $\mathcal{H}$ as in Figure 1, there there exists no probabilistic t-time adversary $\mathcal{A}$ for which $\mathsf{Adv}^{sCI-dist}_{\mathcal{A},\mathcal{H},\{C\}}(\lambda)$ is at least ϵ provided that*

$$d \leq q+1,\ \epsilon \geq 2(n+1)\epsilon' \text{ and } t \leq t' - \Theta(nq\tau)$$

where $d = d(\lambda)$ upper bounds the sum of the degrees of the polynomials that $\mathcal{A}$ queries upon and τ is the maximum time for an exponentiation in $\mathbb{G}$ and $\mathbb{Z}_p$.

Let there exist a polynomial-time adversary $\mathcal{A}$ with an advantage ϵ in the selective Correlated-Input Distinguishing game with respect to the class of uniform-output polynomials over $\mathbb{Z}_p$. We build an algorithm $\mathcal{B}$ that interacts with $\mathcal{A}$ and solves a given random instance of decisional q-DHI problem with the advantage at least $\frac{1}{2(n+1)} \cdot \epsilon$, as follows.

Algorithm $\mathcal{B}$ is given a $q+2$-tuple $(g', (g')^s, (g')^{s^2}, \ldots, (g')^{s^q}, R) \in (\mathbb{G}^\star)^{q+2}$ for some unknown $s \in \mathbb{Z}_p^\star$, where the tuple is either sampled from $\mathcal{P}_{\mathcal{DDHI}}$ (i.e., $R = (g')^{\frac{1}{s}}$) or from $\mathcal{R}_{\mathcal{DDHI}}$ (i.e., R is uniform and independent in $\mathbb{G}$). The objective of $\mathcal{B}$ is to output 1 if $R = (g')^{\frac{1}{s}}$ and 0 otherwise. $\mathcal{B}$ invokes $\mathcal{A}$ and interacts with it as follows.

The reduction is similar to that for the selective Correlated-Input Predicting game in Theorem 3. So we shall only highlight the differences in the following.

Setup Phase 1, **Query Phase**, **Setup Phase 2**, and **Partially Adaptive Query-Response Phase** remain exactly the same as in the proof of Theorem 3, but in any $\mathbf{Query}_i$, if $k_i = k$ then $\mathcal{B}$ terminates the simulation and concludes its own game by outputting a random bit $b^\star \xleftarrow{\$} \{0,1\}$.

Next, $\mathcal{B}$ presents the challenge output as follows.

Challenge Phase:

We define polynomials $l'_k(X) := f(X)/l_k(X)$ and $f'_k(X) := f(X)/(f_k(X)+c)$. Observe that $f'_k(X) = \frac{f(X)}{l_k(X)\cdot(X-r)} = \frac{l_k(X)l'_k(X)}{l_k(X)(X-r)} = \frac{l'_k(X)}{(X-r)}$. Let $l'_k(X)$ be rewritten as $l'_k(X) = Q(X)\cdot(X-r) + c^\star$, where $Q(X) \in \mathbb{Z}_p[X]$ and $c^\star \in \mathbb{Z}_p$. Hence, $f'_k(X) = Q(X) + \frac{c^\star}{(X-r)}$. As seen in the proof of Theorem 3, we know that $c^\star \neq 0$ with all but negligible probability.

Let $Q(X)$ be expanded as $Q(X) = \sum_{j=0}^{deg[Q(X)]} \mu_j X^j$, where $\mu_j \in \mathbb{Z}_p$. $\mathcal{B}$ first computes $(g')^{Q(x)}$ as $\prod_{j=0}^{deg[Q(X)]} (g_j)^{\mu_j}$ and then sets

$$z \longleftarrow (g')^{Q(x)} \cdot R^{c^\star}$$

$\mathcal{B}$ gives z to $\mathcal{A}$.

Observe that if $R = (g')^{\frac{1}{s}} = (g')^{\frac{1}{x-r}}$ (when the input $q+2$-tuple is sampled from $\mathcal{P}_{\mathcal{DDHI}}$), then

$$\begin{aligned} z &= (g')^{Q(x)} \cdot (g')^{\frac{c^\star}{x-r}} \\ &= (g')^{f'_k(x)} \\ &= (g')^{\frac{f(x)}{f_k(x)+c}} \\ &= g^{\frac{1}{f_k(x)+c}} \\ &= h(f_k(x)) \end{aligned}$$

That is, if $R = (g')^{\frac{1}{s}}$, then z is a valid image of $f_k(x)$ under $h(\cdot)$. On the other hand, if R is uniform and independent in $\mathbb{G}$ (when the input $q+2$-tuple is sampled from $\mathcal{R}_{\mathcal{DDHI}}$), then z is a random element in $\mathbb{G}$.

Guess: $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$.

Eventually, $\mathcal{B}$ concludes its own game by outputting its guess as b' itself, meaning $R = z_{b'}$ where $z_1 = (g')^{\frac{1}{s}}$ and z_0 is a random element in $\mathbb{Z}_p$.

Now to compute the probability with which $\mathcal{B}$ outputs the right guess, let δ denote the probability that $\mathcal{B}$ does not abort the simulation.

If $\mathcal{B}$ aborts, then it outputs the right guess with probability $\frac{1}{2}$ and the corresponding advantage, Adv_1, of $\mathcal{B}$ in solving the given decisional q-DHI problem instance is

$$\begin{aligned}\mathsf{Adv}_1 &= |\Pr[b = b'] - \frac{1}{2}| \\ &= \frac{1}{2} - \frac{1}{2} \\ &= 0\end{aligned}$$

If it does not abort, then the probability that $\mathcal{B}$ outputs the right guess is at least $\frac{1}{2} + \delta \cdot \epsilon$, and the corresponding advantage, Adv_2, of $\mathcal{B}$ is

$$\begin{aligned}\mathsf{Adv}_2 &= |\Pr[b = b'] - \frac{1}{2}| \\ &\geq |(\frac{1}{2} + \delta \cdot \epsilon) - \frac{1}{2}| \\ &= \delta \cdot \epsilon\end{aligned}$$

.

Further, by the same argument as in Theorem 3, we have that $\delta \geq \frac{1}{2(n+1)}$.

Thus,

$$\begin{aligned}\mathsf{Adv}^{sCI-dist}_{\mathcal{B},\mathcal{H},\{C\}}(\lambda) =& |\frac{1}{n}\sum_{i=1}^{n}\Pr[\mathcal{B}(\{h_c(C_i(k))\}_{i\in[w-1]}, h_c(C_w(k))) \longrightarrow 1|w = i] \\ &- \frac{1}{n}\sum_{i=1}^{n}\Pr[\mathcal{B}(\{h_c(C_i(k))\}_{i\in[w-1]}, r_w) \longrightarrow 1|w = i]| \\ =& \frac{1}{n}|\Pr[\mathcal{B}(\{h_c(C_i(k))\}_{i\in[w-1]}, h_c(C_w(k))) \longrightarrow 1|w = n] \\ &- \Pr[\mathcal{B}(\{h_c(C_i(k))\}_{i\in[w-1]}, r_w) \longrightarrow 1|w = 1]| \\ =& \frac{1}{n}|\Pr[k \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathcal{O}^{weak}_{F(k,\cdot)}(\cdot)} \longrightarrow 1] \\ &- \Pr[k \xleftarrow{\$} \mathcal{K}, G \xleftarrow{\$} \mathsf{FF}(\mathcal{K},\mathcal{D},\mathcal{R}) : \mathcal{A}^{\mathcal{O}^{weak}_{G(k,\cdot)}(\cdot)} \longrightarrow 1]| \\ =& \frac{1}{n}\mathsf{Adv}^{aRKA-wPRF}_{\mathcal{A},\{F\},\{C\},1}(\lambda)\end{aligned} \tag{2}$$

$$\begin{aligned}&|\Pr[\mathcal{A}(g', (g')^s, (g')^{s^2}, \ldots, (g')^{s^q}, (g')^{\frac{1}{s}}) = 1] \\ &- \Pr[\mathcal{A}(g', (g')^s, (g')^{s^2}, \ldots, (g')^{s^q}, R) = 1]| \\ &= \mathsf{Adv}_2 + \mathsf{Adv}_2 \\ &\geq 2(n+1)\epsilon\end{aligned} \tag{3}$$

and hence, the total advantage of $\mathcal{B}$ in solving the given decisional q-DHI problem instance is at least $\frac{1}{2(n+1)} \cdot \epsilon \geq \epsilon'$.

The claimed bounds are evident by the construction of the reduction. This completes the proof of Theorem 4. ■

5 Relations between CI-Security and RKA-Security

In this section we examine relations between correlated-input secure hash functions and security under related-key attacks, whose formal treatment was initiated by [BK03]. The latter asks that security of a cryptographic primitive (e.g., a pseudorandom function) maintains security when used with *related secret keys*. In this section we only consider our notion of pseudorandomness under correlated-inputs, so "CI-security" below refers by default to this notion.

5.1 Relations to RKA-Secure Weak PRFs

We start by showing an equivalence between CI-secure hash functions and some form of RKA-security for *weak* PRFsthat we introduce later. The idea, following [HLO10], is to "switch" the input and key for these primitives.

RKA-Secure Weak PRFs Recall that weak PRFs [NR99], as opposed to normal ones, handle only random inputs. In defining RKA-security for this primitive, a modeling choice we need to consider is whether, when the adversary queries for a value of the function under a related key, a *new* random input is chosen (or the previous random-input re-used). We give general definitions that capture the possibilities. (However, for our results we only use a notion where the same random input is re-used.) We also consider both "selective" and "adaptive" security notions; in the former, the adversary chooses the relations applied to the secret key before receiving any responses.

Note that although some of the related works (e.g., [BC10]) represent the correlations by functions (called "related-key deriving (RKD) functions therein"), we continue to represent the correlations by circuits for ease of comparison with CI-security. For example, by $\{C\}$-RKA-PRF we refer to an RKA-PRF where the correlated secret keys are sampled according to circuits in $\{C\}$ (executed on a common random input). A subtlety to note is that while it is meaningful to consider CI-security for a class of circuits that do not contain circuit for the identity function, for certain RKA-secure primitives, we require that the circuit for the identity function be always included. (Correlated-input security is more general in this sense.)

We also note that a PRF function family is specified by an efficient probabilistic parameter-generation algorithm Gen_{prf} which takes as input a security parameter 1^λ and outputs the description of a function including a description of the function's keyspace, domain and range. However, for simplicity of exposition, we only consider a single PRF function in most part of the following discussion as long as there is no ambiguity.

Definition 8 ($q_{inp} - \{C\}$-sRKA-wPRF.) *Let $F : \mathcal{K} \times \mathcal{D} \longrightarrow \mathcal{R}$ be an efficiently computable function. Let $\{C\} \subseteq \mathsf{Fun}(\mathcal{K}, \mathcal{K})$ be a set of RKD circuits. F is said to be $q_{inp} - \{C\}$-sRKA-PRF, if, $\forall \mathcal{A} \in \mathsf{PPT}$, $\mathsf{Adv}^{sRKA-wPRF}_{\mathcal{A},F,\{C\},q_{inp}}(\lambda)$ is negligible, where,*

$$
\begin{aligned}
&\mathsf{Adv}^{sRKA-wPRF}_{\mathcal{A},F,\{C\},q_{inp}}(\lambda) \\
&:= \big| \Pr[k \xleftarrow{\$} \mathcal{K}, x_j \xleftarrow{\$} \mathcal{D} : \{(ind_i, C_i)\}_{i\in[n]} (\subset ([q_{inp}], \{C\})) \leftarrow \mathcal{A}, \\
&\quad \mathcal{A}(\{(x_{ind_i}, F(C_i(k), x_{ind_i}))\})_{i\in[n]} \longrightarrow 1] \\
&- \Pr[k \xleftarrow{\$} \mathcal{K}, G \xleftarrow{\$} \mathsf{FF}(\mathcal{K}, \mathcal{D}, \mathcal{R}), x_j \xleftarrow{\$} \mathcal{D} : \\
&\quad \{(ind_i, C_i)\}_{i\in[n]} (\subset ([q_{inp}], \{C\})) \leftarrow \mathcal{A}, \mathcal{A}(\{(x_{ind_i}, G(C_i(k), x_{ind_i}))\})_{i\in[n]} \longrightarrow 1] \big|
\end{aligned}
\tag{4}
$$

Definition 9 ($q_{inp} - \{C\}$-aRKA-wPRF.) *Let $F : \mathcal{K} \times \mathcal{D} \longrightarrow \mathcal{R}$ be an efficiently computable function. Let $\{C\} \subseteq \mathsf{Fun}(\mathcal{K}, \mathcal{K})$ be a set of RKD circuits. F is said to be $q_{inp} - \{C\}$-aRKA-wPRF, if, $\forall \mathcal{A} \in \mathsf{PPT}$,*

$\mathsf{Adv}^{aRKA-wPRF}_{\mathcal{A},F,\{C\},q_{inp}}(\lambda)$ *is negligible, where,*

$$\begin{aligned}\mathsf{Adv}^{aRKA-wPRF}_{\mathcal{A},F,\{C\},q_{inp}}(\lambda) :=& |\Pr[k \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathcal{O}^{weak}_{F(k,\cdot)}(\cdot,\cdot)} \longrightarrow 1] \\ & - \Pr[k \xleftarrow{\$} \mathcal{K}, G \xleftarrow{\$} \mathsf{FF}(\mathcal{K},\mathcal{D},\mathcal{R}) : \mathcal{A}^{\mathcal{O}^{weak}_{G(k,\cdot)}(\cdot,\cdot)} \longrightarrow 1]|\end{aligned} \tag{5}$$

Here, the related-key-wprf oracle $\mathcal{O}^{weak}_{f(k,\cdot)}(\cdot)$ *takes* $(ind_i, C_i) \in ([q_{inp}], \{C\})$ *as input and outputs* $(x_{ind_i}, f(C_i(k), x_{ind_i}))$*, where* $x_{ind_i} \xleftarrow{\$} \mathcal{D}$*.*

The Equivalence In the following we show an equivalence between a one-input RKA-Secure wPRF for $\{C\}$ and a CI-secure hash function for $\{C\}$. (The equivalence holds respectively in the cases of selective and adaptive security notions.) First, we construct a family of functions $\{F\}$ from a family of functions $\mathcal{H}$ and show that if $\mathcal{H}$ is a $\{C\}$-aCI-pseudorandom function family (resp. $\{C\}$-sCI-pseudorandom function family) then $\{F\}$ is a 1-$\{C\}$-aRKA-wPRF (resp. 1-$\{C\}$-sRKA-wPRF).

Let $\mathcal{H}$ be a family of functions specified by Gen. A family of functions $\{F\}$ is defined by the following parameter-generation algorithm:

$\mathsf{Gen}_{prf}(1^\lambda)$. Gen_{prf} runs $\mathsf{Gen}(1^\lambda)$ that outputs the description of a parameter set $\mathcal{D}$, $c \in \mathcal{D}$, and a function $h_c : \mathcal{K} \longrightarrow \mathcal{R}$. Gen_{prf} outputs $\mathcal{K}$, $\mathcal{D}$ and $\mathcal{R}$ for keyspace, domain and range, respectively, of a function F defined by,

$$F(k, x) := h_x(k)$$

for any $k \in \mathcal{K}$ and $x \in \mathcal{D}$.

Figure 2: Construction of 1-$\{C\}$-(s/a)RKA-wPRF Family from $\{C\}$-(s/a)CI-pseudorandom Function Family

Theorem 5 *$\{C\}$-aCI-pseudorandom function family (resp. $\{C\}$-sCI-pseudorandom function family) implies 1-$\{C\}$-aRKA-wPRF (resp. 1-$\{C\}$-sRKA-wPRF).*

Proof: We shall only present the proof for the adaptive case. The proof for the selective case follows on similar lines.

Suppose $\mathcal{A} \in \mathsf{PPT}$ is a $\{C\}$-restricted adversary against 1-$\{C\}$-RKA-wPRF security of $\{F\}$ as in Figure 2. Then, we show that there exists a $\{C\}$-restricted adversary $\mathcal{B}$ against adaptive CI-pseudorandomness of $\mathcal{H}$ such that

$$\mathsf{Adv}^{aRKA-wPRF}_{\mathcal{B},\{F\},\{C\},1}(\lambda) = n.\mathsf{Adv}^{aCI-dist}_{\mathcal{A},\mathcal{H},\{C\}}(\lambda)$$

$\mathcal{B}$ makes at most as many queries as $\mathcal{A}$ does and has the same running time as that of $\mathcal{A}$.

Let $b_{ci} \xleftarrow{\$} \{0,1\}$. Let $c \xleftarrow{\$} \mathcal{D}$ and $k \xleftarrow{\$} \mathcal{K}$. In its own game, $\mathcal{B}$ is given c, $\mathcal{D}$, $\mathcal{K}$ and $\mathcal{R}$. It can present an adaptive ith query $C_i \in \{C\}$ to which it receives $h_c(C_i(k))$. Let n' denote its total number of queries. It presents a challenge query $C_{n'+1}$ to which it receives $z_{b_{ci}}$, where $z_0 \xleftarrow{\$} \mathcal{R}$ and $z_1 := h_c(C_{n'+1}(r))$. The objective of $\mathcal{B}$ is to guess b_{ci}. $\mathcal{B}$ invokes $\mathcal{A}$ and interacts with it as follows:

Let n be the maximum number of queries of $\mathcal{A}$. We define $n+1$ hybrids, H^i_n, for $0 \leq i \leq n$ as follows. H^i_n: a sequence of n elements, where the first i elements are outputs of the PRF function and the rest are uniform and independent random elements in $\mathcal{R}$. Since $q_{inp} = 1$, we shall represent $\mathcal{A}$'s queries only by the circuits and ignore the co-ordinate for index.

1. Choose $w \xleftarrow{\$} [n]$. Give $\mathcal{K}$, $\mathcal{D}$ and $\mathcal{R}$ to $\mathcal{A}$ to specify the wPRF.

2. For $i < w$, when $\mathcal{A}$ presents its ith query C_i, query the challenger with the same, and respond to $\mathcal{A}$ via $(c, h_c(C_i(k))$, where $(h_c(C_i(k))$ is output by the challenger.

3. For $i = w$, when $\mathcal{A}$ presents its wth query C_w, present the same as the challenge query to the challenger, and respond to $\mathcal{A}$ via $(c, z_{b_{ci}})$, where $z_{b_{ci}}$ is output by the challenger.

4. For $i > w$, respond to $\mathcal{A}$'s query C_i via (c, r_i), where $r_i \xleftarrow{\$} \mathcal{R}$.

5. When $\mathcal{A}$ finally outputs its guess $\hat{b}_{ci}$, output the same.

Note that, for $i < w$, $(c, h_c(C_i(k)) = (c, F(C_i(k), c))$ is a valid wPRF output, and for $i > w$, (c, r_i) is a random output. For $i = w$, $(c, z_{b_{ci}})$ is a valid wPRF output if $z_{b_{ci}} = 1$ and a random output, otherwise. Thus, the resulting hybrid is H_n^w if $z_{b_{ci}} = 1$, otherwise it is H_n^{w-1}.

Let $r_w \xleftarrow{\$} \mathcal{R}$. Observe that

$$\begin{aligned} &\Pr[\mathcal{B}(\{h_c(C_i(k))\}_{i\in[w-1]}, r_w) \longrightarrow 1 | w = 1] \\ &= \Pr[k \xleftarrow{\$} \mathcal{K}, G \xleftarrow{\$} \mathsf{FF}(\mathcal{K}, \mathcal{D}, \mathcal{R}) : \mathcal{A}^{\mathcal{O}^{weak}_{G(k,\cdot)}(\cdot,\cdot)} \longrightarrow 1], \end{aligned} \tag{6}$$

and, for any $1 < j \leq n$,

$$\begin{aligned} &\Pr[\mathcal{B}(\{h_c(C_i(k))\}_{i\in[w-1]}, r_w) \longrightarrow 1 | w = j] \\ &= \Pr[\mathcal{B}(\{h_c(C_i(k))\}_{i\in[w-1]}, h_c(C_w(k))) \longrightarrow 1 | w = j-1] \end{aligned} \tag{7}$$

Thus, we have,

$$\begin{aligned} \mathsf{Adv}^{sCI-dist}_{\mathcal{B},\mathcal{H},\{C\}}(\lambda) =& |\frac{1}{n}\sum_{i=1}^{n} \Pr[\mathcal{B}(\{h_c(C_i(k))\}_{i\in[w-1]}, h_c(C_w(k))) \longrightarrow 1 | w = i] \\ &- \frac{1}{n}\sum_{i=1}^{n} \Pr[\mathcal{B}(\{h_c(C_i(k))\}_{i\in[w-1]}, r_w) \longrightarrow 1 | w = i]| \\ =& \frac{1}{n}|\Pr[\mathcal{B}(\{h_c(C_i(k))\}_{i\in[w-1]}, h_c(C_w(k))) \longrightarrow 1 | w = n] \\ &- \Pr[\mathcal{B}(\{h_c(C_i(k))\}_{i\in[w-1]}, r_w) \longrightarrow 1 | w = 1]| \\ =& \frac{1}{n}|\Pr[k \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathcal{O}^{weak}_{F(k,\cdot)}(\cdot)} \longrightarrow 1] \\ &- \Pr[k \xleftarrow{\$} \mathcal{K}, G \xleftarrow{\$} \mathsf{FF}(\mathcal{K}, \mathcal{D}, \mathcal{R}) : \mathcal{A}^{\mathcal{O}^{weak}_{G(k,\cdot)}(\cdot)} \longrightarrow 1]| \\ =& \frac{1}{n}\mathsf{Adv}^{aRKA-wPRF}_{\mathcal{A},\{F\},\{C\},1}(\lambda) \end{aligned} \tag{8}$$

This completes the proof of Theorem 5. ■

Next, we construct a family of functions $\mathcal{H}$ from a family of functions $\{F\}$ and show that if $\{F\}$ is 1-$\{C\}$-aRKA-wPRF (resp. (resp. 1-$\{C\}$-sRKA-wPRF) then $\mathcal{H}$ is $\{C\}$-aCI-pseudorandom function family (resp. $\{C\}$-sCI-pseudorandom function family).

Theorem 6 *1-$\{C\}$-aRKA-wPRF (resp. 1-$\{C\}$-sRKA-wPRF) implies $\{C\}$-aCI-pseudorandom function family (resp. $\{C\}$-sCI-pseudorandom function family).*

Proof: We shall only present the proof for the adaptive case. The proof for the selective case follows on similar lines.

Suppose $\mathcal{A} \in \mathsf{PPT}$ be a $\{C\}$-restricted adversary against adaptive CI-security of $\mathcal{H}$. Then, we show that there exists a $\{C\}$-restricted adversary $\mathcal{B}$ against aRKA-wPRF security of F such that

$$\mathsf{Adv}^{aCI-dist}_{\mathcal{A},\mathcal{H},\{C\}}(\lambda) \leq 2\mathsf{Adv}^{aRKA-wPRF}_{\mathcal{B},\{F\},\{C\},1}(\lambda)$$

Let $\{F\}$ be a family of functions specified by Gen_{prf}. A family of functions $\mathcal{H}$ is defined through the following parameter-generation algorithm:

$\mathsf{Gen}(1^\lambda)$. Gen runs $\mathsf{Gen}_{prf}(1^\lambda)$ to get $F : \mathcal{K} \times \mathcal{D} \longrightarrow \mathcal{R}$. It outputs $\mathcal{D}$ as the parameter set, chooses $i \xleftarrow{\$} \mathcal{D}$, and outputs i, $\mathcal{K}$ and $\mathcal{R}$ to specify the function $h_i : \mathcal{K} \longrightarrow \mathcal{R}$ defined by,

$$h_i(k) := F(k, i)$$

for any $k \in \mathcal{K}$.

Figure 3: Construction of $\{C\}$-(s/a)CI-pseudorandom Function Family from 1-$\{C\}$-(s/a)RKA-wPRF Family

$\mathcal{B}$ makes one query more than $\mathcal{A}$ does and has the same running time as that of $\mathcal{A}$.

$\mathcal{B}$ is given access to a related-key-wPRF oracle $\mathcal{O}^{weak}_{f(k,\cdot)}(\cdot,\cdot)$, which either instantiates the PRF $F : \mathcal{K} \times \mathcal{D} \longrightarrow \mathcal{R}$ for a random key $k \xleftarrow{\$} \mathcal{K}$ or instantiates a random function G, where $k \xleftarrow{\$} \mathcal{K}$ and $G \xleftarrow{\$} \mathsf{FF}(\mathcal{K}, \mathcal{D}, \mathcal{R})$. The objective of $\mathcal{B}$ is to output 1 in the former case and 0 in the latter case. $\mathcal{B}$ invokes $\mathcal{A}$ and interacts with it as follows.

1. Choose $C' \xleftarrow{\$} \{C\}$ and query the oracle with C'. Let $(c, f(C'(k), c))$ be the response received. Give c, $\mathcal{K}$ and $\mathcal{R}$ to $\mathcal{A}$ as a description of the function $h_c : \mathcal{K} \longrightarrow \mathcal{R}$.
2. When $\mathcal{A}$ presents an ith query $C_i \in \{C\}$, respond via a_i, where $a_i \leftarrow \mathcal{O}^{weak}_{f(k,\cdot)}(1, C_i)$.
3. When $\mathcal{A}$ finally outputs its guess $\hat{b}$, output the same.

Let n denote the total number of $\mathcal{A}$'s queries. Let $r \xleftarrow{\$} \mathcal{D}$ and $r_1, \ldots, r_{n+1} \xleftarrow{\$} \mathcal{R}$.

$$\begin{aligned}
\mathsf{Adv}^{sCI-dist}_{\mathcal{A},\mathcal{H},\{C\}}(\lambda) = \; & |\Pr[\mathcal{A}(\{h_c(C_i(r))\}_{i\in[n]}, h_c(C_{n+1}(r))) \longrightarrow 1] \\
& - \Pr[\mathcal{A}(\{h_c(C_i(r))\}_{i\in[n]}, r_{n+1}) \longrightarrow 1]| \\
\leq \; & |\Pr[\mathcal{A}(\{h_c(C_i(r))\}_{i\in[n]}, h_c(C_{n+1}(r))) \longrightarrow 1] \\
& - \Pr[\mathcal{A}(\{r_i\}_{i\in[n]}, r_{n+1}) \longrightarrow 1]| \qquad (9) \\
& + |\Pr[\mathcal{A}(\{r_i\}_{i\in[n]}, r_{n+1}) \longrightarrow 1] \\
& - \Pr[\mathcal{A}(\{h_c(C_i(r))\}_{i\in[n]}, r_{n+1}) \longrightarrow 1]| \qquad (10)
\end{aligned}$$

From the above reduction, the Term (9) is exactly the advantage of $\mathcal{B}$ in its own adaptive PRF-RKA game; i.e.,

$$\begin{aligned}
& |\Pr[\mathcal{A}(\{h_c(C_j(r))\}_{j\in[n]}, h_c(C_{n+1}(r))) \longrightarrow 1] - \Pr[\mathcal{A}(\{r_j\}_{j\in[n]}, r_{n+1}) \longrightarrow 1]| \\
& = |\Pr[k \xleftarrow{\$} \mathcal{K} : \mathcal{B}^{\mathcal{O}^{weak}_{F(k,\cdot)}(\cdot,\cdot)} \longrightarrow 1] \\
& \quad - \Pr[k \xleftarrow{\$} \mathcal{K}, G \xleftarrow{\$} \mathsf{FF}(\mathcal{K}, \mathcal{D}, \mathcal{R}) : \mathcal{B}^{\mathcal{O}^{weak}_{G(k,\cdot)}(\cdot,\cdot)} \longrightarrow 1]| \\
& = \mathsf{Adv}^{aRKA-wPRF}_{\mathcal{B},\{F\},\{C\},1}(\lambda)
\end{aligned}$$

Now, let $b_{prf} \xleftarrow{\$} \{0, 1\}$. Term (10) is exactly the advantage of $\mathcal{A}$ in a game $\mathsf{Exp}^{aCI-dist-rand}_{\mathcal{A},\mathcal{H},\{C\}}(b_{prf})$ defined as follows: If $b_{prf} = 1$, then $\forall i \in [n]$, the ith query C_i is responded via $h_c(C_i(r))$, and the final challenge query is responded via $r_{n+1} \xleftarrow{\$} \mathcal{R}$; otherwise, $\forall i \in [n]$, the ith query C_i is responded via $r_i \xleftarrow{\$} \mathcal{R}$, and the final challenge query is responded via $r_{n+1} \xleftarrow{\$} \mathcal{R}$; the advantage of any adversary

$\mathcal{A}'$ in this game is $|\Pr[\mathsf{Exp}^{aCI-dist-rand}_{\mathcal{A}',\mathcal{H},\{C\}}(1) \longrightarrow 1] - \Pr[\mathsf{Exp}^{aCI-dist-rand}_{\mathcal{A}',\mathcal{H},\{C\}}(0) \longrightarrow 1]|$. As this game can be easily simulated by $\mathcal{B}$ such that the simulated game is for $b_{prf} = 1$ if its related-key-wPRF oracle instantiates the PRF and for $b_{prf} = 0$ if its related-key-wPRF oracle instantiates a random function, Term (10) is also upper-bounded by the advantage of $\mathcal{B}$ in its own game; i.e.,

$$\begin{aligned} &|\Pr[\mathcal{A}(\{r_i\}_{i\in[n]}, r_{n+1}) \longrightarrow 1] - \Pr[\mathcal{A}(\{h_c(C_i(r))\}_{i\in[n]}, r_{n+1}) \longrightarrow 1]| \\ &\leq \mathsf{Adv}^{aRKA-wPRF}_{\mathcal{B},\{F\},\{C\},1}(\lambda) \end{aligned} \tag{11}$$

leading us to

$$\mathsf{Adv}^{aCI-dist}_{\mathcal{A},\mathcal{H},\{C\}}(\lambda) \leq 2\mathsf{Adv}^{aRKA-wPRF}_{\mathcal{B},\{F\},\{C\},1}(\lambda)$$

This concludes the proof of Theorem 6. ■

New CI-Secure Hash Functions As an application of the above equivalence, we obtain new CI-secure hash functions. In particular, note that an adaptive one-input RKA-secure wPRF for a class of circuits $\{C\}$ is trivially implied by an RKA-Secure PRF for $\{C\}$ (Here, the latter is defined as expected, namely as in prior work except cast in our framework of circuits). We can therefore use the recent constructions of RKA-Secure PRFs by Bellare and Cash [BC10] to obtain adaptive CI-secure hash functions. Namely, the latter are secure under the standard DDH assumption for class of circuits computing multiplications by a group elements or under exponential-hardness of DDH for additions by a group elements.

Though the resulting CI-secure hash functions are secure for weaker classes of relations as compared to our main construction, they are remarkable in that they are both adaptively secure and do not need a public key. (They do not even need any randomly-generated global parameters, as the constructions of [BC10] work in a fixed group.) The latter is because in the case that we start with an RKA-secure PRF (rather than wPRF), our construction of CI-secure hash can be modified by applying the PRF to any fixed value in the domain of the latter (still using the input as the key).

5.2 CI-secure Functions Imply Other RKA-secure Primitives

In this section we discuss a general technique for building RKA-secure cryptographic primitives from a CI-secure hash function. The idea is to hash the coins used to generate keys for the the former, using a CI-secure hash functions.

Informally, let Ψ be a scheme for a cryptographic primitive. Let KeyGen be a PPT algorithm for Ψ, and let $l(\lambda)$ be the length of the randomness it uses. Our transformation uses a $\{C\}$-CI-pseudorandom function family $\mathcal{H}$ specified by Gen, and the transformation involves modifying $\mathsf{KeyGen}(1^\lambda; r)$ where $r \xleftarrow{\$} \{0,1\}^{l(\lambda)}$ to $\mathsf{KeyGen}(1^\lambda; h(r'))$ where $r' \xleftarrow{\$} \{0,1\}^{t(\lambda)}$ and $(h : \{0,1\}^{t(\lambda)} \longrightarrow \{0,1\}^{l(\lambda)}) \leftarrow \mathsf{Gen}(1^\lambda)$.

More concretely, we exemplify the above technique for digital signatures. We give our formalization of RKA-security for signatures in Appendix B.

In what follows we show that $\{C\}$-aCI-pseudorandom function family (resp. $\{C\}$-sCI-pseudorandom function family) implies $\{C\}$-aRKA-unforgeable scheme (resp. $\{C\}$-sRKA-unforgeable scheme). The transformation is given in Figure 4.

Theorem 7 *$\{C\}$-aCI-pseudorandom function family (resp. $\{C\}$-sCI-pseudorandom function family) implies $\{C\}$-aRKA-unforgeable scheme (resp. $\{C\}$-sRKA-unforgeable scheme).*

Proof: We shall only present the proof for the adaptive case. The proof for the selective case is similar, and is hence skipped.

Consider the construction in Figure 4. For any $\{C\}$-restricted adversary $\mathcal{A} \in \mathsf{PPT}$ against aRKA-security of Σ, we show that there exists a $\{C\}$-restricted adversary $\mathcal{B}_{ci} \in \mathsf{PPT}$ against adaptive correlated-input security of $\mathcal{H}$ and an adversary $\mathcal{B}_{forge} \in \mathsf{PPT}$ against existential unforgeability of Σ', such that

$$\mathsf{Adv}^{aRKA-forge}_{\mathcal{A},\Sigma,\{C\}}(\lambda) \leq 2n(Adv^{aCI-dist}_{\mathcal{B}_{ci},\mathcal{H},\{C\}}(\lambda) + \mathsf{Adv}^{forge}_{\mathcal{B}_{forge},\Sigma'}(\lambda)),$$

Let $\mathcal{H}$ be a function family specified by $\mathsf{Gen}$with output length $l(\lambda)$. Let $\Sigma' = (\mathsf{KeyGen}', \mathsf{Sign}', \mathsf{Verify}')$ be a signature scheme such that $l(\lambda)$ is the length of the randomness used by KeyGen'. The signature scheme $\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is defined by:

- $\mathsf{KeyGen}(1^\lambda)$: $(h : \{0,1\}^{t(\lambda)} \longrightarrow \{0,1\}^{l(\lambda)}) \leftarrow \mathsf{Gen}(1^\lambda)$; $sk \xleftarrow{\$} \{0,1\}^{t(\lambda)}$; $(sk', pk') \leftarrow \mathsf{KeyGen}'(1^\lambda; h(sk))$; output sk as the secret key, and $pk := (h, pk')$ as the public key.
- $\mathsf{Sign}(sk, m)$: Run $\mathsf{KeyGen}'(1^\lambda, h(sk))$ to obtain sk'. The signature on message m is set as $\sigma \leftarrow \mathsf{Sign}'(sk', m)$.
- $\mathsf{Verify}(pk, m, \sigma)$: Output valid if $\mathsf{Verify}'(pk', m, \sigma) = \mathsf{valid}$ and output $\mathsf{invalid}$ otherwise.

Figure 4: Construction of RKA-secure Signature Scheme from CI-secure Pseudorandom Functions

where $\mathsf{Adv}^{forge}_{\mathcal{B}_{forge},\Sigma'}(\lambda)$ denotes the advantage of $\mathcal{B}_{forge}$ against the existential unforgeability of Σ'.

Let $b_{ci} \xleftarrow{\$} \{0,1\}$. In its own game, for $(h_c : \{0,1\}^{t(\lambda)} \longrightarrow \{0,1\}^{l(\lambda)}) \leftarrow \mathsf{Gen}(1^\lambda)$ and $sk \xleftarrow{\$} \{0,1\}^{t(\lambda)}$, $\mathcal{B}_{ci}$ can present $\{C_i\}_{i\in[n'+1]}$ to its challenger who responds via $(\{h_c(C_i(sk))\}_{i\in n'}, z_{b_{ci}})$, where $z_1 = h_c(C_{n'+1}(sk))$ and $z_0 \xleftarrow{\$} \{0,1\}^{l(\lambda)}$. The objective of $\mathcal{B}_{ci}$ is to guess b_{ci}. $\mathcal{B}_{ci}$ invokes $\mathcal{A}$ and interacts with it as follows:

1. Let n be the total number of distinct RKDs on which $\mathcal{A}$ queries. Choose $w \xleftarrow{\$} [0, n-1]$. Choose $r_i \xleftarrow{\$} \{0,1\}^{l(\lambda)}$ for $i \in [w+2, n]$. Query the challenger with the circuit C_{id} corresponding to the identity function. Let $a_{id} := h_c(C_{id}(sk)) = h_c(sk)$. Compute $(sk', pk') \leftarrow \mathsf{KeyGen}'(1^\lambda; a_{id})$; give $pk := (h, pk')$ as the public key to $\mathcal{A}$.
2. When $\mathcal{A}$ presents a query (C_i, m_{ij}), if $i \leq w$, query the challenger with C_i. Let $a_i := h_c(C_i(sk))$ be the response received from the challenger. Compute $(sk'_i, pk'_i) \leftarrow \mathsf{KeyGen}'(1^\lambda; a_i)$. Respond via $\sigma_{ij} \leftarrow \mathsf{Sign}'(sk'_i, m_{ij})$.
3. When $\mathcal{A}$ presents a query (C_{w+1}, m_{w+1j}), present the challenge query C_{w+1}. Let a_{w+1} be the response received from the challenger. Compute $(sk'_{w+1}, pk'_{w+1}) \leftarrow \mathsf{KeyGen}'(1^\lambda; a_{w+1})$. Respond via $\sigma_{(w+1)j} \leftarrow \mathsf{Sign}'(sk'_{w+1}, m_{(w+1)j})$.
4. When $\mathcal{A}$ presents a query (C_i, m_{ij}), if $i > w+1$, compute $(sk'_i, pk'_i) \leftarrow \mathsf{KeyGen}'(1^\lambda; r_i)$. Respond via $\sigma_{ij} \leftarrow \mathsf{Sign}'(sk'_i, m_{ij})$.
5. When $\mathcal{A}$ outputs $\hat{b}_{ci}$, output the same.

Let $\mathsf{Exp}^{sRKA-forge-rand}_{\mathcal{A},\Sigma,\{C\}}(b)$ be an experiment which is similar to $\mathsf{Exp}^{aRKA-forge}_{\mathcal{A},\Sigma,\{C\}}$ except that a query (C_i, m_{ij}) is responded via $\sigma_{ij} \leftarrow \mathsf{Sign}'(sk'_i, m_{ij})$ where $(sk'_i, pk'_i) \leftarrow \mathsf{KeyGen}'(1^\lambda; h_c(C_i(sk)))$ if $b = 1$, and via $\sigma_{ij} \leftarrow \mathsf{Sign}'(sk'_i, m_{ij})$ where $(sk'_i, pk'_i) \leftarrow \mathsf{KeyGen}'(1^\lambda; r_i)$ otherwise. Observe that, in view of $\mathcal{A}$, $\mathsf{Exp}^{sRKA-forge-rand}_{\mathcal{A},\Sigma,\{C\}}(1)$ is identical to $\mathsf{Exp}^{aRKA-forge}_{\mathcal{A},\Sigma,\{C\}}$. Let $r_{w+1} \xleftarrow{\$} \{0,1\}^{l(\lambda)}$. Observe that

$$
\begin{aligned}
&\mathsf{Adv}^{sRKA-wPRF}_{\mathcal{A},F,\{C\},q_{inp}}(\lambda) \\
&:= |\Pr[k \xleftarrow{\$} \mathcal{K}, x_j \xleftarrow{\$} \mathcal{D} : \{(ind_i, C_i)\}_{i\in[n]} (\subset ([q_{inp}], \{C\})) \leftarrow \mathcal{A}, \\
&\quad \mathcal{A}(\{(x_{ind_i}, F(C_i(k), x_{ind_i}))\})_{i\in[n]} \longrightarrow 1] \\
&- \Pr[k \xleftarrow{\$} \mathcal{K}, G \xleftarrow{\$} \mathsf{FF}(\mathcal{K}, \mathcal{D}, \mathcal{R}), x_j \xleftarrow{\$} \mathcal{D} : \\
&\quad \{(ind_i, C_i)\}_{i\in[n]} (\subset ([q_{inp}], \{C\})) \leftarrow \mathcal{A}, \mathcal{A}(\{(x_{ind_i}, G(C_i(k), x_{ind_i}))\})_{i\in[n]} \longrightarrow 1]|
\end{aligned}
\tag{12}
$$

$$\begin{aligned}&\Pr[\mathcal{B}_{ci}(\{h_c(C_i(sk))\}_{i\in[w]}, h_c(C_{w+1}(sk))) \longrightarrow 1 | w = n-1]\\ &= \Pr[\mathsf{Exp}^{sRKA-forge-rand}_{\mathcal{A},\Sigma,\{C\}}(1) \longrightarrow 1],\end{aligned} \tag{13}$$

$$\begin{aligned}&\Pr[\mathcal{B}_{ci}(\{h_c(C_i(sk))\}_{i\in[w]}, r_{w+1}) \longrightarrow 1 | w = 0]\\ &= \Pr[\mathsf{Exp}^{sRKA-forge-rand}_{\mathcal{A},\Sigma,\{C\}}(0) \longrightarrow 1],\end{aligned} \tag{14}$$

and, for any $1 \leq j < n$,

$$\begin{aligned}&\Pr[\mathcal{B}_{ci}(\{h_c(C_i(sk))\}_{i\in[w]}, r_{w+1}) \longrightarrow 1 | w = j]\\ &= \Pr[\mathcal{B}_{ci}(\{h_c(C_i(sk))\}_{i\in[w]}, h_c(C_{w+1}(sk))) \longrightarrow 1 | w = j-1]\end{aligned} \tag{15}$$

Thus, we have,

$$\begin{aligned}&\mathsf{Adv}^{sCI-dist}_{\mathcal{B}_{ci},\mathcal{H},\{C\}}(\lambda)\\ &=|\Pr[\mathsf{Exp}^{sCI-dist}_{\mathcal{B}_{ci},\mathcal{H},\{C\}}(1) \longrightarrow 1],\\ &\quad - \Pr[\mathsf{Exp}^{sCI-dist}_{\mathcal{B}_{ci},\mathcal{H},\{C\}}(0) \longrightarrow 1]|\\ &=|\frac{1}{n}\sum_{i=0}^{n-1}\Pr[\mathcal{B}_{ci}(\{h_c(C_i(sk))\}_{i\in[w]}, h_c(C_{w+1}(sk))) \longrightarrow 1 | w = i]\\ &\quad - \frac{1}{n}\sum_{i=0}^{n-1}\Pr[\mathcal{B}_{ci}(\{h_c(C_i(sk))\}_{i\in[w]}, r_{w+1}) \longrightarrow 1 | w = i]|\\ &=\frac{1}{n}|\Pr[\mathcal{B}_{ci}(\{h_c(C_i(sk))\}_{i\in[w]}, h_c(C_{w+1}(sk))) \longrightarrow 1 | w = n-1]\\ &\quad - \Pr[\mathcal{B}_{ci}(\{h_c(C_i(sk))\}_{i\in[w]}, r_{w+1}) \longrightarrow 1 | w = 0]|\\ &=\frac{1}{n}|\Pr[\mathsf{Exp}^{sRKA-forge-rand}_{\mathcal{A},\Sigma,\{C\}}(1) \longrightarrow 1] - \Pr[\mathsf{Exp}^{sRKA-forge-rand}_{\mathcal{A},\Sigma,\{C\}}(0) \longrightarrow 1]|\end{aligned}$$

From the triangle inequality, it holds that

$$\begin{aligned}\mathsf{Adv}^{aRKA-forge}_{\mathcal{A},\Sigma,\{C\}}(\lambda) = &\quad |\Pr[\mathsf{Exp}^{sCI-dist}_{\mathcal{B}_{ci},\mathcal{H},\{C\}}(1) \longrightarrow 1],\\ = &\quad \Pr[\mathsf{Exp}^{aRKA-forge}_{\mathcal{A},\Sigma,\{C\}} \longrightarrow 1]\\ = &\quad \Pr[\mathsf{Exp}^{sRKA-forge-rand}_{\mathcal{A},\Sigma,\{C\}}(1) \longrightarrow 1]\\ \leq &\quad |\Pr[\mathsf{Exp}^{sRKA-forge-rand}_{\mathcal{A},\Sigma,\{C\}}(1) \longrightarrow 1]\\ &\quad - \Pr[\mathsf{Exp}^{sRKA-forge-rand}_{\mathcal{A},\Sigma,\{C\}}(0) \longrightarrow 1]|\\ &\quad + \Pr[\mathsf{Exp}^{sRKA-forge-rand}_{\mathcal{A},\Sigma,\{C\}}(0) \longrightarrow 1]\end{aligned} \tag{16}$$

Evidently, there exists an adversary $\mathcal{B}_{forge}$ against the existential unforgeability of Σ' such that Term (16) is upper bounded by $\mathsf{Adv}^{forge}_{\mathcal{B}_{forge},\Sigma'}(\lambda)$. From the above reductions, we have,

$$\mathsf{Adv}^{aRKA-forge}_{\mathcal{A},\Sigma,\{C\}}(\lambda) \leq 2n(Adv^{aCI-dist}_{\mathcal{B}_{ci},\mathcal{H},\{C\}}(\lambda) + \mathsf{Adv}^{forge}_{\mathcal{B}_{forge},\Sigma'}(\lambda)).$$

This concludes the proof of the Theorem 7. ■

Remark 7.1 *Note that we implicitly assume the existence of a existentially unforgeable scheme Σ' in the above consturctions of $\{C\}$-(s/a)RKA-unforgeable schemes, since the former is implied by any $\{C\}$-(s/a)CI-speudorandom function family.*

Discussion In the case that the starting CI-secure hash function has a public key, the above transformation results in a cryptographic primitive for which algorithms operating on the secret key also need to access an authentic public key. In some scenarios, e.g. smart cards, this may not always be practical. Moreover, our main construction of CI-secure hash is only selectively-secure, resulting in a selectively RKA-secure the cryptographic primitive. On the other hand, it is sometimes possible to use our techniques (in a "non-blackbox" way) to design an RKA-secure scheme without a public key and that is adaptively secure. In particular, we show this for RKA-secure symmetric-key encryption recently introduced in [AHI10] in the full version. We also mention that using the CI-secure hash functions derived from the Bellare-Cash RKA-secure PRFs [BC10] avoid these issues, but for a weaker class of relations.

Relation to Tampering Attacks We also note that RKA-security for a cryptographic primitive can also be used to to protect against tampering attacks [GLM$^+$04], where, for instance, the secret key stored by a smart card is tampered with and its behavior is observed while it acts using the tampered secret, with an objective of gaining advantage against the security of the functionality of the smart card when using the original secret. However, as discussed in [AHI10], security against tampering attacks is easier to achieve in general, through some kind of "sanity check" on the secret key (for instance, by including a signature on the secret key as a part of the public key, which is verified by any algorithm using the former); although, as discussed above, this approach may not always be practical. This does not work for RKA-security, since we actually want related secret keys to function like *independently generated* ones.

6 Adaptive RKA-secure wPRF and Symmetric-Key Encryption Schemes

In this section, we give a number-theoretic construction of $\{C\}$-aRKA-CPA symmetric-key encryption scheme, where $\{C\}$ is the set of all non-zero polynomials over $\mathbb{Z}_p$. This, we do so, by first giving a construction of $poly(\lambda) - \{C\}$-aRKA-wPRF for the same class of related-key circuits and then showing how to tranform the same to $\{C\}$-aRKA-CPA symmetric-key encryption scheme.

Notations. Let $\mathcal{M}$ denote a message space and let m_{fix} be any fixed message in $\mathcal{M}$.

We next state our complexity assumption that will be used in the proofs that follow.

Decisonal Truncated q-ADHE Problem The Decisonal Truncated q-ADHE problem in $\mathbb{G}$ is defined as follows: given a $(q+1)$-tuple $(g, g^x, g^{x^2}, \ldots, g^{x^q}) \in (\mathbb{G}^\star)^{q+1}$ for some unknown $x \in \mathbb{Z}_p^\star$, distinguish between $g^{x^{q+1}}$ and a random element $R \xleftarrow{\$} \mathbb{G}$.
An algorithm $\mathcal{A}$ solves the decisional truncated q-ADHE problem in $\mathbb{G}$ with advantage ϵ if

$$\begin{aligned}\mathsf{DTADHE\text{-}Adv}_{\mathcal{A},q} := |\Pr[\mathcal{A}(g, g^x, g^{x^2}, \ldots, g^{x^q}, g^{x^{q+1}}) = 1] \\ - \Pr[\mathcal{A}(g, g^x, g^{x^2}, \ldots, g^{x^q}, R) = 1]| \geq \epsilon\end{aligned}$$

where the probability is over the random choice of generator $g \in \mathbb{G}^\star$, the random choice of $x \in \mathbb{Z}_p^\star$, the random choice of $R \in \mathbb{G}^\star$, and the random bits consumed by $\mathcal{A}$. The distribution on the left is referred to as $\mathcal{P}_{\mathcal{DTADHE}}$ and the distribution on the right as $\mathcal{R}_{\mathcal{DTADHE}}$.

Definition 10 *We say that the decisional truncated (q, t, ϵ)-ADHE assumption holds in $\mathbb{G}$ (or* Grp-Gen *satisfies the decisional truncated (q, t, ϵ)-ADHE assumption) if no probabilistic t-time algorithm has advantage at least ϵ in solving the decisional truncated q-ADHE problem in $\mathbb{G}$.*

6.1 Construction of $\{C\}$-aRKA-CPA Symmetric-Key Encryption Scheme

In the following, we give our number-theoretic construction of $\{C\}$-aRKA-CPA symmetric-key encryption scheme Π, where $\{C\}$ is the set of all non-zero polynomials over a finite field $\mathbb{Z}_p$.

The symmetric-key encryption scheme $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is defined by:

- $\mathsf{KeyGen}(1^\lambda)$: Let $(\mathbb{G}, p, g) \xleftarrow{\$} \mathsf{GrpGen}(1^\lambda)$. Sample $k \xleftarrow{\$} \mathbb{Z}_p$. Set $\mathbb{G}$ as the message space. Output $(\mathbb{G}, p, g)$ as public parameters *params* and k as the secret key.
- $\mathsf{Enc}_{params}(k, m_i)$: Sample $g_i \xleftarrow{\$} \mathbb{G}$. Output $(g_i, {g_i}^k.m_i)$.
- $\mathsf{Dec}_{params}(sk, c)$: For $c = (c_1, c_2)$, output $({c_1}^k)^{-1}.c_2$.

Figure 5: Our Construction of $\{C\}$-aRKA-CPA Symmetric-Key Encryption Scheme

Before we prove the adaptive RKA-security of the above scheme, for the sake of simplicity of exposition, we shall first give a construction of a function which we prove to be $poly(\lambda) - \{C\}$-aRKA-wPRF, and then prove that the above scheme is $\{C\}$-aRKA-CPA secure as a corollary.

Construction of $poly(\lambda) - \{C\}$-aRKA-wPRF for Polynomial RKD Circuits. The construction of the proposed function follows.

Let $(\mathbb{G}, p, g') \xleftarrow{\$} \mathsf{GrpGen}(1^\lambda)$. We define a family of functions $F : \mathbb{Z}_p \times \mathbb{G} \longrightarrow \mathbb{G}$ as follows. The function with key k, $F_k(\cdot)$, is defined by:

for any input $g \in \mathbb{G}$,

$$F_k(g) := g^k.$$

The distribution of functions in F is induced by the uniform random distribution of the key k in $\mathbb{Z}_p$.

Figure 6: Our Construction of $poly(\lambda) - \{C\}$-aRKA-wPRF

We now show that F as defined in the above construction is a weak-PRF which is secure against adaptive related-key-attacks with respect to $\{C\}$, where $\{C\}$ is the set of all non-zero polynomials over a finite field $\mathbb{Z}_p$, and the adversary is allowed to query for polynomially-many RKD circuits. Infact, we show this for a slightly weaker form of security than of $poly(\lambda) - \{C\}$-aRKA-wPRF where each query is responded for a freshly chosen random input. However, as we shall see, this suffices for us to give $\{C\}$-aRKA-CPA symmetric-key encryption scheme. The proof of security follows.

Theorem 8 *Let F be a family of functions as defined in Figure 6 and $\{C\}$ be the set of all non-zero polynomials over $\mathbb{Z}_p$. If Decisonal Truncated q-ADHE assumption holds in $\mathbb{G}$, then F is a $poly(\lambda) - \{C\}$-aRKA-wPRF, provided q upper bounds the degree of the polynomials in $\{C\}$.*

Proof: Suppose there exists a $\{C\}$-restricted adversary $\mathcal{A} \in \mathsf{PPT}$ against RKA-wPRF security of F with advantage greater than $\epsilon(\lambda)$ (i.e., $\mathsf{Adv}^{aRKA-wPRF}_{\mathcal{A},F,\{C\},poly(\lambda)}(\lambda) > \epsilon(\lambda)$). Then we build an algorithm $\mathcal{B}$ that interacts with $\mathcal{A}$ and solves a given random instance of Decisonal Truncated q-ADHE problem such that

$$\mathsf{DTADHE\text{-}Adv}_{\mathcal{B},q}(\lambda) > \frac{1}{n}.\epsilon(\lambda).$$

Algorithm $\mathcal{B}$ is given a $q+2$-tuple $(g', (g')^x, (g')^{x^2}, \ldots, (g')^{x^q}, T) \in (\mathbb{G}^\star)^{q+2}$ for some unknown $x \in \mathbb{Z}_p^\star$, where the tuple is either sampled from $\mathcal{P}_{\mathcal{DTADHE}}$ (i.e., $T = (g')^{x^{q+1}}$) or from $\mathcal{R}_{\mathcal{DTADHE}}$ (i.e., T is a uniform and independent random element in $\mathbb{G}$). The objective of $\mathcal{B}$ is to output 1 if $T = (g')^{x^{q+1}}$ and 0 otherwise. Denote $(g')^{x^i}$ by g'_i for $i \in [q]$, and g' by g'_0.

Let n be the maximum number of queries of $\mathcal{A}$. We define $n+1$ hybrids, H^i_n, for $0 \leq i \leq n$: a sequence of n elements, where the first i elements are outputs of the PRF function and the rest are uniform and independent random elements in $\mathbb{G}$.

$\mathcal{B}$ invokes $\mathcal{A}$ and interacts with it as follows:

1. Choose $w \xleftarrow{\$} [n]$.

2. For $i < w$, respond to a query $f_i(X) = \sum_{j=0}^{q} \alpha_j X^j$ as follows: choose $r_i \xleftarrow{\$} \mathbb{Z}_p$ and compute the pair (g_i, prf_i) as

$$g_i \longleftarrow (g'_0)^{r^i}$$
$$prf_i \longleftarrow (\prod_{j=0}^{q} (g'_j)^{\alpha_j})^{r_i}$$

3. For $i = w$, respond to a query $f_w(X) = \sum_{j=0}^{q} \alpha_j X^j$ as follows: choose a random polynomial $f(X) \xleftarrow{\$} \mathbb{Z}_p[X]$ such that $d := deg(f(X)) = q + 1 - deg(f_w(X))$. (Note that $d \geq 1$). Let $f(X) = \sum_{j=0}^{d} \beta_j X^j$ and $f(X)f_w(X) = \sum_{j=0}^{q+1} \delta_j X^j$. Choose $r_w \xleftarrow{\$} \mathbb{Z}_p$ and compute the pair (g_w, prf_w) as

$$g_w \longleftarrow (\prod_{j=0}^{d} (g'_j)^{\beta_j})^{r_w}$$
$$prf_w \longleftarrow (\prod_{j=0}^{q+1} (g'_j)^{\delta_j})^{r_w}$$

4. For $i > w$, respond to a query $f_i(X) = \sum_{j=0}^{q} \alpha_j X^j$ as follows: set the pair (g_i, prf_i) as

$$g_i \xleftarrow{\$} \mathbb{G}$$
$$prf_i \xleftarrow{\$} \mathbb{G}$$

Note that for $i < w$, $g_i \xleftarrow{\$} \mathbb{G}$ and $prf_i = (g_i)^{f_i(x)}$, and for $i > w$, g_i and prf_i are uniformly and independently random in $\mathbb{G}$. If $T = (g')^{x^{q+1}}$, then $g_w = (g'_0)^{f(x)}$ and $prf_w = (g_w)^{f_w(x)}$. As $f(X)$ is a random non-zero polynomial, g_w is uniformly and independently random in $\mathbb{G}$. If T is independently random element in $\mathbb{G}$, then g_w and prf_w are uniformly and independently random elements in $\mathbb{G}$. Thus, the resulting hybrid is H_n^w if $T = (g')^{x^{q+1}}$, otherwise, it is H_n^{w-1}.

Observe that

$$\Pr[\mathcal{B}(\mathcal{P}_{\mathcal{DTADHE}}) \longrightarrow 1 | w = n] = \Pr[k \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathcal{O}_{F(k,\cdot)}(\cdot)} \longrightarrow 1],$$

$$\Pr[\mathcal{B}(\mathcal{R}_{\mathcal{DTADHE}}) \longrightarrow 1 | w = 1] = \Pr[k \xleftarrow{\$} \mathcal{K}, G \xleftarrow{\$} \mathsf{FF}(\mathcal{K}, \mathcal{D}, \mathcal{R}) : \mathcal{A}^{\mathcal{O}_{G(k,\cdot)}(\cdot)} \longrightarrow 1],$$

and, for any $1 \leq j < n$,

$$\Pr[\mathcal{B}(\mathcal{P}_{\mathcal{DTADHE}}) \longrightarrow 1 | w = j] = \Pr[\mathcal{B}(\mathcal{R}_{\mathcal{DTADHE}}) \longrightarrow 1 | w = j + 1]$$

Thus, we have,

$$
\begin{aligned}
\mathsf{DTADHE}\ \mathsf{Adv}_{\mathcal{B},q}(\lambda) &= |\Pr[\mathcal{B}(\mathcal{P}_{\mathcal{DTADHE}}) \longrightarrow 1] - \Pr[\mathcal{B}(\mathcal{R}_{\mathcal{DTADHE}}) \longrightarrow 1]| \\
&= |\frac{1}{n}\sum_{i=1}^{n} \Pr[\mathcal{B}(\mathcal{P}_{\mathcal{DTADHE}}) \longrightarrow 1|w=i] \\
&- \frac{1}{n}\sum_{i=1}^{n} \Pr[\mathcal{B}(\mathcal{R}_{\mathcal{DTADHE}}) \longrightarrow 1|w=i]| \\
&= \frac{1}{n}|\Pr[\mathcal{B}(\mathcal{P}_{\mathcal{DTADHE}}) \longrightarrow 1|w=n] - \Pr[\mathcal{B}(\mathcal{R}_{\mathcal{DTADHE}}) \longrightarrow 1|w=1]| \\
&= \frac{1}{n}|\Pr[k \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathcal{O}_{F(k,\cdot)}(\cdot)} \longrightarrow 1] \\
&- \Pr[k \xleftarrow{\$} \mathcal{K}, G \xleftarrow{\$} \mathsf{FF}(\mathcal{K},\mathcal{D},\mathcal{R}) : \mathcal{A}^{\mathcal{O}_{G(k,\cdot)}(\cdot)} \longrightarrow 1]| \\
&> \frac{1}{n}.\epsilon(\lambda).
\end{aligned}
$$

This completes the proof of Theorem 8. ■

Using the above theorem, we now show that the above scheme Π (as defined in Figure 5) is $\{C\}$-aRKA-CPA symmetric-key encryption scheme, where $\{C\}$ is the set of all non-zero polynomials over $\mathbb{Z}_p$.

Corollary 8.1 *Let Π be a symmetric-key encryption scheme as defined in Figure 5 and $\{C\}$ be the set of all non-zero polynomials over $\mathbb{Z}_p$. If Decisonal Truncated q'-ADHE assumption holds in $\mathbb{G}$, then Π is a $\{C\}$-aRKA-CPA symmetric-key encryption scheme, provided q upper bounds the degree of the polynomials in $\{C\}$.*

Proof: Suppose there exists a $\{C\}$-restricted adversary $\mathcal{A} \in \mathsf{PPT}$ against aRKA-CPA security of Π. Then we build an algorithm $\mathcal{B}$ against aRKA-wPRF security of F such that

$$\mathsf{Adv}^{aRKA-CPA-symm}_{\mathcal{A},\Pi,\{C\}}(\lambda) \leq 2Adv^{aRKA-wPRF}_{\mathcal{B},F,\{C\},poly(\lambda)}(\lambda)$$

Algorithm $\mathcal{B}$ is given an oracle $\mathcal{O}_{F'(k,\cdot)}(\cdot)$ which either instantiates the wPRF F (specified by $(\mathbb{G},p,g') \leftarrow \mathsf{GrpGen}(1^\lambda)$) for a uniformly random key $k \xleftarrow{\$} \mathcal{K}$ or a random function G for $k \xleftarrow{\$} \mathcal{K}$ and $G \xleftarrow{\$} \mathsf{FF}(\mathcal{K},\mathcal{D},\mathcal{R})$. The objective of $\mathcal{B}$ is to output 1 in the former case represented by $b_{prf} = 1$ and 0 in the latter case represented by $b_{prf} = 0$. $\mathcal{B}$ chooses $b_{msg} \xleftarrow{\$} \{0,1\}$, invokes $\mathcal{A}$, and runs the experiment $\mathsf{Exp}^{aRKA-CPA}_{\mathcal{A},\Pi,\{C\}}(b_{prf},b_{msg})$ defined as follows:

1. Give $(\mathbb{G},p,g')$ to $\mathcal{A}$ as the specification of the public parameters.
2. When $\mathcal{A}$ presents its ith query (C_i,m_i), query the oracle with C_i. Let (g_i,z_i) be the response of the oracle. Respond to $\mathcal{A}$ via $c_i := (g_i, z_i.m_i)$.
3. When $\mathcal{A}$ outputs its guess $\hat{b}_{msg}$, output the same.

$$
\begin{aligned}
&\mathsf{Adv}^{aRKA-CPA-symm}_{\mathcal{A},\Pi,\{C\}}(\lambda) \\
&= |\Pr[\mathsf{Exp}^{aRKA-CPA-symm}_{\mathcal{A},\Pi,\{C\}}(1) \longrightarrow 1] - \Pr[\mathsf{Exp}^{aRKA-CPA-symm}_{\mathcal{A},\Pi,\{C\}}(0) \longrightarrow 1]| && (17)\\
&= |\Pr[Exp^{aRKA-CPA}_{\mathcal{A},\Pi,\{C\}}(1,1) \longrightarrow 1] - \Pr[Exp^{aRKA-CPA}_{\mathcal{A},\Pi,\{C\}}(1,0) \longrightarrow 1]| && (18)\\
&\leq |\Pr[Exp^{aRKA-CPA}_{\mathcal{A},\Pi,\{C\}}(1,1) \longrightarrow 1] - \Pr[Exp^{aRKA-CPA}_{\mathcal{A},\Pi,\{C\}}(0,1) \longrightarrow 1]| && (19)\\
&+|\Pr[Exp^{aRKA-CPA}_{\mathcal{A},\Pi,\{C\}}(0,1) \longrightarrow 1] - \Pr[Exp^{aRKA-CPA}_{\mathcal{A},\Pi,\{C\}}(0,0) \longrightarrow 1]| && (20)\\
&+|\Pr[Exp^{aRKA-CPA}_{\mathcal{A},\Pi,\{C\}}(0,0) \longrightarrow 1] - \Pr[Exp^{aRKA-CPA}_{\mathcal{A},\Pi,\{C\}}(1,0) \longrightarrow 1]| && (21)
\end{aligned}
$$

Equation (18) follows from the fact that, in the view of $\mathcal{A}$, $\mathsf{Exp}^{aRKA-CPA-symm}_{\mathcal{A},\Pi,\{C\}}(b)$ and $\mathsf{Exp}^{aRKA-CPA}_{\mathcal{A},\Pi,\{C\}}(1,b)$ are identical.

In $Exp^{aRKA-CPA}_{\mathcal{A},\Pi,\{C\}}(0,b)$, each response $c_i := (g_i, z_i.m'_i)$ is independent of the message encrypted in the view of $\mathcal{A}$, since g_i and z_i are uniform and independent random elements in $\mathbb{G}$. Hence, $Exp^{aRKA-CPA}_{\mathcal{A},\Pi,\{C\}}(0,1)$ is information-theoretically indistinguisable from $Exp^{aRKA-CPA}_{\mathcal{A},\Pi,\{C\}}(0,0)$. Hence, Term (20) is 0. Further, it is easy to see that Term (19) and Term (21) are upper-bounded by $\mathsf{Adv}^{aRKA-wPRF}_{\mathcal{B},F,\{C\},poly(\lambda)}(\lambda)$. This completes the proof of Corollary 8.1. ■

Acknowledgements

We thank the anonymous reviewers of TCC 2011 for detailed comments and suggestions. The second author thanks David Cash for discussions about related-key security, and was supported in part by Brent Waters' grants NSF CNS-0915361 and CNS-0952692. Thanks also to Mihir Bellare and Ran Canetti for useful comments.

References

[AHI10] Benny Applebaum, Danny Harnik, and Yuval Ishai. Semantic security under related-key attacks and applications. Cryptology ePrint Archive, Report 2010/544, 2010. `http://eprint.iacr.org/`.

[BB04a] Dan Boneh. and Xavier Boyen. Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles. In *Advances in Cryptology – Eurocrypt*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004.

[BB04b] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73. Springer, 2004.

[BBO07] Mihir Bellare, Alexandra Boldyreva, and Adam O'Neill. Deterministic and efficiently searchable encryption. In *CRYPTO*, pages 535–552, 2007.

[BC10] Mihir Bellare and David Cash. Pseudorandom functions and permutations provably secure against related-key attacks. In *CRYPTO*, pages 666–684, 2010.

[BCFW09] Alexandra Boldyreva, David Cash, Marc Fischlin, and Bogdan Warinschi. Foundations of non-malleable hash and one-way functions. In *ASIACRYPT*, pages 524–541, 2009.

[BCM11] Mihir Bellare, David Cash, and Rachel Miller. A broader perspective on related-key attacks and a connection with key-dependent message security. Manuscript, 2011.

[BCOP04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In *EUROCRYPT*, pages 506–522, 2004.

[BFO08] Alexandra Boldyreva, Serge Fehr, and Adam O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In *CRYPTO*, pages 335–359, 2008.

[BFOR08] Mihir Bellare, Marc Fischlin, Adam O'Neill, and Thomas Ristenpart. Deterministic encryption: Definitional equivalences and constructions without random oracles. In *CRYPTO*, pages 360–378, 2008.

[BK03] Mihir Bellare and Tadayoshi Kohno. A theoretical treatment of related-key attacks: Rka-prps, rka-prfs, and applications. In *EUROCRYPT*, pages 491–506, 2003.

[BO81] Michael Ben-Or. Probabilistic algorithms in finite fields. In *FOCS*, pages 394–398, 1981.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[Can97] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In *CRYPTO*, pages 455–469, 1997.

[CFPR96] Don Coppersmith, Matthew K. Franklin, Jacques Patarin, and Michael K. Reiter. Low-exponent rsa with related messages. In *EUROCRYPT*, pages 1–9, 1996.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004.

[CMR98] Ran Canetti, Daniele Micciancio, and Omer Reingold. Perfectly one-way probabilistic hash functions (preliminary version). In *STOC*, pages 131–140, 1998.

[DP08] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *FOCS*, pages 293–302, 2008.

[DY05] Yevgeniy Dodis and Aleksandr Yampolskiy. A verifiable random function with short proofs and keys. In Serge Vaudenay, editor, *Public Key Cryptography*, volume 3386 of *Lecture Notes in Computer Science*, pages 416–431. Springer, 2005.

[Fei73] Horst Feistel. Cryptography and computer privacy. *Scientific American*, 228(5), 1973.

[GL10] David Goldenberg and Moses Liskov. On related-secret pseudorandomness. In *TCC*, pages 255–272, 2010.

[GLM+04] Rosario Gennaro, Anna Lysyanskaya, Tal Malkin, Silvio Micali, and Tal Rabin. Algorithmic tamper-proof (atp) security: Theoretical foundations for security against hardware tampering. In *TCC*, pages 258–277, 2004.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[HLO10] Brett Hemenway, Steve Lu, and Rafail Ostrovsky. Correlated product security from any one-way function and the new notion of decisional correlated product security. Cryptology ePrint Archive, Report 2010/100, 2010. `http://eprint.iacr.org/`.

[IKNP03] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. Extending oblivious transfers efficiently. In *CRYPTO*, pages 145–161, 2003.

[KPSY09] Eike Kiltz, Krzysztof Pietrzak, Martijn Stam, and Moti Yung. A new randomness extraction paradigm for hybrid encryption. In *EUROCRYPT*, pages 590–609, 2009.

[NR99] Moni Naor and Omer Reingold. Synthesizers and their application to the parallel construction of pseudo-random functions. *J. Comput. Syst. Sci.*, 58(2):336–375, 1999.

[O'N10] Adam O'Neill. Deterministic public-key encryption revisited. Cryptology ePrint Archive, Report 2010/533, 2010. `http://eprint.iacr.org/`.

[RS09] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. In *TCC*, pages 419–436, 2009.

[RTTV08] Omer Reingold, Luca Trevisan, Madhur Tulsiani, and Salil P. Vadhan. Dense subsets of pseudorandom sets. In *FOCS*, pages 76–85, 2008.

[Sha49] Claude E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4):656–715, 1949.

[WG00] David Wagner and Ian Goldberg. Proofs of security for the unix password hashing algorithm. In *ASIACRYPT*, pages 560–572, 2000.

A Adaptive Correlated-Input Security Definitions

The Adaptive Correlated-Input Inverting experiment $\mathsf{Exp}^{aCI-inv}_{\mathcal{A},\mathcal{H},\{C\}}$. For a family of deterministic functions $\mathcal{H}$, an adversary $\mathcal{A}$, and a family of efficiently-computable correlated-input circuits $\{C\}$, we define the following game between a challenger and the adversary $\mathcal{A}$.

- **Setup Phase**. Challenger runs the Gen algorithm of $\mathcal{H}$ for a security parameter input 1^λ and gets $h_c : \mathcal{D}_h \longrightarrow \mathcal{R}_h$. It gives h_c to $\mathcal{A}$ and chooses r, a uniform random string of appropriate length.
- **Adaptive Query-Response Phase**. $\mathcal{A}$ presents polynomially-many adaptive queries, where, an ith query is $C_i \in \{C\}$. Challenger responds to it via $h_c(C_i(r))$.
- **Invert Phase**. Let n denote the total number of $\mathcal{A}$'s queries. $\mathcal{A}$ outputs $(\hat{k}, \hat{y})$ for $\hat{k} \in [n]$ and $\hat{y} \in \mathcal{D}_h$.

The output of the experiment is defined to be 1 if $h_c(\hat{y}) = h_c(C_{\hat{k}}(r))$ and 0 otherwise.
We define the advantage of an adversary $\mathcal{A}$ in the above game as:

$$\mathsf{Adv}^{aCI-inv}_{\mathcal{A},\mathcal{H},\{C\}}(\lambda) = \Pr[\mathsf{Exp}^{aCI-inv}_{\mathcal{A},\mathcal{H},\{C\}} = 1]$$

The probability is over the random bits used by the challenger and the adversary.

Definition 11 *A family of functions $\mathcal{H}$ is said to be adaptive correlated-input one-way with respect to a family of correlated-input circuits $\{C\}$, if for all $\mathcal{A} \in \mathsf{PPT}$ there exists a negligible function $negl$, such that:*

$$Adv^{aCI-inv}_{\mathcal{A},\mathcal{H},\{C\}}(\lambda) \leq negl(\lambda)$$

The Adaptive Correlated-Input Predicting experiment $\mathsf{Exp}^{aCI-pred}_{\mathcal{A},\mathcal{H},\{C\}}$. For a family of deterministic functions $\mathcal{H}$, an adversary $\mathcal{A}$, and a family of efficiently-computable correlated-input circuits $\{C\}$, we define the following game between a challenger and the adversary $\mathcal{A}$.

- **Setup Phase**. Challenger runs the Gen algorithm of $\mathcal{H}$ for a security parameter input 1^λ and gets $h_c : \mathcal{D}_h \longrightarrow \mathcal{R}_h$. It gives h_c to $\mathcal{A}$ and chooses r, a uniform random string of appropriate length.
- **Adaptive Query-Response Phase**. $\mathcal{A}$ presents polynomially-many adaptive queries, where, an ith query is $C_i \in \{C\}$. Challenger responds to it via $h_c(C_i(r))$.
- **Predict Phase**. Let n denote the total number of $\mathcal{A}$'s queries. $\mathcal{A}$ outputs $(C_{n+1}, \hat{y}) \in (\{C\}, \mathcal{R}_h)$.

The output of the experiment is defined to be 1 if $C_{n+1} \neq C_i$ for all $i \in [n]$ and $\hat{y} = h_c(C_{n+1}(r))$, and 0 otherwise.
We define the advantage of an adversary $\mathcal{A}$ in the above game as:

$$\mathsf{Adv}^{aCI-pred}_{\mathcal{A},\mathcal{H},\{C\}}(\lambda) = \Pr[\mathsf{Exp}^{aCI-pred}_{\mathcal{A},\mathcal{H},\{C\}} = 1]$$

The probability is over the random bits used by the challenger and the adversary.

Definition 12 *A family of functions $\mathcal{H}$ is said to be adaptive correlated-input unpredictable with respect to a family of correlated-input circuits $\{C\}$, if for all $\mathcal{A} \in \mathsf{PPT}$ there exists a negligible function negl, such that:*

$$\mathsf{Adv}^{aCI-pred}_{\mathcal{A},\mathcal{H},\{C\}}(\lambda) \leq negl(\lambda)$$

The Adaptive Correlated-Input Distinguishing experiment $\mathsf{Exp}^{aCI-dist}_{\mathcal{A},\mathcal{H},\{C\}}(b)$. For a family of deterministic functions $\mathcal{H}$, an adversary $\mathcal{A}$, and a family of efficiently-computable correlated-input circuits $\{C\}$, and a random bit b, we define the following game between a challenger and the adversary $\mathcal{A}$.

- **Setup Phase**. Challenger runs the Gen algorithm of $\mathcal{H}$ for a security parameter input 1^λ and gets $h_c : \mathcal{D}_h \longrightarrow \mathcal{R}_h$. It gives h_c to $\mathcal{A}$ and chooses r, a uniform random string of appropriate length.
- **Adaptive Query-Response Phase**. $\mathcal{A}$ presents polynomially-many adaptive queries, where, an ith query is $C_i \in \{C\}$. Challenger responds to it via $h_c(C_i(r))$.
- **Challenge Phase**. $\mathcal{A}$ presents a challenge circuit $C_{n+1} \in \{C\}$ such that $C_{n+1} \neq C_i$ for all $i \in [n]$. Let $z_0 \xleftarrow{\$} \mathcal{R}_h$ and $z_1 := h_c(C_{n+1}(r))$. The challenger responds via z_b.
- **Guess Phase**. The adversary outputs a guess $\hat{b}$ of b.

$\hat{b}$ is defined to be the output of the experiment.
We define the advantage of an adversary $\mathcal{A}$ in the above game as:

$$\mathsf{Adv}^{aCI-dist}_{\mathcal{A},\mathcal{H},\{C\}}(\lambda) = |\Pr[\mathsf{Exp}^{aCI-dist}_{\mathcal{A},\mathcal{H},\{C\}}(1) \longrightarrow 1] - \Pr[\mathsf{Exp}^{aCI-dist}_{\mathcal{A},\mathcal{H},\{C\}}(0) \longrightarrow 1]|$$

The probability is over the random bits used by the challenger and the adversary.

Definition 13 *A family of functions $\mathcal{H}$ is said to be adaptive correlated-input pseudorandom with respect to a family of correlated-input circuits $\{C\}$, if for all $\mathcal{A} \in \mathsf{PPT}$ there exists a negligible function negl, such that:*

$$\mathsf{Adv}^{aCI-dist}_{\mathcal{A},\mathcal{H},\{C\}}(\lambda) \leq negl(\lambda)$$

B Definitions of RKA-Secure Primitives

We now give definitions of RKA-secure PRFs.

Definition 14 ($q_{inp} - \{C\}$-aRKA-PRF) *Let $F : \mathcal{K} \times \mathcal{D} \longrightarrow \mathcal{R}$ be an efficiently computable function. Let $\{C\} \subseteq \mathsf{Fun}(\mathcal{K},\mathcal{K})$ be a set of RKD circuits. F is said to be $q_{inp} - \{C\}$-aRKA-PRF, if, $\forall \mathcal{A} \in \mathsf{PPT}$, $\mathsf{Adv}^{aRKA-PRF}_{\mathcal{A},F,\{C\},q_{inp}}(\lambda)$ is negligible, where,*

$$\begin{aligned}\mathsf{Adv}^{aRKA-PRF}_{\mathcal{A},F,\{C\},q_{inp}}(\lambda) :=& |\Pr[k \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathcal{O}_{F(k,\cdot)}(\cdot,\cdot)} \longrightarrow 1] \\ & - \Pr[k \xleftarrow{\$} \mathcal{K}, G \xleftarrow{\$} \mathsf{FF}(\mathcal{K},\mathcal{D},\mathcal{R}) : \mathcal{A}^{\mathcal{O}_{G(k,\cdot)}(\cdot,\cdot)} \longrightarrow 1]|\end{aligned} \tag{22}$$

Here, the related-key-prf oracle $\mathcal{O}_{f(k,\cdot)}(\cdot,\cdot)$ takes two inputs, $C_i \in \{C\}$ and $x_i \in \mathcal{D}$, and outputs $f(C_i(k), x_i)$.

Definition 15 ($q_{inp} - \{C\}$**-sRKA-PRF.)** *Let $F : \mathcal{K} \times \mathcal{D} \longrightarrow \mathcal{R}$ be an efficiently computable function. Let $\{C\} \subseteq \mathsf{Fun}(\mathcal{K}, \mathcal{K})$ be a set of RKD circuits. F is said to be $q_{inp} - \{C\}$-sRKA-PRF, if, $\forall \mathcal{A} \in \mathsf{PPT}$:* $\mathsf{Adv}^{sRKA-PRF}_{\mathcal{A},F,\{C\},q_{inp}}(\lambda) := |\Pr[k \xleftarrow{\$} \mathcal{K} : \{(C_i, x_i)\}_{i\in[n]} \leftarrow \mathcal{A}, \mathcal{A}(\{F(C_i(k), x_i))\}_{i\in[n]} \longrightarrow 1] - \Pr[k \xleftarrow{\$} \mathcal{K}, G \xleftarrow{\$} \mathsf{FF}(\mathcal{K}, \mathcal{D}, \mathcal{R}) : \{(C_i, x_i)\}_{i\in[n]} \leftarrow \mathcal{A}, \mathcal{A}(G(C_i(k), x_i)) \longrightarrow 1]|$
is negligible in λ, where $C_i \in \{C\}$ and $x_i \in \mathcal{D}$, and $|\{(x_i)\}_i| \leq q_{inp}$.

We now give definitions of RKA-secure symmetric-key encryption.

The Selective RKA - CPA Symmetric-Key experiment $\mathsf{Exp}^{sRKA-CPA-symm}_{\mathcal{A},\Pi,\{C\}}(b)$**.** For a symmetric-key encryption scheme $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, an adversary $\mathcal{A}$, a family of efficiently-computable RKD circuits $\{C\}$, and a random bit b, we define the following game between a challenger and the adversary $\mathcal{A}$.

- **Setup Phase 1**. Challenger runs $\mathsf{KeyGen}(1^\lambda)$, gets *params* including a description of a keyspace $\mathcal{K}_\lambda$, and chooses $k \xleftarrow{\$} \mathcal{K}_\lambda$. The challenger specifies $\mathcal{K}_\lambda$ to $\mathcal{A}$.
- **Challenge Phase**. $\mathcal{A}$ chooses a positive integer $n(= poly(\lambda))$, and gives to the challenger n circuits $C_i \in \{C\}$, $i \in [n]$.
- **Setup Phase 2**. The challenger gives *params* to $\mathcal{A}$.
- **Partially Adaptive Query-Response Phase**. The adversary $\mathcal{A}$ presents the adaptive ith query, $(j, m_i) \in ([n], \mathcal{M})$ and the challenger responds via:

$$c_i = \begin{cases} \mathsf{Enc}_{params}(C_j(k), m_i), & \text{if b=1} \\ \mathsf{Enc}_{params}(C_j(k), m_{fix}), & \text{if b=0} \end{cases}$$

- **Guess Phase**. The adversary outputs a guess $\hat{b}$ of b, which is also the output of the experiment.

We define the advantage of an adversary $\mathcal{A}$ in the above game as:

$$\mathsf{Adv}^{sRKA-CPA-symm}_{\mathcal{A},\Pi,\{C\}}(\lambda) = |\Pr[\mathsf{Exp}^{sRKA-CPA-symm}_{\mathcal{A},\Pi,\{C\}}(1) \longrightarrow 1] - \Pr[\mathsf{Exp}^{sRKA-CPA-symm}_{\mathcal{A},\Pi,\{C\}}(0) \longrightarrow 1]|$$

The probability is over the random bits used by the challenger and the adversary.

Definition 16 *A symmetric-key encryption scheme Π is said to be secure against selective RKA-CPA attack with respect to a family of correlated-input circuits $\{C\}$, if for all $\mathcal{A} \in \mathsf{PPT}$ there exists a negligible function negl, such that:*

$$\mathsf{Adv}^{sRKA-CPA-symm}_{\mathcal{A},\Pi,\{C\}}(\lambda) \leq negl(\lambda)$$

We shall call such a scheme, $\{C\}$-sRKA-CPA symmetric-key encryption scheme.

The Adaptive RKA - CPA Symmetric-Key experiment $\mathsf{Exp}^{aRKA-CPA-symm}_{\mathcal{A},\Pi,\{C\}}(b)$**.** For a symmetric-key encryption scheme $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, an adversary $\mathcal{A}$, a family of efficiently-computable correlated-input circuits $\{C\}$, and a random bit b, we define the following game between a challenger and the adversary $\mathcal{A}$.

- **Setup Phase**. The challenger runs $\mathsf{KeyGen}(1^\lambda)$, gets *params* including a description of a keyspace $\mathcal{K}_\lambda$, and chooses $k \xleftarrow{\$} \mathcal{K}_\lambda$. The challenger specifies $\mathcal{K}_\lambda$ to $\mathcal{A}$.

- **Adaptive Query-Response Phase**. $\mathcal{A}$ presents the adaptive ith query, $(C_i, m_i) \in (\{C\}, \mathcal{M})$, for $1 \leq i \leq n$, and the challenger responds via c_i:

$$c_i = \begin{cases} \mathsf{Enc}_{params}(C_i(k), (m_i) \text{ if b=1} \\ \mathsf{Enc}_{params}(C_i(k), m_{fix}) \text{ if b=0} \end{cases}$$

- **Guess Phase**. The adversary outputs a guess $\hat{b}$ of b, which is also the output of the experiment.

We define the advantage of an adversary $\mathcal{A}$ in the above game as:

$$\begin{aligned} &\mathsf{Adv}^{aRKA-CPA-symm}_{\mathcal{A},\Pi,\{C\}}(\lambda) \\ &= |\Pr[\mathsf{Exp}^{aRKA-CPA-symm}_{\mathcal{A},\Pi,\{C\}}(1) \longrightarrow 1] - \Pr[\mathsf{Exp}^{aRKA-CPA-symm}_{\mathcal{A},\Pi,\{C\}}(0) \longrightarrow 1]| \end{aligned}$$

The probability is over the random bits used by the challenger and the adversary.

Definition 17 *A symmetric-key encryption scheme Π is said to be secure against adaptive RKA-CPA attack with respect to a family of correlated-input circuits $\{C\}$, if for all $\mathcal{A} \in \mathsf{PPT}$ there exists a negligible function negl, such that:*

$$\mathsf{Adv}^{aRKA-CPA-symm}_{\mathcal{A},\Pi,\{C\}}(\lambda) \leq negl(\lambda)$$

We shall call such a scheme, $\{C\}$-aRKA-CPA symmetric-key encryption scheme.

Remark 8.1 *Our definitions of $\{C\}$-sRKA-CPA symmetric-key encryption (resp. $\{C\}$-aRKA-CPA symmetric-key encryption) are stronger than the adaptive (resp. passive) RKA-secure symmetric-key encryption of [AHI10] in the following sense: While in [AHI10], the challenger chooses the RKD circuits, in our definitions, we let the adversary choose them in both selective and adaptive cases (with the adversary declaring the number of distinct related-key circuits that it would adaptively query, before it presents those queries).*

We now give definitions of RKA-secure signature scheme.

For a signature scheme $\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ we denote by $\mathcal{SK}_\lambda$ and $\mathcal{PK}_\lambda$ the sets of secret keys and public keys that are produced by $\mathsf{KeyGen}(1^\lambda)$. Let $\{C\} \subseteq \mathsf{Fun}(\mathcal{SK}_\lambda, \mathcal{SK}_\lambda)$ be a set of RKD functions.

The Adaptive RKA Signature-Forging experiment $\mathsf{Exp}^{aRKA-forge}_{\mathcal{A},\Sigma,\{C\}}$. For a signature scheme Σ, an adversary $\mathcal{A}$, and a family of efficiently-computable RKD circuits $\{C\}$, we define the following game between a challenger and the adversary $\mathcal{A}$.

- $(sk, pk) \longleftarrow \mathsf{KeyGen}(1^\lambda)$.
- $(m, \sigma) \longleftarrow \mathcal{A}^{\mathbf{aRKA\text{-}Sign}_{sk}(\cdot,\cdot)}(pk)$

where, $\mathbf{aRKA\text{-}Sign}_{sk}(\cdot,\cdot)$, the aRKA-Signing oracle, takes as input a function $C \in \{C\}$ and a message m and outputs $\mathsf{Sign}(C(sk), m)$.

The output of the experiment is defined as 1 if $\mathsf{Verify}(pk, m, \sigma) = \mathsf{valid}$ and if for no $\mathcal{A}$'s oracle query of the form (id, m), the response was signature σ; otherwise, the output is 0.

Definition 18 *A signature scheme $\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is existentially unforgeable against adaptive related-key-attacks with respect to $\{C\}$, if for all $\mathcal{A} \in \mathsf{PPT}$ there exists a negligible function negl, such that:*

$$\mathsf{Adv}^{aRKA-forge}_{\mathcal{A},\Sigma,\{C\}}(\lambda) := \Pr[\mathsf{Exp}^{aRKA-forge}_{\mathcal{A},\Sigma,\{C\}} \longrightarrow 1] \leq negl(\lambda).$$

We call such a scheme $\{C\}$-aRKA-unforgeable.

The Selective RKA Signature-Forging experiment $\mathsf{Exp}^{sRKA-forge}_{\mathcal{A},\Sigma,\{C\}}$. For a signature scheme Σ, an adversary $\mathcal{A}$, and a family of efficiently-computable RKD circuits $\{C\}$, we define the following game between a challenger and the adversary $\mathcal{A}$.

- $(sk, pk) \longleftarrow \mathsf{KeyGen}(1^\lambda)$.
- $\{(C_i, m_i)\}_i \longleftarrow \mathcal{A}(pk)$
- $(m, \sigma) \longleftarrow \mathcal{A}(\{\sigma_i\}_i)$

where, $\sigma_i \leftarrow \mathsf{Sign}(C_i(sk), m_i)$.

The output of the experiment is defined as 1 if $\mathsf{Verify}(pk, m, \sigma) = \mathsf{valid}$ and if for no $\mathcal{A}$'s oracle query of the form (id, m), the response was signature σ; otherwise, the output is 0.

Definition 19 *A signature scheme $\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is existentially unforgeable against selective related-key-attacks with respect to $\{C\}$, if for all $\mathcal{A} \in \mathsf{PPT}$ there exists a negligible function negl, such that:*

$$\mathsf{Adv}^{sRKA-forge}_{\mathcal{A},\Sigma,\{C\}}(\lambda) := \Pr[\mathsf{Exp}^{sRKA-forge}_{\mathcal{A},\Sigma,\{C\}} \longrightarrow 1] \leq negl(\lambda).$$

We call such a scheme $\{C\}$-sRKA-unforgeable.

C Inadequacy of Existing Security Notions

Here, we demonstrate that the primitives, *one-way functions* and *pseudo-random generators* are not necessarily correlated-input secure, by illustrating an example each of:

1. a one-way function that is not correlated-input one-way;
2. a pseudo-random generator that is neither correlated-input unpredictable nor correlated-input uniform.

One-Way Function vs Correlated-Input one-way functions: Let $f : \{0,1\}^n \longrightarrow \{0,1\}^n$ be any one-way function, where $n \in \mathbb{N}$. Consider a function $\hat{f} : \{0,1\}^{2n} \longrightarrow \{0,1\}^{2n}$ defined as $\hat{f}(x_1 \parallel x_2) = x_1 \parallel f(x_2)$, where $x_1, x_2 \in \{0,1\}^n$. Evidently, the function, $\hat{f}(\cdot)$ is one-way.

For any $x_1, x_2 \xleftarrow{\$} \{0,1\}^n$, let $y_1 = x_1 \parallel x_2$ and $y_2 = x_2 \parallel x_1$. We have, $\hat{f}(y_1) = x_1 \parallel f(x_2)$ and $\hat{f}(y_2) = x_2 \parallel f(x_1)$. Given $\hat{f}(y_1)$ and $\hat{f}(y_2)$, one can easily recover x_1 and x_2, and hence, invert $\hat{f}(Y)$ at both $Y = y_1$ and $Y = y_2$.

Hence, given the images of any two inputs of $\hat{f}(\cdot)$ that are correlated such that the first n bits of one input is equal to the last n bits of the other, it is possible to invert both the images, thus demonstrating that the one-way function, $\hat{f}(\cdot)$, is not correlated-input one-way for the aforementioned class of correlations.

One-Way Function vs Correlated-Input Unpredictable and Pseudorandom Functions: Let $G : \{0,1\}^n \longrightarrow \{0,1\}^{l(n)}$ be any pseudo-random generator, where $n \in \mathbb{N}$ and $l : \mathbb{N} \longrightarrow \mathbb{N}$ such that $l(n) > n$, $\forall n \in \mathbb{N}$. Consider a function $\hat{G} : \{0,1\}^{3n} \longrightarrow \{0,1\}^{2n+l(n)}$ defined as $\hat{G}(x_1 \parallel x_2 \parallel x_3) = x_1 \parallel x_2 \parallel G(x_3)$, where $x_1, x_2, x_3 \in \{0,1\}^n$. Evidently, the function, $\hat{G}(\cdot)$ is a pseudo-random generator.

For any $x_1, x_2, x_3 \xleftarrow{\$} \{0,1\}^n$, let$y_1 = x_1 \parallel x_2 \parallel x_3$, $y_2 = x_2 \parallel x_3 \parallel x_1$ and $y_3 = x_1 \parallel x_3 \parallel x_2$. We have, $\hat{G}(y_1) = x_1 \parallel x_2 \parallel G(x_3)$ and $\hat{G}(y_2) = x_2 \parallel x_3 \parallel G(x_1)$. Given $\hat{G}(y_1)$ and $\hat{G}(y_2)$, one can easily recover x_1, x_2 and x_3, and hence, can recover the seed y_3, with which $\hat{G}(y_3)$ can be generated.

Hence, given the outputs $\hat{G}(\cdot)$ for any two seeds, y_1 and y_2, one can not only distinguish the output of $\hat{G}(\cdot)$ for a seed y_3 that is correlated to y_1 and y_2 as aforementioned (each of the three n-bit parts of y_3 is in at least one of first $2n$ bits of y_1 and y_2), but also can recover the seed y_3 itself, and hence can predict its output. This demonstrates that the pseudo-random generator, $\hat{G}(\cdot)$, is neither correlated-input unpredictable nor correlated-input pseudorandom.

D sRKA-security of Boneh-Boyen Signature Scheme

We now discuss one implications of our main results on designing primitives secure against related key attacks. Using our techniques, it can be proven that a slight modification of Boneh-Boyen [BB04b] signature scheme gives a selectively secure RKA-secure signature scheme where the correlations are polynomials over a finite field $\mathbb{Z}_p$. We require that the polynomials be distinct even "ignoring the constant term" (i.e., the difference between any two polynomials should not just be in the constant term). For the modified scheme, the signature on a message $m \in \mathbb{Z}_p$ is defined as $g^{\frac{1}{x+m+c}}$ for the public parameters g and c and for the secret signing key x (instead of $g^{\frac{1}{x+m}}$ as in the Boneh-Boyen scheme). Further analysis follows.

The adversary gives to the challenger n polynomial - message pairs $\{(f_i(X), m_i)\}_i$, [10] where $\{f_i(X)\}_i$ are uniform-output polynomials over $\mathbb{Z}_p$ and distinct even ignoring the constant term (otherwise, it can be seen that the signatures on two different messages with two different keys may be identical). Now the challenger uniformly samples a random generator $g \xleftarrow{\$} \mathbb{G}$ and a random element $c \xleftarrow{\$} \mathbb{Z}_p$ for the public parameters and $x \xleftarrow{\$} \mathbb{Z}_p$ for the signing key.

To respond to the queries, the challenger computes $\{g^{\frac{1}{f_i(x)+m_i+c}}\}_{i\in[n]}$. Note that:

1. $\{g^{\frac{1}{f_i(x)+m_i+c}}\}_{i\in[n]}$ represent the n signatures on the messages $m_1, \ldots, m_n$ using correlated keys $\{f_i(x)\}_{i\in[n]}$ under the modified Boneh-Boyen signature scheme.
2. These values further represent the output of the hash function (specified by (g, c)) on n correlated inputs for the correlations specified by distinct polynomials $\{f_i(X) + m_i\}_{i\in[n]}$.

Observation (2) above implies that, by theorem 3, each of above n strings is unpredictable given the rest. This implies security (unforgeability) of the modified Boneh-Boyen signature scheme under such correlated key attacks.

A similar result can be shown for the Dodis and Yampolskiy [DY05] verifiable pseudo-random function.

[10] We require the messages to be signed to be fixed in advance as in the original signature scheme [BB04b].