Yvo G. Desmedt (Ed.)

LNCS 2567

Public Key Cryptography – PKC 2003

6th International Workshop
on Practice and Theory in Public Key Cryptography
Miami, FL, USA, January 2003
Proceedings

Springer

Yvo G. Desmedt (Ed.)

Public Key Cryptography – PKC 2003

6th International Workshop
on Practice and Theory in Public Key Cryptography
Miami, FL, USA, January 6-8, 2003
Proceedings

Springer

Volume Editor

Yvo G. Desmedt
Florida State University
Department of Computer Science
253 Love Building, Tallahassee, FL 32306-4530, USA
E-mail: desmedt@cs.fsu.edu

Cataloging-in-Publication Data applied for

Bibliographic information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the DeutscheNationalbibliografie; detailed bibliographic data is available in the Internet at <http://dnb.ddb.de>.

CR Subject Classification (1998): E.3, F.2.0, C.2.0, K.4.4, K.6.5

ISSN 0302-9743
ISBN 3-540-00324-X Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Typesetting: Camera-ready by author, data conversion by DA-TeX Gerd Blumenstein
Printed on acid-free paper SPIN 10871877 06/3142 5 4 3 2 1 0

Preface

PKC 2003 was the Sixth International Workshop on Practice and Theory in Public Key Cryptography and was sponsored by IACR, the International Association for Cryptologic Research (www.iacr.org). This year the workshop was organized in cooperation with the Department of Computer Science, Florida State University. The General Chair, Mike Burmester was responsible for local organization, registration, etc.

There were 105 submitted papers which were considered by the Program Committee. This is an increase of 52% compared to PKC 2002, which took place in Paris, France, February 2002, and which was incorrectly identified on the cover of the proceedings as being the fourth workshop. Due to the large number of submissions, some papers that contained new ideas had to be rejected. Priority was given to novel papers. Of the 105 submissions, 26 were selected for the proceedings. These contain the revised versions of the accepted papers. Each paper was sent to at least 3 members of the program committee for comments. Revisions were not checked for correctness of their scientific aspects and the authors bear full responsibility for the contents of their papers. Some authors will write final versions of their papers for publication in refereed journals.

I am very grateful to the members of the Program Committee for their hard work in the difficult task of selecting roughly 1 out of 4 of the submitted papers. Submissions to PKC 2003 were required to be anonymous. A Program Committee member could only present one accepted paper, or co-author at most two accepted papers without being allowed to present these. Papers submitted by members of the Program Committee were sent to at least 4 referees (and, of course, no Program Committee member reviewed his or her own paper).

The following external referees helped the Program Committee in reaching its decisions: Mehdi-Laurent Akkar, Joonsang Baek, Endre Bangerter, Régis Bevan, Daniel Bleichenbacher, Emmanuel Bresson, Eric Brier, Jan Camenisch, Matthew Campagna, Dario Catalano, Benoit Chevallier-Mames, Koji Chida, Nicolas Courtois, Annalisa De Bonis, Yevgeniy Dodis, Thomas Dübendorfer, Jacques Fournier, Atsushi Fujioka, Jun Furukawa, Clemente Galdi, Rosario Gennaro, Christophe Giraud, Louis Granboulan, Louis Goubin, Stuart Haber, Thomas Holenstein, Nick Howgrave-Graham, Stanislaw Jarecki, Antoine Joux, Jonathan Katz, Wataru Kishimoto, Erik Woodward Knudsen, Takeshi Koshiba, Hugo Krawczyk, Ben Lynn, Anna Lysyanskaya, Kazuto Matsuo, Patrick McDaniel, Phong Nguyen, Jesper Buus Nielsen, Satoshi Obana, Benny Pinkas, David Pointcheval, Bartosz Przydatek, Hervé Sibert, Francesco Sica, Nigel Smart, Markus Stadler, Martijn Stam, Reto Strobl, Koutarou Suzuki, Mike Szydlo, Tsuyoshi Takagi, Katsuyuki Takashima, Eran Tromer, Christophe Tymen, Salil Vadhan, Stefan Wolf, Jürg Wullschleger, and Akihiro Yamamura. (I apologize for any possible omission.) The Program Committee appreciates their efforts.

Thanks to Hoang Ha, Haizhi Chen, and Wayman E. Luy for secretarial work and for partially maintaining the WWW page of the conference, and to Wayne Sprague for setting up the e-mail addresses for PKC. Several people helped the General Chair with sending out the call for papers, registration, registration at the conference, etc.

Finally, I would like to thank everyone who submitted to PKC 2003, and IACR for its sponsorship.

October 2002 Yvo Desmedt

PKC 2003

Sixth International Workshop on Practice and Theory in Public Key Cryptography

Miami Convention Center, Miami, Florida, USA
January 6–8, 2003

Sponsored by the

International Association for Cryptologic Research
in cooperation with the
Department of Computer Science, Florida State University

General Chair

Mike Burmester, Florida State University, USA

Program Chair

Yvo Desmedt, Florida State University, USA

Program Committee

Masayuki Abe	NTT Laboratories, Japan
Feng Bao	Laboratories for Information Technology, Singapore
Giovanni Di Crescenzo	Telcordia, USA
Marc Joye	Gemplus, France
Kaoru Kurosawa	Ibaraki University, Japan
Arjen Lenstra	Citicorp, USA
Tal Malkin	AT&T Research, USA
Ueli Maurer	ETH, Zurich, Switzerland
Moni Naor	Weizmann Institute of Science, Israel
Tatsuaki Okamoto	NTT Laboratories, Japan
Jacques Patarin	Université de Versailles, France
Tal Rabin	IBM Research Lab., USA
Kazue Sako	NEC, Japan
Jacques Stern	École Normale Supérieure, France
Serge Vaudenay	ETH, Lausanne, Switzerland
Yongge Wang	University of North Carolina, USA
Michael Wiener	Canada
Moti Yung	Columbia University, USA
Yuliang Zheng	University of North Carolina, USA

Table of Contents

Diffie-Hellman Based Schemes

Threshold Cryptography

Reduction Proofs

Broadcast and Tracing

Digital Signatures

Specialized Multiparty Cryptography

Cryptanalysis I

Elliptic Curves: Implementation Attacks

Implementation and Hardware Issues

New Public Key Schemes

Elliptic Curves: General Issues

Cryptanalysis II

Efficient Construction of (Distributed) Verifiable Random Functions

Yevgeniy Dodis

Department of Computer Science
New York University, USA
dodis@cs.nyu.edu

Abstract. We give the first simple and efficient construction of *verifiable random functions* (VRFs). VRFs, introduced by Micali et al. [13], combine the properties of regular pseudorandom functions (PRFs) (i.e., indistinguishability from a random function) and digital signatures (i.e., one can provide an unforgeable proof that the VRF value is correctly computed). The efficiency of our VRF construction is only slightly worse than that of a regular PRF construction of Naor and Reingold [16]. In contrast to our direct construction, all previous VRF constructions [13, 12] involved an expensive generic transformation from verifiable unpredictable functions (VUFs).

We also provide the first construction of *distributed* VRFs. Our construction is more efficient than the only known construction of distributed (non-verifiable) PRFs [17], but has more applications than the latter. For example, it can be used to distributively implement the random oracle model in a *publicly verifiable* manner, which by itself has many applications.

Our construction is based on a new variant of decisional Diffie-Hellman (DDH) assumption on certain groups where the regular DDH assumption does *not* hold [10, 9]. Nevertheless, this variant of DDH seems to be plausible based on our *current* understanding of these groups. We hope that the demonstrated power of our assumption will serve as a motivation for its closer study.

1 Introduction

As a motivating example for our discussion, consider the problem of implementing the *random oracle model* [2]. Recall that in this model one assumes the existence of a publicly verifiable random function $\mathcal{O}$ (over some suitable domain and range). Each value $\mathcal{O}(x)$ is random and independent from the other values, and evaluating $\mathcal{O}$ on the same input twice yields the same (random) output. This model has found numerous applications in cryptography, which we do not even attempt to enumerate. It was shown by Canetti et al. [5], though, that no fixed public function can generically replace the random oracle, so more elaborate solutions are needed.

Y.G. Desmedt (Ed.): PKC 2003, LNCS 2567, pp. 1–17, 2003.

Pseudorandom Functions. As the first attempt, we may assume the existence of a trusted (but computationally bounded) party T. Since a function is an exponential sized object, T cannot store it explicitly. While maintaining a dynamically growing look-up table is a possibility, it is very inefficient as it requires large storage and growing complexity. A slightly better option is to use a *pseudorandom function* (PRF) $F_{SK}(\cdot)$ [8]. As indicated, this function is fully specified and efficiently computable given its short secret key (or *seed*) SK. However, without the knowledge of SK it looks *computationally* indistinguishable from exponential-sized $\mathcal{O}$.

In terms of constructing PRFs, there are several options. The most relevant to this paper, however, is the number-theoretic construction due to Naor and Reingold [16], which is based on the decisional Diffie-Hellman (DDH) assumption. This assumption in some group $\mathbb{G}$ of prime order q states that given elements g, g^a and g^b of (where g is the generator of $\mathbb{G}$), it is hard to distinguish the value g^{ab} from a truly random value g^c (where a, b, c are random in $\mathbb{Z}_q$). The PRF of [16] is a tree-based construction similar to the PRF construction of [8] from a pseudorandom generator. Namely, the secret key $SK = (g, a_1, \ldots a_\ell)$ consists of a random generator g of $\mathbb{G}$ and ℓ random exponents in $\mathbb{Z}_q$ (where ℓ is the length of the input to our PRF $F_{SK} : \{0,1\}^\ell \rightarrow \mathbb{G}$). Given $x = x_1 \ldots x_\ell \in \{0,1\}^\ell$, the PRF is defined by:

$$F_{g,a_1,\ldots,a_\ell}(x_1 \ldots x_\ell) \stackrel{\text{def}}{=} g^{\prod_{\{i|x_i=1\}} a_i \bmod q} \qquad (1)$$

Verifiable Random Functions. Coming back to our motivating application, replacing random oracle with a PRF has several problems. The first one is the question of verifiability and transferability. Even if everybody trusts T (which we will revisit soon), T has to be contacted not only when the value of F has to be computed for the first time, but even if one party needs to verify that another party has used the correct value of F. Thus, it would be much nicer if each value of $F_{SK}(x)$ would come with a proof $\pi_{SK}(x)$ of correctness, so that the recipient and everybody else can use this proof without the need to contact T again. As a side product, the ability to give such proof will also ensure that T himself cannot "cheat" by giving inconsistent values of F, or denying a correctly computed value of the function. This leads to the notion of *verifiable (pseudo)random functions*, or VRFs [13]. Intuitively, such functions remain (pseudo)random when restricted to all inputs whose function values were not previously revealed (and proved). Notice, the pseudorandomness and verifiability of a VRF immediately imply that a VRF by itself is an unforgeable signature scheme secure against chosen message attack.

Constructions of VRFs. Unfortunately, VRFs are not very well studied yet. Currently, we have two constructions of VRFs: based on RSA [13], and based on a separation between computational and decisional Diffie-Hellman problems in certain groups [12]. Both of these constructions roughly proceed as follows. First, they construct a relatively simple and efficient verifiable *unpredictable* function (VUF) based on the corresponding assumption. Roughly, a VUF is the same verifiable object as a VRF, except each "new" value $F_{SK}(x)$ is only unpredictable (i.e., hard to compute) rather than pseudorandom. From VUFs, a generic construction to VRFs is given, as introduced by [13]. Unfortunately, this construction is very inefficient and also looses a very large factor in its exact security. Essentially, first one uses the Goldreich-Levin theorem [7] to construct a VRF with very small (slightly super-logarithmic) input size and output size 1 (and pretty dramatic security loss too!).[1] Then one makes enough such computations to amplify the output size to roughly match that of the input. Then one follows another rather inefficient tree-based construction on the resulting VRF to get a VRF with arbitrary input size and small output size. Finally, one evaluates the resulting convoluted VRF several times to increase the output size to the desired level. In some sense, the inefficiency of the above construction is expected given its generality and the fact that it has to convert pure unpredictability into a much stronger property of pseudorandomness. Still, this means that the resulting VRF constructions are very bulky and inelegant. In this work we present the first simple, efficient and "direct" VRF construction.

DISTRIBUTED PRFs. Returning to our target application of implementing the random oracle, the biggest problem of both PRF/VRF-based solutions is the necessity to fully trust the honest party T holding the secret key for F. Of course, VRFs slightly reduced this trust level, but T still singlehandedly knows all the values of F. Clearly, this approach (1) puts to much trust into T, (2) makes T is bottleneck of all the computations; (3) makes T is "single point of failure": compromising T will break the security of any application which depends on the random oracle assumption.

The natural solution to this problem is to distribute the role of T among n servers. This leads to the notion of *distributed* PRF*s* (DPRFs) and *distributed* VRF*s* (DVRFs). Since the latter concept was not studied prior to our work, we start with DPRFs, thus ignoring the issue of verifiability for now. Intuitively, DPRFs with threshold $1 \leq t < n$ allow any $(t+1)$ out of n servers to jointly compute the function using their shares, while no coalition of up to t servers to be in a better situation that any outside party. Namely, the function remains pseudorandom to any such coalition.

DPRFs first originate in the work of Micali and Sidney [14]. However, their construction (later improved by [15]) can tolerate only a moderate number of servers or a small threshold, since its complexity is proportional to n^t. The next influential work is that of Naor et al. [15], who give several efficient constructions of certain weak variants of DPRFs. Ironically, one of the constructions (namely, that of distributed *weak* PRF) can be turned into an efficient DPRF by utilizing random oracles. Even though this is non-trivial (since nobody should compute the value of a DPRF without the cooperation of $t+1$ servers), we would certainly prefer a solution in the plain model, since elimination of the random oracle was one of the main motivation for DPRFs!

[1] The latter is the reason for such a small input size. One can make a very strong exponential assumption to increase the input size, like was done in [12], but the construction still loses a lot in security, and still goes through an intermediate VUF.

The first regular DPRF was recently constructed by Nielsen [17] by distributing a slightly modified variant of the Naor-Reingold PRF [16], given in Equation (1) (in the final version of their work, [16] also give essentially the same construction). Unfortunately, the resulting DPRF in highly *interactive* among the servers (while ideally the servers would only talk to the user requesting the function value) and requires a lot of rounds (proportional to the length of the input). In particular, the question of non-interactive DPRF construction remained open prior to this work.

DISTRIBUTED VRFS. Even though DVRFs were not explicitly studied prior to this work, they seem to provide the most satisfactory solution to our original problem of implementing the random oracle. Indeed, distributing the secret key ensures that no coalition of up to t servers can compromise the security (i.e., pseudorandomness) of the resulting random oracle. On the other hand, verifiability ensures that one does not need to contact the servers again after the random oracle was computed once: the proof can convince any other party of the correctness of the VRF value. For example, DVRFs by themselves provide an ordinary threshold signature scheme, which can be verified without further involvement of the servers. And, of course, using DVRFs are likely to enhance the security, robustness or functionality of many applications originally designed for plain PRFs, VRFs and DPRFs.

OUR CONTRIBUTIONS. We give the first simple and direct construction of VRFs, based on a new "DDH-like" assumption which seems to be plausible on certain recently proposed elliptic and hyper-elliptic groups (e.g., [10]). We call this assumption *sum-free decisional Diffie-Hellman* (sf-DDH) assumption. While we will discuss this assumption later, we mention that in the proposed groups the regular regular DDH assumption is *false* (in fact, this is what gives us verifiability!), and yet the sf-DDH or some similar assumption seems plausible. Our construction is similar to the Naor-Reingold (NR) construction given by Equation (1), except we utilize some carefully chosen encoding C before applying the NR-construction. Specifically, if $C : \{0,1\}^\ell \rightarrow \{0,1\}^L$ is some injective encoding, we consider the function of the form

$$F_{g,a_1,\ldots,a_L}(x_1 \ldots x_\ell) \stackrel{\text{def}}{=} g^{\prod_{\{i|C(x)_i=1\}} a_i \bmod q} \tag{2}$$

Identifying the properties of the encoding C and constructing C satisfying these properties will be one of the main technical challenges we will have to face. At the end we will achieve $L = O(\ell)$ (specifically, $L = 2\ell$ to get a regular PRF, and $L = 3\ell+2$ to get a VRF), making our efficiency very close to the NR-construction.

Our second main contribution is the first construction of a distributed VRF(DVRF). Namely, we show that our VRF construction can be made distributed and *non-interactive* (although multi-round). This is the first non-interactive construction of a distributed PRF (let alone VRF), since the only previous DPRF construction of [17, 16] is highly interactive among the servers. In fact, our DVRF construction is more efficient than the above mention DPRF construction, despite achieving the extra verifiability. We already mentioned the big saving in communication complexity (roughly, from $n^2\ell k$ to $n\ell k$, where k is the security parameter). Another important advantage, though, is that we dispense with the need to perform somewhat expensive (concurrently composable) zero knowledge proofs for the equality of discrete logs. This is because in our groups the DDH problem is easy, so it can be locally checked by each party without the need for the proof. In particular, even though we need to apply the encoding C to the message, while the construction of [17, 16] does not, the lack of ZK-proofs makes our round complexity again slightly better. Finally, we remark that the same distributed construction can be applied to distribute the VUF of Lysyanskaya [12] (which results in a threshold "unique signature" scheme under a different assumption than the one we propose).

2 Definitions

2.1 Verifiable Random Functions and Friends

Definition 1. *A function family* $F_{(\cdot)}(\cdot) : \{0,1\}^{\ell(k)} \to \{0,1\}^{m(k)}$ *is a family of* VRF*s, if there exists a probabilistic polynomial time algorithm* Gen *and deterministic algorithms* Prove *and* Verify *such that:* $\mathsf{Gen}(1^k)$ *outputs a pair of keys* (PK, SK)*;* $\mathsf{Prove}_{SK}(x)$ *outputs a pair* $\langle F_{SK}(x), \pi_{SK}(x)\rangle$*, where* $\pi_{SK}(x)$ *is the proof of correctness; and* $\mathsf{Verify}_{PK}(x, y, \pi)$ *verifies that* $y = F_{SK}(x)$ *using the proof* π*. We require:*

1. *Uniqueness: no values* $(PK, x, y_1, y_2, \pi_1, \pi_2)$ *can satisfy* $\mathsf{Verify}_{PK}(x, y_1, \pi_1) = \mathsf{Verify}_{PK}(x, y_2, \pi_2)$ *when* $y_1 \neq y_2$*.*
2. *Provability: if* $(y, \pi) = \mathsf{Prove}_{SK}(x)$*, then* $\mathsf{Verify}_{PK}(x, y, \pi) = 1$*.*
3. *Pseudorandomness: for any* PPT $A = (A_1, A_1)$ *who did not call its oracle on* x *(see below), the following probability is at most* $\frac{1}{2} + \mathsf{negl}(k)$ *(here and everywhere,* $\mathsf{negl}()$ *stands for some negligible function in the security parameter* k*):*

$$\Pr\left[\, b = b' \;\middle|\; \begin{array}{c} (PK, SK) \leftarrow \mathsf{Gen}(1^k);\ (x, st) \leftarrow A_1^{\mathsf{Prove}(\cdot)}(PK);\ y_0 = F_{SK}(x); \\ y_1 \leftarrow \{0,1\}^{m(k)};\ b \leftarrow \{0,1\};\ b' \leftarrow A_2^{\mathsf{Prove}(\cdot)}(y_b, st) \end{array} \right]$$

Intuitively, the definition states that no "new" value of the function can be distinguished from a random string, even after seeing any other function values together with the corresponding proofs. Regular PRFs form the non-verifiable analogs of VRFs. Namely, $PK = \emptyset$, $\pi_{SK}(\cdot) = \emptyset$, there is no algortihm Verify, no uniqueness and provability properties, and pseudorandomness is the only remaining property. We notice that the resulting definition is not the typical definition for PRFs [8]: namely, that no adversary can tell having oracle access to a truly random function from having oracle access to a pseudorandom function. However, it is easy to see that our definition is equivalent to that usual one, so will we use it as the more convenient in the context of VRFs.

2.2 Diffie-Hellman Assumptions

Assume $\mathsf{Setup}(1^k)$ outputs the description of some cyclic group $\mathbb{G}$ of prime order q together with its random generator g. Let $L = L(k)$ be some integer and $a_1 \dots a_L$ be random elements of $\mathbb{Z}_q$. Let $[L]$ denote $\{1 \dots L\}$, and given a subset $I \subseteq [L]$, we denote $a_I = \prod_{i \in I} a_i \bmod q$ (where $a_\emptyset = 1$), $G(I) = G_I = g^{a_I}$. Finally, we will often view an element $z \in \{0,1\}^L$ as either a subset $\{i \mid z_i = 1\}$, or an L-dimensional vector over $GF(2)$ (and vice versa).

Generalized Diffie-Hellman Assumptions. The security of ours, as well as the previous related constructions [16, 12], will rely on various assumptions of the following common flavor. The adversary A has oracle access to $G(\cdot)$, and tries to "obtain information" about some value $G(J)$. The meaning of obtaining information depends on whether we are making a computational or a decisional assumption. In the former case, A has to compute $G(J)$, while in the latter case A has to distinguish $G(J)$ from a random element of $\mathbb{G}$. While the decisional assumption is stronger, it has a potential of yielding a (verifiable) *pseudorandom* function, while the computational assumption can yield at best[2] a (verifiable) *unpredictable* function.

In either case, we require that it should be hard to any polynomial time adversary to succeed. Of course, one has to make some non-trivial restrictions on when the adversary is considered suceessful. Formally, given that the adversary called its oracle on subsets $I_1, \dots, I_t$ and "obtained information" about $G(J)$, we can define a predicate $\mathcal{R}(J, I_1, \dots I_t)$ which indicates whether the adversary's actions are "legal". For example, at the very least the predicate should be false if $J \in \{I_1 \dots I_t\}$. We call any such predicate *non-trivial.* We will certainly restrict ourselves to non-trivial predicates, but may sometimes place some more restrictions on $\mathcal{R}$ in order to make a more plausible and weaker assumption (see below).

Definition 2. *Given $L = L(k)$, we say that the group $\mathbb{G}$ satisfies the* generalized decisional Diffie-Hellman *(gDDH) assumption of order* L relative to a non-trivial predicate $\mathcal{R}$, *if for any* PPT *adversary $A = (A_1, A_1)$ who called its oracle on subsets $I_1 \dots I_t$ satisfying $\mathcal{R}(J, I_1, \dots, I_t) = 1$, the probability below is at most $\frac{1}{2} + \mathsf{negl}(k)$:*

$$\Pr\left[\, b = b' \;\middle|\; \begin{array}{l} (\mathbb{G}, q, g) \leftarrow \mathsf{Setup}(1^k);\ (a_1 \dots a_L) \leftarrow \mathbb{Z}_q,\ (J, st) \leftarrow A_1^{G(\cdot)}(\mathbb{G}, q); \\ \qquad y_0 = G(J);\ y_1 \leftarrow \mathbb{G};\ b \leftarrow \{0,1\};\ b' \leftarrow A_2^{G(\cdot)}(y_b, st) \end{array} \right]$$

Very similarly one can define the *generalized computational Diffie-Hellman* (gCDH) assumption of order L *relative to* $\mathcal{R}$, where the job of A is to *compute* $G(J)$. We notice that the more restrictions $\mathcal{R}$ places on the I_i's and the "target" set J, the harder it is for the adversary to succeed, so the assumption becomes weaker (and more preferable). Thus, the strongest possible assumption of the above type is to put no further restrictions on $\mathcal{R}$ other than non-triviality (i.e., $J \notin \{I_1, \dots I_t\}$). We call the two resulting assumptions simply gDDH and gCDH (without specifying $\mathcal{R}$). A slightly weaker assumption results when we

[2] Unless a generic inefficient conversion is used, or one assumes random oracles.

require that the target set is equal to the full set $J = [L]$, i.e. the adversary has to obtain information about $g^{a_1 \cdots a_L}$. We call the resulting assumptions *full target* gDDH/gCDH (where $L = 2$ yields regular DDH/CDH). Finally, making L larger generally makes the assumption stronger, since the adversary can always choose to concentrate on some subset of L. Thus, it is preferable to base the security of some contsruction on as small L and as restrictive $\mathcal{R}$ as possible.

Before moving to our new sum-free gDDH assumption, let us briefly state some simple facts about gDDH/gCDH. It was already observed by [19] that gDDH assumption of any polynomial order $L(k)$ (with or without full target) follows from the regular DDH assumption (which corresponds to $L = 2$). Unfortunately, we do not know of the same result for the gCDH problem. The best analog of this result was implicitly obtained by [12], who more or less showed that the regular gCDH assumption of logarithmic order $O(\log k)$ (even with full target) implies the gCDH assumption of any polynomial order $L(k)$, *provided* in the latter we restrict the adversary to operate on the codewords of any good error-correcting code.

SUM-FREE gDDH. We already saw that the regular DDH assumption is a very strong security assumption in that it implies the gDDH assumption. This useful fact almost immediately implies, for example, that the Naor-Reingold construction in Equation (1) is a PRF under DDH, illustrating the power of DDH for proving pseudorandomness. Unfortunately, groups were DDH is true are not convenient for making *verifiable* random functions, since nobody can verify the equality of discrete logs. On the other hand, we will see shortly that it is very easy to obtain verifiability in groups where DDH is solvable in polynomial time (such as the group suggested by [10]). Unfortunately, such groups certainly do not satisfy the gDDH assumption too, which seems to imply that we have to settle for the computational assumption (like gCDH) in such groups, which in turn implies that we settle only for the VUF construction rather than the desired VRF. Indeed, obtaining such a VUF is exactly what was recently done by Lysyanskaya [12] in groups where DDH is easy but gCDH is hard.

However, we make the crucial observation that the easiness of regular DDH does *not* mean that no version of gDDH assumption can be true: it only means *we might have to put more restrictions on the predicate* $\mathcal{R}$ in order to make it hard for the adversary to break the gDDH assumption relative to $\mathcal{R}$. Indeed, for the current elliptic groups for which we believe in a separation between DDH and CDH, we only know how to test if (h, u, v, w) is of the form $u = h^a, v = h^b, w = h^{ab}$ (this is called a DDH-tuple). This is done by means of a certain bilinear mapping (details are not important), for which we do not know a multi-linear variant. In fact, Boneh and Silverberg [4] observe that a multi-linear variant of such mapping seems unlikely to exist in the currently proposed groups, and pose as a major open problem to exhibit groups where such mappings exist. This suggests that many natural, but more restrictive flavors of DDH seem to hold in the currently proposed groups (where regular DDH is easy). For example, as was mentioned by Boneh and Franklin [3], it seems reasonable to assume that it is hard to distinguidh a tuple $(h, h^a, h^b, h^c, h^{abc})$ from a random tuple

(h, h^a, h^b, h^c, h^d). Put differently, when $a_1 \ldots a_L$ are chosen at random and given a sample $g = G(\emptyset), G(I_1) \ldots G(I_t)$, the only way we know how to distinguish $G(J)$ from a random element of such groups is by exhibiting three sets I_m, I_p, I_s (where $0 \leq m, p, s \leq t$, and I_0 denotes the empty set) such that $a_J \cdot a_{I_m} \equiv a_{I_p} \cdot a_{I_s} \bmod q$.[3] The last equation implies that "$J + I_m = I_p + I_s$", where we view the sets as L-bit 0/1-vectors, and the addition is bitwise over the integers. In other words, one has to explicitly find a DDH-tuple among the samples $G(I_i)$'s and the target $G(J)$.

We formalize this intuition into the following predicate $\mathcal{R}(J, I_1, \ldots, I_t)$. Let us denote $I_0 = \emptyset$. We say that J is DDH-*dependent* on $I_1 \ldots I_t$ if there are indices $0 \leq m, p, s \leq t$ satisfying $J + I_m = I_p + I_s$ (see explanation above). For example, 10101 is DDH-dependent on 01010, 00001 and 11111, since $10101 + 01011 = 11111 + 00001 = 11112$. Then we define the DDH-*free* relation $\mathcal{R}$ to be true if and only if J is DDH-independent from $I_1 \ldots I_t$.

Definition 3. *Given $L = L(k)$, we say that the group $\mathbb{G}$ (where regular* DDH *is easy) satisfies the* sum-free decisional Diffie-Hellman *(*sf-DDH*) assumption of order L if it satisfies the* gDDH *assumption of order L relative to the* DDH*-free relation $\mathcal{R}$ above.*

For our purposes we notice that DDH-dependence also implies that $J \oplus I_m = I_p \oplus I_s$, where $\oplus$ indicates the bitwise addition moduo 2 (i.e., we make "$2 = 0$"), or $J \oplus I_m \oplus I_p \oplus I_s = 0$. Let us call J 4-*wise independent* from $I_1 \ldots I_t$ if no three sets I_m, I_p, I_s yield $J \oplus I_m \oplus I_p \oplus I_s = 0$. Hence, if we let $\mathcal{R}'(J, I_1, \ldots, I_t) = 1$ if and only if J is 4-wise independent from the I_i's, we get that $\mathcal{R}'$ is a stricter relation than our DDH-free $\mathcal{R}$. But this means that gDDH assumption relative to $\mathcal{R}'$ is a *weaker* assumption than sf-DDH, so we call it *weak* sf-DDH. Our actual construction will in fact be based on weak sf-DDH.

To summarize, sf-DDH is the strongest assumption possible in groups were regular DDH is false. We chose this assumption to get the simplest and most efficient VRF construction possible when DDH is false. However, even if the ambitious sf-DDH assumption we propose turns out to be false in the current groups where DDH is easy — which we currently have no indication of — it seems plausible that some reasonable weaker gDDH assumptions (relative to more restrictive $\mathcal{R}$) might still hold. And our approach seems to be general enough to allow some easy modification to our construction (at slight efficiency loss) meet many such weaker gDDH assumptions.

3 Constructions

Assume $\mathbb{G}$ is the group where DDH is easy while some version of sf-DDH holds. We consider the natural the type of functions given by Equation (2); in our new

[3] One can also try to find the additive relations, but since the a_i's are all random, it seems that the only such relations one can find would trivially follow from some multiplicative relations.