A preliminary version of this paper appears in the proceedings of PKC 2018. This is the full version.

Public-Key Encryption Resistant to Parameter Subversion and its Realization from Efficiently-Embeddable Groups

Benedikt Auerbach* Mihir Bellare† Eike Kiltz‡

January 6, 2018

Abstract

We initiate the study of public-key encryption (PKE) schemes and key-encapsulation mechanisms (KEMs) that retain security even when public parameters (primes, curves) they use may be untrusted and subverted. We define a strong security goal that we call ciphertext pseudo-randomness under parameter subversion attack (CPR-PSA). We also define indistinguishability (of ciphertexts for PKE, and of encapsulated keys from random ones for KEMs) and public-key hiding (also called anonymity) under parameter subversion attack, and show they are implied by CPR-PSA, for both PKE and KEMs. We show that hybrid encryption continues to work in the parameter subversion setting to reduce the design of CPR-PSA PKE to CPR-PSA KEMs and an appropriate form of symmetric encryption. To obtain efficient, elliptic-curve-based KEMs achieving CPR-PSA, we introduce efficiently-embeddable group families and give several constructions from elliptic-curves.

*Horst-Görtz Institute for IT Security and Faculty of Mathematics, Ruhr-University Bochum, Germany. Email: benedikt.auerbach@rub.de. Supported by NRW Research Training Group SecHuman.

†Department of Computer Science & Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: mihir@eng.ucsd.edu. URL: http://cseweb.ucsd.edu/~mihir/. Supported in part by NSF grants CNS-1526801 and CNS-1717640, ERC Project ERCC FP7/615074 and a gift from Microsoft corporation.

‡Horst-Görtz Institute for IT Security and Faculty of Mathematics, Ruhr-University Bochum, Germany. Email: eike.kiltz@rub.de. Supported in part by ERC Project ERCC FP7/615074 and by DFG SPP 1736 Big Data.

Contents

1 Introduction

This paper initiates a study of public-key encryption (PKE) schemes, and key-encapsulation mechanisms (KEMs), resistant to subversion of public parameters. We give definitions, and efficient, elliptic-curve-based schemes. As a tool of independent interest, we define efficiently-embeddable group families and construct them from elliptic curves.

Parameter subversion. Many cryptographic schemes rely on some trusted, public parameters common to all users and implementations. Sometimes these are specified in standards. The Oakley primes [Orm98], for example, are a small number of fixed prime numbers widely used for discrete-log-based systems. For ECC (Elliptic Curve Cryptography), the parameters are particular curves. Examples include the P-192, P-224, ... curves from the FIPS-186-4 [NIS13] standard and Ed25519 [BDL+11].

There are many advantages to such broad use of public parameters. For example, it saves implementations from picking their own parameters, a task that can be error-prone and difficult to do securely. It also makes key-generation faster and allows concrete-security improvements in the multi-user setting [BBM00]. Recent events indicate, however, that public parameters also bring a risk, namely that they can be *subverted*. The representative example is Dual-EC. We refer to [BLN15] for a comprehensive telling of the story. Briefly, Dual EC was a PRG whose parameters consisted of a description of a cyclic group and two generators of the group. If the discrete logarithm of one generator to base the other were known, security would be compromised. The Snowden revelations indicate that NIST had adopted parameters provided by the NSA and many now believe these parameters had been subverted, allowing the NSA to compromise the security of Dual EC. Juniper's use of Dual EC further underscores the dangers [CCG+16].

Security in the face of parameter subversion. DGGJR [DGG+15] and BFS [BFS16] initiated the study of cryptography that retains security in the face of subverted parameters, the former treating PRGs and the latter treating NIZKs, where the parameter is the common reference string. In this paper we treat encryption. We define what it means for parameter-using PKE schemes and KEMs to retain security in the face of subversion of their parameters. With regard to schemes, ECC relies heavily on trusted parameters. Accordingly we focus here, providing various efficient elliptic-curve-based schemes that retain security in the face of parameter subversion.

Current mitigations. In practice, parameters are sometimes specified in a verifiable way, for example derived deterministically (via a public algorithm) from publicly-verifiable coins. The coins could be obtained by applying a hash function like SHA1 to some specified constants (as is in fact done for the FIPS-186-4 curves [NIS13] and in the ECC brainpool project), via the first digits of the irrational number π, or via lottery outcomes [BDF+15]. This appears to reduce the possibility of subversion, but BCCHLN [BCC+14] indicate that the potential of subverting elliptic curves still remains, so there is cause for caution even in this regard. Also, even if such mechanisms might "work" in some sense, we need definitions to understand what "work" means, and proofs to ensure definitions are met. Our work gives such definitions.

Background. A PKE scheme specifies a parameter generation algorithm that returns parameters π, a key-generation algorithm that takes π and returns a public key pk and matching secret key sk, an encryption algorithm that given π, pk and message m returns a ciphertext c, and a decryption algorithm that given π, sk, c recovers m. We denote the classical notions of security by IND —indistinguishability of ciphertexts under chosen-ciphertext attack [BDPR98, CS98]— and PKH —public-key hiding, also called anonymity, this asks that ciphertexts not reveal the public key under which they were created [BBDP01]. For KEMs, parameter and key generation are the same, encryption is replaced by encapsulation —it takes π, pk to return an encapsulated

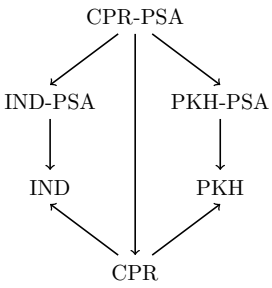


Figure 1: **Relations between notions of security.** The notions are defined, and the relations hold, for both PKE schemes and KEMs. An arrow A $\rightarrow$ B is an implication: if a scheme meets A then it also meets B.

---

key K and a ciphertext c that encapsulates K— and decryption is replaced by decapsulation —given π, sk, c it recovers K. We continue to denote the classical goals by IND —this now asks for indistinguishability of encapsulated keys from random under chosen-ciphertext attack [CS03]— and PKH. We stress that these classical notions assume *honest parameter generation*, meaning *the parameters are trusted*.

We know that, in this setting, IND PKE is reduced, via hybrid encryption, to IND KEMs and ind-cpa symmetric encryption [CS03]. To the best of our knowledge, no analogous result exists for PKH.

Mass surveillance activities have made apparent the extent to which privacy can be violated purely by access to meta-data, including who is communicating with whom. PKE and KEMs providing PKH are tools towards systems that do more to hide identities of communicants. We will thus target this goal in the parameter subversion setting as well.

Definitions and relations. For both PKE and KEMs, we formulate a goal called ciphertext pseudorandomness under parameter subversion attack, denoted CPR-PSA. It asks that ciphertexts be indistinguishable from strings drawn randomly from the ciphertext space, even under a chosen-ciphertext attack (CCA). We also extend the above-discussed classical goals to the parameter subversion setting, defining IND-PSA and PKH-PSA. For both PKE (Proposition 3.1) and KEMs (Proposition 3.2) we show that CPR-PSA implies both IND-PSA and PKH-PSA. We thus get the relations between the new and classical notions summarized in Figure 1. (Here CPR is obtained by dropping the PSA in CPR-PSA, meaning it is our definition with honest parameter generation. This extends the notions of [Möl04, DGG$^+$15] to allow a CCA.)

We ask whether we can reduce the design of CPR-PSA PKE to the design of CPR-PSA KEMs via hybrid encryption. Proposition 3.3 says the answer is yes, but, interestingly, requires that the KEM has an extra property of well-distributed ciphertexts that we denote WDC-PSA. (The symmetric encryption scheme is required to have pseudo-random ciphertexts. Such symmetric schemes are easily obtained.) We now have a single, strong target for constructions, namely CPR-PSA+WDC-PSA KEMs. (By the above they imply CPR-PSA PKE, which in turn implies IND-PSA PKE and PKH-PSA PKE.) Our goal thus becomes to build efficient KEMs that are CPR-PSA+WDC-PSA.

Parameter-free schemes. We say that a scheme (PKE or KEM) is parameter free if there are no parameters. (Formally, the parameters are the empty string ε.) Note that a parameter-free scheme that is XXX secure is trivially also XXX-PSA secure. (XXX $\in \{\text{CPR}, \text{IND}, \text{PKH}\}$.) This is an important observation, and some of our schemes will indeed be parameter-free, but, as we discuss next, this observation does not trivialize the problem.

Issues and challenges. In an attempt to achieve PSA security through the above observation, we could consider the following simple way to eliminate parameters. Given a XXX-secure parameter-using scheme, build a parameter-free version of it as follows: the new scheme sets its parameters to the empty string; key generation runs the old parameter generation algorithm to get π, then the old key generation algorithm to get pk and sk, setting the new public and secret keys to (π, pk) and (π, sk), respectively; encryption and decryption can then follow the old scheme. This trivial construction, however, has drawbacks along two dimensions that we expand on below: (1) security and (2) efficiency.

With regard to security, the question is, if the old scheme is XXX, is the new one too? (If so, it is also XXX-PSA, since it is parameter free, so we only need to consider the classical notions.) The answer to the question is yes if XXX = IND, but *no if* XXX $\in \{\text{PKH}, \text{CPR}\}$. Imagine, as typical, that the parameters describe a group. Then in the new scheme, different users use different, independent groups. This will typically allow violation of PKH [BBDP01]. For example, in the El Gamal KEM, a ciphertext is a group element, so if two users have groups $\mathbb{G}_0$ and $\mathbb{G}_1$, respectively, one can determine which user generated a ciphertext by seeing to which of the two groups it belongs. The same is true for RSA where the group $\mathbb{G}_i = \mathbb{Z}_{N_i}$ is determined by the modulus N_i in the key of user i. Even when the moduli have the same bit length, attacks in [BBDP01] show how to violate PKH-security of the simple RSA KEM.

With regard to efficiency, the drawback is that we lose the benefits of parameter-using schemes noted above. In particular, key-generation is less efficient (because it involves parameter generation for the old scheme, which can be costly), and public keys are longer (because they contain the parameters of the old scheme). We aim to retain, as much as possible, the efficiency benefits of parameters while adding resistance to PSA.

BBDP [BBDP01] give (1) parameter-free IND+PKH RSA-based PKE schemes and (2) parameter-using discrete-log based IND+PKH PKE schemes. The former, since parameter-free, are IND-PSA+PKH-PSA, but they are not CPR-PSA and they are not as efficient as ECC-based schemes. The latter, while ECC-based and fast, are not secure against PSA.

The open question that emerges is thus to design efficient, ECC-based KEMs that are CPR-PSA+WDC-PSA. The technical challenge is to achieve CPR-PSA (and thus PKH-PSA) even though the groups of different users may be different.

Overview of the approach. We introduce and formalize *efficiently-embeddable group (eeg) families* and identify desirable security properties for them. We give two transforms constructing CPR-PSA+WDC-PSA KEMs from secure eeg families. This reduces our task to finding secure eeg families. We propose several instantiations of eeg families from elliptic curves with security based on different assumptions. An overview of the resulting KEMs is given in Table 1. We discuss our results in greater detail below.

Efficiently-embeddable group families. As described above, having users utilize different groups typically enables linking ciphertexts to the intended receiver and hence violating CPR-PSA. However, certain families of groups allow to efficiently map group elements to a space, which is independent of the particular group of the family. Building on these types of group families it is possible to achieve CPR-PSA secure encryption while still allowing each user to choose his own group.

We formalize the required properties via *efficiently embeddable group families*, a novel abstraction that we believe is of independent interest. An eeg family EG specifies a parameter generation algorithm $\mathsf{EG.P}$ sampling parameters to be used by the other algorithms, and a group generation algorithm $\mathsf{EG.G}$ sampling a group from the family. Embedding algorithm $\mathsf{EG.E}$ embeds elements of the group into some embedding space $\mathsf{EG.ES}$. The group element can be recovered using inversion algorithm $\mathsf{EG.I}$. An important property is that the embedding space

eeg family	Transform	Parameter	Assumption	Efficiency			Key size
				KE.G	KE.E	KE.D	
$\mathsf{EG}_{\text{twist}}$	**eegToKE1**	p	sCDH-PSA	t_{TGen}	2,2	2	$10k$
$\mathsf{EG}_{\text{twist}}$	**eegToKE2**	p	CDH-PSA	t_{TGen}	3,3	3	$12k$
$\mathsf{EG}^{\ell}_{\text{twist-rs}}$	**eegToKE1**	—	sCDH-PSA	t_{TGen}	3, ℓ+1	1	$9k$
$\mathsf{EG}^{\ell}_{\text{twist-rs}}$	**eegToKE2**	—	CDH-PSA	t_{TGen}	4, ℓ+2	2	$11k$
$\mathsf{EG}_{\text{twist-re}}$	**eegToKE1**	—	sCDH-PSA	t_{TGen}	3, 3	1	$9k$
$\mathsf{EG}_{\text{twist-re}}$	**eegToKE2**	—	CDH-PSA	t_{TGen}	4, 4	2	$11k$
$\mathsf{EG}^{\ell}_{\text{ell1}}$, $\mathsf{EG}^{\ell}_{\text{ell2}}$	**eegToKE1**	p	sCDH-PSA	t_{EllGen}	$3, \ell+1$	1	$6k$
$\mathsf{EG}^{\ell}_{\text{ell1-rs}}$, $\mathsf{EG}^{\ell}_{\text{ell2-rs}}$	**eegToKE1**	—	sCDH-PSA	t_{EllGen}	5, ℓ+1	1	$5k$

Table 1: **Our elliptic curve based CPR-PSA+WDC-PSA KEMs.** The modulus of the used field is denoted by p. Efficiency of KE.G is dominated by the sampling time of the curve-twist pair or the Elligator curve respectively. Efficiency of KE.E (average, worst case) and KE.D (worst case) is given as the number of exponentiations on the curves. The key size is measured in bits, $k = \lceil |\mathbb{F}_p| \rceil$ being the bit length of the used modulus. For the rejection sampling based constructions, ℓ denotes the cut-off bound.

only depends on the parameters and in particular not on the used group. Looking ahead, the KEM's public key will contain a group sampled with EG.G and ciphertexts will be embeddings. We require two security properties for EG in order to achieve CPR-PSA+WDC-PSA KEMs. Both assume parameter subversion attacks and are defined with respect to a sampling algorithm EG.S, which samples (not necessarily uniformly distributed) group elements. The first, embedding pseudorandomness (EPR-PSA), is that embeddings of group elements sampled with EG.S are indistinguishable from uniform. Further we give definitions of the computational Diffie-Hellman assumption and the strong computational Diffie-Hellman assumption with respect to EG, the latter being an adaption of the interactive assumption introduced in [ABR01] to our setting. The definitions differ from the usual (strong) computational Diffie-Hellman assumption in two points. The group used for the challenge is sampled using EG.G on a parameter of the adversary's choice and additionally one of the exponents used in the challenge is sampled with sampling algorithm EG.S. We denote the assumptions by CDH-PSA and sCDH-PSA.

Key ecapsulation mechanisms from eeg families. We give two transforms of eeg families to CPR-PSA+WDC-PSA-secure KEMs. The difference is in the computational assumptions made on the eeg family in order to achieve CPR-PSA. The first transform **eegToKE1** is applicable to any eeg family EG. If EG is both EPR-PSA and sCDH-PSA the resulting KEM is CPR-PSA. The second transform, **eegToKE2**, is only applicable to eeg families consisting of groups, which order has no small prime factors. Its security is based on the computational Diffie-Hellman assumption instead of the strong computational Diffie-Hellman assumption, i.e. it achieves CPR-PSA KEMs under the weaker assumption that EG is both EPR-PSA and CDH-PSA. However, this comes at the cost of larger key size and slower encryption and decryption. Both transforms yield WDC-PSA KEMs if the used eeg family is EPR-PSA.

Instantiations from elliptic curves. We propose several instantiations of eeg families from elliptic curves. It is well known that elliptic curves are not all equal in security. We target elliptic-curve groups over the field $\mathbb{F}_p$ for a large odd prime p since they are less vulnerable to discrete-log-finding attacks than groups over fields of characteristic two [Fre98, PQ12]. While the usage of standardized primes allows for more efficient implementations, several cryptanalysts

eeg family	Curve type	Parameter	$\Delta_{\text{EPR-PSA}}$	See
$\mathsf{EG}_{\text{twist}}$	twist	p	0	§ 5.2
$\mathsf{EG}^{\ell}_{\text{twist-rs}}$	twist	—	$(1/2)^{\ell}$	§ 5.3
$\mathsf{EG}_{\text{twist-re}}$	twist	—	0	§ 5.4
$\mathsf{EG}^{\ell}_{\text{ell1}}$, $\mathsf{EG}^{\ell}_{\text{ell2}}$	Elligator	p	$(2/3)^{\ell}$	§ 6.2
$\mathsf{EG}^{\ell}_{\text{ell1-rs}}$, $\mathsf{EG}^{\ell}_{\text{ell2-rs}}$	Elligator	—	$(4/5)^{\ell}$	§ 6.3

Table 2: **Security of our eeg families.** The modulus of the used field is denoted by p. $\Delta_{\text{EPR-PSA}}$ denotes the maximal advantage of an (unbounded) adversary in breaking EPR-PSA. ℓ denotes the cut-off bound used in the construction based on rejection sampling.

---

further suggest that p should be as random as possible for maximal security, see for example Brainpool's RFC on ECC [LM10]. These constraints make building eeg families more challenging. We offer solutions for both cases. Our construction differ a) in the type of curves the family consists of and b) whether the modulus p of the used field serves as parameter or whether it is sampled randomly alongside the curves. Regarding a), we give eeg families consisting of curve-twist pairs and eeg families consisting of Elligator1 or Elligator2 curves [BHKL13]. Curves of this type allow for an efficiently computable and efficiently invertible map between (a subset of) the curve points and a set, which only depends on the modulus of the underlying prime field. This enables the definition of sampling, embedding and inversion algorithms. Our eeg families $\mathsf{EG}_{\text{twist}}$, $\mathsf{EG}^{\ell}_{\text{twist-rs}}$ and $\mathsf{EG}_{\text{twist-re}}$ rely on pairs of a curve and its quadratic twist, solutions $\mathsf{EG}^{\ell}_{\text{ell1}}$, $\mathsf{EG}^{\ell}_{\text{ell2}}$,$\mathsf{EG}^{\ell}_{\text{ell1-rs}}$ and $\mathsf{EG}^{\ell}_{\text{ell2-rs}}$ are based on Elligator1 or Elligator2 curves respectively. The latter result in smaller key sizes of the corresponding KEM but are only applicable to transform **eegToKE1** since they consist of groups of even order. The eeg families $\mathsf{EG}_{\text{twist}}$, $\mathsf{EG}^{\ell}_{\text{ell1}}$, $\mathsf{EG}^{\ell}_{\text{ell2}}$ were implicitly given in prior work [Kal91, Möl04, BHKL13]. In these constructions the modulus p of the field serves as parameter and also determines the embedding space.

Regarding b), we provide alternatives to these constructions, which no longer rely on a fixed modulus. The constructions have empty parameters and p is sampled at random in the group generation algorithm. The technical challenge is to still achieve pseudorandom embeddings in an embedding space independent of the group. Our solutions $\mathsf{EG}^{\ell}_{\text{twist-rs}}$, $\mathsf{EG}^{\ell}_{\text{ell1-rs}}$, $\mathsf{EG}^{\ell}_{\text{ell2-rs}}$ build on the corresponding eeg families and achieve this by using rejection sampling with cut-off parameter ℓ. The corresponding embedding spaces consist of bit strings of length only dependent on the security parameter. The sampling algorithms have a worst-case running time of ℓ exponentiations, but the average cost is either two or three exponentiations independently of ℓ. Eeg family $\mathsf{EG}_{\text{twist-re}}$ building on a range expansion technique from [HOT04] improves on $\mathsf{EG}^{\ell}_{\text{twist-rs}}$ both in terms of efficiency and security. As in the other constructions embeddings are bit strings, but sampling only requires a single exponentiation.

Security of the instantiations. We now discuss the security properties of our instantiations in greater detail. An overview is given in Table 2. All of our constructions achieve EPR-PSA statistically. Embeddings in eeg families $\mathsf{EG}_{\text{twist}}$, and $\mathsf{EG}_{\text{twist-re}}$ are perfectly random, i.e. any (unbounded) adversary has advantage 0 in breaking EPR-PSA. For families $\mathsf{EG}^{\ell}_{\text{ell1}}$, $\mathsf{EG}^{\ell}_{\text{ell2}}$, $\mathsf{EG}^{\ell}_{\text{ell1-rs}}$, $\mathsf{EG}^{\ell}_{\text{ell2-rs}}$ and $\mathsf{EG}^{\ell}_{\text{twist-rs}}$ the advantage decays exponentially in the cut-off bound ℓ.

Diffie-Hellman problems CDH-PSA and sCDH-PSA are non standard. They are defined with respect to the eeg family's sampling algorithm and assume parameter subversion attacks. However, for all of our proposed instantiations we are able to show that CDH-PSA and sCDH-PSA can be reduced to assumptions, which no longer depend on the sampling algorithms, but use uniformly

sampled exponents instead. The assumptions differ in the type of curves and whether the modulus p of the used field serves as parameter or is chosen at random. Constructions $\mathsf{EG}_{\text{twist}}$, $\mathsf{EG}^{\ell}_{\text{twist-rs}}$ and $\mathsf{EG}_{\text{twist-re}}$ use curve-twist pairs of elliptic curves. Correspondingly, in this cases CDH-PSA security of both the curve and its twist is required. Families $\mathsf{EG}^{\ell}_{\text{ell1}}$, $\mathsf{EG}^{\ell}_{\text{ell2}}$, $\mathsf{EG}^{\ell}_{\text{ell1-rs}}$, $\mathsf{EG}^{\ell}_{\text{ell2-rs}}$ use Elligator1 or Elligator2 curves. So in this case each group in the family corresponds to a single curve, which has to be secure. Considering the parameters of our constructions, they belong to one of two classes. Eeg families $\mathsf{EG}_{\text{twist}}$, $\mathsf{EG}^{\ell}_{\text{ell1}}$ and $\mathsf{EG}^{\ell}_{\text{ell2}}$ use the modulus p as parameter, which might be subject to subversion. Accordingly CDH-PSA in this case corresponds to the assumption that the adversary's possibility to choose p does not improve its capacities in solving Diffie-Hellman instances on either curve-twist pairs or Elligator curves respectively. Eeg families $\mathsf{EG}^{\ell}_{\text{twist-rs}}$, $\mathsf{EG}_{\text{twist-re}}$, $\mathsf{EG}^{\ell}_{\text{ell1-rs}}$ and $\mathsf{EG}^{\ell}_{\text{ell2-rs}}$ serve as more conservative alternatives. Each of these eeg families is parameter-free and each user choses his own modulus at random, resulting in the weaker assumption that solving Diffie-Hellman instances over curves sampled with respect to a randomly chosen modulus is hard.

Related work. One might consider generating parameters via a multi-party computation protocol so that no particular party controls the outcome. It is unclear however what parties would perform this task and why one might trust any of them. PKE resistant to parameter subversion provides greater security.

Parameter subversion as we consider it allows the adversary full control of the parameters. This was first considered for NIZKs [BFS16] and (under the term backdoored) for PRGs [DGG+15, DPSW16]. Various prior works, in various contexts, considered relaxing the assumptions on parameters in some way [CPs07, GO07, GGJS11, KKZZ14], but these do not allow the adversary full control of the parameters and thus do not provide security against what we call parameter subversion.

Algorithm-substitution attacks, studied in [BPR14, BH15, DFP15, BJK15, AMV15], are another form of subversion, going back to the broader framework of kleptography [YY96, YY97]. The cliptography framework of RTYZ [RTYZ16] aims to capture many forms of subversion. In [RTYZ17] the same authors consider PKE that retains security in the face of substitution of any of its algorithms, but do not consider parameter subversion.

2 Preliminaries

Notation. We let ε denote the empty string. If X is a finite set, we let $x \leftarrow\!\!\$\ X$ denote picking an element of X uniformly at random and assigning it to x. All our algorithms are randomized and polynomial time (PT) unless stated otherwise. An adversary is an algorithm. Running time is worst case. If A is an algorithm, we let $y \leftarrow A(x_1, \ldots; r)$ denote running A with random coins r on inputs $x_1, \ldots$ and assigning the output to y. We let $y \leftarrow\!\!\$\ A(x_1, \ldots)$ be the result of picking r at random and letting $y \leftarrow A(x_1, \ldots; r)$. We let $[A(x_1, \ldots)]$ denote the set of all possible outputs of A when invoked with inputs $x_1, \ldots$. We use the code based game playing framework of [BR06]. (See Figure 4 for an example.) By Pr[G] we denote the probability that the execution of game G results in the game returning true. We also adopt the convention that the running time of an adversary refers to the worst case execution time of the game with the adversary. This means that the time taken for oracles to compute replies to queries is included. The random oracle model [BR93] is captured by a game procedure RO that implements a variable output length random oracle. It takes a string x and an integer m and returns a random m-bit string. We denote by $\mathcal{P}_k$ the set of primes of bit length k and by $[d]$ the set $\{0, \ldots, d-1\}$. Furthermore, the uniform distribution on M is denoted by U_M. If two random variables X and Y are equal in distribution we write $X \sim Y$. The statistical distance between X and Y is denoted by $\Delta(X; Y)$.

If $\Delta(X;Y) \leq \delta$ we say X is δ-close to Y.

3 Public-Key encryption resistant to parameter subversion

In this section we recall public-key encryption schemes and key encapsulation mechanisms. For both primitives we define the strong security notion of pseudorandomness of ciphertexts in the setting of parameter subversion and show that it implies both indistinguishability of encryptions and public-key hiding. We further define the security notion of well-distributedness of ciphertexts for key encapsulation mechanisms. Finally, we recall symmetric encryption schemes and revisit the hybrid encryption paradigm in the setting of ciphertext pseudorandomness under parameter subversion attacks.

3.1 Public-Key encryption schemes

Below we give a syntax for public-key encryption schemes. It follows [CS03], but uses slightly different notation and includes an additional algorithm setting up global parameters to be utilized by all users. We then formalize a novel security requirement of pseudorandomness of ciphertexts under parameter subversion attacks (CPR-PSA), which says that even if the parameters of the scheme are controlled by the adversary, ciphertexts obtained under any public key are indistinguishable from random elements of the ciphertext space, which depends only on the security parameter, the message length and the global parameters. We then recall two existing requirements of public-key encryption schemes adapting them to the setting of parameter subversion attacks. The first is the well-known notion of indistinguishability of encryptions [GM84], the second, from [BBDP01, ABC$^+$05], is that ciphertexts under different public keys are indistinguishable, which they called anonymity or key hiding and we call public-key hiding. In Proposition 3.1 we show that the first requirement implies the other two, allowing us to focus on it subsequently. We model the possibility of subverted parameters by having the adversary provide the parameters, which are used in the security games.

Public-Key Encryption. A *public-key encryption scheme* (PKE) PE specifies the following. Parameter generation algorithm $\mathsf{PE.P}$ takes input 1^k, where $k \in \mathbb{N}$ is the security parameter, and returns global parameters π. Key-generation algorithm $\mathsf{PE.G}$ takes input $1^k, \pi$ and returns a tuple (pk, sk) consisting of the public (encryption) key pk and matching secret (decryption) key sk. $\mathsf{PE.CS}$ associates to k, π and message length $m \in \mathbb{N}$ a finite set $\mathsf{PE.CS}(k, \pi, m)$ that is the *ciphertext space* of PE. Encryption algorithm $\mathsf{PE.E}$ takes $1^k, \pi, pk$ and a message $M \in \{0,1\}^*$ and returns a ciphertext $c \in \mathsf{PE.CS}(k, \pi, |M|)$. Deterministic decryption algorithm $\mathsf{PE.D}$ takes $1^k, \pi, sk$ and a ciphertext c and returns either a message $M \in \{0,1\}^*$ or the special symbol $\perp$ indicating failure. The correctness condition requires that for all $k \in \mathbb{N}$, all $\pi \in [\mathsf{PE.P}(1^k)]$, all $(pk, sk) \in [\mathsf{PE.G}(1^k, \pi)]$ and all $M \in \{0,1\}^*$ we have $\Pr\left[\mathsf{PE.D}(1^k, \pi, sk, c) = M\right] \geq 1 - \mathsf{PE.de}(k)$, where the probability is over $c \leftarrow\!\!\$\ \mathsf{PE.E}(1^k, \pi, pk, M)$ and $\mathsf{PE.de} : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is the *decryption error* of PE. Our PKEs will be in the ROM [BR93], which means the encryption and decryption algorithms have access to a random oracle specified in the security games. Correctness must then hold for all choices of the random oracle. We say a PKE is *parameter-free* if $\mathsf{PE.P}$ returns ε on every input 1^k.

Ciphertext pseudorandomness. Consider game $\mathbf{G}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{A}}(k)$ of Figure 2 associated to PKE PE, adversary $\mathcal{A}$ and security parameter k, and let

$$\mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{A}}(k) = 2\Pr[\mathbf{G}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{A}}(k)] - 1 \ .$$

We say that PE has pseudorandom ciphertexts under parameter subversion attacks (also called

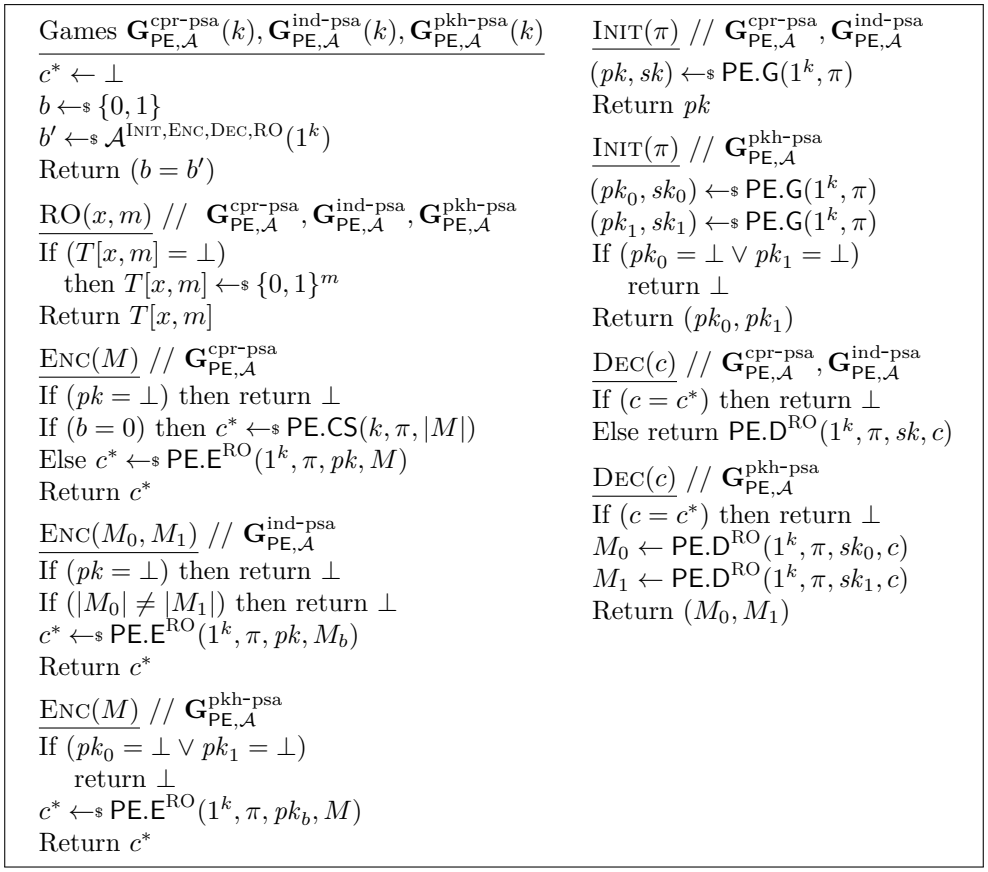


Figure 2: Games defining security of PKEs. In each game the adversary is given access to oracles. The game, to which an oracle belongs, is indicated behind the oracle's name. In each game oracles Init and Enc may be queried only once. Further Init has to be queried before using any of the other oracles.

CPR-PSA) if the function $\mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{A}}(\cdot)$ is negligible for every $\mathcal{A}$. In the game, b is a challenge bit. When $b = 1$, the challenge ciphertext c^* is an encryption of a message of the adversary's choice, but if $b = 0$ it is chosen at random from the ciphertext space. Given the public key and challenge ciphertext, the adversary outputs a guess b' and wins if b' equals b, the game returning true in this case and false otherwise. The adversary has access to an oracle Init, which sets up the public key using parameters of the adversary's choice, and an oracle Enc to generate the challenge ciphertext. Furthermore it has access to the random oracle and a decryption oracle crippled to not work on the challenge ciphertext. We require that the adversary queries the oracles Init and Enc only once. Furthermore Init has to be queried before using any of the other oracles.

Indistinguishability of encryptions. Consider game $\mathbf{G}^{\text{ind-psa}}_{\mathsf{PE},\mathcal{A}}(k)$ of Figure 2 associated to PKE PE, adversary $\mathcal{A}$ and security parameter k, and let

$$\mathbf{Adv}^{\text{ind-psa}}_{\mathsf{PE},\mathcal{A}}(k) = 2\Pr[\mathbf{G}^{\text{ind-psa}}_{\mathsf{PE},\mathcal{A}}(k)] - 1 \ .$$

We say that PE has indistinguishable encryptions under parameter subversion attacks (also called IND-PSA) if the function $\mathbf{Adv}^{\text{ind-psa}}_{\mathsf{PE},\mathcal{A}}(\cdot)$ is negligible for every $\mathcal{A}$. In the game, b is a challenge bit. The adversary has access to an oracle Init, which sets up the public key using parameters of the adversary's choice, and an oracle Enc, which receives as input two messages M_0, M_1 of the

same length and outputs the challenge ciphertext c^*. When $b = 0$, the challenge ciphertext is an encryption of M_0, if $b = 1$ an encryption of M_1. Given the public key and challenge ciphertext, the adversary outputs a guess b' and wins if b' equals b, the game returning true in this case and false otherwise. Again, the adversary has access to the random oracle and a decryption oracle crippled to not work on the challenge ciphertext. We require that the adversary queries the oracles Init and Enc only once. Furthermore Init has to be queried before using any of the other oracles.

Public-key hiding. Consider game $\mathbf{G}^{\text{pkh-psa}}_{\mathsf{PE},\mathcal{A}}(k)$ of Figure 2 associated to PKE PE, adversary $\mathcal{A}$ and security parameter k, and let

$$\mathbf{Adv}^{\text{pkh-psa}}_{\mathsf{PE},\mathcal{A}}(k) = 2\Pr[\mathbf{G}^{\text{pkh-psa}}_{\mathsf{PE},\mathcal{A}}(k)] - 1 \ .$$

We say that PE is public-key hiding under parameter subversion attacks (also called PKH-PSA) if the function $\mathbf{Adv}^{\text{pkh-psa}}_{\mathsf{PE},\mathcal{A}}(\cdot)$ is negligible for every $\mathcal{A}$. In the game, b is a challenge bit. Unlike the prior games, two key pairs are generated, not one. The challenge ciphertext c^* is an encryption of a message of the adversary's choice under pk_b. Given the public keys and the challenge ciphertext, the adversary outputs a guess b' and wins if b' equals b. This time the crippled decryption oracle returns decryptions under both secret keys. The adversary sets up the public keys with its call to oracle Init, and an uses oracle Enc to generate the challenge ciphertext. Again we require that the adversary queries the oracles Init and Enc only once. Furthermore Init has to be queried before using any of the other oracles.

Relations. The following says that pseudorandomness of ciphertexts implies both indistinguishable encryptions and anonymity. We give both asymptotic and concrete statements of the results.

Proposition 3.1. *Let* PE *be a PKE that has pseudorandom ciphertexts under parameter subversion attacks. Then:*

1. PE *is* IND-PSA. *Concretely, given an adversary* $\mathcal{A}$ *the proof specifies an adversary* $\mathcal{B}_0$ *such that* $\mathbf{Adv}^{\text{ind-psa}}_{\mathsf{PE},\mathcal{A}}(k) \le 2 \cdot \mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{B}_0}(k)$ *for every* $k \in \mathbb{N}$*, and* $\mathcal{B}_0$ *has the same running time and query counts as* $\mathcal{A}$*.*
2. PE *is* PKH-PSA. *Concretely, given an adversary* $\mathcal{A}$ *the proof specifies an adversary* $\mathcal{B}_1$ *such that* $\mathbf{Adv}^{\text{pkh-psa}}_{\mathsf{PE},\mathcal{A}}(k) \le 2 \cdot \mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{B}_1}(k)$ *for every* $k \in \mathbb{N}$*, and* $\mathcal{B}_0$ *has the same running time and query counts as* $\mathcal{A}$*.*

Proof. We first prove statement 1. Let $k \in \mathbb{N}$ and let $\mathcal{A}$ be an adversary against the encryption indistinguishability game of Figure 2. We construct an adversary $\mathcal{B}_0$ against the ciphertext uniformity game. The definition of $\mathcal{B}_0$ may be found in Figure 3. To analyze $\mathcal{B}_0$'s advantage note that

$$\mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{B}_0}(k) = \Pr[\mathbf{G}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{B}_0}(k) = \mathsf{true} \mid b = 1] - \Pr[\mathbf{G}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{B}_0}(k) = \mathsf{false} \mid b = 0].$$

Assume that b sampled in $\mathbf{G}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{B}_0}(k)$ equals 0. In this case all input that $\mathcal{A}$ receives as answer to its oracle queries is independent of d. Hence $\Pr[\mathbf{G}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{B}_0}(k) = \mathsf{false} \mid b = 0] = 1/2$. On the other hand if $b = 1$, $\mathcal{A}$'s query to SimEnc is answered with an encryption of message M_d. Hence in this case $\mathcal{B}_0$ provides $\mathcal{A}$ with a perfect simulation of $\mathbf{G}^{\text{ind-psa}}_{\mathsf{PE},\mathcal{A}}(k)$. Since $\mathcal{B}_0$ returns 1 exactly if $d = d'$, this implies $\Pr[\mathbf{G}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{B}_0}(k) = \mathsf{true} \mid b = 1] = 1/2\mathbf{Adv}^{\text{ind-psa}}_{\mathsf{PE}}(\mathcal{A})(k) + 1/2$ Combining both equalities yields $\mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{B}_0}(k) = 1/2 \cdot \mathbf{Adv}^{\text{ind-psa}}_{\mathsf{PE},\mathcal{A}}(k)$ as desired.

We now prove statement 2. Let $k \in \mathbb{N}$ and let $\mathcal{A}$ be an adversary against the public-key hiding game of Figure 2. We construct an adversary $\mathcal{B}_1$ against the ciphertext uniformity game. The definition of $\mathcal{B}_1$ may be found in Figure 3. To analyze $\mathcal{B}_1$'s advantage note that

$$\mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{B}_1}(k) = \Pr[\mathbf{G}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{B}_1}(k) = \mathsf{true} \mid b = 1] - \Pr[\mathbf{G}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{B}_1}(k) = \mathsf{false} \mid b = 0].$$

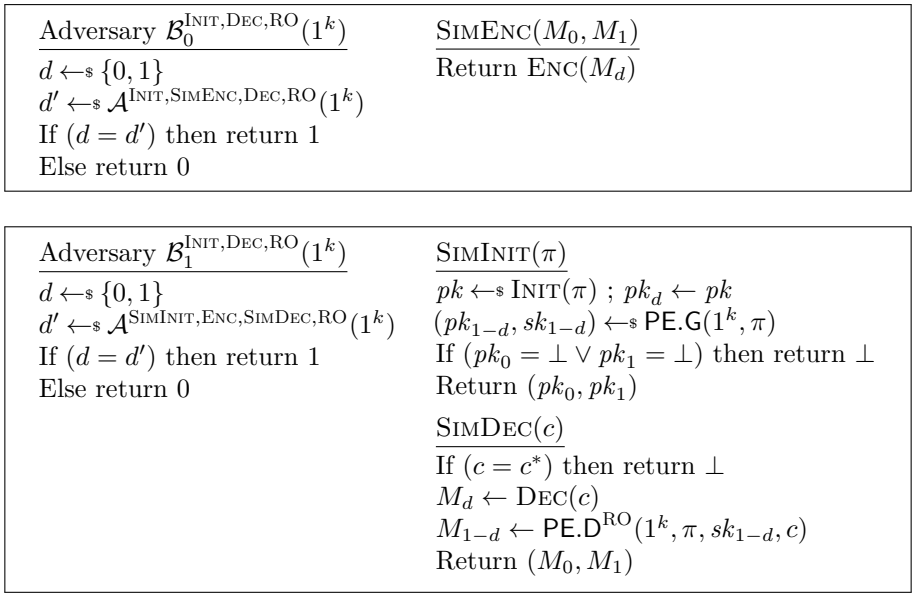


Figure 3: Adversaries for the proof of Proposition 3.1.

Assume that b sampled in $\mathbf{G}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{B}_1}(k)$ equals 0. The answer (pk_0, pk_1) to $\mathcal{A}$'s initialization query is independent of d, since pk_0 and pk_1 are equal in distribution and independently generated. Furthermore the answer to $\mathcal{A}$'s query to SimEnc is a uniformly sampled ciphertext, which is independent of d. Hence $\Pr[\mathbf{G}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{B}_1}(k) = \mathsf{false} \mid b = 0] = 1/2$.

On the other hand, if $b = 1$ the answer to $\mathcal{A}$'s query to SimEnc(M) was generated as $c^* \leftarrow\$ \mathsf{PE.E}(1^k, pk_d, M)$. Hence in this case $\mathcal{B}_1$ provides $\mathcal{A}$ with a perfect simulation of game $\mathbf{G}^{\text{pkh-psa}}_{\mathsf{PE},\mathcal{A}}(k)$. Since $\mathcal{B}_1$ outputs 1 exactly if $d = d'$, we obtain $\Pr[\mathbf{G}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{B}_1}(k) = \mathsf{true} \mid b = 1] = \Pr[d = d' \mid b = 1] = 1/2 \cdot \mathbf{Adv}^{\text{pkh-psa}}_{\mathsf{PE},\mathcal{A}}(k) + 1/2$. Plugging the results into the equation from above yields $\mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{B}_1}(k) = 1/2 \cdot \mathbf{Adv}^{\text{pkh-psa}}_{\mathsf{PE},\mathcal{A}}(k)$ as desired. □

3.2 Key encapsulation mechanisms

Below we first give a syntax for key encapsulation mechanisms. It follows [CS03] but with notation a bit different and including an additional algorithm setting up global parameters to be utilized by all users. As for public-key encryption schemes we formalize the security requirement of pseudorandomness of ciphertexts under parameter subversion attacks (CPR-PSA). We then adapt the two existing KEM requirements of indistinguishability of encryptions [CS03] and public-key hiding [BBDP01, ABC+05] to the setting of parameter subversion attacks. In Proposition 3.2 we show that —as in the case of public-key encryption— the first requirement implies the other two. We furthermore define a new security requirement called well-distributedness of ciphertexts, which is necessary to achieve CPR-PSA in the hybrid PKE construction. It states that key-ciphertext pairs generated using the KEM's encapsulation algorithm are indistinguishable from choosing a ciphertext at random and then computing its decapsulation.

KEMs. A *key encapsulation mechanism* (KEM) KE specifies the following. Parameter generation algorithm $\mathsf{KE.P}$ takes input 1^k, where $k \in \mathbb{N}$ is the security parameter, and returns global parameters π. Key-generation algorithm $\mathsf{KE.G}$ takes input $1^k, \pi$ and returns a tuple (pk, sk) consisting of the public (encryption) key pk and matching secret (decryption) key sk. $\mathsf{KE.KS}$ associates to k a finite set $\mathsf{KE.KS}(k)$ only depending on the security parameter that is the *key space*

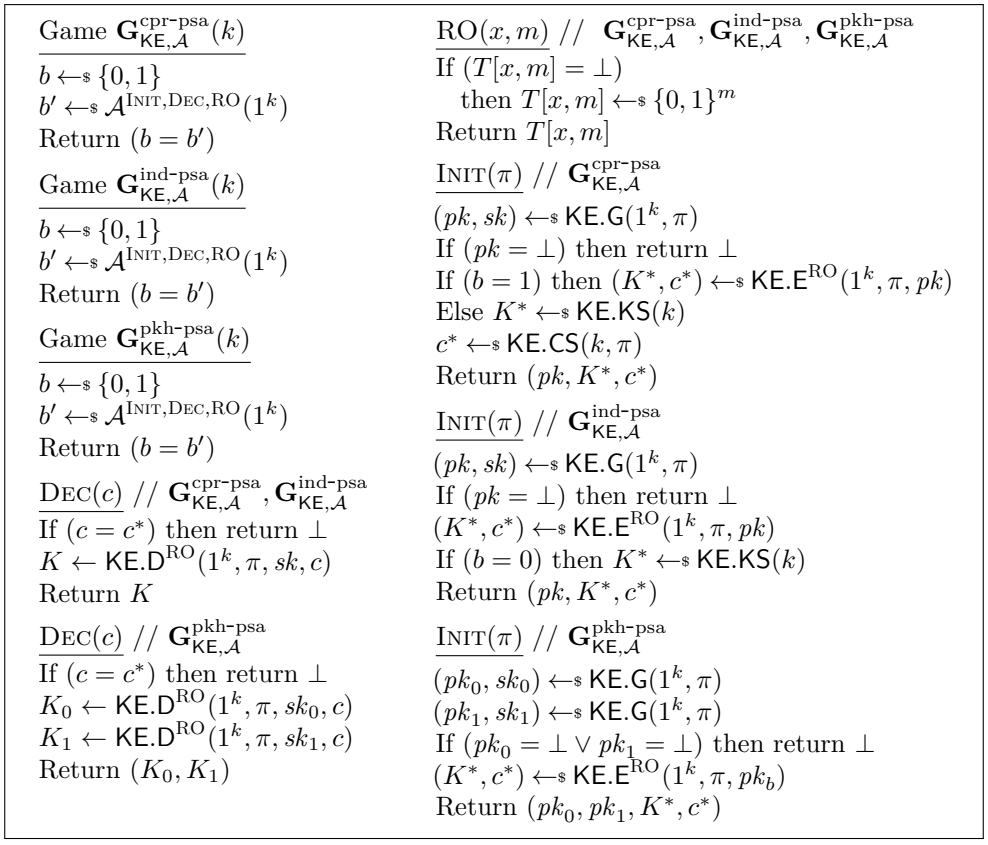


Figure 4: Games defining security of key encapsulation mechanism KE. In each game the adversary is given access to oracles. The game, to which an oracle belongs, is indicated behind the oracle's name. In each game oracle Init must be queried only once, which has to be done before using any of the other oracles.

---

of KE. KE.CS associates to k and parameters π a finite set $\mathsf{KE.CS}(k, \pi)$ that is the *ciphertext space* of KE. Encapsulation algorithm KE.E takes $1^k, \pi, pk$ and returns (K, c) where $K \in \mathsf{KE.KS}(k)$ is the *encapsulated key* and $c \in \mathsf{KE.CS}(k, \pi)$ is a ciphertext encapsulating K. Deterministic decapsulation algorithm KE.D takes $1^k, \pi, sk$ and a ciphertext c and returns either a key $K \in \mathsf{KE.KS}(k)$ or the special symbol $\perp$ indicating failure. The correctness condition requires that for all $k \in \mathbb{N}$, all $\pi \in [\mathsf{KE.P}(1^k)]$ and all $(pk, sk) \in [\mathsf{KE.G}(1^k, \pi)]$ we have $\Pr\left[\mathsf{KE.D}(1^k, \pi, sk, c) = K\right] \geq 1 - \mathsf{KE.de}(k)$, where the probability is over $(K, c) \leftarrow\!\!\$\ \mathsf{KE.E}(1^k, \pi, pk)$ and $\mathsf{KE.de} : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is the *decryption error* of KE. Our KEMs will be in the ROM [BR93], which means the encapsulation and decapsulation algorithms have access to a random oracle specified in the security games. Correctness must then hold for all choices of the random oracle. We say a KEM is *parameter-free* if KE.P returns ε on every input 1^k.

Ciphertext pseudorandomness. Consider game $\mathbf{G}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{A}}(k)$ of Figure 4 associated to KEM KE, adversary $\mathcal{A}$ and security parameter k, and let

$$\mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{A}}(k) = 2\Pr[\mathbf{G}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{A}}(k)] - 1\ .$$

We say that KE has pseudorandom ciphertexts under parameter subversion attacks (also called CPR-PSA) if the function $\mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{A}}(\cdot)$ is negligible for every $\mathcal{A}$. In the game, b is a challenge bit. When $b = 1$, the challenge key K^* and ciphertext c^* are generated via the encapsulation algorithm, but if $b = 0$ they are chosen at random, from the key space and ciphertext space,

respectively. Given the public key, challenge key and challenge ciphertext, the adversary outputs a guess b' and wins if b' equals b, the game returning true in this case and false otherwise. The adversary has access to an oracle INIT, which sets up the challenge. We require that the adversary queries INIT before using any of the other oracles and that it queries INIT only once. Further the adversary has access to an oracle for decapsulation under sk, crippled to not work when invoked on the challenge ciphertext. It, and the encapsulation and decapsulation algorithms, have access to the random oracle RO. The parameters used in the game are provided by the adversary via its call to INIT.

INDISTINGUISHABILITY OF ENCAPSULATED KEYS FROM RANDOM. Consider game $\mathbf{G}^{\text{ind-psa}}_{\mathsf{KE},\mathcal{A}}(k)$ of Figure 4 associated to KEM KE, adversary $\mathcal{A}$ and security parameter k, and let

$$\mathbf{Adv}^{\text{ind-psa}}_{\mathsf{KE},\mathcal{A}}(k) = 2\Pr[\mathbf{G}^{\text{ind-psa}}_{\mathsf{KE},\mathcal{A}}(k)] - 1 \ .$$

We say that KE has encapsulated keys indistinguishable from random under parameter subversion attacks (also called IND-PSA) if the function $\mathbf{Adv}^{\text{ind-psa}}_{\mathsf{KE},\mathcal{A}}(\cdot)$ is negligible for every $\mathcal{A}$. In the game, b is a challenge bit. When $b = 1$, the challenge key K^* and ciphertext c^* are generated via the encapsulation algorithm, while if $b = 0$ the key is switched to one drawn randomly from the key space, the ciphertext remaining real. Given the public key, challenge key and challenge ciphertext, the adversary outputs a guess b' and wins if b' equals b. Again the adversary has access to a crippled decapsulation oracle and the random oracle and provides the parameters used in the game via his call to the oracle INIT, which has to be queried before using any of the other oracles.

PUBLIC-KEY HIDING. Consider game $\mathbf{G}^{\text{pkh-psa}}_{\mathsf{KE},\mathcal{A}}(k)$ of Figure 4 associated to KEM KE, adversary $\mathcal{A}$ and security parameter k, and let

$$\mathbf{Adv}^{\text{pkh-psa}}_{\mathsf{KE},\mathcal{A}}(k) = 2\Pr[\mathbf{G}^{\text{pkh-psa}}_{\mathsf{KE},\mathcal{A}}(k)] - 1 \ .$$

We say that KE is public-key hiding under parameter subversion attacks (also called PKH-PSA) if the function $\mathbf{Adv}^{\text{pkh-psa}}_{\mathsf{KE},\mathcal{A}}(\cdot)$ is negligible for every $\mathcal{A}$. In the game, b is a challenge bit. Unlike the prior games, two key pairs are generated, not one. The challenge key K^* and ciphertext c^* are generated via the encapsulation algorithm under pk_b. Given the public keys, challenge key and challenge ciphertext, the adversary outputs a guess b' and wins if b' equals b. This time the crippled decapsulation oracle returns decapsulations under both secret keys. Again the adversary provides the parameters to be used in the game via his single call to the oracle INIT, which has to be queried before using any of the other oracles.

RELATIONS. The following says that in the parameter subversion setting CPR-PSA implies both IND-PSA and PKH-PSA. We give both the asymptotic and concrete statements of the results.

Proposition 3.2. *Let KE be a KEM that has pseudorandom ciphertexts under parameter subversion attacks. Then:*

1. KE *is* IND-PSA. *Concretely, given an adversary $\mathcal{A}$ the proof specifies an adversary $\mathcal{B}$ such that $\mathbf{Adv}^{\text{ind-psa}}_{\mathsf{KE},\mathcal{A}}(k) \leq 2 \cdot \mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}}(k)$ for every $k \in \mathbb{N}$, and $\mathcal{B}$ has the same running time and query counts as $\mathcal{A}$.*
2. KE *is* PKH-PSA. *Concretely, given an adversary $\mathcal{A}$ the proof specifies an adversary $\mathcal{B}$ such that $\mathbf{Adv}^{\text{pkh-psa}}_{\mathsf{KE},\mathcal{A}}(k) \leq 2 \cdot \mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}}(k)$ for every $k \in \mathbb{N}$, and $\mathcal{B}$ has the same running time and query counts as $\mathcal{A}$.*

Proof. We first give a proof for statement 1. Consider the sequence of games $\mathbf{G}_0$, $\mathbf{G}_1$, $\mathbf{G}_2$ of

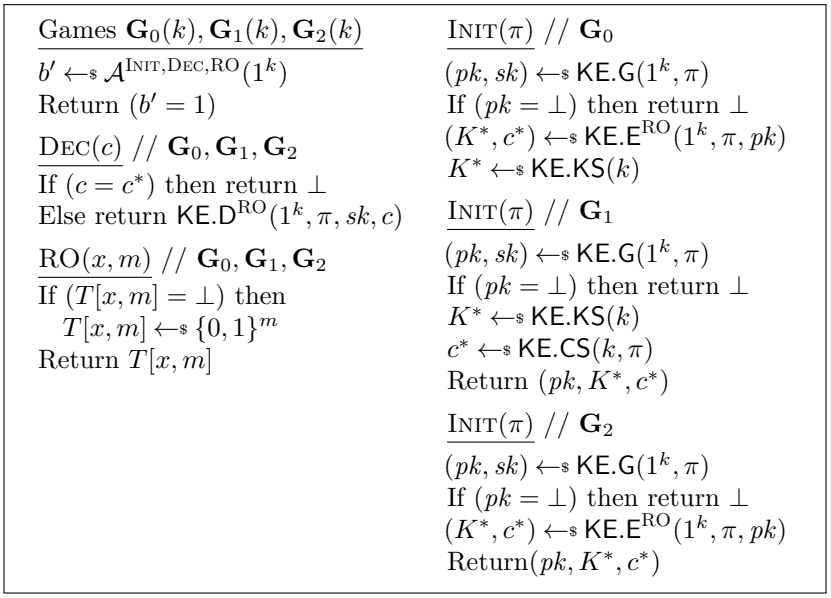


Figure 5: Games for the proof of Proposition 3.2.

Figure 5. Using $\mathbf{G}_0$ and $\mathbf{G}_2$ we may rewrite $\mathcal{A}$'s advantage in the CPR-PSA game as

$$\begin{aligned}\mathbf{Adv}^{\text{ind-psa}}_{\mathsf{KE},\mathcal{A}}(k) &= \Pr[\mathbf{G}_2(k)] - \Pr[\mathbf{G}_0(k)] \\ &= \Pr[\mathbf{G}_2(k)] - \Pr[\mathbf{G}_1(k)] + \Pr[\mathbf{G}_1(k)] - \Pr[\mathbf{G}_0(k)] \ .\end{aligned}$$

We give two adversaries $\mathcal{B}_0$ and $\mathcal{B}_1$ satisfying $\mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}_0}(k) = \Pr[\mathbf{G}_2(k)] - \Pr[\mathbf{G}_1(k)]$ and $\mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}_1}(k) = \Pr[\mathbf{G}_1(k)] - \Pr[\mathbf{G}_0(k)]$. This yields $\mathbf{Adv}^{\text{ind-psa}}_{\mathsf{KE},\mathcal{A}}(k) = \mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}_0}(k) + \mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}_1}(k)$. Hence defining $\mathcal{B}$ as an adversary which samples $d \leftarrow\!\!\$\ \{0,1\}$ and runs $\mathcal{B}_d$ yields the claim.

The definition of the adversaries may be found in Figure 6. Consider adversary $\mathcal{B}_0$. If b of game $\mathbf{G}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}_0}(k)$ equals 1 , $\mathcal{B}_0$ provides $\mathcal{A}$ with a perfect simulation of $\mathbf{G}_2(k)$. If $b = 0$ it provides $\mathcal{A}$ with a perfect simulation of $\mathbf{G}_1(k)$. Thus

$$\mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}_0}(k) = \Pr[\mathbf{G}_2(k)] - \Pr[\mathbf{G}_1(k)] \ .$$

Now consider asversary $\mathcal{B}_1$. If $b = 1$, $\mathcal{B}_1$ provides $\mathcal{A}$ with a perfect simulation of $\mathbf{G}_0(k)$. If $b = 0$ it provides $\mathcal{A}$ with a perfect simulation of $\mathbf{G}_1(k)$. $\mathcal{B}_1$ returns $1 - b'$, which yields

$$\begin{aligned}\Pr[\mathbf{G}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}_1}(k) = \mathsf{true} \mid b = 1] &= \Pr[\mathbf{G}_0(k) = \mathsf{false}] = 1 - \Pr[\mathbf{G}_0(k)] \text{ and} \\ \Pr[\mathbf{G}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}_1}(k) = \mathsf{false} \mid b = 0] &= \Pr[\mathbf{G}_1(k) = \mathsf{false}] = 1 - \Pr[\mathbf{G}_1(k)] \ .\end{aligned}$$

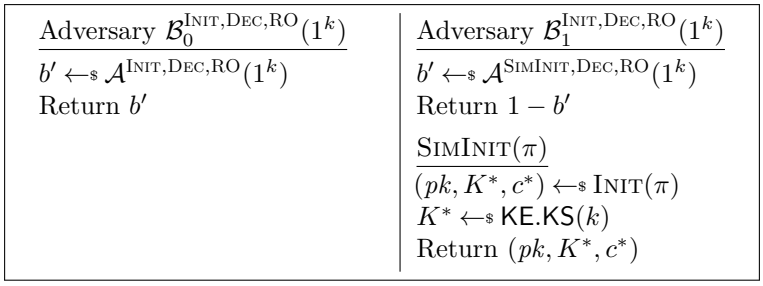
We obtain

$$\begin{aligned}\mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}_1}(k) &= \Pr[\mathbf{G}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}_1}(k) = \mathsf{true} \mid b = 1] - \Pr[\mathbf{G}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}_1}(k) = \mathsf{false} \mid b = 0] \\ &= \Pr[\mathbf{G}_1(k)] - \Pr[\mathbf{G}_0(k)] \ .\end{aligned}$$

As stated above, taking an appropriate combination of $\mathcal{B}_0$ and $\mathcal{B}_1$ yields an adversary as described in the claim.

Now we prove statement 2. Let $k \in \mathbb{N}$ and let $\mathcal{A}$ be an adversary against the public-key hiding game of Figure 4. We construct an adversary $\mathcal{B}$ against the ciphertext uniformity game. The definition of $\mathcal{B}$ may be found in Figure 6. To analyze $\mathcal{B}$'s advantage note that

$$\mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}}(k) = \Pr[\mathbf{G}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}}(k) = \mathsf{true} \mid b = 1] - \Pr[\mathbf{G}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}}(k) = \mathsf{false} \mid b = 0].$$



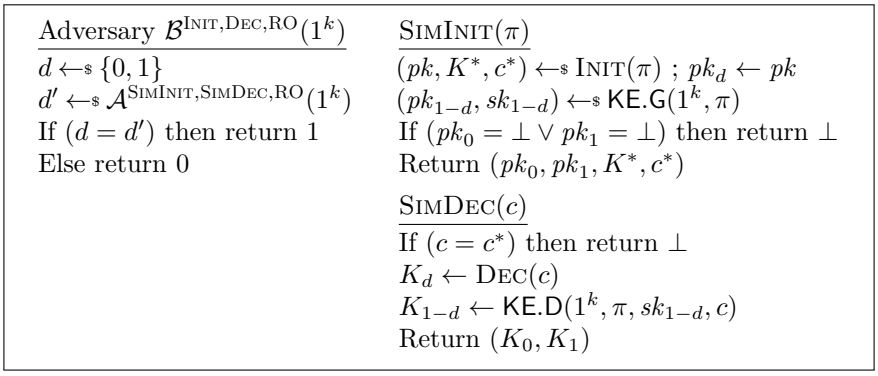


Figure 6: Adversaries for the proof of Proposition 3.2.

$\mathcal{B}$ returns 1 exactly if the bit d' returned by $\mathcal{A}$ equals d. Assume that b sampled in the definition of $\mathbf{G}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}}(k)$ equals 0. In this situation the answer (pk_0, pk_1, c^*, K^*) to $\mathcal{A}$'s initialization query is independent of d, since pk_0 and pk_1 are equal in distribution and K^* and c^* are sampled independently of the keys. Hence $\Pr[\mathbf{G}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}}(k) = \mathsf{false} \mid b = 0] = 1/2$.

On the other hand, if $b = 1$ the key-ciphertext pair was generated as $(K^*, c^*) \leftarrow\!\!\$\ \mathsf{KE.E}(1^k, pk_d)$. Hence in this case (pk_0, pk_1, K^*, c^*) is distributed as in Game $\mathbf{G}^{\text{pkh-psa}}_{\mathsf{KE},\mathcal{A}}(k)$. Since $\mathcal{B}$ outputs 1 exactly if $d = d'$, we obtain $\Pr[\mathbf{G}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}}(k) = \mathsf{true} \mid b = 1] = \Pr[d = d' \mid b = 1] = 1/2 \cdot \mathbf{Adv}^{\text{pkh-psa}}_{\mathsf{KE},\mathcal{A}}(k) + 1/2$.

Plugging the results into the equation from above yields $\mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}}(k) = 1/2 \cdot \mathbf{Adv}^{\text{pkh-psa}}_{\mathsf{KE},\mathcal{A}}(k)$ as desired.

□

Well-distributed ciphertexts. Consider game $\mathbf{G}^{\text{wdc-psa}}_{\mathsf{KE},\mathcal{A}}(k)$ of Figure 11 associated to KEM KE, adversary $\mathcal{A}$ and security parameter k, and let

$$\mathbf{Adv}^{\text{wdc-psa}}_{\mathsf{KE},\mathcal{A}}(k) = 2\Pr[\mathbf{G}^{\text{wdc-psa}}_{\mathsf{KE},\mathcal{A}}(k)] - 1.$$

We say KE has well distributed ciphertexts under parameter subversion attacks (also called WDC-PSA), if the function $\mathbf{Adv}^{\text{wdc-psa}}_{\mathsf{KE},\mathcal{A}}(\cdot)$ is negligible for every adversary $\mathcal{A}$. In the game b is a challenge bit. If b equals 1 the adversary as response to querying the initialization procedure, which may be done at most once, receives a key-ciphertext pair generated using $\mathsf{KE.E}$. If b equals 0 it receives a pair (c^*, K^*) generated by choosing c^* at random and then setting K^* to be the decapsulation of c^*. The adversary has access to a decryption oracle. We require that the adversary queries Init before querying any of the other oracles. Looking ahead, all of our instantiations achieve this notion statistically.

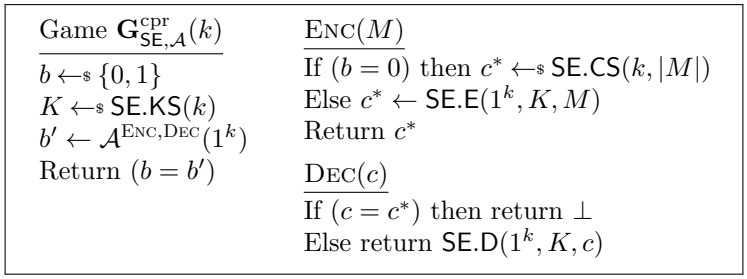


Figure 7: Games defining one-time security notions of SKEs.

3.3 Symmetric encryption

Below, we recall symmetric encryption. Our definition follows [CS03] but uses different notation. We further define the security notion of ciphertext pseudorandomness for symmetric key encryption.

One-Time symmetric-Key Encryption. A symmetric-key encryption scheme (SKE) specifies the following. $\mathsf{SE.KS}$ associates to security parameter k key space $\mathsf{SE.KS}(k)$. $\mathsf{SE.CS}$ associates to security parameter k and message length $m \in \mathbb{N}$ the ciphertext space $\mathsf{SE.CS}(k, m)$. Deterministic encryption algorithm $\mathsf{SE.E}$ takes as input 1^k, key $K \in \mathsf{SE.KS}(k)$ and a message $M \in \{0,1\}^*$ and returns ciphertext $c \in \mathsf{SE.CS}(k, |M|)$. Deterministic decryption algorithm $\mathsf{SE.D}$ on input $1^k, K \in \mathsf{SE.KS}(k), c \in \mathsf{SE.CS}(k, m)$ returns either a message $M \in \{0,1\}^m$ or the special symbol $\perp$ indicating failure. For correctness we require that $M = \mathsf{SE.D}(1^k, K, c)$ for all k, all $K \in \mathsf{SE.KS}(k)$ and all $M \in \{0,1\}^*$, where $c \leftarrow \mathsf{SE.E}(1^k, K, M)$.

One-time security Consider game $\mathbf{G}^{\text{cpr}}_{\mathsf{SE},\mathcal{A}}(k)$ of Figure 7 associated to SKE SE, adversary $\mathcal{A}$ and security parameter k, and let

$$\mathbf{Adv}^{\text{cpr}}_{\mathsf{SE},\mathcal{A}}(k) = 2\Pr[\mathbf{G}^{\text{cpr}}_{\mathsf{SE},\mathcal{A}}(k)] - 1 \ .$$

We say that SE has pseudorandom ciphertexts (also called CPR) if the function $\mathbf{Adv}^{\text{cpr}}_{\mathsf{SE},\mathcal{A}}(\cdot)$ is negligible for every $\mathcal{A}$. We require that Enc is queried at most once.

3.4 PKE from key encapsulation and symmetric-key encryption

Below, we analyze hybrid encryption in the setting of parameter subversion. Formally we give a transform **KEMToPE** that associates to KEM KE and symmetric-key encryption scheme SE a public-key encryption scheme PE. The construction essentially is the hybrid encryption scheme of [CS03] including an additional parameter generation algorithm. The scheme's parameter generation, key generation encryption and decryption algorithms are in Figure 8. PE's ciphertext space is given by $\mathsf{PE.CS}(k, \pi, m) = \mathsf{KE.CS}(k, \pi) \times \mathsf{SE.CS}(k, m)$. It is easy to verify that PE has decryption error $\mathsf{PE.de}(k) = \mathsf{KE.de}(k)$. The following essentially states that hybrid encryption also works in setting of ciphertext pseudorandomness under parameter subversion attacks, i.e., combining a KEM that is both CPR-PSA and WDC-PSA with a SKE that is CPR yields a CPR-PSA PKE, where the well-distributedness of the KEM's ciphertext is necessary to correctly simulate the decryption oracle in the CPR-PSA game with respect to PE.

Proposition 3.3. *Let* KE *a KEM and* SE *a SE such that* $\mathsf{KE.KS}(k) = \mathsf{SE.KS}(k)$ *for all* $k \in \mathbb{N}$. *Let* $\mathsf{PE} = \mathbf{KEMToPE}[\mathsf{KE}, \mathsf{SE}]$ *be the PKE associated to* KE *and* SE. *If* KE *is* CPR-PSA *and* WDC-PSA *and if* SE *is* CPR *then* PE *is* CPR-PSA *Concretely, given adversary* $\mathcal{A}$ *against* $\mathbf{G}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{A}}(k)$, *there exist adversaries* $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ *having the same running time and query count as*

$\underline{\mathsf{PE.P}(1^k)}$	$\underline{\mathsf{PE.E}(1^k, \pi, pk, M)}$
$\pi \leftarrow\!\!\$\ \mathsf{KE.P}(1^k)$	$(K, c_1) \leftarrow\!\!\$\ \mathsf{KE.E}^{\mathrm{RO}}(1^k, \pi, pk)$
Return π	$c_2 \leftarrow \mathsf{SE.E}(1^k, K, M)$
$\underline{\mathsf{PE.G}(1^k, \pi)}$	Return (c_1, c_2)
$(pk, sk) \leftarrow\!\!\$\ \mathsf{KE.G}(1^k, \pi)$	$\underline{\mathsf{PE.D}(1^k, \pi, sk, c)}$
Return (pk, sk)	$(c_1, c_2) \leftarrow c$
	$K \leftarrow \mathsf{KE.D}^{\mathrm{RO}}(1^k, \pi, sk, c_1)$
	$M \leftarrow \mathsf{SE.D}(1^k, K, c_2)$
	Return M

Figure 8: PKE $\mathbf{KEMToPE}[\mathsf{KE}, \mathsf{SE}]$ associated to KEM KE and SE SE.

$\mathcal{A}$, which satisfy

$$\mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{A}}(k) \leq 2\ \mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}_1}(k) + \mathbf{Adv}^{\text{wdc-psa}}_{\mathsf{KE},\mathcal{B}_2}(k) + \mathbf{Adv}^{\text{cpr}}_{\mathsf{SE},\mathcal{B}_3}(k) + \mathsf{KE.de}(k)\ .$$

Proof. Let $\mathsf{PE} = \mathbf{KEMToPE}[\mathsf{KE}, \mathsf{SE}]$, and $\mathcal{A}$ be an adversary against $\mathbf{G}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{A}}(k)$. Consider the sequence of games $\mathbf{G}_0, \ldots, \mathbf{G}_6$ of Figure 9 associated to PE, $\mathcal{A}$ and security parameter k. We have

$$\mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{PE},\mathcal{A}}(k) = \Pr[\mathbf{G}_6(k)] - \Pr[\mathbf{G}_0(k)]\ .$$

Games $\mathbf{G}_0$ and $\mathbf{G}_1$ only differ by a conceptual change in the way decryption queries are are answered. Hence

$$\Pr[\mathbf{G}_1(k)] - \Pr[\mathbf{G}_0(k)] = 0\ . \tag{1}$$

As a next step we give an adversary $\mathcal{B}_2$ such that

$$\Pr[\mathbf{G}_2(k)] - \Pr[\mathbf{G}_1(k)] \leq \mathbf{Adv}^{\text{wdc-psa}}_{\mathsf{KE},\mathcal{B}_2}(k)\ . \tag{2}$$

The definition of $\mathcal{B}_2$ may be found in Figure 10. If the challenge bit b of game $\mathbf{G}^{\text{wdc-psa}}_{\mathsf{KE},\mathcal{B}_2}(k)$ equals 1, adversary $\mathcal{B}_2$ provides $\mathcal{A}$ with a perfect simulation of $\mathbf{G}_2$, if b equals 0 it provides $\mathcal{A}$ with a perfect simulation of $\mathbf{G}_1$. This establishes Equation (2).

We now give an adversary $\mathcal{B}'_1$ such that

$$\Pr[\mathbf{G}_3(k)] - \Pr[\mathbf{G}_2(k)] = \mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}'_1}(k)\ . \tag{3}$$

The definition of adversary $\mathcal{B}'_1$ may be found in Figure 10. If the challenge bit b in game $\mathbf{G}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}'_1}(k)$ equals 0, adversary $\mathcal{B}'_1$ provides $\mathcal{A}$ with a perfect simulation of game $\mathbf{G}_3$. If b equals 1, it provides $\mathcal{A}$ with a perfect simulation of game $\mathbf{G}_2$. Since $\mathcal{B}'_1$ returns $1 - b'$, we have $\mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}'_1}(k) = (1 - \Pr[\mathbf{G}_2(k)]) - (1 - \Pr[\mathbf{G}_3(k)])$. Equation (3) follows.

As a next step we give an adversary $\mathcal{B}_3$ such that

$$\Pr[\mathbf{G}_4(k)] - \Pr[\mathbf{G}_3(k)] = \mathbf{Adv}^{\text{cpr}}_{\mathsf{SE},\mathcal{B}_3}(k)\ . \tag{4}$$

The definition of adversary $\mathcal{B}_3$ may be found in Figure 10. If the challenge bit b in game $\mathbf{G}^{\text{cpr}}_{\mathsf{SE},\mathcal{B}_3}(k)$ equals 0, adversary $\mathcal{B}_2$ provides $\mathcal{A}$ with a perfect simulation of games $\mathbf{G}_3$. If b equals 1 it provides $\mathcal{A}$ with a perfect simulation of game $\mathbf{G}_4$. Hence Equation (4) follows.

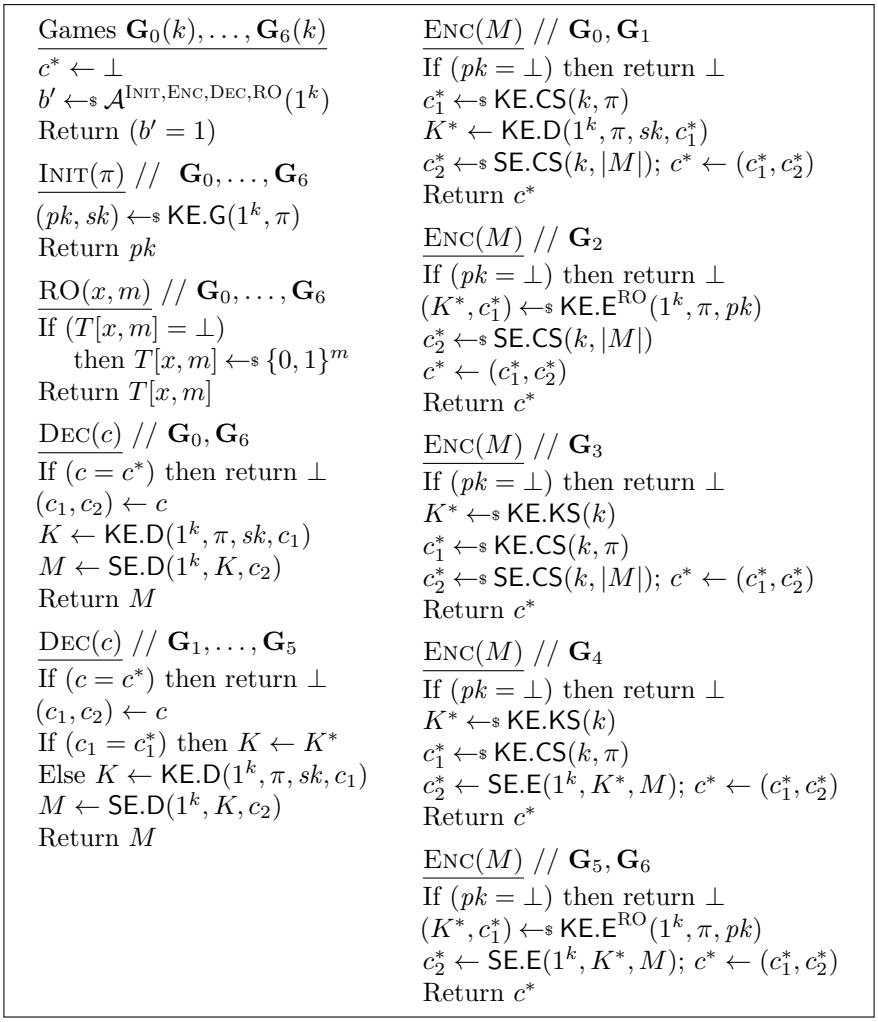


Figure 9: Games for the proof of Proposition 3.3.

We continue by giving an adversary $\mathcal{B}_1''$ such that

$$\Pr[\mathbf{G}_5(k)] - \Pr[\mathbf{G}_4(k)] = \mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}_1''}(k) \ . \tag{5}$$

The definition of adversary $\mathcal{B}_1''$ may be found in Figure 10. If the challenge bit b in game $\mathbf{G}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{B}_1''}(k)$ equals 0, adversary $\mathcal{B}_1''$ provides $\mathcal{A}$ with a perfect simulation of game $\mathbf{G}_4$. If b equals 1, it provides $\mathcal{A}$ with a perfect simulation of game $\mathbf{G}_5$. Equation (5) follows.

Note that games $\mathbf{G}_5$ and $\mathbf{G}_6$ only differ in the way decryption queries having first ciphertext component $c_1 = c_1^*$ are answered. The games only differ if c_1^* does not decrypt to K^*. Hence

$$\Pr[\mathbf{G}_6(k)] - \Pr[\mathbf{G}_5(k)] \leq \mathsf{KE.de}(k) \ . \tag{6}$$

Combining Equations (1) to (6) and defining $\mathcal{B}_1$ to be the adversary that flips a coin and then either runs $\mathcal{B}_1'$ or $\mathcal{B}_1''$ yields the claim of the proposition. □

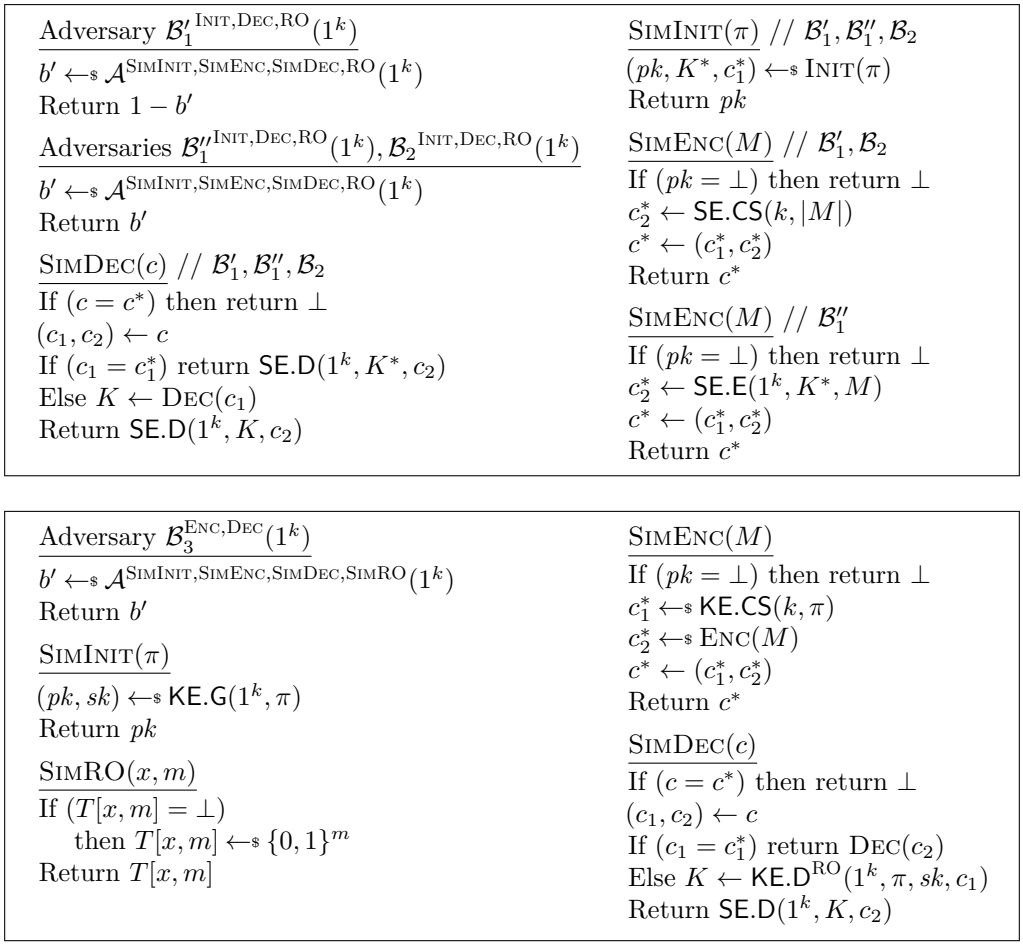


Figure 10: Adversaries for the proof of Proposition 3.3.

4 KEMs from efficiently embeddable groups

In this section we define efficiently embeddable group families (eeg). We define the security notion of pseudorandom embeddings under parameter subversion attacks (EPR-PSA) and adapt the computational Diffie-Hellman problem (CDH-PSA) and the strong computational Diffie-Hellman problem (sCDH-PSA) to the setting of efficiently embeddable group families and parameter subversion. Further we give two generic constructions of key encapsulation mechanisms from eeg families. The first construction achieves security assuming sCDH-PSA and EPR-PSA, the second requires only CDH-PSA and EPR-PSA.

4.1 Efficiently embeddable group families

Efficiently embeddable group families. Let $k \in \mathbb{N}$ denote the security parameter. An embeddable group family EG specifies the following. Parameter generation algorithm $\mathsf{EG.P}$ takes as input 1^k and returns parameters π to be utilized by all users. If $\mathsf{EG.P}$ returns ε on every input 1^k, i. e. if no parameters are used, we say that EG is *parameter-free*. Group generation algorithm $\mathsf{EG.G}$ is used to generate a group of the family. Formally, on input $1^k, \pi$ it returns a tuple $G = (\langle\mathbb{G}\rangle, n, g)$, were $\langle\mathbb{G}\rangle$ is a description of a cyclic group $\mathbb{G}$ of order n, and g is a generator of $\mathbb{G}$. $\mathsf{EG.ES}$ associates to k a finite set $\mathsf{EG.ES}(k, \pi)$ called the embedding space that

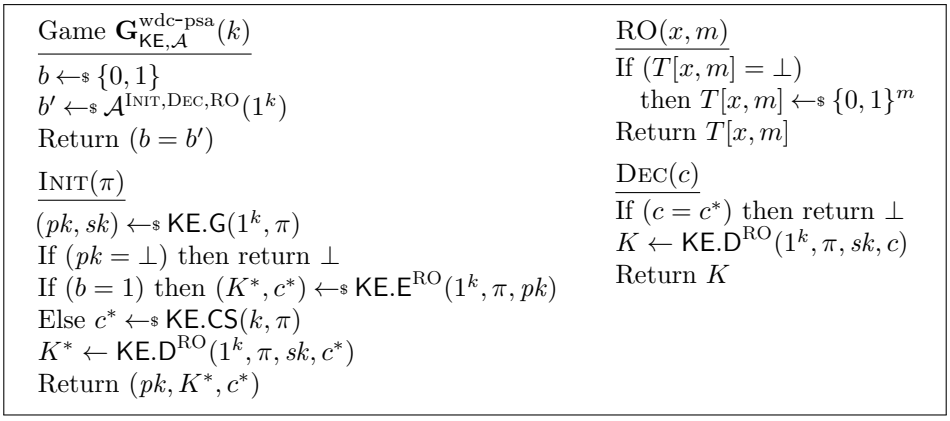


Figure 11: Games defining well-distributedness of ciphertexts of KEs.

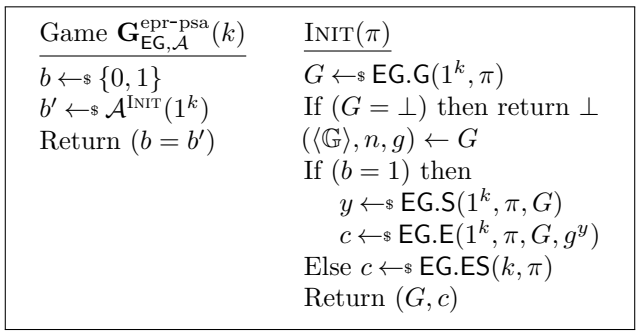


Figure 12: Game defining embedding pseudorandomness of eeg family EG.

is only dependent on k and π. Sampling algorithm $\mathsf{EG.S}$ is used to sample exponents for the group generator. Formally, it receives as input $1^k, \pi$ and $G \in [\mathsf{EG.G}(1^k, \pi)]$ and outputs $y \in \mathbb{Z}_n$. (We do *not* require y to be uniformly distributed.) Embedding algorithm $\mathsf{EG.E}$ is used to embed group elements into the embedding space. It receives as input 1^k, π, $G \in [\mathsf{EG.G}(1^k, \pi)]$ and $h \in \mathbb{G}$ and returns an element $c \in \mathsf{EG.ES}(k, \pi)$. Deterministic inversion algorithm $\mathsf{EG.I}$ is used to invert the embedding. formally, on input of 1^k, π, $G \in [\mathsf{EG.G}(1^k, \pi)]$ and $c \in \mathsf{EG.ES}(k, \pi)$ it returns an element of $\mathbb{G}$. For correctness we require that

$$\Pr\Big[\mathsf{EG.I}(1^k, \pi, G, c) = g^y\Big] \geq 1 - \mathsf{EG.ie}(k)$$

holds for all $k \in \mathbb{N}$, all $\pi \in \mathsf{EG.P}(1^k)$ and all $G \in [\mathsf{EG.G}(1^k, \pi)]$, where the probability is over $y \leftarrow\!\!\$\ \mathsf{EG.S}(1^k, \pi, G)$ and $c \leftarrow\!\!\$\ \mathsf{EG.E}(1^k, \pi, G, g^y)$. $\mathsf{EG.ie} : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is called the *inversion error* of EG.

Embedding Pseudorandomness. Consider game $\mathbf{G}^{\text{epr-psa}}_{\mathsf{EG},\mathcal{A}}(k)$ of Figure 12 associated to eeg family EG, adversary $\mathcal{A}$ and security parameter k. Let

$$\mathbf{Adv}^{\text{epr-psa}}_{\mathsf{EG},\mathcal{A}}(k) = 2\Pr[\mathbf{G}^{\text{epr-psa}}_{\mathsf{EG},\mathcal{A}}(k)] - 1.$$

We say that EG has pseudorandom embeddings under parameter subversion attacks (also called EPR-PSA) if the function $\mathbf{Adv}^{\text{epr-psa}}_{\mathsf{EG},\mathcal{A},\cdot}$ is negligible for every $\mathcal{A}$. In the game, b is a challenge bit. When $b = 1$, the challenge embedding c^* is generated by sampling an exponent using $\mathsf{EG.S}$ and embedding the group generator raised to the exponent with $\mathsf{EG.E}$. If $b = 0$ the adversary is given an embedding sampled uniformly from the embedding space. Given the group and

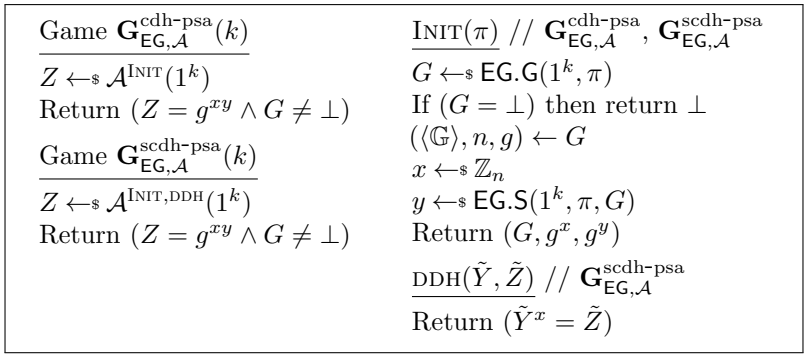


Figure 13: Experiments for the computational Diffie-Hellman problem and the strong computational Diffie-Hellman problem with respect to eeg family EG. In both games oracle INIT may be queried only once and in the sCDH game it has to be queried before using oracle DDH.

---

the embedding, the adversary outputs a guess b' and wins if b' equals b. The parameters used in the game are provided by the adversary making a single call to the oracle INIT. All of our instantiations sample exponents such that the resulting embeddings are statistically close to uniform on $\mathsf{EG.ES}(k, \pi)$, and hence achieve this notion statistically.

COMPUTATIONAL PROBLEMS ASSOCIATED TO EG. The computational Diffie-Hellman problem for a cyclic group $\mathbb{G}$ of order n, which is generated by g, asks to compute g^{xy} given g^x and g^y, where $x, y \leftarrow\!\!\$\ \mathbb{Z}_n$. In the strong computational Diffie-Hellman problem introduced by Abdalla *et al.* in [ABR01] the adversary additionally has access to an oracle, which may be used to check whether $Y^x = Z$ for group elements $Y, Z \in \mathbb{G}$. We give definitions for the (strong) computational Diffie-Hellman problem with respect to eeg families EG, which allow parameter subversion. An additional difference is that y is not chosen uniformly from $\mathbb{Z}_n$ but instead sampled using $\mathsf{EG.S}$.

Thus, consider games $\mathbf{G}^{\text{cdh-psa}}_{\mathsf{EG},\mathcal{A}}(k)$ and $\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG},\mathcal{A}}(k)$ of Figure 13. The games are associated to eeg family EG, adversary $\mathcal{A}$ and security parameter k. In both games the adversary has access to an oracle INIT setting up a problem instance according to the parameters it is provided. The oracle may be queried only once. In game $\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG},\mathcal{A}}(k)$ we require that INIT is queried before using DDH. Let

$$\mathbf{Adv}^{\text{cdh-psa}}_{\mathsf{EG},\mathcal{A}}(k) := \Pr\left[\mathbf{G}^{\text{cdh-psa}}_{\mathsf{EG},\mathcal{A}}(k)\right] \ , \quad \mathbf{Adv}^{\text{scdh-psa}}_{\mathsf{EG},\mathcal{A}}(k) := \Pr\left[\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG},\mathcal{A}}(k)\right].$$

We say that the computational Diffie-Hellman problem under parameter subversion (also called CDH-PSA) is hard with respect to EG if $\mathbf{Adv}^{\text{cdh-psa}}_{\mathsf{EG},\mathcal{A}}(\cdot)$ is negligible for every adversary $\mathcal{A}$ and that the strong computational Diffie-Hellman problem under parameter subversion (also called sCDH-PSA) is hard with respect to EG if $\mathbf{Adv}^{\text{scdh-psa}}_{\mathsf{EG},\mathcal{A}}(\cdot)$ is negligible for every adversary $\mathcal{A}$.

4.2 Key encapsulation from efficiently embeddable groups

Below, we give two generic constructions of a key encapsulation mechanism from an eeg family EG. The security of the first construction is based on the strong Diffie-Hellman problem. I.e. if sCDH-PSA is hard with respect to EG, the KEM is IND-PSA. If additionally EG has pseudorandom embeddings, the KEM has pseudorandom and well-distributed ciphertexts.

The construction is similar to the standard El Gamal based key encapsulation mechanism as for example used in [ABR01, CS03]. As an intermediate step in the proof that the construction is CPR-PSA we obtain that it is IND-PSA. The proof of this property follows the outlines of the proofs given in [ABR01, CS03]. Afterwards we use the pseudorandomness of the eeg

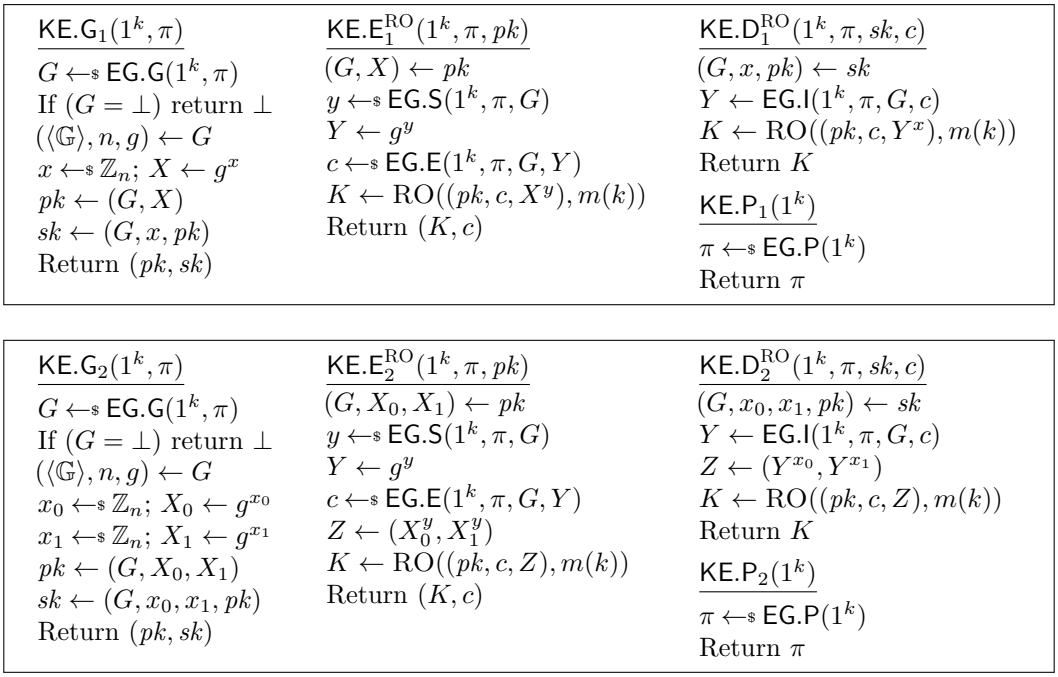


Figure 14: KEMs $\mathsf{KE}_1 = \mathbf{eegToKE1}[\mathsf{EG}, m]$ and $\mathsf{KE}_2 = \mathbf{eegToKE2}[\mathsf{EG}, m]$ built from eeg family EG and polynomial m via our transform. Both KEs have key space $\mathsf{KE.KS}(k) = \{0,1\}^{m(k)}$ and ciphertext space $\mathsf{KE.CS}(k,\pi) = \mathsf{EG.ES}(k,\pi)$.

---

family's embeddings to show, that our construction achieves pseudorandom and well-distributed ciphertexts.

The second construction uses the twin Diffie-Hellman technique introduced in [CKS08] to achieve security under the weaker CDH-PSA-assumption. It is applicable to eeg families consisting of groups, which orders do not have small prime factors.

Construction 1. Formally we define a transform **eegToKE1** that associates to an eeg family EG and a polynomial $m : \mathbb{N} \to \mathbb{N}$ a KEM $\mathsf{KE} = \mathbf{eegToKE1}[\mathsf{EG}, m]$. The parameter generation, key generation, encryption and decryption algorithms of KE are in Figure 14. The construction is in the ROM, so that encryption and decryption invoke the RO oracle. The key space is $\mathsf{KE.KS}(k) = \{0,1\}^{m(k)}$. The ciphertext space $\mathsf{KE.CS}(k,\pi) = \mathsf{EG.ES}(k,\pi)$ is the embedding space of EG. It is easy to verify that $\mathsf{KE.de} = \mathsf{EG.ie}$, meaning the decryption error of the KEM equals the inversion error of the eeg family.

Construction 2. The second construction defines a transform **eegToKE2** that associates to an eeg family EG and a polynomial $m : \mathbb{N} \to \mathbb{N}$ a KEM $\mathsf{KE} = \mathbf{eegToKE2}[\mathsf{EG}, m]$. KE's algorithms may be found in Figure 14. As in the first construction the key encapsulation mechanism has key space $\mathsf{KE.KS}(k) = \{0,1\}^{m(k)}$, ciphertext space $\mathsf{KE.CS}(k,\pi) = \mathsf{EG.ES}(k,\pi)$ and decryption error equal to EG's inversion error. Again the construction is in the random oracle model.

Security of the constructions. The following says that if sCDH-PSA is hard with respect to eeg family EG then $\mathbf{eegToKE1}[\mathsf{EG}, m]$ has desirable security properties.

Theorem 4.1. *Let* $\mathsf{KE} = \mathbf{eegToKE1}[\mathsf{EG}, m]$ *be the KEM associated to eeg family* EG *and polynomial* $m : \mathbb{N} \to \mathbb{N}$ *as defined in Figure 14. Assume that* EG *is* EPR-PSA *and that* sCDH-PSA *is hard with respect to* EG. *Then*

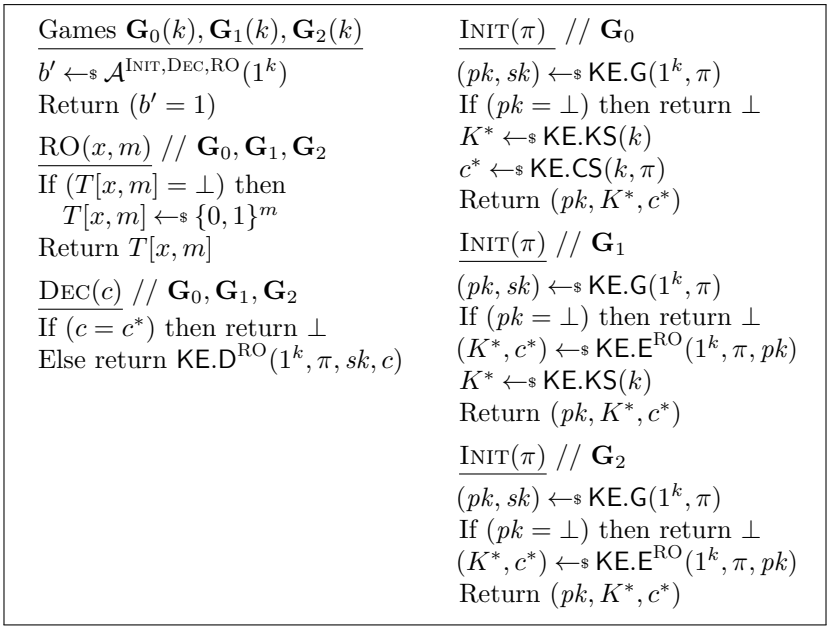


Figure 15: Games for the proof of Theorem 4.1 (i).

(i) KE *has pseudorandom ciphertexts under parameter subversion attacks.*

(ii) KE *has well-distributed ciphertexts under parameter subversion attacks.*

Moreover, if EG *is parameter-free so is* KE*. Concretely, given an adversary* $\mathcal{A}$ *making at most* $q(k)$ *queries to* RO *the proof specifies adversaries* $\mathcal{B}_1$ *and* $\mathcal{B}_2$ *having the same running time as* $\mathcal{A}$ *satisfying*

$$\mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{KE}}(\mathcal{A})(k) \leq \mathbf{Adv}^{\text{scdh-psa}}_{\mathsf{EG},\mathcal{B}_1}(k) + \mathbf{Adv}^{\text{epr-psa}}_{\mathsf{EG},\mathcal{B}_2}(k) \ ,$$

where $\mathcal{B}_2$ *makes at most* $q(k)$ *queries to* DDH*. Furthermore given an adversary* $\mathcal{A}'$ *the proof specifies an adversary* $\mathcal{B}'$ *having the same running time as* $\mathcal{A}'$ *such that,*

$$\mathbf{Adv}^{\text{wdc-psa}}_{\mathsf{KE},\mathcal{A}'}(k) \leq \mathbf{Adv}^{\text{epr-psa}}_{\mathsf{EG},\mathcal{B}'}(k) + \mathsf{EG.ie}(k) \ .$$

Proof. Let $k \in \mathbb{N}$ and $\mathcal{A}$ be an adversary against the ciphertext pseudorandomness game defined in Figure 4 making at most $q(k)$ random oracle queries. Consider the sequence of games $\mathbf{G}_0, \mathbf{G}_1, \mathbf{G}_2$ of Figure 15. By definition of the games

$$\mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{A}}(k) = \Pr[\mathbf{G}_2(k)] - \Pr[\mathbf{G}_0(k)].$$

To prove the theorem we construct adversaries $\mathcal{B}_0, \mathcal{B}_1$ such that $\Pr[\mathbf{G}_2(k)] - \Pr[\mathbf{G}_1(k)] \leq \mathbf{Adv}^{\text{scdh-psa}}_{\mathsf{EG},\mathcal{B}_0}(k)$ and $\Pr[\mathbf{G}_1(k)] - \Pr[\mathbf{G}_0(k)] = \mathbf{Adv}^{\text{epr-psa}}_{\mathsf{EG},\mathcal{B}_1}$. Plugging both equations into the equation from above yields the claim. First we prove the bound on $\Pr[\mathbf{G}_2(k)] - \Pr[\mathbf{G}_1(k)]$. Note that the answer (pk, K^*, c^*), which $\mathcal{A}$ receives as response to its call to Init, has the same distribution in $\mathbf{G}_1$ and $\mathbf{G}_2$. Furthermore from $\mathcal{A}$'s view the games are equally distributed until it queries its random oracle for $\text{RO}((pk, c^*, g^{xy}), m(k)) = \mathsf{KE.D}(1^k, \pi, sk, c^*)$. Denote by $\mathcal{Q}$ the event that $\mathcal{A}$ queries RO on this input. Since $\mathbf{G}_1$ and $\mathbf{G}_2$ are equal in distribution until $\mathcal{Q}$ occurs, the probability of $\mathcal{Q}$ is the same in both $\mathbf{G}_1$ and $\mathbf{G}_2$. This implies

$$\Pr[\mathbf{G}_2(k)] - \Pr[\mathbf{G}_1(k)] = (\Pr[\mathbf{G}_2(k) \mid \mathcal{Q}] - \Pr[\mathbf{G}_1(k) \mid \mathcal{Q}]) \Pr[\mathcal{Q}] \leq \Pr[\mathcal{Q}].$$

We construct an adversary $\mathcal{B}_0$ against sCDH-PSA providing $\mathcal{A}$ with a perfect simulation of games $\mathbf{G}_0$ and $\mathbf{G}_1$ until $\mathcal{A}$ queries for $\text{RO}((pk, c^*, g^{xy}), m(k))$. $\mathcal{B}_0$ furthermore returns a valid

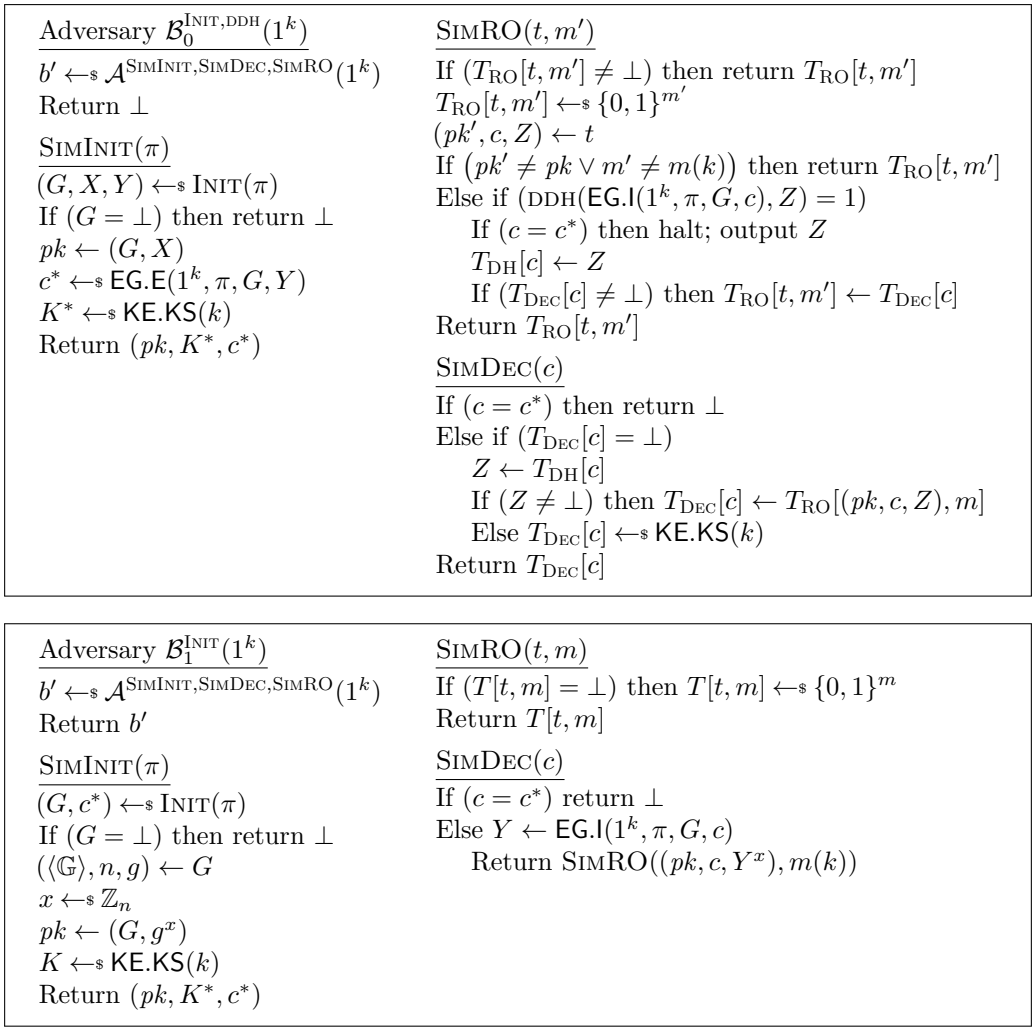


Figure 16: Adversaries for the proof of Theorem 4.1.

solution to its sCDH-PSA-challenge exactly if $\mathcal{Q}$ occurs. This implies $\mathbf{Adv}^{\text{scdh-psa}}_{\mathsf{EG},\mathcal{B}_0}(k) = \Pr[\mathcal{Q}]$. The definition of $\mathcal{B}_0$ may be found in Figure 16.

$\mathcal{B}_0$ provides $\mathcal{A}$ with a perfect simulation SimInit of oracle Init for both $\mathbf{G}_1$ and $\mathbf{G}_2$. $\mathcal{B}_0$ is able to detect whether $\mathcal{Q}$ occurs using its oracle DDH and in this case returns a valid answer to its sCDH-PSA-challenge. $\mathcal{B}_0$ furthermore uses DDH to provide $\mathcal{A}$ with perfect simulations SimDec, SimRO of the oracles Dec and RO. Note that $\mathcal{B}_0$ queries its DDH-oracle at most once to respond a SimRO-query of $\mathcal{A}$. Summing up we constructed an adversary $\mathcal{B}_0$ against $\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG},\mathcal{B}}(k)$ having the same running time as $\mathcal{A}$, making at most $q(k)$ queries to DDH, which furthermore satisfies $\Pr[\mathbf{G}_2(k)] - \Pr[\mathbf{G}_1(k)] \leq \Pr[\mathcal{Q}] = \mathbf{Adv}^{\text{scdh-psa}}_{\mathsf{EG},\mathcal{B}}(k)$.

We proceed by constructing an adversary $\mathcal{B}_1$ against game $\mathbf{G}^{\text{epr-psa}}_{\mathsf{EG},\mathcal{A}}(k)$, which satisfies $\Pr[\mathbf{G}_1(k)] - \Pr[\mathbf{G}_0(k)] = \mathbf{Adv}^{\text{epr-psa}}_{\mathsf{EG},\mathcal{A}}(k)$. The definition of $\mathcal{B}_1$ may be found in Figure 16. If the bit b in game $\mathbf{G}^{\text{epr-psa}}_{\mathsf{EG},\mathcal{A}}(k)$ equals 1, adversary $\mathcal{B}_1$ provides $\mathcal{A}$ with a perfect simulation of $\mathbf{G}_1(k)$, if $b = 0$ it provides $\mathcal{A}$ with a perfect simulation of $\mathbf{G}_0(k)$. We obtain

$$\mathbf{Adv}^{\text{epr-psa}}_{\mathsf{EG},\mathcal{A}}(k) = \Pr[\mathbf{G}_1(k)] - \Pr[\mathbf{G}_0(k)] \ ,$$

which, as pointed out above, concludes the proof of statement (i).

We now prove (ii). Consider the sequence of games of Figure 17 defined with respect to

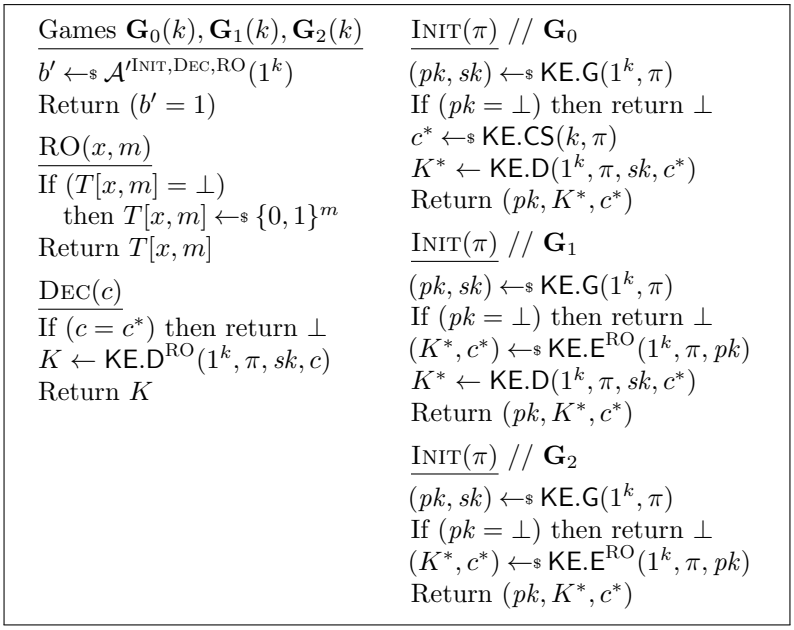


Figure 17: Games for the proof of Theorem 4.1 (ii).

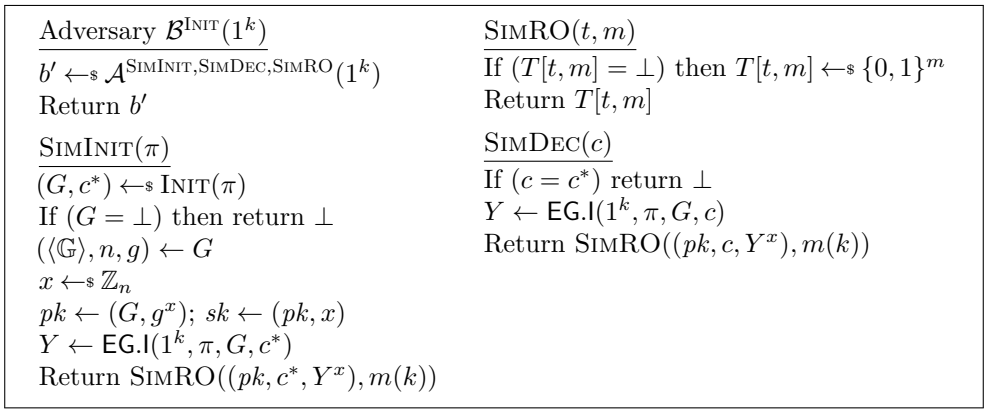


Figure 18: Adversary for the proof of Theorem 4.1 (ii).

KEM KE, adversary $\mathcal{A}'$ and security parameter k. We have

$$\mathbf{Adv}^{\text{wdc-psa}}_{\mathsf{KE},\mathcal{A}'}(k) = \Pr[\mathbf{G}_2(k)] - \Pr[\mathbf{G}_0(k)] \ .$$

Note that games $\mathbf{G}_2$ and $\mathbf{G}_1$ only differ if a decryption error occurs. Since for KE we have $\mathsf{KE.de}(k) = \mathsf{EG.ie}(k)$, we obtain

$$\Pr[\mathbf{G}_2(k)] - \Pr[\mathbf{G}_1(k)] \leq \mathsf{KE.de}(k) = \mathsf{EG.ie}(k) \ .$$

We now give an adversary $\mathcal{B}'$ satisfying

$$\Pr[\mathbf{G}_1(k)] - \Pr[\mathbf{G}_0(k)] = \mathbf{Adv}^{\text{epr-psa}}_{\mathsf{EG},\mathcal{B}'}(k) \ .$$

Combining the two equations yields statement (ii). The definition of $\mathcal{B}'$ may be found in Figure 18. If the challenge bit b in $\mathbf{G}^{\text{epr-psa}}_{\mathsf{EG},\mathcal{B}'}(k)$ equals 1, $\mathcal{B}'$ provides $\mathcal{A}'$ with a perfect simulation of game $\mathbf{G}_1$, if b equals 0 it provides $\mathcal{A}'$ with a perfect simulation of game $\mathbf{G}_0$. This implies $\Pr[\mathbf{G}_1(k)] - \Pr[\mathbf{G}_0(k)] = \mathbf{Adv}^{\text{epr-psa}}_{\mathsf{EG},\mathcal{B}'}(k)$ as desired.

□

Game $\mathbf{G}^{\text{twin-scdh-psa}}_{\mathsf{EG},\mathcal{A}}(k)$	$\underline{\textsc{Init}(\pi)}$
$(Z_0, Z_1) \leftarrow\!\!\$\ \mathcal{A}^{\textsc{Init},\textsc{ddh}}(1^k)$	$G \leftarrow\!\!\$\ \mathsf{EG.G}(1^k, \pi)$
Return $(Z_0 = g^{x_0 y} \wedge Z_1 = g^{x_1 y})$	If $(G = \bot)$ then return $\bot$
$\underline{\textsc{ddh}(\tilde{Y}, \tilde{Z}_0, \tilde{Z}_1)}$	$(\langle\mathbb{G}\rangle, n, g) \leftarrow G$
Return $(\tilde{Y}^{x_0} = \tilde{Z}_0 \wedge \tilde{Y}^{x_1} = \tilde{Z}_1 \wedge G \neq \bot)$	$x_0 \leftarrow\!\!\$\ \mathbb{Z}_n$; $x_1 \leftarrow\!\!\$\ \mathbb{Z}_n$
	$y \leftarrow\!\!\$\ \mathsf{EG.S}(1^k, \pi, G)$
	Return $(G, g^{x_0}, g^{x_1}, g^y)$

Figure 19: Experiment for the strong twin Diffie-Hellman problem with respect to eeg family EG. Oracle INIT may be queried only once and it has to be queried before using oracle DDH.

Our second construction used with an appropriate eeg family achieves security under the weaker CDH-PSA-assumption.

Theorem 4.2. *Let $m, \eta\colon \mathbb{N} \to \mathbb{N}$ be polynomials and EG an eeg family, such that for all k, all $\pi \in \mathsf{EG.P}(1^k)$ and all $G = (\langle\mathbb{G}\rangle, n, g) \in [\mathsf{EG.G}(1^k, \pi)]$ the smallest prime factor of n is larger than $2^{\eta(k)}$. Furthermore let $\mathsf{KE} = \mathbf{eegToKE2}[\mathsf{EG}, m]$ be the KEM associated to EG and polynomial $\ell : \mathbb{N} \to \mathbb{N}$ as defined in Figure 14. Assume that EG is* EPR-PSA *and that* CDH-PSA *is hard with respect to EG.*

(i) KE has pseudorandom ciphertexts under parameter subversion attacks.

(ii) KE has well-distributed ciphertexts under parameter subversion attacks.

Moreover, if EG is parameter-free so is KE. Concretely, given an adversary $\mathcal{A}$ making at most $q(k)$ queries to RO *there exist adversaries $\mathcal{B}_1$, $\mathcal{B}_2$ having the same running time as $\mathcal{A}$ making at most $q(k)$ queries to* DDH *satisfying*

$$\mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{KE}}(\mathcal{A})(k) \leq \mathbf{Adv}^{\text{cdh-psa}}_{\mathsf{EG},\mathcal{B}_1}(k) + \mathbf{Adv}^{\text{epr-psa}}_{\mathsf{EG},\mathcal{B}_2}(k) + \frac{q(k)}{2^{\eta(k)}} \ .$$

Furthermore given an adversary $\mathcal{A}'$ the proof specifies an adversary $\mathcal{B}'$ having the same running time as $\mathcal{A}'$ such that,

$$\mathbf{Adv}^{\text{wdc-psa}}_{\mathsf{KE},\mathcal{A}'}(k) \leq \mathbf{Adv}^{\text{epr-psa}}_{\mathsf{EG},\mathcal{B}'}(k) + \mathsf{EG.ie}(k) \ .$$

The proof of the theorem is analogous to the proof of Theorem 4.1 but additionally relies on the twin-Diffie-Hellman technique from [CKS08]. We give a sketch of the proof below:

In [CKS08] the authors show that for groups of prime order the computational Diffie-Hellman assumption is equivalent to the strong twin Diffie-Hellman assumption. In this variant of the strong Diffie-Hellman problem an adversary given group elements g^y, g^{x_0}, g^{x_1} has to compute $g^{x_0 y}$ and $g^{x_1 y}$. We give an adaption of this problem to the setting of eeg families and parameter subversion in Figure 19. Thus for security parameter k, eeg family EG and adversary $\mathcal{A}$ consider game $\mathbf{G}^{\text{twin-scdh-psa}}_{\mathsf{EG},\mathcal{A}}(k)$ of Figure 19. The adversary has access to oracles INIT and DDH, where we require that INIT is queried only once and that it is queried before using DDH. Let

$$\mathbf{Adv}^{\text{twin-scdh-psa}}_{\mathsf{EG},\mathcal{A}}(k) := \Pr\Big[\mathbf{G}^{\text{twin-scdh-psa}}_{\mathsf{EG},\mathcal{A}}(k)\Big].$$

We say the strong twin Diffie-Hellman problem under parameter subversion (also called twinsCDH-PSA) is hard with respect to EG if $\mathbf{Adv}^{\text{twin-scdh-psa}}_{\mathsf{EG},\mathcal{A},k}$ is negligible for every adversary $\mathcal{A}$.

Theorem 3 of [CKS08] states that for groups of prime order the computational Diffie-Hellman assumption is equivalent to the strong twin Diffie-Hellman assumption. It is easily adapted to the setting of eeg families consisting of groups of (potentially) composite order:

Lemma 4.3. *Let $\eta\colon \mathbb{N} \to \mathbb{N}$ and EG be an eeg family, such that for all $\pi \in [\mathsf{EG.P}(1^k)]$ and all $G = (\langle\mathbb{G}\rangle, n, g) \in [\mathsf{EG.G}(1^k, \pi)]$ the smallest prime factor of n is larger than $2^{\eta(k)}$. Further let $\mathcal{A}$ an adversary against* twinsCDH-PSA *with respect to EG making at most $q(k)$ queries to its* DDH*-oracle. Then there exists an adversary $\mathcal{B}$ against* CDH-PSA *with respect to EG, which has the same running time as $\mathcal{A}$ and furthermore satisfies*

$$\mathbf{Adv}^{\text{twin-scdh-psa}}_{\mathsf{EG},\mathcal{A}}(k) \leq \mathbf{Adv}^{\text{cdh-psa}}_{\mathsf{EG},\mathcal{B}}(k) + \frac{q(k)}{2^{\eta(k)}}.$$

Using this result Theorem 4.2 can be derived as follows. Analogously to the proof of Theorem 4.1 it is possible to construct from adversary $\mathcal{A}$ an adversary $\mathcal{B}'_1$ against twinsCDH-PSA running in the same time as $\mathcal{A}$, making at most $q(k)$ queries to DDH and an adversary $\mathcal{B}_2$ against the embedding pseudorandomness game such that

$$\mathbf{Adv}^{\text{cpr-psa}}_{\mathsf{KE},\mathcal{A}}(k) \leq \mathbf{Adv}^{\text{twin-scdh-psa}}_{\mathsf{EG},\mathcal{B}'_1}(k) + \mathbf{Adv}^{\text{epr-psa}}_{\mathsf{EG},\mathcal{B}_2}(k) \ .$$

Now an application of Lemma 4.3 yields the statement *i*). Statement *ii*) can be shown as in the proof of Theorem 4.1.

5 Efficiently embeddable group families from curve-twist pairs

In this section we give instantiations of eeg-families based on elliptic curves. The main tool of the constructions is a bijection of [Kal91] mapping points of an elliptic curve and its quadratic twist to an interval of integers. We first give a construction using parameters, the parameter being a prime p of length k serving as the modulus of the prime field the curves are defined over. The construction has embedding space $[2p + 1]$. Since we assume, that the parameter shared by all users might be subject to subversion, security of this construction corresponds to the assumption that there exist no inherently bad choices for p, i.e. that for *any* sufficiently large prime p it is possible to find elliptic curves defined over $\mathbb{F}_p$ on which the computational Diffie-Hellman assumption holds.

As an alternative we also give parameter-free eeg-families whose security is based on the weaker assumption that for *random* k-bit prime p it is possible to find elliptic curves defined over $\mathbb{F}_p$, such that the computational Diffie-Hellman assumption holds. Since in this construction the modulus p is sampled along with the curve, it is no longer possible to use $[2p + 1]$ as the embedding space of the eeg family. We propose two solutions to overcome this, one using rejection sampling to restrict the embedding space to the set $[2^k]$, the other one is based on a technique from [HOT04] and expands the embedding space to $[2^{k+1}]$.

5.1 Elliptic curves

Let $p \geq 5$ be prime and $\mathbb{F}_p$ a field of order p. An elliptic curve over $\mathbb{F}_p$ can be expressed in short Weierstrass form, that is as the set of projective solutions of an equation of the form

$$YZ^2 = X^3 + aXZ^2 + bZ^3,$$

where $a, b \in \mathbb{F}_p$ with $4a^3 + 27b^2 \neq 0$. We denote the elliptic curve generated by p, a, b by $E(p, a, b)$. $E(p, a, b)$ possesses exactly one point with Z-coordinate 0, the so called point at infinity $\mathcal{O} = (0 : 1 : 0)$. After normalizing by $Z = 1$ the curve's other points can be interpreted as the solutions $(x, y) \in \mathbb{F}_p^2$ of the affine equation $y^2 = x^3 + ax + b$. It is possible to establish an efficiently computable group law on $E(p, a, b)$ with $\mathcal{O}$ serving as the neutral element of the group. We use multiplicative notation for the group law to be consistent with the rest of the paper.

Twists of Elliptic Curves. In [Kal91, section 4] Kaliski establishes the following one-to-one correspondence between two elliptic curves defined over $\mathbb{F}_p$ which are related by twisting and a set of integers.

Lemma 5.1. *Let $p \in \mathbb{N}_{\geq 5}$ be prime. Let $u \in \mathbb{Z}_p$ be a quadratic nonresidue modulo p and $a, b \in \mathbb{Z}_p$ such that $4a^3 + 27b^2 \neq 0$. Consider the elliptic curves $E_0 := E(p, a, b)$ and $E_1 := E(p, au^2, bu^3)$. Then $|E_0| + |E_1| = 2p+2$. Furthermore, the functions $l_0 : E_0 \longrightarrow [2p+2]$ and $l_1 : E_1 \longrightarrow [2p+2]$ defined as*

$$l_0(P) = \begin{cases} p & \text{if } P = \mathcal{O}_0 \\ (ux \bmod p) & \text{if } (P = (x,y)) \wedge (0 \leq y \leq (p-1)/2) \\ (ux \bmod p) + p + 1 & \text{if } (P = (x,y)) \wedge ((p-1)/2 < y) \end{cases},$$

and

$$l_1(P) = \begin{cases} 2p+1 & \text{if } P = \mathcal{O}_1 \\ x & \text{if } (P = (x,y)) \wedge (0 < y \leq (p-1)/2) \\ x + p + 1 & \text{if } (P = (x,y)) \wedge ((y = 0) \vee ((p-1)/2 < y)) \end{cases}$$

are injective with nonintersecting ranges, where $\mathcal{O}_0$ and $\mathcal{O}_1$ denote the neutral elements of E_0 and E_1 respectively.

Lemma 5.2. *The functions l_0 and l_1 can be efficiently inverted. That is, given $z \in [2p+1]$, one can efficiently compute the unique $(P, \delta) \in E_0 \cup E_1 \times \{0,1\}$ such that $l_\delta(P) = z$.*

Proof. Note that $z \in [p]$ satisfies $z \in \mathrm{im}(l_0)$ exactly if $(u^{-1}x)^3 + au^{-1}x + b$ is a square modulo p. Further for $z \in \{p+1, \ldots, 2p\}$ we have $z \in \mathrm{im}(l_1)$ exactly if $u^3\left((z-p-1)^3 + a(z-p-1) + b\right)$ is a square modulo p. Hence for all elements of $[2p+2]$ its is possible to efficiently determine, whether they lie in $\mathrm{im}(l_0)$ or $\mathrm{im}(l_1)$. Furthermore both l_0 and l_1 can be efficiently inverted by additions and multiplications modulo p. More precisely, let $z \in \mathrm{im}(l_0)\backslash\{p\}$. If $z < p$ we are able to recover its preimage as $l_0^{-1}(z) = (u^{-1}z, y)$, where y is the unique solution of the equation $y^2 = ((u^{-1}z)^3 + au^{-1}z + b)$ in $\mathbb{Z}_p$, which furthermore satisfies $y \leq (p-1)/2$. On the other hand, if $z > p$ we have $l_0^{-1}(z) = (u^{-1}(z-p-1), y)$, where y is the unique solution of the equation $y^2 = ((u^{-1}(z-p-1))^3 + au^{-1}(z-p-1) + b)$in $\mathbb{Z}_p$, which furthermore satisfies $y > (p-1)/2$.

Analogously for $z \in \mathrm{im}(l_1)\backslash\{2p+1\}$ with $z < p$ we have $l_1^{-1}(z) = (z, y)$, where y is the unique solution of the equation $(y^2 = z^3 + az + b)$in $\mathbb{Z}_p$, which satisfies $y \leq (p-1)/2$. Finally let $z \in \mathrm{im}(l_1) \setminus \{2p+1\}$ with $z < p$. Then $l_1^{-1}(z) = (z-p-1, y)$, where y is the unique solution of the equation $y^2 = ((z-p-1)^3 + a(z-p-1) + b)$in $\mathbb{Z}_p$, which satisfies $y > (p-1)/2$. □

Definition 5.3. *A curve-twist generator* TGen *on input of security parameter 1^k and a k-bit prime p returns (G_0, G_1), where $G_0 = (\langle E_0 \rangle, n_0, g_0)$ and $G_1 = (\langle E_1 \rangle, n_1, g_1)$ are secure cyclic elliptic curves defined over the field $\mathbb{F}_p$. More precisely we require $E_0 := E(p, a, b)$ and $E_1 := E(p, au^2, bu^3)$ for $a, b \in \mathbb{F}_p$ such that $(4a^3 + 27b^2) \neq 0$ and quadratic nonresidue u. Furthermore we require that g_0 generates E_0 and g_1 generates E_1 as well as $|E_0| = n_0$, $|E_1| = n_1$ and $\gcd(n_0, n_1) = 1$.*

Generation of secure Twisted Elliptic Curves. There exist several proposals for properties an elliptic curve over a prime field $\mathbb{F}_p$ should have to be considered secure (e.g., [BL, FPRE15]). Firstly, the elliptic curve's order is required to be either the product of a big prime and a small cofactor — or preferably prime. Secondly, several conditions preventing the transfer of discrete logarithm problems on the curve to groups, where faster algorithms to

compute discrete logarithms may be applied, should be fulfilled. Finally, for our applications we need both the elliptic curve and its quadratic twist to be secure, a property usually called twist security. For concreteness, we suggest to implement $\mathsf{TGen}(1^k, p)$ by sampling the necessary parameters a, b, u with rejection sampling such that the resulting curve $E(p, a, b)$ fulfills the three security requirement mentioned above. This way, TGen can be implemented quite efficiently[1] and furthermore, with overwhelming probability, the resulting curve fulfills all relevant security requirements from [BL, FPRE15] that are not covered by the three security properties explicitly mentioned above.

Computational problems associated to TGen. Let TGen a curve-twist generator. We give two versions of the (strong) computational Diffie-Hellman assumption with respect to TGen. In the first version the prime p on which TGen is invoked is chosen by the adversary, while in the second version p is sampled uniformly at random from all k-bit primes. For $d \in \{0, 1\}$ consider games $\mathbf{G}^{\text{twist}_d\text{-cp-cdh}}_{\mathsf{TGen},\mathcal{A}}(\cdot)$, $\mathbf{G}^{\text{twist}_d\text{-cp-scdh}}_{\mathsf{TGen},\mathcal{A}}(\cdot)$, $\mathbf{G}^{\text{twist}_d\text{-up-cdh}}_{\mathsf{TGen},\mathcal{A}}(\cdot)$, $\mathbf{G}^{\text{twist}_d\text{-up-scdh}}_{\mathsf{TGen},\mathcal{A}}(\cdot)$ of Figure 20. We define advantage functions

$$\begin{aligned}
\mathbf{Adv}^{\text{twist}_d\text{-cp-cdh}}_{\mathsf{TGen},\mathcal{A}}(k) &= \Pr\left[\mathbf{G}^{\text{twist}_d\text{-cp-cdh}}_{\mathsf{TGen},\mathcal{A}}(k)\right] , \\
\mathbf{Adv}^{\text{twist}_d\text{-cp-scdh}}_{\mathsf{TGen},\mathcal{A}}(k) &= \Pr\left[\mathbf{G}^{\text{twist}_d\text{-cp-scdh}}_{\mathsf{TGen},\mathcal{A}}(k)\right] , \\
\mathbf{Adv}^{\text{twist}_d\text{-up-cdh}}_{\mathsf{TGen},\mathcal{A}}(k) &= \Pr\left[\mathbf{G}^{\text{twist}_d\text{-up-cdh}}_{\mathsf{TGen},\mathcal{A}}(k)\right] , \\
\mathbf{Adv}^{\text{twist}_d\text{-up-scdh}}_{\mathsf{TGen},\mathcal{A}}(k) &= \Pr\left[\mathbf{G}^{\text{twist}_d\text{-up-scdh}}_{\mathsf{TGen},\mathcal{A}}(k)\right] .
\end{aligned}$$

Definition 5.4. *Let TGen be a curve-twist generator. We say the computational Diffie-Hellman assumption for chosen (uniform) primes holds with respect to TGen, if both $\mathbf{Adv}^{\text{twist}_0\text{-cp-cdh}}_{\mathsf{TGen},(P_k)_k,\mathcal{A}}(\cdot)$ and $\mathbf{Adv}^{\text{twist}_1\text{-cp-cdh}}_{\mathsf{TGen},(P_k)_k,\mathcal{A}}(\cdot)$ (or $\mathbf{Adv}^{\text{twist}_0\text{-up-cdh}}_{\mathsf{TGen},(P_k)_k,\mathcal{A}}(\cdot)$ and $\mathbf{Adv}^{\text{twist}_1\text{-up-cdh}}_{\mathsf{TGen},(P_k)_k,\mathcal{A}}(\cdot)$ respectively) are negligible for all adversaries $\mathcal{A}$.*

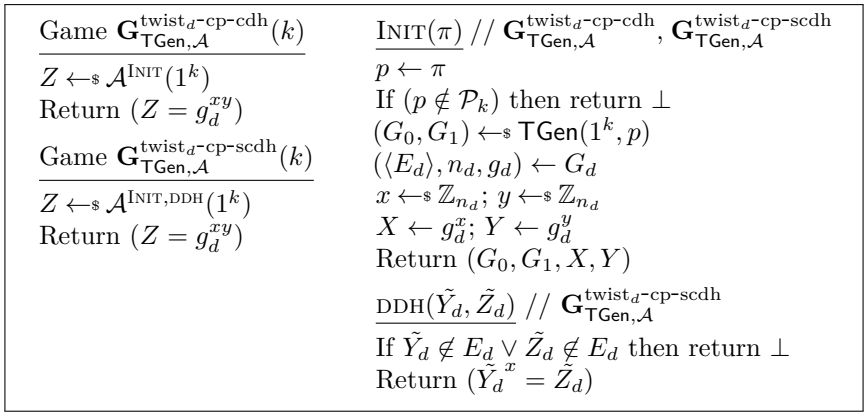
Furthermore we say the strong computational Diffie-Hellman assumption for chosen (uniform) primes holds with respect to TGen, if both $\mathbf{Adv}^{\text{twist}_0\text{-cp-scdh}}_{\mathsf{TGen},\mathcal{A}}(\cdot)$ and $\mathbf{Adv}^{\text{twist}_1\text{-cp-scdh}}_{\mathsf{TGen},\mathcal{A}}(\cdot)$ (or $\mathbf{Adv}^{\text{twist}_0\text{-up-scdh}}_{\mathsf{TGen},(P_k)_k,\mathcal{A}}(\cdot)$ and $\mathbf{Adv}^{\text{twist}_1\text{-up-scdh}}_{\mathsf{TGen},(P_k)_k,\mathcal{A}}(\cdot)$ respectively) are negligible for all adversaries $\mathcal{A}$.

5.2 An eeg family from elliptic curves

In [Kal91] Kaliski implicitly gives an eeg family based on elliptic curves. The family is parameter-using, the parameter being a prime p serving as the modulus of the field the elliptic curves are defined over. The definition of eeg family $\mathsf{EG}_{\text{twist}}$ may be found in Figure 21. Parameter generation algorithm $\mathsf{EG}_{\text{twist}}.\mathsf{P}$ on input of security parameter 1^k returns a randomly sampled k-bit prime[2] p. Group generation algorithm $\mathsf{EG}_{\text{twist}}.\mathsf{G}$ on input of parameter $\pi = p$ checks, whether p is indeed a prime of appropriate length, and —if so— runs a curve-twist generator $\mathsf{TGen}(1^k, \pi)$ to obtain the description of two cyclic secure cyclic elliptic curves $G_0 = (\langle E_0\rangle, n_0, g_0)$ and $G_1 = (\langle E_1\rangle, n_1, g_1)$. Its output is $(\langle\mathbb{G}\rangle, n, g)$, where $\mathbb{G} \leftarrow E_0 \times E_1$ is the direct product of

[1] In [GM00] Galbraith and McKee consider elliptic curves E chosen uniformly from the set of elliptic curves over a fixed prime field $\mathbb{F}_p$. They give a conjecture (together with some experimental evidence) for a lower bound on the probability of $|E|$ being prime. Using a similar technique [FPRE15] argue, that the probability of a uniformly chosen elliptic curve over a fixed prime field $\mathbb{F}_p$ to be both secure and twist secure is bounded from below by $0.5/\log^2(p)$. Since their definition of security of an elliptic curve includes primality of the curve order and since due to Lemma 5.1 the orders of curve and twist sum up to $2p + 2$, this in particular implies that the curve and its twist are cyclic and have coprime group order.

[2] In practice one would preferably instantiate $\mathsf{EG}_{\text{twist}}$ with a standardized prime.



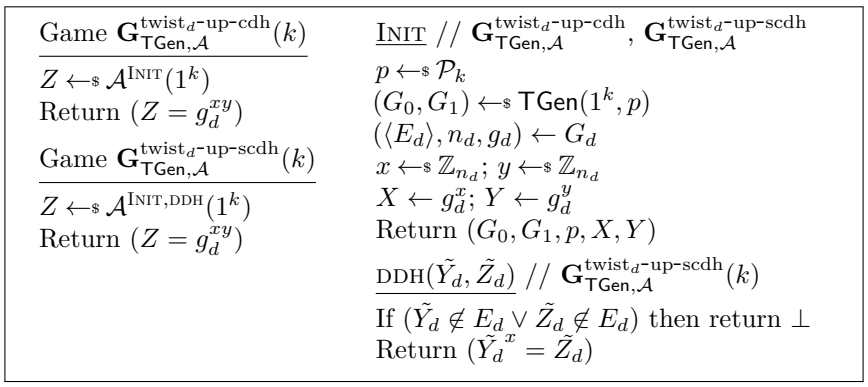


Figure 20: Experiments for the (strong) CDH problem for chosen (uniform) primes with respect to $d \in \{0, 1\}$, adversary $\mathcal{A}$ and curve-twist generator TGen.

---

the two elliptic curves, $n \leftarrow n_0 \cdot n_1$ and $g \leftarrow (g_0, g_1)$. Here we assume that the description $\langle \mathbb{G} \rangle$ of $\mathbb{G}$ includes the values n_0 and n_1, which are used by $\mathsf{EG}_{\text{twist}}$'s other algorithms. Note that $|\mathbb{G}| = n$ and since n_0 and n_1 are coprime, g generates $\mathbb{G}$. Furthermore, if we regard E_0 and E_1 as subgroups of $\mathbb{G} = E_0 \times E_1$ in the natural way, we may rewrite the set $E_0 \cup E_1 \subseteq \mathbb{G}$ as

$$\begin{aligned} E_0 \cup E_1 &= \{(h_0, \mathcal{O}_1) \mid h_0 \in E_0\} \cup \{(\mathcal{O}_0, h_1) \mid h_1 \in E_1\} \\ &= \{(g_0, g_1)^y \mid y \in \mathbb{Z}_n \colon y \equiv 0 \mod n_0 \text{ or } y \equiv 0 \mod n_1\} \end{aligned}$$

Algorithm $\mathsf{EG}_{\text{twist}}.\mathsf{S}$ uses this property to efficiently sample $y \in \mathbb{Z}_n$ such that $g^y \sim U_{E_0 \cup E_1}$. It first samples $z \leftarrow\!\!\$\ \mathbb{Z}_{2p+1}$. If $z < n_0$ it returns $\varphi_{\text{crt}}(z, 0)$. Else it returns $\varphi_{\text{crt}}(0, z - n_0 - 1)$. Here φ_{crt} denotes the canonical isomorphism $\varphi_{\text{crt}} : \mathbb{Z}_{n_0} \times \mathbb{Z}_{n_1} \to \mathbb{Z}_n$. As a result $y \leftarrow\!\!\$\ \mathsf{EG}_{\text{twist}}.\mathsf{S}(1^k, G)$ satisfies $y \sim U_M$, where $M := \{y \in \mathbb{Z}_n \mid y \equiv 0 \mod n_0 \text{ or } y \equiv 0 \mod n_1\}$. Embedding algorithm $\mathsf{EG}_{\text{twist}}.\mathsf{E}$ receives as input 1^k, π, G and $h = (h_0, h_1) \in \mathbb{G}$. It first checks, whether h lies outside of the support $[\mathsf{EG}_{\text{twist}}.\mathsf{S}(1^k, \pi, G)]$ of the sampling algorithm, i.e. whether both $h_0 \neq \mathcal{O}_0$ and $h_1 \neq \mathcal{O}_1$. In this case the element is mapped to 0. If h is an element of $[\mathsf{EG}_{\text{twist}}.\mathsf{S}(1^k, \pi, G)]$, algorithm $\mathsf{EG}_{\text{twist}}.\mathsf{E}$ returns $l_0(h_0)$ if $h_1 = \mathcal{O}_1$, and $l_1(h_1)$ if $h_1 \neq \mathcal{O}_1$. Here $l_0 \colon E_0 \to [2p+2]$ and $l_1 \colon E_1 \to [2p+2]$ denote the maps of Lemma 5.1. By Lemma 5.1 the map $\mathsf{EG}_{\text{twist}}.\mathsf{E}(1^k, G, \cdot)|_{E_0 \cup E_1}$ is a bijection between $E_0 \cup E_1$ and $[2p+1]$ and we obtain $\mathsf{EG}_{\text{twist}}.\mathsf{E}(1^k, G, g^y) \sim U_{[2p+1]}$ for y sampled with $\mathsf{EG}_{\text{twist}}.\mathsf{S}(1^k, G)$.

Let $\mathcal{A}$ be a (potentially unbounded) adversary against the embedding pseudorandomness game of Figure 12 with respect to $\mathsf{EG}_{\text{twist}}$. Denote the parameter $\mathcal{A}$ calls the procedure INIT on by π. If π is not a prime of length k, we have $\perp \leftarrow\!\!\$\ \mathsf{EG}_{\text{twist}}.\mathsf{G}(1^k, \pi)$ and procedure INIT returns

$\underline{\mathsf{EG}_{\text{twist}}.\mathsf{P}(1^k)}$
$p \leftarrow\!\!\$\ \mathcal{P}_k$
$\pi \leftarrow p$
Return π

$\underline{\mathsf{EG}_{\text{twist}}.\mathsf{G}(1^k, \pi)}$
$p \leftarrow \pi$
If $(p \notin \mathcal{P}_k)$ return $\perp$
$(G_0, G_1) \leftarrow\!\!\$\ \mathsf{TGen}(1^k, p)$
$(\langle E_0\rangle, g_0, n_0) \leftarrow G_0$; $(\langle E_1\rangle, g_1, n_1) \leftarrow G_1$
$\mathbb{G} \leftarrow E_0 \times E_1$; $g \leftarrow (g_0, g_1)$; $n \leftarrow n_0 \cdot n_1$
$G \leftarrow (\langle \mathbb{G}\rangle, n, g)$
Return G

$\underline{\mathsf{EG}_{\text{twist}}.\mathsf{S}(1^k, \pi, G)}$
$p \leftarrow \pi$
$z \leftarrow\!\!\$\ \mathbb{Z}_{2p+1}$
If $(z < n_0)$ return $\varphi_{\text{crt}}(z, 0)$
Else return $\varphi_{\text{crt}}(0, z - n_0 - 1)$

$\underline{\mathsf{EG}_{\text{twist}}.\mathsf{E}(1^k, \pi, G, (h_0, h_1))}$
If $(h_0 \neq \mathcal{O}_0 \wedge h_1 \neq \mathcal{O}_1)$ return 0
Elseif $h_1 = \mathcal{O}_1$ return $l_0(h_0)$
Else return $l_1(h_1)$

$\underline{\mathsf{EG}_{\text{twist}}.\mathsf{I}(1^k, \pi, G, z)}$
If $(z \in \text{im}(l_0))$ return $l_0^{-1}(z)$
Else return $l_1^{-1}(z)$

Figure 21: Definition of eeg family $\mathsf{EG}_{\text{twist}}$ with embedding space $\mathsf{EG}_{\text{twist}}.\mathsf{ES}(k, \pi) = [2p+1]$. l_0 and l_1 denote the maps from Lemma 5.1, φ_{crt} the canonical isomorphism $\mathbb{Z}_{n_0} \times \mathbb{Z}_{n_1} \to \mathbb{Z}_n$.

$\perp$. On the other hand if π is a prime of appropriate length, as discussed above INIT returns (G, c), where $g \leftarrow\!\!\$\ \mathsf{EG}_{\text{twist}}.\mathsf{G}(1^k, \pi)$ and $c \sim U_{[2p+1]}$ for both $b = 0$ and $b = 1$. Hence for any choice of π adversary $\mathcal{A}$'s call to INIT is answered with a response, which is independent of the challenge bit b. This implies $\mathbf{Adv}^{\text{epr-psa}}_{\mathsf{EG}_{\text{twist}},\mathcal{A}}(k) = 0$. Furthermore, for all $\pi \in [\mathsf{EG}_{\text{twist}}.\mathsf{P}(1^k)]$, $G \in [\mathsf{EG}_{\text{twist}}.\mathsf{G}(1^k, \pi)]$, $y \in [\mathsf{EG}_{\text{twist}}.\mathsf{S}(1^k, \pi, G)]$ and $c = \mathsf{EG}_{\text{twist}}.\mathsf{E}(1^k, \pi, G, g^y)$, algorithm $\mathsf{EG}_{\text{twist}}.\mathsf{I}$ efficiently reconstructs g^y from c using Lemma 5.2. Summing up we have the following.

Lemma 5.5. *EG_{twist} from Figure 21 is an eeg family with embedding space $\mathsf{EG}_{twist}.\mathsf{ES}(k, G) = [2p+1]$ and inversion error $\mathsf{EG}_{twist}.\mathsf{ie}(k) = 0$. Furthermore EG_{twist} has pseudorandom embeddings. More precisely, for every (potentially unbounded) adversary $\mathcal{A}$ we have*

$$\mathbf{Adv}^{\text{epr-psa}}_{\mathsf{EG}_{twist},\mathcal{A}}(k) = 0 \ .$$

Concerning the hardness of CDH-PSA with respect to $\mathsf{EG}_{\text{twist}}$ we obtain the following.

Lemma 5.6. *Let EG_{twist} be the embeddable group generator constructed with respect to twisted elliptic curve generator TGen as described above. If the (strong) Diffie-Hellman assumption for chosen primes holds with respect to TGen, then the (strong) Diffie-Hellman assumption holds with respect to EG_{twist}.*

Concretely for every adversary $\mathcal{A}$ against game $\mathbf{G}^{\text{cdh-psa}}_{\mathsf{EG}_{twist},\mathcal{A}}(\cdot)$ there exist adversaries $\mathcal{B}_0$, $\mathcal{B}_1$ against games $\mathbf{G}^{\text{twist}_0\text{-cp-cdh}}_{\mathsf{TGen},\mathcal{B}_0}(\cdot)$ or $\mathbf{G}^{\text{twist}_1\text{-cp-cdh}}_{\mathsf{TGen},\mathcal{B}_1}(\cdot)$ respectively satisfying

$$\mathbf{Adv}^{\text{cdh-psa}}_{\mathsf{EG}_{twist},\mathcal{A}}(k) \leq \mathbf{Adv}^{\text{twist}_0\text{-cp-cdh}}_{\mathsf{TGen},\mathcal{B}_0}(k) + \mathbf{Adv}^{\text{twist}_1\text{-cp-cdh}}_{\mathsf{TGen},\mathcal{B}_1}(k).$$

Furthermore if $\mathcal{A}$ is an adversary against $\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG}_{twist},\mathcal{A}}(\cdot)$, which makes at most Q queries to its DDH*-oracle, then there exist adversaries $\mathcal{B}_0$, $\mathcal{B}_1$ against games $\mathbf{G}^{\text{twist}_0\text{-cp-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(\cdot)$ or $\mathbf{G}^{\text{twist}_1\text{-cp-scdh}}_{\mathsf{TGen},\mathcal{B}_1}(\cdot)$ respectively making at most Q queries to their* DDH*-oracles, satisfying*

$$\mathbf{Adv}^{\text{scdh-psa}}_{\mathsf{EG}_{twist},\mathcal{A}}(k) \leq \mathbf{Adv}^{\text{twist}_0\text{-cp-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(k) + \mathbf{Adv}^{\text{twist}_1\text{-cp-scdh}}_{\mathsf{TGen},\mathcal{B}_1}(k).$$

Proof. We show the statement on sCDH-PSA. The statement on CDH-PSA can be shown analogously. Let $\mathcal{A}$ be an adversary against the sCDH-PSA game with respect to $\mathsf{EG}_{\text{twist}}$. Let

Game $\mathbf{G}_0(k)$	$\textsc{Init}(\pi)$
$Z \leftarrow\!\!\$\ \mathcal{A}^{\textsc{Init},\textsc{ddh}}(1^k)$	$p \leftarrow\!\!\$\ \pi$
Return $(Z = g^{xy} \wedge G \neq \bot)$	If $(p \notin \mathcal{P}_k)$ then return $\bot$
Game $\mathbf{G}_1(k)$	$(G_0, G_1) \leftarrow\!\!\$\ \mathsf{TGen}(1^k, p)$
$Z \leftarrow\!\!\$\ \mathcal{A}^{\textsc{Init},\textsc{ddh}}(1^k)$	$(\langle E_0\rangle, g_0, n_0) \leftarrow G_0$; $(\langle E_1\rangle, g_1, n_1) \leftarrow G_1$
Return $(Z = g^{xy} \wedge Y \in E_1 \wedge G \neq \bot)$	$\mathbb{G} \leftarrow E_0 \times E_1$; $g \leftarrow (g_0, g_1)$; $n \leftarrow n_0 \cdot n_1$
$\textsc{ddh}(\tilde{Y}, \tilde{Z})$	$G \leftarrow (\langle\mathbb{G}\rangle, n, g)$
Return $(\tilde{Y}^x = \tilde{Z})$	$x \leftarrow\!\!\$\ \mathbb{Z}_n$
	$y \leftarrow\!\!\$\ \mathsf{EG}_{\mathrm{twist}}.\mathsf{S}(1^k, \pi, G)$
	$X \leftarrow g^x$, $Y \leftarrow g^y$
	Return (G, X, Y)

Figure 22: Games for the proof of Lemma 5.6. Both games use the same procedures $\textsc{Init}$ and $\textsc{ddh}$. In $\mathbf{G}_1$ we see E_1 as a subset of $E_0 \times E_1$ in the natural way.

Adversary $\mathcal{B}_0^{\textsc{Init},\textsc{ddh}}(1^k)$	$\textsc{SimInit}(\pi)$
$(Z_0, Z_1) \leftarrow\!\!\$\ \mathcal{A}^{\textsc{SimInit},\textsc{Simddh}}(1^k)$	$p \leftarrow\!\!\$\ \pi$
Return Z_0	If $(p \notin \mathcal{P}_k)$ then return $\bot$
$\textsc{Simddh}((\tilde{Y}_0, \tilde{Y}_1), (\tilde{Z}_0, \tilde{Z}_1))$	$(G_0, G_1, X_0, Y_0) \leftarrow\!\!\$\ \textsc{Init}(\pi)$
If $(\tilde{Y}_1^{x_1} = \tilde{Z}_1)$	$(\langle E_0\rangle, n_0, g_0) \leftarrow G_0$; $(\langle E_1\rangle, n_1, g_1) \leftarrow G_1$
then return $\textsc{ddh}(\tilde{Y}_0, \tilde{Z}_0)$	$G \leftarrow (\langle E_0 \times E_1\rangle, n_0 n_1, (g_0, g_1))$
Return false	$x_1 \leftarrow\!\!\$\ \mathbb{Z}_{n_1}$; $X_1 \leftarrow g_1^{x_1}$
	$X \leftarrow (X_0, X_1)$; $Y \leftarrow (Y_0, \mathcal{O}_1)$
	Return (G, X, Y)

Figure 23: Adversary for the proof of Lemma 5.6.

$k \in \mathbb{N}$. Consider games $\mathbf{G}_0$ and $\mathbf{G}_1$ defined in Figure 22. Note that $\mathbf{G}_0$ is the usual sCDH-PSA game with respect to $\mathsf{EG}_{\mathrm{twist}}$ and adversary $\mathcal{A}$ as defined in Figure 13. Hence

$$\mathbf{Adv}^{\text{scdh-psa}}_{\mathsf{EG}_{\mathrm{twist}},\mathcal{A}}(k) = \Pr[\mathbf{G}_0(k)] \ . \tag{7}$$

In game $\mathbf{G}_0$ let d' denote the indicator random variable taking value 0 if $Y \in E_0$ and 1 if $Y \in E_1$. This yields

$$\Pr[\mathbf{G}_1(k)] = \Pr\big[\mathbf{G}_0(k) \wedge d' = 1\big] \leq \Pr\big[\mathbf{G}_0(k) \mid d' = 1\big] \ , \tag{8}$$

$$\Pr[\mathbf{G}_0(k)] - \Pr[\mathbf{G}_1(k)] = \Pr\big[\mathbf{G}_0(k) \wedge d' = 0\big] \leq \Pr\big[\mathbf{G}_0(k) \mid d' = 0\big] \ . \tag{9}$$

We construct an adversary $\mathcal{B}_0$ such that

$$\Pr\big[\mathbf{G}_0(k) \mid d' = 0\big] \leq \mathbf{Adv}^{\text{twist}_0\text{-cp-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(k) \ . \tag{10}$$

The definition of adversary $\mathcal{B}_0$ may be found in Figure 23. It provides $\mathcal{A}$ with a perfect simulation of game $\mathbf{G}_0(k)$ conditioned on the events $d' = 0$. Since $\mathcal{A}$ solving its sCDH challenge implies $\mathcal{B}_0$ solving its sCDH challenge, we obtain $\Pr[\mathbf{G}_0(k) \mid d' = 0] \leq \mathbf{Adv}^{\text{twist}_0\text{-cp-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(k)$.

Analogous to the case above there exists an adversary $\mathcal{B}_1$ against game $\mathbf{G}^{\text{twist}_1\text{-cp-scdh}}_{\mathsf{TGen},\mathcal{B}_1}(k)$ satisfying

$$\Pr\big[\mathbf{G}_0(k) \mid d' = 1\big] \leq \mathbf{Adv}^{\text{twist}_1\text{-cp-scdh}}_{\mathsf{TGen},\mathcal{B}_1}(k) \ . \tag{11}$$

$\underline{\mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{P}(1^k)}$
Return ε

$\underline{\mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{G}(1^k, \pi)}$
$p \leftarrow\!\!\$\ \mathcal{P}_k$
$G' \leftarrow\!\!\$\ \mathsf{EG}_{\text{twist}}.\mathsf{G}(1^k, p)$
$G \leftarrow (G', p)$
Return G

$\underline{\mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{S}(1^k, \pi, G)}$
$(G', p) \leftarrow G$
For $\ell^* = 1$ to ℓ
  Do $y \leftarrow \mathsf{EG}_{\text{twist}}.\mathsf{S}(1^k, p, G')$
  If $(\mathsf{EG}_{\text{twist}}.\mathsf{E}(1^k, p, G, g^y) < 2^k)$
    return y
Return $\perp$

$\underline{\mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{E}(1^k, \pi, G, h)}$
$(G', p) \leftarrow G'$
$z \leftarrow\!\!\$\ \mathsf{EG}_{\text{twist}}.\mathsf{E}(1^k, p, G', h)$
Return z

$\underline{\mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{I}(1^k, \pi, G, z)}$
$(G', p) \leftarrow G'$
$h \leftarrow \mathsf{EG}_{\text{twist}}.\mathsf{I}(1^k, p, G', z)$
Return h

Figure 24: Parameter-free eeg family $\mathsf{EG}^{\ell}_{\text{twist-rs}}$.

Now Equations (7) to (11) yield

$$\mathbf{Adv}^{\text{scdh-psa}}_{\mathsf{EG}_{\text{twist}},\mathcal{A}}(k) \le \mathbf{Adv}^{\text{twist}_0\text{-cp-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(k) + \mathbf{Adv}^{\text{twist}_1\text{-cp-scdh}}_{\mathsf{TGen},\mathcal{B}_1}(k)$$

as desired. □

5.3 A parameter-free eeg family using rejection sampling

Eeg family $\mathsf{EG}_{\text{twist}}$ of Section 5.2 is parameter-using, the parameter being the size p of the field $\mathbb{F}_p$. Correspondingly, hardness of the CDH problem with respect to $\mathsf{EG}_{\text{twist}}$ follows from the assumption, that the elliptic curves output by curve-twist generator TGen are secure, independently of the prime p the curve-twist generator TGen is instantiated with. In this section we show how $\mathsf{EG}_{\text{twist}}$ can be used to construct an eeg family $\mathsf{EG}^{\ell}_{\text{twist-rs}}$ for which hardness of CDH-PSA follows from the weaker assumption that TGen instantiated with a *randomly* chosen prime is able to sample secure elliptic curves.

We now discuss eeg family $\mathsf{EG}^{\ell}_{\text{twist-rs}}$. The construction is parameter-free and has embedding space $[2^k]$. The size p of the field over which the elliptic curves are defined is now sampled as part of the group generation. The embedding algorithm uses rejection sampling to ensure that embeddings of group elements g^y for y sampled with $\mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{S}$ are elements of $[2^k]$. The specification of $\mathsf{EG}^{\ell}_{\text{twist-rs}}$'s algorithms may be found in Figure 24.

Theorem 5.7. *Let $\ell : \mathbb{N} \to \mathbb{N}$ be a polynomial. $\mathsf{EG}^{\ell}_{\textit{twist-rs}}$ as described above is an eeg family with embedding space $\mathsf{EG}^{\ell}_{\textit{twist-rs}}.\mathsf{ES}(k, \pi) = [2^k]$ and inversion error $\mathsf{EG}^{\ell}_{\textit{twist-rs}}.\mathsf{ie}(k) \le 2^{-\ell(k)}$. Furthermore $\mathsf{EG}^{\ell}_{\textit{twist-rs}}$ has pseudorandom embeddings. More precisely, for every (potentially unbounded) adversary $\mathcal{A}$ we have*

$$\mathbf{Adv}^{\text{epr-psa}}_{\mathsf{EG}^{\ell}_{\textit{twist-rs}},\mathcal{A}}(k) \le 2^{-\ell(k)} \quad .$$

Proof. The polynomial ℓ ensures that the sampling algorithms runs in polynomial time by serving as an upper bound on the number of rounds the algorithm tries to sample y satisfying $\mathsf{EG}_{\text{twist}}.\mathsf{E}(1^k, G, g^y) < 2^k$.

We first bound $\mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{ie}$. Fix $k \in \mathbb{N}$ and $G = (\langle\mathbb{G}\rangle, n, g) \in [\mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{G}(k)]$. Note that $\mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{I}$ only fails to invert $\mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{E}$, if $\perp \leftarrow\!\!\$\ \mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{S}(1^k, G)$. Let $\mathcal{L}$ denote the event that $\mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{S}(1^k, G)\}$ returns $\perp$ and let $M := \{y \in [\mathsf{EG}_{\text{twist}}.\mathsf{S}(1^k, G)] \mid \mathsf{EG}_{\text{twist}}.\mathsf{E}(1^k, G, g^y) < 2^k\}$. We have

$$|M| = 2^k \ge (2p+1)/2 = \left|[\mathsf{EG}_{\text{twist}}.\mathsf{S}(1^k, G)]\right|/2.$$

Hence $\Pr[\mathcal{L}] \le 2^{-\ell(k)}$, which implies $\mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{ie}(k) \le 2^{-\ell(k)}$.

Let $\mathcal{A}$ be an adversary against the embedding pseudorandomness game of Figure 12. Since $\mathsf{EG}^{\ell}_{\text{twist-rs}}$ is parameter-free, the parameter π that $\mathcal{A}$ provides to Init does not influence its success

probability in the security game. Hence to prove the bound on $\mathbf{Adv}^{\text{epr-psa}}_{\mathsf{EG}^{\ell}_{\text{twist-rs}},\mathcal{A}}(k)$ it suffices to bound the statistical distance between elements of $\mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{ES}(k,\pi)$, which are sampled and embedded using the algorithms of $\mathsf{EG}^{\ell}_{\text{twist-rs}}$, and elements of $\mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{ES}(1^k,\pi)$, which are distributed uniformly on $\mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{ES}(k,\pi)$. Conditioned on the event $\neg\mathcal{L}$ the output y of $\mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{S}(1^k,G)$ is uniformly distributed on M. For all $y' \in M$ we obtain

$$\Pr[y = y'] = \Pr[y = y' \mid \mathcal{L}]\Pr[\mathcal{L}] + \Pr[y = y' \mid \neg\mathcal{L}]\Pr[\neg\mathcal{L}] = 2^{-k}\Pr[\neg\mathcal{L}].$$

Since the map $M \to \mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{ES}(k) = [2^k]; y \mapsto \mathsf{EG}_{\text{twist}}.\mathsf{E}(1^k,G,g^y)$ is a bijection,

$$\begin{aligned}
\Delta(y;U_M) &= \frac{1}{2}\left(\Pr[y=\perp] + \sum_{y'\in M}\left|\Pr[y=y'] - \frac{1}{2^k}\right|\right)\\
&= \frac{1}{2}\left(\Pr[\mathcal{L}] + \sum_{y\in M}\left|\Pr[\neg\mathcal{L}]\frac{1}{2^k} - \frac{1}{2^k}\right|\right)\\
&= \frac{1}{2}\left(\Pr[\mathcal{L}] + \Pr[\mathcal{L}]\sum_{y\in M}\frac{1}{2^k}\right) = \Pr[\mathcal{L}] \le 2^{-\ell(k)}.
\end{aligned}$$

This completes the proof. □

As discussed above, we obtain that —assuming that TGen invoked on randomly sampled prime p returns a secure curve-twist pair— the CDH-problem with respect to eeg family $\mathsf{EG}^{\ell}_{\text{twist-rs}}$ is hard.

Lemma 5.8. *Let $\ell : \mathbb{N} \to \mathbb{N}$ be a polynomial and $\mathsf{EG}^{\ell}_{twist\text{-}rs}$ the eeg family with underlying curve-twist generator TGen as described above. If the (strong) computational Diffie-Hellman assumption for uniform primes holds with respect to TGen, then the (strong) computational Diffie-Hellman assumption holds with respect to $\mathsf{EG}^{\ell}_{twist\text{-}rs}$.*

Concretely, for every adversary $\mathcal{A}$ against game $\mathbf{G}^{\text{cdh-psa}}_{\mathsf{EG}^{\ell}_{twist\text{-}rs},\mathcal{A}}(\cdot)$ there exist adversaries $\mathcal{B}_0$, $\mathcal{B}_1$ against games $\mathbf{G}^{\text{twist}_0\text{-up-cdh}}_{\mathsf{TGen},\mathcal{B}_0}(\cdot)$ or $\mathbf{G}^{\text{twist}_1\text{-up-cdh}}_{\mathsf{TGen},\mathcal{B}_1}(\cdot)$ respectively running in the same time as $\mathcal{A}$ and satisfying

$$\mathbf{Adv}^{\text{cdh-psa}}_{\mathsf{EG}^{\ell}_{twist\text{-}rs},\mathcal{A}}(k) \le 3\left(\mathbf{Adv}^{\text{twist}_0\text{-up-cdh}}_{\mathsf{TGen},\mathcal{B}_0}(k) + \mathbf{Adv}^{\text{twist}_1\text{-up-cdh}}_{\mathsf{TGen},\mathcal{B}_1}(k)\right) + 2^{-\ell(k)}$$

for all $k \in \mathbb{N}_{\ge 6}$.

Further for every adversary $\mathcal{A}$ against game $\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{twist\text{-}rs},\mathcal{A}}(\cdot)$ making at most Q queries to its DDH*-oracle there exist adversaries $\mathcal{B}_0$, $\mathcal{B}_1$ against $\mathbf{G}^{\text{twist}_0\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(\cdot)$ or $\mathbf{G}^{\text{twist}_1\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_1}(\cdot)$ respectively, making at most Q queries to their* DDH*-oracles and running in the same time as $\mathcal{A}$, which satisfy*

$$\mathbf{Adv}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{twist\text{-}rs},\mathcal{A}}(k) \le 3\left(\mathbf{Adv}^{\text{twist}_0\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(k) + \mathbf{Adv}^{\text{twist}_1\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_1}(k)\right) + 2^{-\ell(k)}$$

for all $k \in \mathbb{N}_{\ge 6}$.

Proof. We prove the statement on sCDH-PSA. The statement on CDH-PSA can be shown analogously. Let $k \in \mathbb{N}_{\ge 6}$ and $\mathcal{A}$ be an adversary against the strong CDH game with respect to $\mathsf{EG}^{\ell}_{\text{twist-rs}}$. Consider games $\mathbf{G}_0$ and $\mathbf{G}_1$ defined in Figure 25. Note that $\mathbf{G}_0$ is the usual sCDH-PSA game with respect to eeg family $\mathsf{EG}^{\ell}_{\text{twist-rs}}$ and adversary $\mathcal{A}$ and that condition $G \neq \perp$ always holds. We obtain

$$\mathbf{Adv}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{\text{twist-rs}},\mathcal{A}}(k) = \Pr[\mathbf{G}_0(k)] \ . \tag{12}$$

Game $\mathbf{G}_0(k)$	$\underline{\textsc{Init}(\pi)}$
$Z \leftarrow\!\!\$\ \mathcal{A}^{\textsc{Init},\textsc{ddh}}(1^k)$	$p \leftarrow\!\!\$\ \mathcal{P}_k$
Return $(Z = g^{xy} \wedge G \neq \bot)$	$(G_0, G_1) \leftarrow\!\!\$\ \mathsf{TGen}(1^k, p)$
Game $\mathbf{G}_1(k)$	$(\langle E_0\rangle, g_0, n_0) \leftarrow G_0$; $(\langle E_1\rangle, g_1, n_1) \leftarrow G_1$
$Z \leftarrow\!\!\$\ \mathcal{A}^{\textsc{Init},\textsc{ddh}}(1^k)$	$\mathbb{G} \leftarrow E_0 \times E_1$; $g \leftarrow (g_0, g_1)$; $n \leftarrow n_0 \cdot n_1$
Return $(Z = g^{xy} \wedge Y \in E_1 \wedge G \neq \bot)$	$G' \leftarrow (\langle\mathbb{G}\rangle, n, g)$; $G \leftarrow (G', p)$
$\underline{\textsc{ddh}(\tilde{Y}, \tilde{Z})}$	$x \leftarrow\!\!\$\ \mathbb{Z}_n$
Return $(\tilde{Y}^x = \tilde{Z})$	$y \leftarrow\!\!\$\ \mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{S}(1^k, \pi, G)$
	$X \leftarrow g^x$; $Y \leftarrow g^y$
	Return (G, X, Y)

Figure 25: Games for the proof of Lemma 5.8. Both games use the same procedures Init and ddh. In $\mathbf{G}_1$ we see E_1 as a subset of $E_0 \times E_1$ in the natural way.

$\underline{\text{Adversary } \mathcal{B}_0^{\textsc{Init},\textsc{ddh}}(1^k)}$	$\underline{\textsc{SimInit}(\pi)}$
$(Z_0, Z_1) \leftarrow\!\!\$\ \mathcal{A}^{\textsc{SimInit},\textsc{Simddh}}(1^k)$	$(G_0, G_1, p, X_0, Y_0) \leftarrow\!\!\$\ \textsc{Init}$
Return Z_0	$(\langle E_0\rangle, n_0, g_0) \leftarrow G_0$; $(\langle E_1\rangle, n_1, g_1) \leftarrow G_1$
$\underline{\textsc{Simddh}((\tilde{Y}_0, \tilde{Y}_1), (\tilde{Z}_0, \tilde{Z}_1))}$	$G' \leftarrow (\langle E_0 \times E_1\rangle, n_0 n_1, (g_0, g_1))$; $G \leftarrow (G', p)$
If $(\tilde{Y}_1^{x_1} = \tilde{Z}_1)$	$x_1 \leftarrow\!\!\$\ \mathbb{Z}_{n_1}$; $X_1 \leftarrow g_1^{x_1}$
then return $\textsc{ddh}(\tilde{Y}_0, \tilde{Z}_0)$	$X \leftarrow (X_0, X_1)$; $Y \leftarrow (Y_0, \mathcal{O}_1)$
Return false	Return (G, X, Y)

Figure 26: Adversary for the proof of Lemma 5.8.

In game $\mathbf{G}_0$ let d' denote the indicator random variable taking value 0 if $g^y \in E_0$, 1 if $g^y \in E_1$ and $\bot$ if $y = \bot$. We have $\Pr[\mathbf{G}_1] = \Pr[\mathbf{G}_0 \wedge d' = 1]$. This yields

$$\begin{aligned}\Pr[\mathbf{G}_0(k)] - \Pr[\mathbf{G}_1(k)] &\leq \Pr\big[\mathbf{G}_0(k) \wedge d' = 0\big] + \Pr\big[d' = \bot\big] \\ &\leq \Pr\big[\mathbf{G}_0(k) \mid d' = 0\big] + 2^{-\ell(k)} \end{aligned} \tag{13}$$

and

$$\Pr[\mathbf{G}_1(k)] = \Pr\big[\mathbf{G}_0(k) \wedge d' = 1\big] \leq \Pr\big[\mathbf{G}_0(k) \mid d' = 1\big] \ . \tag{14}$$

We construct adversaries $\mathcal{B}_0$ and $\mathcal{B}_1$ against $\mathbf{G}^{\text{twist}_0\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(\cdot)$ or $\mathbf{G}^{\text{twist}_1\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_1}(\cdot)$ respectively, which satisfy

$$\Pr\big[\mathbf{G}_0(k) \mid d' = 0\big] \leq 3\mathbf{Adv}^{\text{twist}_0\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(k) \ , \tag{15}$$

$$\Pr\big[\mathbf{G}_0(k) \mid d' = 1\big] \leq 3\mathbf{Adv}^{\text{twist}_1\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_1}(k) \ . \tag{16}$$

Plugging this into Equations (12) to (14) yields

$$\mathbf{Adv}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{\text{twist-rs}},\mathcal{A}}(k) \leq 3\left(\mathbf{Adv}^{\text{twist}_0\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(k) + \mathbf{Adv}^{\text{twist}_1\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_1}(k)\right) + 2^{-\ell(k)}$$

as desired.

We show how to prove equation (15); equation (16) can be obtained in a similar way. Consider adversary $\mathcal{B}_0$ of Figure 26, which provides $\mathcal{A}$ with a simulation of game $\mathbf{G}_0(k)$ conditioned on

the event $d' = 0$. Note that $\mathcal{A}$ solving its challenge implies that Z_0 is a correct solution to $\mathcal{B}_0$'s challenge.

We now proceed to analyze $\mathcal{B}_0$'s success probability. By definition of $\mathcal{B}_0$, the value G generated by $\mathcal{B}_0$ consisting of prime p and the description of group G' is distributed as in $\mathbf{G}_0(k)$. Furthermore —as in game $\mathbf{G}_0(k)$— the group element X generated by $\mathcal{B}$ is uniformly distributed on $E_0 \times E_1$ and procedure SIMDDH is a perfect simulation of the DDH oracle of $\mathbf{G}_0(k)$. However by definition of $\mathbf{G}^{\text{twist}_0\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(k)$ the value Y is uniformly distributed on $E_0 \subseteq E_0 \times E_1$, while in $\mathbf{G}_0$ conditioned on $d' = 0$ it is distributed as $Y = g^y$, where $y \leftarrow\!\!\$\ \mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{S}(1^k, \pi, G)$, with the additional condition that $Y \in E_0$. We conclude the proof by showing that $\mathcal{A}$ — even on input (G, X, Y) with $Y \sim U_{E_0}$ — looses at most a factor of 3 in its success probability of computing g^{xy}.

To prove this, we condition on G. We write $\Pr_{\mathbf{G}_0}$ and $\Pr_{\mathbf{G}_{\text{twist}_0}}$ to indicate, whether we consider probabilities in game $\mathbf{G}_0(k)$ or $\mathbf{G}^{\text{twist}_0\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(k)$. Let $\tilde{G} \in [\mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{G}(1^k, \pi)]$, where $\tilde{G} = (\tilde{G}', \tilde{p})$ and $\tilde{G}' = (\langle \tilde{E}_0 \times \tilde{E}_1 \rangle, \tilde{n_0}\tilde{n_1}, (\tilde{g_0}, \tilde{g_1}))$. In a first step we show that for all $\tilde{Y_0} \in \tilde{E_0}$

$$\Pr_{\mathbf{G}_0}\Big[Y = (\tilde{Y_0}, \tilde{\mathcal{O}_1}) \mid d' = 0 \wedge G = \tilde{G}\Big] \leq 3 \Pr_{\mathbf{G}_{\text{twist}_0}}\Big[Y_0 = \tilde{Y_0} \mid G = \tilde{G}\Big] \ . \tag{17}$$

Let $M := \{\tilde{Y} \in \tilde{E_0} \times \tilde{E_1} \mid \tilde{Y} \in \tilde{E_0} \wedge \mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{E}(1^k, \tilde{G}, \tilde{Y}) < 2^k\}$, where we interpret $\tilde{E_0}$ as subset of $\tilde{E_0} \times \tilde{E_1}$ in the natural way. In game $\mathbf{G}_0$ conditioned on the events $G = \tilde{G}$ and $d' = 0$ we have $Y \sim U_M$ since rejection sampling is used to generate y. We compute a bound on $|M|$. Since $\mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{E}(1^k, \tilde{G}, \tilde{\mathcal{O}}) = \tilde{p} < 2^k$, we have $\tilde{\mathcal{O}} \in M$. Now let $\tilde{Y_0} \in \tilde{E_0} \setminus \{\tilde{\mathcal{O}_0}\}$ and $\tilde{Y} = (\tilde{Y_0}, \tilde{\mathcal{O}_1})$. Write $\tilde{Y_0}$ in its coordinate form $\tilde{Y_0} = (c_x, c_y) \in \mathbb{F}_p^2$. Then $\tilde{Y_0}^{-1} = (c_x, -c_y)$ and $\tilde{Y}^{-1} = (\tilde{Y_0}^{-1}, \tilde{\mathcal{O}_1})$. If we assume that $c_y \neq 0$, this by definition of $\mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{E}$ implies that either $\mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{E}(1^k, \pi, \tilde{G}, \tilde{Y}) = l_0(\tilde{Y_0}) < \tilde{p} < 2^k$ or $\mathsf{EG}^{\ell}_{\text{twist-rs}}.\mathsf{E}(1^k, \pi, \tilde{G}, \tilde{Y}^{-1}) = l_0(\tilde{Y_0}^{-1}) < \tilde{p} < 2^k$, where $l_0 : E_0 \to [2\tilde{p} + 2]$ is the function of Lemma 5.1. Hence either $\tilde{Y}$ or $\tilde{Y}^{-1}$ is an element of M. Since there are at most 3 points on $\tilde{E_0}$ having y-coordinate 0, we obtain $|M| \geq (\tilde{n_0} - 3)/2$.

For all $\tilde{Y_0} \in \tilde{E_0}$ this yields

$$\begin{aligned} &\Pr_{\mathbf{G}_0}\Big[Y = (\tilde{Y_0}, \tilde{\mathcal{O}_1}) \mid d' = 0 \wedge G = \tilde{G}\Big] / \Pr_{\mathbf{G}_{\text{twist}_0}}\Big[Y_0 = \tilde{Y_0} \mid G = \tilde{G}\Big] \\ \leq\ & \frac{\tilde{n_0}}{(\tilde{n_0} - 3)/2} = 2\frac{1}{1 - 3/\tilde{n_0}} \leq 2\frac{1}{1 - 2^{-(k-4)}} \leq 3 \ . \end{aligned}$$

This establishes (17). Here we use the fact that in $\mathbf{G}^{\text{twist}_0\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(k)$ —conditioned on the event $G = \tilde{G}$— the value Y_0 is uniformly distributed on $\tilde{E_0}$. Furthermore, the second to last inequality uses Hasse's Theorem and the last inequality holds since $k \geq 6$.

Conditioned on the events $G = \tilde{G}$ and $Y_0 = \tilde{Y_0}$ adversary $\mathcal{B}_0$ provides $\mathcal{A}$ with a perfect simulation of $\mathbf{G}_0$ conditioned on $G = \tilde{G}$ and $Y = (\tilde{Y_0}, \tilde{\mathcal{O}_1})$, which in particular implies $d' = 0$. Using Equation (23) we obtain

$$\begin{aligned} &\Pr_{\mathbf{G}_0}\Big[\mathbf{G}_0(k) \mid d' = 0 \wedge G = \tilde{G}\Big] \\ =\ & \sum_{\tilde{Y_0} \in \tilde{E_0}} \Pr_{\mathbf{G}_0}\Big[\mathbf{G}_0(k) \mid Y = (\tilde{Y_0}, \tilde{\mathcal{O}_1}) \wedge d' = 0 \wedge G = \tilde{G}\Big] \cdot \Pr_{\mathbf{G}_0}\Big[Y = (\tilde{Y_0}, \tilde{\mathcal{O}_1}) \mid d' = 0 \wedge G = \tilde{G}\Big] \\ \overset{(17)}{\leq}\ & \sum_{\tilde{Y_0} \in \tilde{E_0}} \Pr_{\mathbf{G}_{\text{twist}_0}}\Big[\mathbf{G}^{\text{twist}_0\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(k) \mid Y_0 = \tilde{Y_0} \wedge G = \tilde{G}\Big] \cdot 3 \Pr_{\mathbf{G}_{\text{twist}_0}}\Big[Y_0 = \tilde{Y_0} \mid G = \tilde{G}\Big] \\ =\ & 3 \Pr_{\mathbf{G}_{\text{twist}_0}}\Big[\mathbf{G}^{\text{twist}_0\text{-up-scdh}}_{\mathsf{TGen}0,\mathcal{B}_0}(k) \mid G = \tilde{G}\Big] \ . \end{aligned}$$

An application of the law of total probability yields (15). □

$\underline{\mathsf{EG}_{\text{twist-re}}.\mathsf{G}(1^k, \pi)}$
$p \leftarrow\!\!\$\ \mathcal{P}_k$
$G' \leftarrow\!\!\$\ \mathsf{EG}_{\text{twist}}.\mathsf{G}(1^k, p)$; $G \leftarrow (G', p)$
Return G

$\underline{\mathsf{EG}_{\text{twist-re}}.\mathsf{S}(1^k, \pi, G)}$
$(G', p) \leftarrow G$
$z \leftarrow\!\!\$\ [2^{k+1}]$
If $(z \leq 2p)$
  $y \leftarrow \psi_G(z)$
  If $(\mathsf{EG}_{\text{twist}}.\mathsf{E}(1^k, p, G', g^y) < 2^{k+1} - (2p+1))$
    return y
  Else $z \leftarrow \mathsf{EG}_{\text{twist}}.\mathsf{E}(1^k, p, G', g^y)$
Else $z \leftarrow z - (2p+1)$
$y \leftarrow \psi_G(z)$
Return y

$\underline{\mathsf{EG}_{\text{twist-re}}.\mathsf{P}(1^k)}$
Return ε

$\underline{\mathsf{EG}_{\text{twist-re}}.\mathsf{E}(1^k, \pi, G, h)}$
$(G', p) \leftarrow G$
$b \leftarrow\!\!\$\ \{0,1\}$
$z \leftarrow \mathsf{EG}_{\text{twist}}.\mathsf{E}(1^k, p, G', h)$
If $z < 2^{k+1} - (2p+1)$
  $z \leftarrow z + b(2p+1)$
Return z

$\underline{\mathsf{EG}_{\text{twist-re}}.\mathsf{I}(1^k, \pi, G, z)}$
$(G', p) \leftarrow G$
If $(z \geq 2p+1)$
  $z \leftarrow z - (2p+1)$
$h \leftarrow \mathsf{EG}_{\text{twist}}.\mathsf{I}(1^k, p, G', z)$
Return h

Figure 27: Definition of eeg family $\mathsf{EG}_{\text{twist-re}}$ with embedding space $\mathsf{EG}_{\text{twist-re}}.\mathsf{ES}(k, \pi) := [2^{k+1}]$. ψ_G denotes the bijection $[2p+1] \to [\mathsf{EG}_{\text{twist}}.\mathsf{S}(1^k, p, G')]$ defined in Section 5.4.

5.4 A parameter-free family using range expansion

In this section we modify the algorithms of $\mathsf{EG}_{\text{twist}}$ to obtain an embeddable group family $\mathsf{EG}_{\text{twist-re}}$ with embedding space $\mathsf{EG}_{\text{twist-re}}.\mathsf{ES}(k, \pi) = [2^{k+1}]$. The eeg family has inversion error $\mathsf{EG}_{\text{twist-re}}.\mathsf{ie}(k) = 0$ and achieves uniformly distributed embeddings. The construction is building on a technique introduced by Hayashi *et al.* [HOT04], where it is used to expand the range of one way permutations. As in Section 5.3, the hardness CDH-PSA with respect to $\mathsf{EG}_{\text{twist-re}}$ is based on the hardness of the CDH problem for uniform primes with respect to TGen.

The sampling algorithm — in contrast to the construction based on rejection sampling — needs access to only one uniformly random sampled integer, performs at most one exponentiation in the group and uses at most one evaluation of $\mathsf{EG}_{\text{twist}}.\mathsf{E}$ to output y with the correct distribution. Furthermore, exponents sampled by $\mathsf{EG}_{\text{twist-re}}.\mathsf{S}$ are distributed such that the eeg family achieves $\mathsf{EG}_{\text{twist-re}}.\mathsf{ie}(k) = 0$ and for every (potentially unbounded) adversary $\mathcal{A}$ we additionally have $\mathbf{Adv}^{\text{epr-psa}}_{\mathsf{EG}_{\text{twist-re}},\mathcal{A}}(k) = 0$.

The description of $\mathsf{EG}_{\text{twist-re}}$ may be found in Figure 27. We now discuss the construction in greater detail. Let $(G', p) = G \in [\mathsf{EG}_{\text{twist-re}}.\mathsf{G}(k, \pi)]$, where $G' = (\langle\mathbb{G}\rangle, n, g)$. The idea of the construction is to partition $[\mathsf{EG}_{\text{twist}}.\mathsf{S}(1^k, p, G')]$ into two sets M_1, M_2 with $M_1 \cup M_2 = [\mathsf{EG}_{\text{twist}}.\mathsf{S}(1^k, p, G')]$, $\{\mathsf{EG}_{\text{twist}}.\mathsf{E}(1^k, p, G', g^y) \mid y \in M_1\} = \{2^{k+1} - (2p+1), \cdots, 2p\}$ and $\{\mathsf{EG}_{\text{twist}}.\mathsf{E}(1^k, p, G', g^y) \mid y \in M_2\} = \{0, \cdots, 2^{k+1} - (2p+2)\}$. The sampling algorithm $\mathsf{EG}_{\text{twist-re}}.\mathsf{S}$ is constructed such that for y sampled by $\mathsf{EG}_{\text{twist-re}}.\mathsf{S}(1^k, \pi, G)$, the probability $\Pr[y = y']$ equals 2^{-k} for all $y' \in M_2$ and $2^{-(k+1)}$ for all $y' \in M_1$. Embedding algorithm $\mathsf{EG}_{\text{twist-re}}.\mathsf{E}$ on input $(1^k, \pi, G, h)$ first computes $c \leftarrow \mathsf{EG}_{\text{twist}}.\mathsf{E}(1^k, p, G', h)$. If $c \in \{2^{k+1} - (2p+1), \cdots, 2p\}$ its output remains unchanged. Otherwise it is shifted to $\{2p+1, \cdots, 2^{k+1} - 1\}$ with probability $1/2$. In this way we achieve embeddings , which are uniformly distributed on $\mathsf{EG}_{\text{twist-re}}.\mathsf{ES}(k, \pi) = [2^{k+1}]$.

Our construction relies on the existence of a bijection $\psi_G : [2p+1] \to [\mathsf{EG}_{\text{twist}}.\mathsf{S}(1^k, p, G')]$ for all $(G', p) = G \in [\mathsf{EG}_{\text{twist-re}}.\mathsf{G}(1^k, \pi)]$. We use the bijection, which was implicitly given in the definition of $\mathsf{EG}_{\text{twist}}.\mathsf{S}$. That is, for $z \in [2p+1]$ we define

$$\psi_G(z) := \begin{cases} \varphi_{\text{crt}}(z, 0) & \text{if } z < n_0 \\ \varphi_{\text{crt}}(0, z - n_0 - 1) & \text{else,} \end{cases}$$

where φ_{crt} denotes the canonical isomorphism $\mathbb{Z}_{n_0} \times \mathbb{Z}_{n_1} \to \mathbb{Z}_n$.

Theorem 5.9. *$\mathsf{EG}_{\textit{twist-re}}$ as specified in Figure 27 is an embeddable group family with embedding space $\mathsf{EG}_{\textit{twist-re}}.\mathsf{ES}(k,\pi) = [2^{k+1}]$ and inverson error $\mathsf{EG}_{\textit{twist-re}}.\mathsf{ie}(k) = 0$. Furthermore $\mathsf{EG}_{\textit{twist-re}}$ has pseudorandom embeddings. More precisely, for every (potentially unbounded) adversary $\mathcal{A}$ we have*

$$\mathbf{Adv}^{\text{epr-psa}}_{\mathsf{EG}_{\textit{twist-re}},\mathcal{A}}(k) = 0 \ .$$

Proof. Let $k \in \mathbb{N}$ and $(G', p) = G \in [\mathsf{EG}_{\text{twist-re}}.\mathsf{G}(1^k)]$, where $G' = (\langle\mathbb{G}\rangle, n, g)$. It is easy to verify that $\mathsf{EG}_{\text{twist-re}}.\mathsf{I}$ is able to retrieve g^y from $\mathsf{EG}_{\text{twist-re}}.\mathsf{E}(1^k, \pi, G, g^y)$ for $\pi = \varepsilon$ and all $G \in [\mathsf{EG}_{\text{twist-re}}.\mathsf{G}(1^k, \pi)]$ and $y \in [\mathsf{EG}_{\text{twist-re}}.\mathsf{S}(1^k, \pi, G)]$. This implies $\mathsf{EG}_{\text{twist-re}}.\mathsf{ie}(k) = 0$.

Let $\mathcal{A}$ be an adversary against the embedding pseudorandomness game of Figure 12 with respect to $\mathsf{EG}_{\text{twist-re}}$. Since $\mathsf{EG}_{\text{twist-re}}$ is parameter-free, the parameter π that $\mathcal{A}$ provides to Init does not influence its success probability in the security game. Hence to show that $\mathbf{Adv}^{\text{epr-psa}}_{\mathsf{EG}_{\text{twist-re}},\mathcal{A}}(k) = 0$ it suffices to show that elements sampled and embedded using $\mathsf{EG}_{\text{twist-re}}$'s algorithms are distributed uniformly on $\mathsf{EG}_{\text{twist-re}}.\mathsf{ES}(k,\pi)$. Let $M := [\mathsf{EG}_{\text{twist}}.\mathsf{S}(1^k, p, G')]$. We partition M into two disjoint sets via

$$M_1 := M \cap \{y \in \mathbb{Z}_n \mid \mathsf{EG}_{\text{twist}}.\mathsf{E}(1^k, p, G', g^y) \geq 2^{k+1} - (2p+1)\}$$
$$M_2 := M \cap \{y \in \mathbb{Z}_n \mid \mathsf{EG}_{\text{twist}}.\mathsf{E}(1^k, p, G', g^y) < 2^{k+1} - (2p+1)\}.$$

We have to show that $\mathsf{EG}_{\text{twist-re}}.\mathsf{E}(1^k, \pi, G, g^y)$ is uniformly distributed on embedding space $\mathsf{EG}_{\text{twist-re}}.\mathsf{ES}(k\pi,) = [2^{k+1}]$ for $y \leftarrow \mathsf{EG}_{\text{twist-re}}.\mathsf{S}$. As an intermediate step we prove the following claim.

Claim 5.10. Let $y' \in M$ and $y \leftarrow \mathsf{EG}_{\text{twist-re}}.\mathsf{S}(1^k, \pi, G)$. Then

$$\Pr[y = y'] = \begin{cases} 2^{-(k+1)} & \text{if} \quad y' \in M_1 \\ 2^{-k} & \text{if} \quad y' \in M_2. \end{cases}$$

Denote by $\mathcal{E}$ the event that $\mathsf{EG}_{\text{twist-re}}.\mathsf{S}$ returns y in line 4 of its definition. Since $z \mapsto \mathsf{EG}_{\text{twist}}.\mathsf{E}(1^k, p, G', g^{\psi_G(z)})$ is a bijection we have $\Pr[\mathcal{E}] = (2^{k+1} - (2p+1))/2^{k+1}$ and $\Pr[\neg\mathcal{E}] = (2p+1)/2^{k+1}$. Hence

$$\Pr[y = y' \mid \mathcal{E}] = \begin{cases} 0 & \text{for } y' \in M_1 \\ 1/(2^{k+1} - (2p+1)) & \text{for } y' \in M_2 \end{cases}.$$

Now assume that $\mathcal{E}$ does not occur. Then y is uniformly distributed on M. This holds since in this case $y = \psi_G(z)$ where $z \sim U_{[2p+1]}$. Summing up we obtain

$$\Pr[y = y'] = \begin{cases} 0 \cdot \frac{2^{k+1}-(2p+1)}{2^{k+1}} + \frac{1}{2p+1}\frac{2p+1}{2^{k+1}} = \frac{1}{2^{k+1}} & \text{if } y' \in M_1 \\ \frac{1}{2^{k+1}-(2p+1)}\frac{2^{k+1}-(2p+1)}{2^{k+1}} + \frac{1}{2p+1}\frac{2p+1}{2^{k+1}} = \frac{2}{2^{k+1}} & \text{if } y' \in M_2 \end{cases},$$

which proves Claim 5.10. Building on Claim 5.10 we show that embeddings under $\mathsf{EG}_{\text{twist-re}}.\mathsf{E}$ are uniformly distributed on the embedding space. For $z \in [2^{k+1}]$ consider the probability $p(z) = \Pr\left[\mathsf{EG}_{\text{twist-re}}.\mathsf{E}(1^k, \pi, G, g^y) = z\right]$ for $y \leftarrow \mathsf{EG}_{\text{twist-re}}.\mathsf{S}(1^k, \pi, G)$.

- Case 1: $z < 2^{k+1} - (2p+1)$.
  Then $p(z) = \Pr\left[\mathsf{EG}_{\text{twist}}.\mathsf{E}(1^k, p, G', g^y) = z \wedge b = 0\right] = 1/2 \cdot 2^{-k} = 2^{-(k+1)}$. The last equality is due to Claim 5.10 using that in this case $y \in M_2$.

Game $\mathbf{G}_0(k)$	$\underline{\textsc{Init}(\pi)}$
$Z \leftarrow\!\!\$\ \mathcal{A}^{\textsc{Init},\textsc{ddh}}(1^k)$	$p \leftarrow\!\!\$\ \mathcal{P}_k$
Return $(Z = g^{xy} \wedge G \neq \bot)$	$(G_0, G_1) \leftarrow\!\!\$\ \mathsf{TGen}(1^k, p)$
Game $\mathbf{G}_1(k)$	$(\langle E_0\rangle, g_0, n_0) \leftarrow G_0$; $(\langle E_1\rangle, g_1, n_1) \leftarrow G_1$
$Z \leftarrow\!\!\$\ \mathcal{A}^{\textsc{Init},\textsc{ddh}}(1^k)$	$\mathbb{G} \leftarrow E_0 \times E_1$; $g \leftarrow (g_0, g_1)$; $n \leftarrow n_0 \cdot n_1$
Return $(Z = g^{xy} \wedge Y \in E_1 \wedge G \neq \bot)$	$G' \leftarrow (\langle\mathbb{G}\rangle, n, g)$, $G \leftarrow (G', p)$
$\underline{\textsc{ddh}(\tilde{Y}, \tilde{Z})}$	$x \leftarrow\!\!\$\ \mathbb{Z}_n$
Return $(\tilde{Y}^x = \tilde{Z})$	$y \leftarrow\!\!\$\ \mathsf{EG}_{\text{twist-re}}.\mathsf{S}(1^k, \pi, G)$
	$X \leftarrow g^x$; $Y \leftarrow g^y$
	Return (G, X, Y)

Figure 28: Games for the proof of Lemma 5.11. Both games use the same procedures Init and ddh. In $\mathbf{G}_1$ we see E_1 as a subset of $E_0 \times E_1$ in the natural way.

- Case 2: $2^{k+1} - (2p+1) \leq z \leq 2p$.
  In this case $p(z) = \Pr\left[\mathsf{EG}_{\text{twist}}.\mathsf{E}(1^k, p, G', g^y) = z\right] = 2^{-(k+1)}$. Again the last equality is due to Claim 5.10 using that in this case $y \in M_1$.

- Case 3: $z > 2p$.
  Then $p(z) = \Pr\left[\mathsf{EG}_{\text{twist}}.\mathsf{E}(1^k, p, G', g^y) = z - (2p+1) \wedge b = 1\right] = 1/2 \cdot 2^{-k} = 2^{-(k+1)}$. Here the last equality is due to Claim 5.10.

Summing up $\mathsf{EG}_{\text{twist-re}}.\mathsf{E}(1^k, \pi, G, g^y)$ is uniformly distributed on $\mathsf{EG}_{\text{twist-re}}.\mathsf{ES}(k, \pi)$ for exponents y sampled with $\mathsf{EG}_{\text{twist-re}}.\mathsf{S}(1^k, \pi, G)$, which completes the proof. □

As in the case of $\mathsf{EG}^{\ell}_{\text{twist-rs}}$, we obtain that —assuming that TGen invoked on randomly sampled prime p returns a secure curve-twist pair— CDH-PSA with respect to eeg family $\mathsf{EG}_{\text{twist-re}}$ is hard.

Lemma 5.11. *Let $\mathsf{EG}_{twist\text{-}re}$ be the eeg family defined above with underlying curve-twist generator TGen. If the (strong) computational Diffie-Hellman assumption holds with respect to TGen, the (strong) computational Diffie-Hellman assumption holds with respect to $\mathsf{EG}_{twist\text{-}re}$.*

Concretely, for every adversary $\mathcal{A}$ against game $\mathbf{G}^{\text{cdh-psa}}_{\mathsf{EG}_{twist\text{-}re},\mathcal{A}}(\cdot)$ there exist adversaries $\mathcal{B}_0$, $\mathcal{B}_1$ against games $\mathbf{G}^{\text{twist}_0\text{-up-cdh}}_{\mathsf{TGen},\mathcal{B}_0}(\cdot)$ or $\mathbf{G}^{\text{twist}_1\text{-up-cdh}}_{\mathsf{TGen},\mathcal{B}_1}(\cdot)$ respectively running in the same time as $\mathcal{A}$ satisfying

$$\mathbf{Adv}^{\text{cdh-psa}}_{\mathsf{EG}_{twist\text{-}re},\mathcal{A}}(k) \leq 2\left(\mathbf{Adv}^{\text{twist}_0\text{-up-cdh}}_{\mathsf{TGen},\mathcal{B}_0}(k) + \mathbf{Adv}^{\text{twist}_1\text{-up-cdh}}_{\mathsf{TGen},\mathcal{B}_1}(k)\right).$$

Furthermore, for every adversary $\mathcal{A}$ against $\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG}_{twist\text{-}re},\mathcal{A}}(\cdot)$ making at most Q queries to its DDH*-oracle there exist adversaries $\mathcal{B}_0$, $\mathcal{B}_1$ against $\mathbf{G}^{\text{twist}_0\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(\cdot)$ or $\mathbf{G}^{\text{twist}_1\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_1}(\cdot)$ respectively running in the same time as $\mathcal{A}$ and making at most Q queries to their* DDH*-oracles, which satisfy*

$$\mathbf{Adv}^{\text{scdh-psa}}_{\mathsf{EG}_{twist\text{-}re},\mathcal{A}}(k) \leq 2\left(\mathbf{Adv}^{\text{twist}_0\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(k) + \mathbf{Adv}^{\text{twist}_1\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_1}(k)\right).$$

Proof. We prove the statement on sCDH-PSA. The statement on CDH-PSA can be shown analogously. Let $k \in \mathbb{N}$ and $\mathcal{A}$ be an adversary against the sCDH-PSA game with respect to $\mathsf{EG}_{\text{twist-re}}$. Consider games $\mathbf{G}_0$ and $\mathbf{G}_1$ defined in Figure 28. Note that $\mathbf{G}_0$ is the usual

Adversary $\mathcal{B}_0^{\text{INIT,DDH}}(1^k)$	SIMINIT(π)
$(Z_0, Z_1) \leftarrow\!\!\$\ \mathcal{A}^{\text{SIMINIT,SIMDDH}}(1^k)$	$(G_0, G_1, p, X_0, Y_0) \leftarrow\!\!\$$ INIT
Return Z_0	$(E_0, n_0, g_0) \leftarrow G_0$; $(E_1, n_1, g_1) \leftarrow G_1$
SIMDDH($(\tilde{Y}_0, \tilde{Y}_1), (\tilde{Z}_0, \tilde{Z}_1)$)	$G' \leftarrow (E_0 \times E_1, n_0 n_1, (g_0, g_1))$, $G \leftarrow (G', p)$
If $(\tilde{Y}_1^{x_1} = \tilde{Z}_1)$	$x_1 \leftarrow\!\!\$\ \mathbb{Z}_{n_1}$; $X_1 \leftarrow g_1^{x_1}$
then return DDH($\tilde{Y}_0, \tilde{Z}_0$)	$X \leftarrow (X_0, X_1)$; $Y \leftarrow (Y_0, \mathcal{O}_1)$
Return false	Return (G, X, Y)

Figure 29: Adversary for the proof of Lemma 5.11.

---

sCDH-PSA game with respect to $\mathsf{EG}_{\text{twist-re}}$ and adversary $\mathcal{A}$, and that condition $G \neq \perp$ always holds. We obtain

$$\mathbf{Adv}^{\text{scdh-psa}}_{\mathsf{EG}_{\text{twist-re}},\mathcal{A}}(k) = \Pr[\mathbf{G}_0(k)] \ . \tag{18}$$

In game $\mathbf{G}_0$ let d' denote the indicator random variable taking value 0 if $Y \in E_0$ and 1 if $Y \in E_1$. This yields

$$\Pr[\mathbf{G}_1(k)] = \Pr\big[\mathbf{G}_0(k) \wedge d' = 1\big] \leq \Pr\big[\mathbf{G}_0(k) \mid d' = 1\big] \ , \tag{19}$$

$$\Pr[\mathbf{G}_0(k)] - \Pr[\mathbf{G}_1(k)] = \Pr\big[\mathbf{G}_0(k) \wedge d' = 0\big] \leq \Pr\big[\mathbf{G}_0(k) \mid d' = 0\big] \ . \tag{20}$$

We construct adversaries $\mathcal{B}_0$ and $\mathcal{B}_1$ against games $\mathbf{G}^{\text{twist}_0\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(\cdot)$ and $\mathbf{G}^{\text{twist}_1\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_1}(\cdot)$ respectively, which satisfy

$$\Pr\big[\mathbf{G}_0(k) \mid d' = 0\big] \leq 2\mathbf{Adv}^{\text{twist}_0\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(k) \ , \tag{21}$$

$$\Pr\big[\mathbf{G}_0(k) \mid d' = 1\big] \leq 2\mathbf{Adv}^{\text{twist}_1\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_1}(k) \ . \tag{22}$$

Plugging this into Equations (18) to (20) yields

$$\mathbf{Adv}^{\text{scdh-psa}}_{\mathsf{EG}_{\text{twist-re}},\mathcal{A}}(k) \leq 2\left(\mathbf{Adv}^{\text{twist}_0\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(k) + \mathbf{Adv}^{\text{twist}_1\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_1}(k)\right)$$

as desired.

We show how to prove equation (21); equation (22) can be obtained in a similar way. Consider adversary $\mathcal{B}_0$ of Figure 29, which provides $\mathcal{A}$ with a simulation of game $\mathbf{G}_0(k)$ conditioned on the event $d' = 0$. Note that $\mathcal{A}$ solving its challenge implies that Z_0 is a correct solution to $\mathcal{B}_0$'s challenge.

We now proceed to analyze $\mathcal{B}_0$'s success probability. By definition of $\mathcal{B}_0$ the value G generated by $\mathcal{B}_0$ consisting of prime p and group G' is distributed as in $\mathbf{G}_0(k)$. Furthermore —as in game $\mathbf{G}_0(k)$— the group element X generated by $\mathcal{B}$ is uniformly distributed on $E_0 \times E_1$ and procedure SIMDDH is a perfect simulation of the DDH oracle of $\mathbf{G}_0(k)$. However by definition of $\mathbf{G}^{\text{twist}_0\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(k)$ the value Y is uniformly distributed on $E_0 \subseteq E_0 \times E_1$, while in $\mathbf{G}_0$ conditioned on $d' = 0$ it is distributed as $Y = g^y$, where $y \leftarrow\!\!\$\ \mathsf{EG}_{\text{twist-re}}.\mathsf{S}(1^k, \pi, G)$, with the additional condition that $Y \in E_0$. We conclude the proof by showing that $\mathcal{A}$ — even on input (G, X, Y) with $Y \sim U_{E_0}$ — looses at most a factor of 2 in its success probability of computing g^{xy}.

To prove this, we condition on G. We write $\Pr_{\mathbf{G}_0}$ and $\Pr_{\mathbf{G}_{\text{twist}_0}}$ to indicate, whether we consider probabilities in game $\mathbf{G}_0(k)$ or $\mathbf{G}^{\text{twist}_0\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(k)$. Let $\tilde{G} \in [\mathsf{EG}_{\text{twist-re}}.\mathsf{G}(1^k, \pi)]$, where $\tilde{G} = (\tilde{G}', \tilde{p})$ and $\tilde{G}' = (\langle \tilde{E}_0 \times \tilde{E}_1 \rangle, \tilde{n}_0\tilde{n}_1, (\tilde{g}_0, \tilde{g}_1))$. In a first step we show that for all $\tilde{Y}_0 \in \tilde{E}_0$

$$\Pr_{\mathbf{G}_0}\Big[Y = (\tilde{Y}_0, \tilde{\mathcal{O}}_1) \mid d' = 0 \wedge G = \tilde{G}\Big] \leq 2\Pr_{\mathbf{G}_{\text{twist}_0}}\Big[Y_0 = \tilde{Y}_0 \mid G = \tilde{G}\Big] \ . \tag{23}$$

As seen in the proof of Theorem 5.9, integers sampled by $\mathsf{EG}_{\text{twist-re}}.\mathsf{S}(1^k, \pi, \tilde{G}')$ either occur with probability 2^{-k} or $2^{-(k+1)}$. This yields $\Pr_{\mathbf{G}_0}\left[d' = 0 \mid G = \tilde{G}\right] \geq \tilde{n_0}/2^{k+1}$. Hence for all $\tilde{Y_0} \in \tilde{E_0}$

$$\begin{aligned}
&\frac{\Pr_{\mathbf{G}_0}\left[Y = (\tilde{Y_0}, \tilde{\mathcal{O}_1}) \mid d' = 0 \wedge G = \tilde{G}\right]}{\Pr_{\mathbf{G}_{\text{twist}_0}}\left[Y = \tilde{Y_0} \mid G = \tilde{G}\right]} \\
&= \frac{\Pr_{\mathbf{G}_0}\left[Y = (\tilde{Y_0}, \tilde{\mathcal{O}_1}) \wedge d' = 0 \mid G = \tilde{G}\right]}{\Pr_{\mathbf{G}_0}\left[d' = 0 \mid G = \tilde{G}\right] \cdot \Pr_{\mathbf{G}_{\text{twist}_0}}\left[Y_0 = \tilde{Y_0} \mid G = \tilde{G}\right]} \\
&\leq \frac{1}{2^k} \cdot \frac{2^{k+1}}{\tilde{n_0}} \cdot \tilde{n_0} = 2 \ .
\end{aligned}$$

This establishes (23).

Conditioned on the events $G = \tilde{G}$ and $Y_0 = \tilde{Y_0}$ adversary $\mathcal{B}_0$ provides $\mathcal{A}$ with a perfect simulation of $\mathbf{G}_0$ conditioned on $G = \tilde{G}$ and $Y = (\tilde{Y_0}, \tilde{\mathcal{O}_1})$, which in particular implies $d' = 0$. Using Equation (23) we obtain

$$\begin{aligned}
&\Pr_{\mathbf{G}_0}\left[\mathbf{G}_0(k) \mid d' = 0 \wedge G = \tilde{G}\right] \\
&= \sum_{\tilde{Y_0} \in \tilde{E_0}} \Pr_{\mathbf{G}_0}\left[\mathbf{G}_0(k) \mid Y = (\tilde{Y_0}, \tilde{\mathcal{O}_1}) \wedge d' = 0 \wedge G = \tilde{G}\right] \cdot \Pr_{\mathbf{G}_0}\left[Y = (\tilde{Y_0}, \tilde{\mathcal{O}_1}) \mid d' = 0 \wedge G = \tilde{G}\right] \\
&\overset{(23)}{\leq} \sum_{\tilde{Y_0} \in \tilde{E_0}} \Pr_{\mathbf{G}_{\text{twist}_0}}\left[\mathbf{G}^{\text{twist}_0\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(k) \mid Y_0 = \tilde{Y_0} \wedge G = \tilde{G}\right] \cdot 2 \Pr_{\mathbf{G}_{\text{twist}_0}}\left[Y_0 = \tilde{Y_0} \mid G = \tilde{G}\right] \\
&= 2 \Pr_{\mathbf{G}_{\text{twist}_0}}\left[\mathbf{G}^{\text{twist}_0\text{-up-scdh}}_{\mathsf{TGen},\mathcal{B}_0}(k) \mid G = \tilde{G}\right] \ .
\end{aligned}$$

An application of the law of total probability yields (21). □

6 Efficiently embeddable group families from Elligator curves

Let p a k-bit prime and $\mathbb{F}_p$ a field of order p. In [BHKL13] Bernstein *et al.* consider elliptic curves E with special properties defined over $\mathbb{F}_p$. They introduce two maps — Elligator 1 and Elligator 2 — mapping the set $[(p+1)/2]$ injectively to a subset $S \subseteq E$. Since by Hasse's theorem $|E| \approx (p+1)$, S covers approximately half of all curve points. To sample elements from S, [BHKL13] proposes an algorithm using rejection sampling. As pointed out in [BHKL13], sampling uniformly from S and applying the inverse of the injective map on the sampled curve point yields an uniformly distributed element of $[(p+1)/2]$.

In this section we discuss how this result may be interpreted as eeg family $\mathsf{EG}^{\ell}_{\text{ell}\delta}$ with p serving as parameter. We furthermore show, that sCDH-PSA with respect to $\mathsf{EG}^{\ell}_{\text{ell}\delta}$ is hard, if one is willing to assume that for *any choice* of p it is possible to find secure Elligator curves. In the last part of the section we construct a variant of $\mathsf{EG}^{\ell}_{\text{ell}\delta}$, which is parameter-free and whose security is based on the weaker assumption that for *randomly sampled* p it is possible to find secure Elligator curves.

6.1 Injective maps into elliptic curves

We first describe the two injective maps Elligator 1 and Elligator 2 of [BHKL13] mapping integers $z \in [(p+1)/2]$ to points on particular elliptic curves defined over $\mathbb{F}_p$, where p is a k-bit prime.

ELLIGATOR 1. The map for Elligator 1, which we denote by ι_1, may be defined for primes p with $p \equiv 3 \bmod 4$. It maps elements of $[(p+1)/2]$ to points of an appropriately chosen complete Edwards curve defined over $\mathbb{F}_p$. Complete Edwards curves are elliptic curves E defined by an equation of the form $x^2 + y^2 = 1 + dx^2y^2$ for some $d \in \mathbb{F}_p \backslash \{0, 1\}$. It is well known that Edwards curves contain a point of order 2, meaning E has even order. Elligator 1 furthermore requires that the curve parameter d can be written as $d = -(c+1)^2/(c-1)^2$, where $c = 2/s^2$ for some $s \in \mathbb{F}_p^*$ with $-2 \neq s^2 \neq 2$. As pointed out in [BHKL13], in this situation we have $d \in \mathbb{F}_p \backslash \{0, 1\}$. Hence E is indeed a complete Edwards curve. ι_1 is derived from an efficiently computable function $\phi \colon \mathbb{F}_p \to E$, which was first introduced in [FJT13], satisfying $\phi^{-1}(\phi(z)) = \{-z, z\}$ for all $z \in \mathbb{F}_p$. Hence restricting ϕ to the set $[(p+1)/2]$ yields an injective map $\iota_1 := \phi|_{[(p+1)/2]} \colon [(p+1)/2] \to E$. Since by Hasse's theorem $||E| - (p+1)| \leq 2\sqrt{p}$, the image of ι_1 covers roughly half of all curve points. Furthermore, ι_1 can be efficiently inverted and containment in the image of ι_1 can be efficiently checked.

ELLIGATOR 2. The map for Elligator 2, denoted by ι_2 is applicable to elliptic curves defined over a field $\mathbb{F}_p$, where p is a prime satisfying $p \equiv 1 \bmod 4$. It maps to a curve E defined by an equation of the form $y^2 = x^3 + ax^2 + bx$, where $a, b \in \mathbb{F}_p$ such that $ab(a^2 - 4b) \neq 0$ and $a^2 - 4b$ is not a square modulo p. Again, such curves contain a point of order 2, meaning E has even order. [BHKL13] constructs an efficiently computable map ψ mapping elements of $\mathbb{F}_p$ to E such that for all $z \in \mathbb{F}_p$ the preimage of $\psi(z)$ is given by $\{z, -z\}$. Analogous to the construction from above setting $\iota_2 := \psi|_{[(p+1)/2]} \colon [(p+1)/2] \to E$ yields an injective map with an image covering roughly half of all curve points to E. Again, ι_2 is efficiently invertible and membership of a point in $\mathrm{im}(\iota_2)$ can be efficiently checked.

ELLIGATOR CURVE GENERATORS. [BHKL13] points out that a fraction of roughly 1/16 of all elliptic curves E over all prime fields can be written as a complete Edwards curve fulfilling the additional conditions necessary for the application of Elligator 1. Elligator 2 on the other hand is applicable for a fraction of curves of even order over prime fields $\mathbb{F}_p$ with $p \equiv 1 \mod 4$. Hence it seems reasonable to assume similar as in Section 5.1 that it is possible to efficiently generate k-bit primes p and elliptic curves E over $\mathbb{F}_p$ compatible with the map Elligator 1 or Elligator 2 respectively. For $\delta \in \{1, 2\}$ we denote by $\mathcal{P}_{\delta,k}$ the set of k-bit primes, which are compatible with the Elligator δ map. That is

$$\mathcal{P}_{1,k} := \{p \in \mathcal{P}_k \mid p \equiv 3 \bmod 4\} \, ; \, \mathcal{P}_{2,k} := \{p \in \mathcal{P}_k \mid p \equiv 1 \bmod 4\} \quad .$$

Definition 6.1. *An Elligator* 1 *curve generator* EllGen_1 *on input of security parameter* 1^k *and prime* $p \in \mathcal{P}_{1,k}$ *returns a secure cyclic elliptic curve* $G = (\langle E \rangle, n, g)$*, where* E *is defined via the equation* $x^2 + y^2 = 1 + dx^2y^2$ *over prime field* $\mathbb{F}_p$*, has even order* n *and is generated by* g*. Here we require that* d *can be written as* $d = -(2/s^2 + 1)^2(2/s^2 - 1)^2$ *for some* $s \in \mathbb{F}_p^*$ *with* $-2 \neq s^2 \neq 2$*.*

An Elligator 2 *curve generator* EllGen_2 *on input of security parameter* 1^k *and prime* $p \in \mathcal{P}_{2,k}$ *returns a elliptic curve* $G = (\langle E \rangle, n, g)$*, where* E *is defined via the equation* $y^2 = x^3 + ax^2 + bx$ *over a primefield* $\mathbb{F}_p$*, has even order* n *and is generated by* g*. Here we require for* a *and* b *it holds that* $ab(a^2 - 4b) \neq 0$ *and* $a^2 - 4b$ *is not a square modulo* p*.*

COMPUTATIONAL PROBLEMS ASSOCIATED TO EllGen_1 AND EllGen_2. Let $\delta \in \{1, 2\}$ and EllGen_δ be an Elligator δ curve generator. Similar to Section 5 we give two variants of the strong computational Diffie-Hellman problem with respect to EllGen_δ. Consider games $\mathbf{G}^{\mathrm{ell}_\delta\text{-cp-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(\cdot)$

Game $\mathbf{G}^{\text{ell}_\delta\text{-cp-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(k)$	$\underline{\textsc{Init}(\pi)}$
$Z \leftarrow\!\!\$\ \mathcal{A}^{\textsc{Init},\textsc{ddh}}(1^k)$	$p \leftarrow \pi$
Return $(Z = g^{xy})$	If $(p \notin \mathcal{P}_{\delta,k})$ then return $\perp$
$\underline{\textsc{ddh}(\tilde{Y},\tilde{Z})}$	$G \leftarrow\!\!\$\ \mathsf{EllGen}_\delta(1^k, p)$
Return $(\tilde{Y}^x = \tilde{Z})$	$(\langle E\rangle, n, g) \leftarrow G$
	$x \leftarrow\!\!\$\ \mathbb{Z}_n,\ y \leftarrow\!\!\$\ \mathbb{Z}_n$
	$X \leftarrow g^x;\ Y \leftarrow g^y$
	Return (G, X, Y)

Game $\mathbf{G}^{\text{ell}_\delta\text{-up-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(k)$	$\underline{\textsc{Init}}$
$Z \leftarrow\!\!\$\ \mathcal{A}^{\textsc{Init},\textsc{ddh}}(1^k)$	$p \leftarrow\!\!\$\ \mathcal{P}_{\delta,k}$
Return $(Z = g^{xy})$	$G \leftarrow\!\!\$\ \mathsf{EllGen}_\delta(1^k, p)$
$\underline{\textsc{ddh}(\tilde{Y},\tilde{Z})}$	$(\langle E\rangle, n, g) \leftarrow G$
Return $(\tilde{Y}^x = \tilde{Z})$	$x \leftarrow\!\!\$\ \mathbb{Z}_n;\ y \leftarrow\!\!\$\ \mathbb{Z}_n$
	$X \leftarrow g^x,\ Y \leftarrow g^y$
	Return (G, p, X, Y)

Figure 30: Games for the strong Diffie-Hellman problem for chosen (uniform) primes with respect to Elligator δ curve generator EllGen_δ and adversary $\mathcal{A}$.

---

and $\mathbf{G}^{\text{ell}_\delta\text{-up-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(\cdot)$ of Figure 30. We define the advantage functions

$$\mathbf{Adv}^{\text{ell}_\delta\text{-cp-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(k) = \Pr\left[\mathbf{G}^{\text{ell}_\delta\text{-cp-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(k)\right] ,$$
$$\mathbf{Adv}^{\text{ell}_\delta\text{-up-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(k) = \Pr\left[\mathbf{G}^{\text{ell}_\delta\text{-up-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(k)\right] .$$

We say that the strong computational Diffie-Hellman assumption for chosen (uniform) primes holds with respect to EllGen_δ if the advantage $\mathbf{Adv}^{\text{ell}_\delta\text{-cp-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(\cdot)$ (or $\mathbf{Adv}^{\text{ell}_\delta\text{-up-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(\cdot)$ respectively) is negligible for every adversary $\mathcal{A}$. Since Elligator curves are always of even order hence transform **eegToKE2** cannot be applied, we do not consider CDH over Elligator curves.

6.2 An eeg family from Elligator curves

Let $k \in \mathbb{N}$, $\delta \in \{1, 2\}$, $p \in \mathcal{P}_{\delta,k}$ and EllGen_δ an Elligator δ generator. For $G = (\langle E\rangle, n, g) \in [\mathsf{EllGen}_\delta(1^k, p)]$ we let $\iota_\delta \colon [(p+1)/2] \to E$ denote the Elligator δ map. In [BHKL13] Bernstein *et al.* propose to use rejection sampling to sample uniformly from $\text{im}(\iota_\delta)$. Curve points sampled in this way then can be embedded in $[(p+1)/2]$ using ι_δ^{-1}. Generating an Elligator δ curve and using the sampling and embedding algorithms described above can be seen as an eeg family with embedding space $[(p+1)/2]$, where the prime p serves as parameter. A comprehensive description of the eeg family $\mathsf{EG}^\ell_{\text{ell}\delta}$ may be found in Figure 31. We obtain the following.

Lemma 6.2. *Let $\ell \colon \mathbb{N} \to \mathbb{N}$ be a polynomial, $\delta \in \{1, 2\}$ and EllGen_δ an Elligator δ curve generator. $\mathsf{EG}^\ell_{ell\delta}$ of Figure 31 is an eeg family with embedding space $\mathsf{EG}^\ell_{ell\delta}.\mathsf{ES}(k, \pi) := [(p+1)/2]$, where $\pi = p$. Further it has pseudorandom embeddings and for $k \geq 4$ its inversion error is bounded by $(2/3)^{\ell(k)}$. More precisely for every (potentially unbounded) adversary $\mathcal{A}$ and $k \geq 4$ we have*

$$\mathbf{Adv}^{\text{epr-psa}}_{\mathsf{EG}^\ell_{ell\delta},\mathcal{A}}(k) \leq (2/3)^{\ell(k)} .$$

Proof. Let $k \in \mathbb{N}_{\geq 4}$, $p = \pi \in \mathsf{EG}^\ell_{\text{ell}\delta}.\mathsf{P}(1^k)$ and $G \in [\mathsf{EG}^\ell_{\text{ell}\delta}.\mathsf{G}(1^k, \pi)]$. By $\mathcal{L}$ we denote the event

that $\mathsf{EG}^{\ell}_{\mathrm{ell}\delta}.\mathsf{S}(1^k,\pi,G)$ outputs $\perp$. We show $\Pr[\mathcal{L}] \leq (2/3)^{\ell(k)}$. Then the result follows analogous to Lemma 5.7.

By Hasse's inequality $n \leq p+1+2\sqrt{p}$. Furthermore $|\mathrm{im}(\iota_\delta)| = (p+1)/2$. Hence

$$\Pr[y \notin \mathrm{im}(\iota_\delta)] \leq 1 - \frac{(p+1)/2}{p+1+2\sqrt{p}} \leq 1 - \frac{1}{2+2^{-k/2+2}} \leq \frac{2}{3} \ ,$$

where the last inequality holds since $k \geq 4$. This implies $\Pr[\mathcal{L}] \leq (2/3)^{\ell(k)}$. □

Assuming that there is no inherently bad choice of the parameter $\pi = p$, i. e. it is possible to find secure Elligator δ curves independently of the (possibly subverted) parameter π, sCDH-PSA is hard with respect to $\mathsf{EG}^{\ell}_{\mathrm{ell}\delta}$. More formally, we obtain the following.

Lemma 6.3. *Let $\ell : \mathbb{N} \to \mathbb{N}$ be a polynomial, $\delta \in \{1,2\}$, EllGen_δ an Elligator δ curve generator If the strong computational Diffie-Hellman assumption for chosen primes holds with respect to EllGen_δ, then the strong computational Diffie-Hellman assumption holds with respect to $\mathsf{EG}^{\ell}_{ell\delta}$.*

More precisely, let $\mathcal{A}$ be an adversary against $\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{ell\delta\text{-}rs},\mathcal{A}}(\cdot)$. Then for all $k \geq 4$

$$\mathbf{Adv}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{ell\delta},\mathcal{A}}(k) \leq 3\mathbf{Adv}^{\text{ell}_\delta\text{-cp-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(k) + (2/3)^{\ell(k)} \ .$$

Proof. Let $k \in \mathbb{N}_{\geq 4}$. We condition on parameter π and group G. Note that $\mathcal{A}$ by definition of $\mathsf{EG}^{\ell}_{\mathrm{ell}\delta}.\mathsf{G}$ cannot succeed in the sCDH-PSA game with respect to $\mathsf{EG}^{\ell}_{\mathrm{ell}\delta}$, if it does not provide a parameter satisfying $\pi \in \mathcal{P}_{\delta,k}$. Hence let $\tilde{p} = \tilde{\pi} \in \mathcal{P}_{\delta,k}$ and $\tilde{G} \in [\mathsf{EG}^{\ell}_{\mathrm{ell}\delta}.\mathsf{G}(k,\tilde{\pi})]$, where $\tilde{G} = (\langle \tilde{E} \rangle, \tilde{n}, \tilde{g})$. Note that games $\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{\mathrm{ell}\delta},\mathcal{A}}(k)$ and $\mathbf{G}^{\text{ell}_\delta\text{-cp-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(k)$ conditioned on the events $\pi = \tilde{\pi}$ and $G = \tilde{G}$ only differ in the distribution of the group element Y. In $\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{\mathrm{ell}\delta},\mathcal{A}}(k)$ conditioned on the events $\pi = \tilde{\pi}$, $G = \tilde{G}$ and $y \neq \perp$ the element Y is distributed uniformly on the set $\mathrm{im}(\iota_\delta)$, which has size $(p+1)/2$. On the other Hand, in $\mathbf{G}^{\text{ell}_\delta\text{-cp-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(k)$ conditioned on the events $\pi = \tilde{\pi}$ and $G = \tilde{G}$ the element Y is distributed uniformly on $\tilde{E}$. We write $\Pr_{\mathbf{G}_{\mathrm{eg}}}$ and $\Pr_{\mathbf{G}_{\mathrm{ellgen}}}$ to indicate whether probabilities are taken in game $\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{\mathrm{ell}\delta},\mathcal{A}}(k)$ or $\mathbf{G}^{\text{ell}_\delta\text{-cp-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(k)$. Using Hasse's inequality we obtain for every $\tilde{Y} \in \tilde{E}$

$$\frac{\Pr_{\mathbf{G}_{\mathrm{eg}}}\left[Y = \tilde{Y} \mid y \neq \perp \wedge G = \tilde{G} \wedge \pi = \tilde{\pi}\right]}{\Pr_{\mathbf{G}_{\mathrm{ellgen}}}\left[Y = \tilde{Y} \mid G = \tilde{G} \wedge \pi = \tilde{\pi}\right]} \leq \frac{p+1+2\sqrt{p}}{(p+1)/2} \leq 3 \ , \tag{24}$$

where the last inequality holds since $k \geq 4$. Conditioning on Y yields

$$\begin{aligned}
&\Pr_{\mathbf{G}_{\mathrm{eg}}}\left[\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{\mathrm{ell}\delta},\mathcal{A}}(k) \mid y \neq \perp \wedge G = \tilde{G} \wedge \pi = \tilde{\pi}\right] \\
&\leq \sum_{\tilde{Y} \in \tilde{E}} \Pr_{\mathbf{G}_{\mathrm{eg}}}\left[\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{\mathrm{ell}\delta},\mathcal{A}}(k) \mid Y = \tilde{Y} \wedge G = \tilde{G} \wedge \pi = \tilde{\pi}\right] \cdot \Pr_{\mathbf{G}_{\mathrm{eg}}}\left[Y = \tilde{Y} \mid y \neq \perp \wedge G = \tilde{G} \wedge \pi = \tilde{\pi}\right] \\
&\overset{(27)}{\leq} 3 \sum_{\tilde{Y} \in \tilde{E}} \Pr_{\mathbf{G}_{\mathrm{ellgen}}}\left[\mathbf{G}^{\text{ell}_\delta\text{-cp-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(k) \mid Y = \tilde{Y} \wedge G = \tilde{G} \wedge \pi = \tilde{\pi}\right] \cdot \Pr_{\mathbf{G}_{\mathrm{ellgen}}}\left[Y = \tilde{Y} \mid G = \tilde{G} \wedge \pi = \tilde{\pi}\right] \\
&= 3 \Pr_{\mathbf{G}_{\mathrm{ellgen}}}\left[\mathbf{G}^{\text{ell}_\delta\text{-cp-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(k) \mid G = \tilde{G} \wedge \pi = \tilde{\pi}\right] \ .
\end{aligned} \tag{25}$$

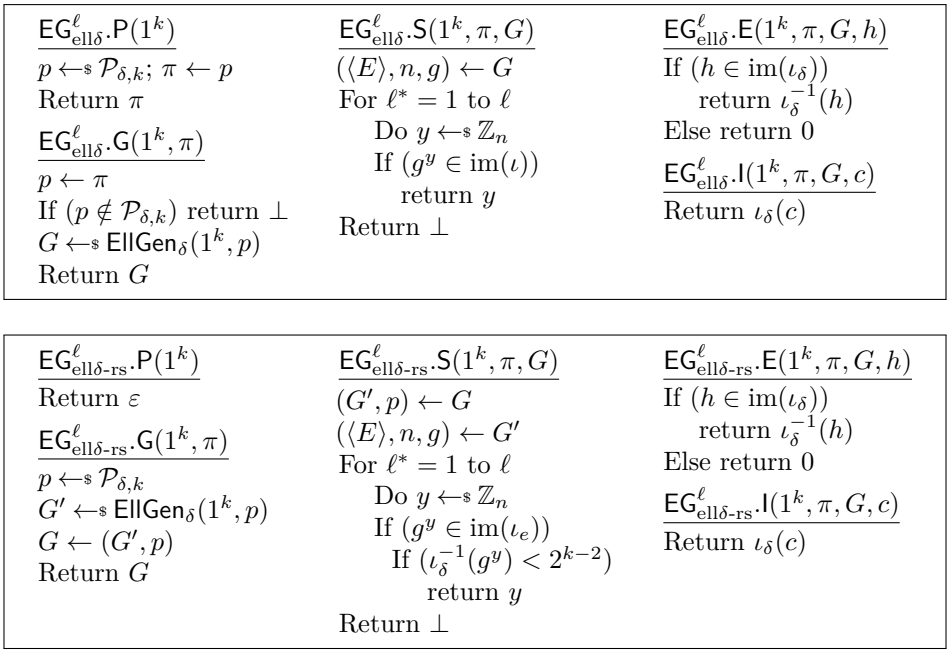


Figure 31: Description of eeg families $\mathsf{EG}^{\ell}_{\mathrm{ell}\delta}$ and $\mathsf{EG}^{\ell}_{\mathrm{ell}\delta\text{-rs}}$ for $\delta \in \{1,2\}$ and Elligator δ generator EllGen_{δ}. ι_δ denotes the injective Elligator δ map $[(p+1)/2] \to E$.

Furthermore,

$$\begin{aligned}
&\Pr_{\mathbf{G}_{\mathrm{eg}}}\left[\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{\mathrm{ell}\delta},\mathcal{A}}(k) \mid G = \tilde{G} \wedge \pi = \tilde{\pi}\right] \\
\leq\ &\Pr_{\mathbf{G}_{\mathrm{eg}}}\left[\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{\mathrm{ell}\delta},\mathcal{A}}(k) \mid y \neq \perp \wedge G = \tilde{G} \wedge \pi = \tilde{\pi}\right] + \Pr_{\mathbf{G}_{\mathrm{eg}}}\left[y \neq \perp \mid G = \tilde{G} \wedge \pi = \tilde{\pi}\right] \\
\leq\ &\Pr_{\mathbf{G}_{\mathrm{eg}}}\left[\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{\mathrm{ell}\delta},\mathcal{A}}(k) \mid y \neq \perp \wedge G = \tilde{G} \wedge \pi = \tilde{\pi}\right] + (2/3)^{\ell(k)}. \qquad (26)
\end{aligned}$$

Here we use, that $\Pr_{\mathbf{G}_{\mathrm{eg}}}\left[y = \perp \mid G = \tilde{G} \wedge \pi = \tilde{\pi}\right] \leq (2/3)^{\ell(k)}$ for all parameters $\tilde{\pi} \in \mathcal{P}_{\delta,k}$ and groups $\tilde{G} \in [\mathsf{EG}^{\ell}_{\mathrm{ell}\delta}.\mathsf{G}\delta(1^k,\tilde{\pi})]$, which was shown in the proof of Lemma 6.2. Combining Equations (25) and (26) and an application of the law of total probability yields the claim. □

6.3 A parameter-free eeg family from Elligator curves

Let $\delta \in \{1,2\}$. In this section we give an alternative construction $\mathsf{EG}^{\ell}_{\mathrm{ell}\delta\text{-rs}}$ of an eeg family from Elligator δ. The construction is parameter-free; prime p used to sample the Elligator curve is now generated during the execution of $\mathsf{EG}^{\ell}_{\mathrm{ell}\delta\text{-rs}}.\mathsf{G}$. The property that embeddings are elements of the new embedding space $\mathsf{EG}^{\ell}_{\mathrm{ell}\delta\text{-rs}}.\mathsf{ES}(k,\pi) = [2^{k-2}]$ is ensured by imposing a second rejection condition in the sampling algorithm. A comprehensive description of the algorithms of $\mathsf{EG}^{\ell}_{\mathrm{ell}\delta\text{-rs}}$ may be found in Figure 31. We obtain the following.

Lemma 6.4. *Let $\ell\colon \mathbb{N} \to \mathbb{N}$ be a polynomial, $\delta \in \{1,2\}$ and EllGen_{δ} an Elligator δ curve generator. $\mathsf{EG}^{\ell}_{ell\delta\text{-}rs}$ of Figure 31 is an eeg family with embedding space $\mathsf{EG}^{\ell}_{ell\delta}.\mathsf{ES}(k,\pi) := [2^{k-2}]$. Further it has pseudorandom embeddings and for $k \geq 6$ its inversion error is bounded by $(4/5)^{\ell(k)}$. More precisely for every (potentially unbounded) adversary $\mathcal{A}$ and $k \geq 6$ we have*

$$\mathbf{Adv}^{\text{epr-psa}}_{\mathsf{EG}^{\ell}_{ell\delta},\mathcal{A}}(k) \leq (4/5)^{\ell(k)} \ .$$

Proof. Let $k \in \mathbb{N}$, $p = \pi \in \mathsf{EG}^{\ell}_{\mathrm{ell}\delta\text{-rs}}.\mathsf{P}(1^k)$ and $G \in [\mathsf{EG}^{\ell}_{\mathrm{ell}\delta\text{-rs}}.\mathsf{G}(1^k, \pi)]$. By $\mathcal{L}$ we denote the event that $\mathsf{EG}^{\ell}_{\mathrm{ell}\delta\text{-rs}}.\mathsf{S}(1^k, \pi, G)$ outputs $\perp$. We show $\Pr[\mathcal{L}] \leq (4/5)^{\ell(k)}$. Then the result follows analogous to Theorem 5.7.

Let $M := \{y \in \mathbb{Z}_n \colon g^y \in \mathrm{im}(\iota_\delta) \wedge \iota_\delta^{-1}(g^y) < 2^{k-2}\}$. Then $|M| = 2^{k-2}$. Further by Hasse's inequality $n \leq p + 1 + 2\sqrt{p} \leq 2^k + 2^{k/2+1}$. We obtain

$$\Pr[y \notin M] \leq 1 - \frac{2^{k-2}}{2^k + 2^{k/2+1}} = 1 - \frac{1}{4 + 2^{-k/2+3}} \leq \frac{4}{5} \ ,$$

where the last inequality holds since $k \geq 6$. This implies $\Pr[\mathcal{L}] \leq (4/5)^{\ell(k)}$. □

Assuming that for randomly chosen p it is possible to find secure Elligator δ curves, sCDH-PSA is hard with respect to $\mathsf{EG}^{\ell}_{\mathrm{ell}\delta}$. More formally, we obtain the following.

Lemma 6.5. *Let $\ell : \mathbb{N} \to \mathbb{N}$ be a polynomial, $\delta \in \{1, 2\}$, EllGen_δ an Elligator δ curve generator. If the strong computational Diffie-Hellman assumption for uniform primes holds with respect to EllGen_δ, then the strong computational Diffie-Hellman assumption holds with respect to $\mathsf{EG}^{\ell}_{ell\delta\text{-}rs}$.*

More precisely, let $\mathcal{A}$ be an adversary against $\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{ell\delta\text{-}rs},\mathcal{A}}(\cdot)$. Then for all $k \in \mathbb{N}_{\geq 6}$

$$\mathbf{Adv}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{ell\delta\text{-}rs},\mathcal{A}}(k) \leq 5\mathbf{Adv}^{\text{ell}_\delta\text{-up-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(k) + (4/5)^{\ell(k)} \ .$$

Proof. Let $k \in \mathbb{N}_{\geq 6}$. Consider $\mathbf{G}^{\text{ell}_\delta\text{-up-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(k)$ of Figure 30. For consistency with the sCDH-PSA game with respect to $\mathsf{EG}^{\ell}_{\mathrm{ell}\delta\text{-rs}}$ we write $G = (G', p)$, where p is the prime and G' the Elligator curve returned on queries to INIT in game $\mathbf{G}^{\text{ell}_\delta\text{-up-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(k)$. Note that the game differs from game $\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{\mathrm{ell}\delta\text{-rs}},\mathcal{A}}(k)$ of Figure 13 only in the distribution of the group element Y. Let $(\tilde{G}', \tilde{p}) = \tilde{G} \in [\mathsf{EG}^{\ell}_{\mathrm{ell}\delta\text{-rs}}.\mathsf{G}(k, \pi)]$, where $\tilde{G}' = (\langle \tilde{E} \rangle, \tilde{n}, \tilde{g})$. In $\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{\mathrm{ell}\delta\text{-rs}},\mathcal{A}}(k)$ conditioned on the events $G = \tilde{G}$ and $y \neq \perp$ the element Y is distributed uniformly on the set $\{\tilde{Y} \in \mathrm{im}(\iota_\delta) \mid \mathsf{EG}^{\ell}_{\mathrm{ell}\delta\text{-rs}}.\mathsf{E}(1^k, \pi, \tilde{G}, \tilde{Y}) < 2^{k-2}\}$, which has size 2^{k-2}. On the other Hand, in $\mathbf{G}^{\text{ell}_\delta\text{-up-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(k)$ conditioned on the event $G = \tilde{G}$ the element Y is distributed uniformly on $\tilde{E}$. We write $\Pr_{\mathbf{G}_{\mathrm{eg}}}$ and $\Pr_{\mathbf{G}_{\mathrm{ellgen}}}$ to indicate whether probabilities are taken in game $\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{\mathrm{ell}\delta\text{-rs}},\mathcal{A}}(k)$ or $\mathbf{G}^{\text{ell}_\delta\text{-up-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(k)$. Using Hasse's inequality we obtain for every $\tilde{Y} \in \tilde{E}$

$$\frac{\Pr_{\mathbf{G}_{\mathrm{eg}}}\left[Y = \tilde{Y} \mid y \neq \perp \wedge G = \tilde{G}\right]}{\Pr_{\mathbf{G}_{\mathrm{ellgen}}}\left[Y = \tilde{Y} \mid G = \tilde{G}\right]} \leq \frac{2^k + 2^{k/2+1}}{2^{k-2}} \leq 5 \ , \tag{27}$$

where the last inequality holds since $k \geq 6$. Conditioning on Y yields

$$\begin{aligned}
&\Pr_{\mathbf{G}_{\mathrm{eg}}}\left[\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{\mathrm{ell}\delta\text{-rs}},\mathcal{A}}(k) \mid y \neq \perp \wedge G = \tilde{G}\right] \\
&\leq \sum_{\tilde{Y} \in \tilde{E}} \Pr_{\mathbf{G}_{\mathrm{eg}}}\left[\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{\mathrm{ell}\delta\text{-rs}},\mathcal{A}}(k) \mid Y = \tilde{Y} \wedge G = \tilde{G}\right] \cdot \Pr_{\mathbf{G}_{\mathrm{eg}}}\left[Y = \tilde{Y} \mid y \neq \perp \wedge G = \tilde{G}\right] \\
&\overset{(27)}{\leq} 5 \sum_{\tilde{Y} \in \tilde{E}} \Pr_{\mathbf{G}_{\mathrm{ellgen}}}\left[\mathbf{G}^{\text{ell}_\delta\text{-up-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(k) \mid Y = \tilde{Y} \wedge G = \tilde{G}\right] \cdot \Pr_{\mathbf{G}_{\mathrm{ellgen}}}\left[Y = \tilde{Y} \mid G = \tilde{G}\right] \\
&= 5 \Pr_{\mathbf{G}_{\mathrm{ellgen}}}\left[\mathbf{G}^{\text{ell}_\delta\text{-up-scdh}}_{\mathsf{EllGen}_\delta,\mathcal{A}}(k) \mid G = \tilde{G}\right] \ .
\end{aligned} \tag{28}$$

Furthermore,

$$\begin{aligned}
&\Pr_{\mathbf{G}_{\mathrm{eg}}}\left[\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{\text{ell}\delta\text{-rs}},\mathcal{A}}(k) \mid G = \tilde{G}\right] \\
\leq\ &\Pr_{\mathbf{G}_{\mathrm{eg}}}\left[\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{\text{ell}\delta\text{-rs}},\mathcal{A}}(k) \mid y \neq \perp \wedge G = \tilde{G}\right] + \Pr_{\mathbf{G}_{\mathrm{eg}}}\left[y \neq \perp \mid G = \tilde{G}\right] \\
\leq\ &\Pr_{\mathbf{G}_{\mathrm{eg}}}\left[\mathbf{G}^{\text{scdh-psa}}_{\mathsf{EG}^{\ell}_{\text{ell}\delta\text{-rs}},\mathcal{A}}(k) \mid y \neq \perp \wedge G = \tilde{G}\right] + (4/5)^{\ell(k)}. \qquad (29)
\end{aligned}$$

Here we use, that $\Pr_{\mathbf{G}_{\mathrm{eg}}}\left[y = \perp \mid G = \tilde{G}\right] \leq (4/5)^{\ell(k)}$ for all $\tilde{G} \in [\mathsf{EllGen}_e(1^k, \pi)]$, which was shown in the proof of Lemma 6.4. Combining Equations (28) and (29) and an application of the law of total probability yields the claim of the lemma. □

Range Expansion? It seems not possible to transform $\mathsf{EG}^{\ell}_{\text{ell}\delta}$ into a parameter-free eeg family using the range expansion technique applied in Section 5.4 to twisted elliptic curves. This technique for each Elligator δ curve with corresponding Elligator map ι_δ would require an efficiently computable bijection ψ mapping elements of the set $[(p+1)/2]$ to $\{y \in \mathbb{Z}_n \mid g^y \in \mathrm{im}(\iota_\delta)\}$. However we are not aware of the existence of maps with this property, which would furthermore give a way of sampling uniformly from $\{y \in \mathbb{Z}_n \mid g^y \in \mathrm{im}(\iota_\delta)\}$ without having to rely on rejection sampling.

References

[ABC+05] Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 205–222, Santa Barbara, CA, USA, August 14–18, 2005. Springer, Heidelberg, Germany. (Cited on page 9, 12.)

[ABR01] Michel Abdalla, Mihir Bellare, and Phillip Rogaway. The oracle Diffie-Hellman assumptions and an analysis of DHIES. In David Naccache, editor, *Topics in Cryptology – CT-RSA 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 143–158, San Francisco, CA, USA, April 8–12, 2001. Springer, Heidelberg, Germany. (Cited on page 6, 22.)

[AMV15] Giuseppe Ateniese, Bernardo Magri, and Daniele Venturi. Subversion-resilient signature schemes. In Indrajit Ray, Ninghui Li, and Christopher Kruegel:, editors, *ACM CCS 15: 22nd Conference on Computer and Communications Security*, pages 364–375, Denver, CO, USA, October 12–16, 2015. ACM Press. (Cited on page 8.)

[BBDP01] Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 566–582, Gold Coast, Australia, December 9–13, 2001. Springer, Heidelberg, Germany. (Cited on page 3, 5, 9, 12.)

[BBM00] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In Bart Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *Lecture Notes in*

Computer Science, pages 259–274, Bruges, Belgium, May 14–18, 2000. Springer, Heidelberg, Germany. (Cited on page 3.)

[BCC+14] Daniel J. Bernstein, Tung Chou, Chitchanok Chuengsatiansup, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, and Christine van Vredendaal. How to manipulate curve standards: a white paper for the black hat. Cryptology ePrint Archive, Report 2014/571, 2014. `http://eprint.iacr.org/2014/571`. (Cited on page 3.)

[BDF+15] Thomas Baignères, Cécile Delerablée, Matthieu Finiasz, Louis Goubin, Tancrède Lepoint, and Matthieu Rivain. Trap me if you can – million dollar curve. Cryptology ePrint Archive, Report 2015/1249, 2015. `http://eprint.iacr.org/2015/1249`. (Cited on page 3.)

[BDL+11] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-speed high-security signatures. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems – CHES 2011*, volume 6917 of *Lecture Notes in Computer Science*, pages 124–142, Nara, Japan, September 28 – October 1, 2011. Springer, Heidelberg, Germany. (Cited on page 3.)

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45, Santa Barbara, CA, USA, August 23–27, 1998. Springer, Heidelberg, Germany. (Cited on page 3.)

[BFS16] Mihir Bellare, Georg Fuchsbauer, and Alessandra Scafuro. NIZKs with an untrusted CRS: Security in the face of parameter subversion. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016, Part II*, volume 10032 of *Lecture Notes in Computer Science*, pages 777–804, Hanoi, Vietnam, December 4–8, 2016. Springer, Heidelberg, Germany. (Cited on page 3, 8.)

[BH15] Mihir Bellare and Viet Tung Hoang. Resisting randomness subversion: Fast deterministic and hedged public-key encryption in the standard model. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part II*, volume 9057 of *Lecture Notes in Computer Science*, pages 627–656, Sofia, Bulgaria, April 26–30, 2015. Springer, Heidelberg, Germany. (Cited on page 8.)

[BHKL13] Daniel J. Bernstein, Mike Hamburg, Anna Krasnova, and Tanja Lange. Elligator: elliptic-curve points indistinguishable from uniform random strings. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 13: 20th Conference on Computer and Communications Security*, pages 967–980, Berlin, Germany, November 4–8, 2013. ACM Press. (Cited on page 7, 42, 43, 44.)

[BJK15] Mihir Bellare, Joseph Jaeger, and Daniel Kane. Mass-surveillance without the state: Strongly undetectable algorithm-substitution attacks. In Indrajit Ray, Ninghui Li, and Christopher Kruegel:, editors, *ACM CCS 15: 22nd Conference on Computer and Communications Security*, pages 1431–1440, Denver, CO, USA, October 12–16, 2015. ACM Press. (Cited on page 8.)

[BL] Daniel J. Bernstein and Tanja Lange. Safecurves: choosing safe curves for elliptic-curve cryptography. `https://safecurves.cr.yp.to`. Accessed: 18 May 2016. (Cited on page 29, 30.)

[BLN15] Daniel J. Bernstein, Tanja Lange, and Ruben Niederhagen. Dual EC: A standardized back door. Cryptology ePrint Archive, Report 2015/767, 2015. `http://eprint.iacr.org/2015/767`. (Cited on page 3.)

[BPR14] Mihir Bellare, Kenneth G. Paterson, and Phillip Rogaway. Security of symmetric encryption against mass surveillance. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 1–19, Santa Barbara, CA, USA, August 17–21, 2014. Springer, Heidelberg, Germany. (Cited on page 8.)

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press. (Cited on page 8, 9, 13.)

[BR06] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Heidelberg, Germany. (Cited on page 8.)

[CCG+16] Stephen Checkoway, Shaanan Cohney, Christina Garman, Matthew Green, Nadia Heninger, Jacob Maskiewicz, Eric Rescorla, Hovav Shacham, and Ralf-Philipp Weinmann. A systematic analysis of the juniper dual ec incident. In *Proceedings of the 23rd ACM conference on Computer and communications security*. ACM, 2016. (Cited on page 3.)

[CKS08] David Cash, Eike Kiltz, and Victor Shoup. The twin Diffie-Hellman problem and applications. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 127–145, Istanbul, Turkey, April 13–17, 2008. Springer, Heidelberg, Germany. (Cited on page 23, 27.)

[CPs07] Ran Canetti, Rafael Pass, and abhi shelat. Cryptography from sunspots: How to use an imperfect reference string. In *48th Annual Symposium on Foundations of Computer Science*, pages 249–259, Providence, RI, USA, October 20–23, 2007. IEEE Computer Society Press. (Cited on page 8.)

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25, Santa Barbara, CA, USA, August 23–27, 1998. Springer, Heidelberg, Germany. (Cited on page 3.)

[CS03] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003. (Cited on page 4, 9, 12, 17, 22.)

[DFP15] Jean Paul Degabriele, Pooya Farshim, and Bertram Poettering. A more cautious approach to security against mass surveillance. In Gregor Leander, editor, *Fast Software Encryption – FSE 2015*, volume 9054 of *Lecture Notes in Computer Science*, pages 579–598, Istanbul, Turkey, March 8–11, 2015. Springer, Heidelberg, Germany. (Cited on page 8.)

[DGG+15] Yevgeniy Dodis, Chaya Ganesh, Alexander Golovnev, Ari Juels, and Thomas Ristenpart. A formal treatment of backdoored pseudorandom generators. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 101–126, Sofia, Bulgaria, April 26–30, 2015. Springer, Heidelberg, Germany. (Cited on page 3, 4, 8.)

[DPSW16] Jean Paul Degabriele, Kenneth G. Paterson, Jacob C. N. Schuldt, and Joanne Woodage. Backdoors in pseudorandom number generators: Possibility and impossibility results. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 403–432, Santa Barbara, CA, USA, August 14–18, 2016. Springer, Heidelberg, Germany. (Cited on page 8.)

[FJT13] Pierre-Alain Fouque, Antoine Joux, and Mehdi Tibouchi. Injective encodings to elliptic curves. In Colin Boyd and Leonie Simpson, editors, *ACISP 13: 18th Australasian Conference on Information Security and Privacy*, volume 7959 of *Lecture Notes in Computer Science*, pages 203–218, Brisbane, Australia, July 1–3, 2013. Springer, Heidelberg, Germany. (Cited on page 43.)

[FPRE15] Jean-Pierre Flori, Jérôme Plût, Jean-René Reinhard, and Martin Ekerå. Diversity and transparency for ecc. Cryptology ePrint Archive, Report 2015/659, 2015. `http://eprint.iacr.org/`. (Cited on page 29, 30.)

[Fre98] Gerhard Frey. How to disguise an elliptic curve (weil descent). Talk given at ECC 1998, 1998. (Cited on page 6.)

[GGJS11] Sanjam Garg, Vipul Goyal, Abhishek Jain, and Amit Sahai. Bringing people of different beliefs together to do UC. In Yuval Ishai, editor, *TCC 2011: 8th Theory of Cryptography Conference*, volume 6597 of *Lecture Notes in Computer Science*, pages 311–328, Providence, RI, USA, March 28–30, 2011. Springer, Heidelberg, Germany. (Cited on page 8.)

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984. (Cited on page 9.)

[GM00] Steven D. Galbraith and James McKee. The probability that the number of points on an elliptic curve over a finite field is prime. *Journal of the London Mathematical Society*, 62(3):671–684, 2000. (Cited on page 30.)

[GO07] Jens Groth and Rafail Ostrovsky. Cryptography in the multi-string model. In Alfred Menezes, editor, *Advances in Cryptology – CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 323–341, Santa Barbara, CA, USA, August 19–23, 2007. Springer, Heidelberg, Germany. (Cited on page 8.)

[HOT04] Ryotaro Hayashi, Tatsuaki Okamoto, and Keisuke Tanaka. An RSA family of trapdoor permutations with a common domain and its applications. In Feng Bao, Robert Deng, and Jianying Zhou, editors, *PKC 2004: 7th International Workshop on Theory and Practice in Public Key Cryptography*, volume 2947 of *Lecture Notes in Computer Science*, pages 291–304, Singapore, March 1–4, 2004. Springer, Heidelberg, Germany. (Cited on page 7, 28, 38.)

[Kal91] Burton S. Kaliski Jr. One-way permutations on elliptic curves. *Journal of Cryptology*, 3(3):187–199, 1991. (Cited on page 7, 28, 29, 30.)

[KKZZ14] Jonathan Katz, Aggelos Kiayias, Hong-Sheng Zhou, and Vassilis Zikas. Distributing the setup in universally composable multi-party computation. In Magnús M. Halldórsson and Shlomi Dolev, editors, *33rd ACM Symposium Annual on Principles of Distributed Computing*, pages 20–29, Paris, France, July 15–18, 2014. Association for Computing Machinery. (Cited on page 8.)

[LM10] M. Lochter and J. Mekle. *RFC 5639: ECC Brainpool Standard Curves & Curve Generation*. Internet Engineering Task Force, March 2010. (Cited on page 7.)

[Möl04] Bodo Möller. A public-key encryption scheme with pseudo-random ciphertexts. In Pierangela Samarati, Peter Y. A. Ryan, Dieter Gollmann, and Refik Molva, editors, *ESORICS 2004: 9th European Symposium on Research in Computer Security*, volume 3193 of *Lecture Notes in Computer Science*, pages 335–351, Sophia Antipolis, French Riviera, France, September 13–15, 2004. Springer, Heidelberg, Germany. (Cited on page 4, 7.)

[NIS13] NIST. Digital signature standard (DSS), 2013. FIPS PUB 186-4. (Cited on page 3.)

[Orm98] H Orman. The oakley key determination protocol, 1998. (Cited on page 3.)

[PQ12] Christophe Petit and Jean-Jacques Quisquater. On polynomial systems arising from a Weil descent. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology – ASIACRYPT 2012*, volume 7658 of *Lecture Notes in Computer Science*, pages 451–466, Beijing, China, December 2–6, 2012. Springer, Heidelberg, Germany. (Cited on page 6.)

[RTYZ16] Alexander Russell, Qiang Tang, Moti Yung, and Hong-Sheng Zhou. Cliptography: Clipping the power of kleptographic attacks. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016, Part II*, volume 10032 of *Lecture Notes in Computer Science*, pages 34–64, Hanoi, Vietnam, December 4–8, 2016. Springer, Heidelberg, Germany. (Cited on page 8.)

[RTYZ17] Alexander Russell, Qiang Tang, Moti Yung, and Hong-Sheng Zhou. Generic semantic security against a kleptographic adversary. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 17: 24th Conference on Computer and Communications Security*, pages 907–922, Dallas, TX, USA, October 31 – November 2, 2017. ACM Press. (Cited on page 8.)

[YY96] Adam Young and Moti Yung. The dark side of "black-box" cryptography, or: Should we trust capstone? In Neal Koblitz, editor, *Advances in Cryptology – CRYPTO'96*, volume 1109 of *Lecture Notes in Computer Science*, pages 89–103, Santa Barbara, CA, USA, August 18–22, 1996. Springer, Heidelberg, Germany. (Cited on page 8.)

[YY97] Adam Young and Moti Yung. Kleptography: Using cryptography against cryptography. In Walter Fumy, editor, *Advances in Cryptology – EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 62–74, Konstanz, Germany, May 11–15, 1997. Springer, Heidelberg, Germany. (Cited on page 8.)