Interdisciplinary Information Sciences Vol. 25, No. 1 (2019) 67–74
©Graduate School of Information Sciences, Tohoku University
ISSN 1340-9050 print/1347-6157 online
DOI 10.4036/iis.2019.R.03

On the Classification of Knowledge-of-exponent Assumptions in Cyclic Groups

Firas KRAIEM*, Shuji ISOBE, Eisuke KOIZUMI and Hiroki SHIZUYA

Graduate School of Information Sciences, Tohoku University, Sendai 980-8576, Japan

Inspired by the work of Ghadafi and Groth (ASIACRYPT 2017) on a certain type of computational hardness assumptions in cyclic groups (which they call "target assumptions"), we initiate an analogous work on another type of hardness assumptions, namely the "knowledge-of-exponent" assumptions (KEAs). Originally introduced by Damgård to construct practical encryption schemes secure against chosen ciphertext attacks, KEAs have subsequently been used primarily to construct succinct non-interactive arguments of knowledge (SNARKs), and proved to be inherent to such constructions. Since SNARKs (and their zero-knowledge variant, zk-SNARKs) are already used in practice in such systems as the Zcash digital currency, it can be expected that the use of KEAs will increase in the future, which makes it important to have a good understanding of those assumptions. Using a proof technique first introduced by Bellare and Palacio (but acknowledged by them as being due to Halevi), we first investigate the internal structure of the q-power knowledge-of-exponent (q-PKE) family of assumptions introduced by Groth, which is thus far the most general variant of KEAs. We then introduce a generalisation of the q-PKE family, and show that it can be simplified.

KEYWORDS: knowledge of exponent, cryptographic assumptions, cyclic groups

1. Introduction

The security of most cryptographic systems cannot currently be proved unconditionally, and must be proved under the assumption that a certain computational task is difficult (in a suitable sense). Of course, in order to increase our confidence in the security of such systems, it is necessary to increase our confidence in the validity of the assumptions under which their security is proved. Traditionally, this was done by assuming that a problem was difficult if a considerable amount of research effort had been devoted to the search of efficient solutions to it without any (or much) success. (The integer factoring or discrete logarithm problems, for instance, fall into this category — at least as far as classical computers are concerned.) In recent years, however, new assumptions are introduced very frequently, and, as pointed out for instance by Naor [12], it is sometimes not clear whether proving the security of a system under a new assumption is much different from simply assuming that the system is secure.

This proliferation of new assumptions raises questions both for cryptographers, who design new cryptographic systems, and for cryptanalysts, who attempt to "break" those systems by showing that the underlying assumptions are in fact false. For the former, what are the best assumptions on which to base their constructions? And for the latter, what are the best assumptions on which to focus their efforts? A solution to these dilemmas was proposed by Ghadafi and Groth [8] for a class of assumptions which they call "target assumptions" and which includes for instance the well-known computational Diffie-Hellman assumption [5]. Their idea was to firstly identify a large class of assumptions (the "target assumptions") which captures many assumptions already used in the literature as well as some which may appear in the future. Secondly, they identify a small subclass of assumptions (called "Uber-assumptions") within the large class, and show that if all the Uber-assumptions hold, then all the target assumptions hold as well. Such a result is useful both to cryptographers and to cryptanalysts. Cryptographers can use any target assumption as the basis of their systems, and be confident that they will remain secure as long as none of the Uber-assumptions is broken. Cryptanalysts, meanwhile, have a higher chance of success if they focus on the Uber-assumptions, since they give a small set of assumptions that is guaranteed to contain at least one false assumption (unless all the assumptions in the large class are true, in which case there is no hope of proving that any assumption is false anyway). Of course, the usefulness of such a result can be increased either by increasing the size of the large class (since then cryptographers have more assumptions at their disposal) or by decreasing the size of the class of Uber-assumptions (since then cryptanalysts can focus their efforts on a smaller number of assumptions). It is for this reason that Ghadafi and Groth apply their analysis to a large class of assumptions (the "target assumptions"), which is a generalisation of existing assumptions.

Received March 15, 2019; Accepted July 3, 2019; J-STAGE Advance published September 6, 2019
2012 ACM Computing Classification System: Theory of Computation → Cryptographic Primitives.

*Corresponding author. E-mail: firas@isl.is.tohoku.ac.jp

In this paper, we attempt to apply a similar analysis to another type of assumptions, called "knowledge-of-exponent assumptions" (KEAs). Namely, we introduce a generalised class of KEAs, which we aim to make as large as possible for the reasons explained above, and we show that it is implied by a smaller subclass. Although KEAs were originally criticised due to their non-falsifiability [12], non-falsifiable assumptions have been proved to be inherent to constructions of succinct non-interactive proof systems [7], and KEAs are commonly used in such constructions. Moreover, since one KEA-based construction (the zk-SNARK construction of [6, 14]) is already being used in practice in such systems as the Zcash digital currency [16], it can be expected that KEAs will become increasingly popular in the future, which makes it all the more important to have a solid understanding of them.

The rest of this paper is organised as follows. In Sect. 2, after reviewing some definitions and notation, we discuss the background surrounding KEAs in the existing literature. In Sect. 3, we investigate the internal structure of the q-power knowledge-of-exponent (q-PKE) family of assumptions introduced by Groth [9], which is the most general instance of KEAs in the literature thus far. In Sect. 4, we propose a generalisation of the q-PKE family which we call "rational knowledge-of-exponent assumptions" (RKEAs) and, as a first step towards identifying Uber-assumptions for RKEAs, we show that all RKEAs are implied by a slightly smaller class of assumptions. Finally, in Sect. 5 we summarise our results and point out possible directions for future work.

2. Preliminary Definitions, Notation and Background

2.1 Definitions and notation

Numbers and strings

$\mathbf{N} = \{0, 1, 2, \ldots\}$, $\mathbf{N}^* = \mathbf{N} \setminus \{0\}$. For any $n \in \mathbf{N}^*$, $|n| := \lfloor \log_2(n) \rfloor + 1$ is the length of the usual binary representation of n, and for simplicity we set $|0| := 0$. (We also use $|x|$ to denote the absolute value of a real number x, but the meaning of such notation should always be clear from the context.) For any $n \in \mathbf{N}$, 1^n is the string of length n with all bits 1. $\mathbf{F}_p$ denotes the finite field with p elements, represented as the integers $\{0, \ldots, p-1\}$ with addition and multiplication modulo p (we only consider prime finite fields). $\mathbf{R}$ denotes the field of real numbers.

Asymptotics

Given a function $f : \mathbf{N} \to \mathbf{R}$, we say that f is *positive* if $f(n) > 0$ for all n. We say that f is *polynomial* (in n), and we write $f(n) \leq \mathrm{poly}(n)$, if there is a positive polynomial p such that $f(n) \leq p(n)$ for all n. We say that f is *negligible* (in n) if for all positive polynomials p and all sufficiently large n we have $f(n) < 1/p(n)$. When such is the case we write $f(n) \leq \mathrm{negl}(n)$, and if n is the security parameter κ, we omit it and write $f \leq \mathrm{negl}$. Given another function $g : \mathbf{N} \to \mathbf{R}$, we note $f = \Theta(g)$ if there are two positive real numbers k_1, k_2 such that for all sufficiently large n we have $k_1 \cdot g(n) \leq f(n) \leq k_2 \cdot g(n)$.

Algorithms

We use the terminology and notation introduced by Abe and Fehr [1]. Unless otherwise stated, all the algorithms in this paper take as input 1^κ, for a security parameter κ, and possibly additional inputs, and run in time polynomial in κ (this implicitly requires all inputs to have size polynomial in κ). Algorithms may be *non-uniform*, meaning that when run on security parameter κ they get as an additional input an *advice string* adv_κ (the sequence of advice strings $(\mathrm{adv}_i)_{i \in \mathbf{N}}$ is fixed for a given algorithm). Algorithms may also be probabilistic. It will always be explicitly stated whether an algorithm is uniform or non-uniform and whether it is deterministic or probabilistic. Regardless, we reiterate that, unless otherwise stated, algorithms *always run in time polynomial in* κ.

To ease notation, 1^κ and adv_κ will often be omitted (e.g., if $\mathcal{A}$ is a non-uniform algorithm, we will often write $\mathcal{A}(x)$ instead of $\mathcal{A}(1^\kappa, x, \mathrm{adv}_\kappa)$ to denote its execution on input x and security parameter κ). For two probabilistic algorithms $\mathcal{A}$ and $\mathcal{B}$ we denote by $\mathcal{A} \| \mathcal{B}$ their joint execution on a common input and random tape, and we write $(u; v) \leftarrow (\mathcal{A} \| \mathcal{B})(x)$ to say that the output of $\mathcal{A}$ on input x is assigned to u and the output of $\mathcal{B}$ on the same input x *and the same random tape* is assigned to v. For a set S, $s \leftarrow S$ means that s is drawn from S uniformly and independently of all other random draws.

2.2 Group generators

Throughout this paper, we will define assumptions relative to a given *group generator*, as defined in [8].

Definition 2.1 (Group generators). A *group generator* is a uniform probabilistic algorithm $\mathcal{G}$, which on security parameter κ outputs group parameters (G_p, g), where

- p is a prime with $|p| = \Theta(\kappa)$;
- G_p is (a description of) a (cyclic) group of order p, with canonical representations of group elements as binary strings and efficient algorithms for performing the group operation and deciding membership; and
- g is a random generator of G_p, chosen uniformly over all the generators.

As in [8], given a group G_p, a generator g, and an element $x \in \mathbf{F}_p$, we will denote by $[x]$ the element of G_p with discrete logarithm x relative to the generator g and the group operation of G_p, i.e., $[x] := g \circ g \circ \cdots \circ g$ for x terms. Thus the generator g is $[1]$ and the identity element is $[0]$. We will also denote the group operation *additively*, so that we have $[x+y] = [x] + [y]$ and $[kx] = k[x]$ (where $k[x] := [x] + [x] + \cdots + [x]$ for k terms).

2.3 KEA1

The first knowledge-of-exponent assumption, which we call KEA1 following [2], was introduced in [4]. Roughly, it says that given a pair $([1], [\alpha])$ of elements of G_p, the only way to generate a pair $([k], [k\alpha])$ is the obvious way: pick k in some fashion, and compute $[k] = k[1]$ and $[k\alpha] = k[\alpha]$. In other words, any algorithm (adversary) which outputs such a pair must "know" k. This is formalised by saying that there must exist another algorithm, called an extractor, which, also given $([1], [\alpha])$, outputs k.

Assumption 2.2 (KEA1)**.** Let $\mathcal{G}$ be a group generator. We say that KEA1 holds (relative to $\mathcal{G}$) if for every non-uniform probabilistic algorithm $\mathcal{A}$ (the *adversary*) there is a non-uniform probabilistic algorithm $\chi_{\mathcal{A}}$ (the *extractor*) such that

$$\begin{aligned}\Pr[&(G_p,[1]) \leftarrow \mathcal{G}; \alpha \leftarrow \mathbf{F}_p; \sigma := (G_p,[1],[\alpha]);\\ &(([u],[v]);k) \leftarrow (\mathcal{A}\|\chi_{\mathcal{A}})(\sigma) :\\ &([v] = \alpha[u]) \wedge ([u] \neq k[1])] \leq \text{negl}.\end{aligned}$$

Remark 2.3. In [6, 14], KEAs are augmented to take into account any prior information the adversary may possess. Namely, the adversary has an additional auxiliary input z, and the condition must hold for all z generated independently of α. (Of course, the extractor is given z as well.)

2.4 The discrete logarithm assumption (DLA)

As in [2], we remark that if the discrete logarithm problem is easy (in groups generated by $\mathcal{G}$), then KEA1 trivially holds (in $\mathcal{G}$), for in that case we can trivially construct a KEA1-extractor $\chi_{\mathcal{A}}$ for any $\mathcal{A}$ as follows. Since $\chi_{\mathcal{A}}$ is given $\mathcal{A}$'s input and random coins, it can compute $\mathcal{A}$'s output $([u],[v])$, and furthermore, since the discrete logarithm problem is easy, it can compute u from $[u]$. It then outputs u, and if the discrete logarithm computation was successful (which happens with high probability since the discrete logarithm problem is easy), it will be successful as well.

Therefore, KEAs are only interesting in groups where the discrete logarithm problem is (believed to be) hard, which are the groups commonly used in cryptographic systems. We will thus assume throughout that the discrete logarithm problem is hard in the group generators we will consider, and we formalise this assumption as follows.

Assumption 2.4 (DLA)**.** We say that DLA holds (relative to the group generator $\mathcal{G}$) if for every non-uniform probabilistic algorithm $\mathcal{A}$ we have

$$\Pr[(G_p,[1]) \leftarrow \mathcal{G}; \alpha \leftarrow \mathbf{F}_p : \mathcal{A}(G_p,[1],[\alpha]) = \alpha] \leq \text{negl}.$$

2.5 KEA2 and KEA3

KEA2 was introduced in [10, 11], and subsequently proved false (under the DLA) in [2]. [2] then introduced KEA3 in order to recover the results of [10, 11], and proved that KEA3 implies KEA1.

Roughly, KEA2 states that given $([1],[x],[\alpha],[\alpha x])$, there are only two ways to produce a pair $([k],[k\alpha])$: generate k in some fashion and output either $(k[1], k[\alpha])$ or $(k[x], k[\alpha x])$. Intuitively, it is easy to see why this assumption should be false under the DLA: what about an adversary which generates k_1, k_2 and outputs $(k_1[1] + k_2[x], k_1[\alpha] + k_2[\alpha x])$? KEA2 asserts that such an adversary should know either $k_1 + k_2 x$ or $k_1 x^{-1} + k_2$, but it seems impossible to compute them without computing x and breaking the DLA. KEA3 addresses this issue in the obvious manner, by asserting that the only way to generate a pair $([k],[k\alpha])$ is as above: generate k_1, k_2 and output $(k_1[1] + k_2[x], k_1[\alpha] + k_2[\alpha x])$. We now turn to the formalisation of both assumptions.

Assumption 2.5 (KEA2)**.** Let $\mathcal{G}$ be a group generator. We say that KEA2 holds (relative to $\mathcal{G}$) if for every non-uniform probabilistic adversary $\mathcal{A}$ there is a non-uniform probabilistic extractor $\chi_{\mathcal{A}}$ such that

$$\begin{aligned}\Pr[&(G_p,[1]) \leftarrow \mathcal{G}; x, \alpha \leftarrow \mathbf{F}_p; \sigma := (G_p,[1],[x],[\alpha],[\alpha x]);\\ &(([u],[v]);k) \leftarrow (\mathcal{A}\|\chi_{\mathcal{A}})(\sigma) :\\ &([v] = \alpha[u]) \wedge ([u] \neq k[1]) \wedge ([u] \neq k[x])] \leq \text{negl}.\end{aligned}$$

Assumption 2.6 (KEA3)**.** Let $\mathcal{G}$ be a group generator. We say that KEA3 holds (relative to $\mathcal{G}$) if for every non-uniform probabilistic adversary $\mathcal{A}$ there is a non-uniform probabilistic extractor $\chi_{\mathcal{A}}$ such that

$$\begin{aligned}&\Pr[(G_p,[1]) \leftarrow \mathcal{G}; x,\alpha \leftarrow \mathbf{F}_p; \sigma := (G_p,[1],[x],[\alpha],[\alpha x]);\\ &\quad(([u],[v]);(k_1,k_2)) \leftarrow (\mathcal{A}\|\chi_{\mathcal{A}})(\sigma):\\ &\quad([v]=\alpha[u]) \wedge ([u] \neq k_1[1]+k_2[x])] \leq \text{negl.}\end{aligned}$$

3. The q-power Knowledge-of-exponent Assumptions

In this section we investigate the internal structure of the q-power knowledge-of-exponent (q-PKE) family of assumptions, which was introduced in [9] as a generalisation of KEA1 and KEA3. These assumptions are as follows.

Assumption 3.1 (q-PKE). Let $\mathcal{G}$ be a group generator, and $q \in \mathbf{N}$. We say that q-PKE holds (relative to $\mathcal{G}$) if for every non-uniform probabilistic adversary $\mathcal{A}$ there is a non-uniform probabilistic extractor $\chi_{\mathcal{A}}$ such that

$$\Pr\Bigg[(G_p,[1]) \leftarrow \mathcal{G}; x,\alpha \leftarrow \mathbf{F}_p; \sigma := (G_p,[1],[x],\ldots,[x^q],[\alpha],[\alpha x],\ldots,[\alpha x^q]);$$
$$(([u],[v]);(k_0,\ldots,k_q)) \leftarrow (\mathcal{A}\|\chi_{\mathcal{A}})(\sigma):$$
$$([v]=\alpha[u]) \wedge \left([u] \neq \sum_{i=0}^{q} k_i[x^i]\right)\Bigg] \leq \text{negl.}$$

Remark 3.2. We can allow the parameter q to be any function of the security parameter κ; in that case, the experiment on security parameter κ has $\sigma := (G_p,[1],[x],\ldots,[x^{q(\kappa)}],[\alpha],[\alpha x],\ldots,[\alpha x^{q(\kappa)}])$. Of course, since our algorithms run in time polynomial in κ, we can assume that $q(\kappa) \leq \text{poly}(\kappa)$. To ease notation, we will always simply write q.

Remark 3.3. It is shown in [9] that, for any q, q-PKE holds in the generic group model [3, 13, 15].

We note that KEA1 is 0-PKE and that KEA3 is 1-PKE. As previously mentioned, it was shown in [2] that 1-PKE implies 0-PKE; the proof there readily extends to show that, for any q, q-PKE implies 0-PKE. We nevertheless include a detailed proof of the latter result, in order to acquaint the reader with all the details of the proof technique that will be used later in this paper.

Theorem 3.4 (Generalisation of Proposition 2 from [2]). *Let $\mathcal{G}$ be a group generator, and $q \in \mathbf{N}$. If q-PKE holds for $\mathcal{G}$, then 0-PKE holds for $\mathcal{G}$.*

Proof. Let $\mathcal{A}$ be an adversary against 0-PKE; we first construct an adversary $\mathcal{B}$ against q-PKE that uses $\mathcal{A}$ in a black-box manner. $\mathcal{B}$ has input

$$(G_p,[1],[x],\ldots,[x^q],[\alpha],[\alpha x],\ldots,[\alpha x^q]);$$

it runs $\mathcal{A}$ on input $(G_p,[1],[\alpha])$ and with its own random tape, and outputs the pair $([u],[v])$ output by $\mathcal{A}$. Since q-PKE holds, there is an extractor $\chi_{\mathcal{B}}$ for $\mathcal{B}$ with negligible error probability ν; we construct an extractor $\chi_{\mathcal{A}}$ for $\mathcal{A}$ that uses $\chi_{\mathcal{B}}$ in a black-box manner. $\chi_{\mathcal{A}}$ proceeds as follows on input $(G_p,[1],[\alpha])$.

- $x \leftarrow \mathbf{F}_p$.
- $\sigma := (G_p,[1],[x],\ldots,[x^q],[\alpha],[\alpha x],\ldots,[\alpha x^q])$.
- $(k_0,\ldots,k_q) \leftarrow \chi_{\mathcal{B}}(\sigma)$.
- Output $\sum_{i=0}^{q} k_i x^i$.

We claim that $\chi_{\mathcal{A}}$ has (negligible) error probability ν.

We run the 0-PKE experiment. Firstly, $\mathcal{A}$ is run on input $(G_p,[1],[\alpha])$, and we let $([u],[v])$ be its output. Then $\chi_{\mathcal{A}}$ is run, again on input $(G_p,[1],[\alpha])$ and with the same random tape as $\mathcal{A}$, and it runs $\chi_{\mathcal{B}}$ on input σ and with its own random tape. Now, observe that σ is distributed identically to the input to $\chi_{\mathcal{B}}$ in the experiment for q-PKE, and so, letting $(([u'],[v']);(k_0,\ldots,k_q))$ be the output of $\mathcal{B}\|\chi_{\mathcal{B}}$ on input σ and with the same random tape as that of $\mathcal{A}$, we have

$$([v']=\alpha[u']) \wedge \left(\sum_{i=0}^{q} k_i[x^i] \neq [u']\right)$$

with probability ν. Since $\mathcal{B}$ on input σ runs $\mathcal{A}$ on input $(G_p,[1],[\alpha])$ and with the same random tape as that with which $\mathcal{A}$ was run originally, we have $([u'],[v'])=([u],[v])$. Observing that

$$\sum_{i=0}^{q} k_i[x^i] = \left(\sum_{i=0}^{q} k_i x^i\right)[1]$$

completes the proof. □

As mentioned, the above result shows in particular that 1-PKE implies 0-PKE. A natural question is then to ask whether this can be generalised to show that in general $(q+1)$-PKE implies q-PKE. The difficulty in showing this along the lines of the proof of Theorem 3.4 is that the q-PKE-extractor is given the group elements $([1],[x],\dots,[x^q],[\alpha],[\alpha x],\dots,[\alpha x^q])$, and seemingly cannot produce the group elements $([1],[x],\dots,[x^{q+1}],[\alpha],[\alpha x],\dots,[\alpha x^{q+1}])$ that are expected by the $(q+1)$-PKE-extractor. (Indeed, the widely-believed computational Diffie-Hellman exponent (CDHE) assumption [17] asserts that this is infeasible.) We circumvent this difficulty by assuming that, for a random group element $[r]$, $([1],[x],\dots,[x^q],[x^{q+1}])$ and $([1],[x],\dots,[x^q],[r])$ are indistinguishable. If such is the case, the q-PKE-extractor can generate a random $[r]$ and use it to generate group elements that are indistinguishable from those expected by the $(q+1)$-PKE-extractor. This assumption is a decisional version of the Diffie-Hellman exponent assumption, which we now define.*

Assumption 3.5 (q-decisional Diffie-Hellman exponent (q-DDHE)). Let $\mathcal{G}$ be a group generator, $\mathcal{A}$ be a non-uniform probabilistic adversary, $q \in \mathbf{N}^*$, and $b \in \{0,1\}$, and consider the following experiment $\mathsf{Exp}_{\mathcal{G},\mathcal{A}}^{q\text{-ddhe-}b}(\kappa)$.

- $(G_p,[1]) \leftarrow \mathcal{G}; x, r \leftarrow \mathbf{F}_p$.
- If $b=0$, then $\sigma := (G_p,[1],[x],\dots,[x^q],[r])$; else, $\sigma := (G_p,[1],[x],\dots,[x^q],[x^{q+1}])$.
- $b' \leftarrow \mathcal{A}(\sigma)$.
- Output b'.

We let

$$\mathsf{Adv}_{\mathcal{G},\mathcal{A}}^{q\text{-ddhe}}(\kappa) := |\Pr[\mathsf{Exp}_{\mathcal{G},\mathcal{A}}^{q\text{-ddhe-}1}(\kappa)=1] - \Pr[\mathsf{Exp}_{\mathcal{G},\mathcal{A}}^{q\text{-ddhe-}0}(\kappa)=1]|$$

be the *advantage* of $\mathcal{A}$ (in q-DDHE) relative to $\mathcal{G}$, and we say that q-DDHE holds in $\mathcal{G}$ if every adversary has negligible advantage, i.e., if for every non-uniform probabilistic adversary $\mathcal{A}$, we have $\mathsf{Adv}_{\mathcal{G},\mathcal{A}}^{q\text{-ddhe}} \le \text{negl}$.

Theorem 3.6. *Let $\mathcal{G}$ be a group generator, and $q \in \mathbf{N}^*$. If q-DDHE and $(q+1)$-PKE hold for $\mathcal{G}$, then q-PKE holds for $\mathcal{G}$.*

Proof. Let $\mathcal{A}$ be an adversary against q-PKE; we first construct an adversary $\mathcal{B}$ against $(q+1)$-PKE that uses $\mathcal{A}$ in a black-box manner. $\mathcal{B}$ has input

$$(G_p,[1],[x],\dots,[x^{q+1}],[\alpha],[\alpha x],\dots,[\alpha x^{q+1}]);$$

it runs $\mathcal{A}$ on input

$$(G_p,[1],[x],\dots,[x^q],[\alpha],[\alpha x],\dots,[\alpha x^q])$$

and with its own random tape, and outputs the pair $([u],[v])$ output by $\mathcal{A}$. Since $(q+1)$-PKE holds, there is an extractor $\chi_{\mathcal{B}}$ for $\mathcal{B}$ with negligible error probability ν; we construct an extractor $\chi_{\mathcal{A}}$ for $\mathcal{A}$ that uses $\chi_{\mathcal{B}}$ in a black-box manner. $\chi_{\mathcal{A}}$ proceeds as follows on input $(G_p,[1],[x],\dots,[x^q],[\alpha],[\alpha x],\dots,[\alpha x^q])$.

- $r \leftarrow \mathbf{F}_p$.
- $\sigma := (G_p,[1],\dots,[x^q],[r],[\alpha],\dots,[\alpha x^q],[\alpha r])$.
- $(k_0,k_1,\dots,k_{q+1}) \leftarrow \chi_{\mathcal{B}}(\sigma)$.
- Output $(k_0 + k_{q+1}r, k_1,\dots,k_q)$.

We claim that $\chi_{\mathcal{A}}$ has negligible error probability.

We run the q-PKE experiment. Firstly, $\mathcal{A}$ is run on input $(G_p,[1],[x],\dots,[x^q],[\alpha],[\alpha x],\dots,[\alpha x^q])$, and we let $([u],[v])$ be its output. Then $\chi_{\mathcal{A}}$ is run, again on input $(G_p,[1],[x],\dots,[x^q],[\alpha],[\alpha x],\dots,[\alpha x^q])$ and with the same random tape as $\mathcal{A}$, and it runs $\chi_{\mathcal{B}}$ on input σ and with its own random tape. We claim that, letting $(([u'],[v']);(k_0,\dots,k_{q+1}))$ be the output of $\mathcal{B}\|\chi_{\mathcal{B}}$ on input σ, we have

$$([v'] = \alpha[u']) \wedge \left(k_{q+1}[r] + \sum_{i=0}^{q} k_i[x^i] \neq [u']\right)$$

with negligible probability. Intuitively, this follows from the fact that, under q-DDHE, σ is indistinguishable from the input to $\mathcal{B}\|\chi_{\mathcal{B}}$ in the $(q+1)$-PKE experiment. To show it formally, we consider the following adversary $\mathcal{Z}$ against q-DDHE, which uses $\mathcal{B}$ and $\chi_{\mathcal{B}}$ in a black-box manner.

$\mathcal{Z}$ proceeds as follows on input $(G_p,[1],[x],\dots,[x^q],[z])$ (where x is random and z is either x^{q+1} or random).

- $\alpha \leftarrow \mathbf{F}_p$.
- $\sigma := (G_p,[1],\dots,[x^q],[z],[\alpha],\dots,[\alpha x^q],[\alpha z])$.
- $(([u],[v]);(k_0,k_1,\dots,k_{q+1})) \leftarrow (\mathcal{B}\|\chi_{\mathcal{B}})(\sigma)$.
- If $[v]=\alpha[u]$ and $k_{q+1}[z] + \sum_{i=0}^{q} k_i[x^i] \neq [u]$, output 1; else, output 0.

If $z = x^{q+1}$, the q-DDHE experiment for $\mathcal{Z}$ is exactly the $(q+1)$-PKE experiment for $\mathcal{B}\|\chi_{\mathcal{B}}$, and so $\mathcal{Z}$ outputs 1 with (negligible) probability ν. On the other hand, if z is random, then σ is distributed identically to the input to $\mathcal{B}\|\chi_{\mathcal{B}}$ when

*Unfortunately, since our proof relies on a decisional assumption, it does not apply in the bilinear setting, which is the setting in which q-PKE was introduced in [9] and subsequently used in [6, 14].

it is run by $\chi_{\mathcal{A}}$ in the q-PKE experiment. Let μ be the probability that $\mathcal{Z}$ outputs 1 in that latter case; then $|\nu - \mu|$ is negligible since q-DDHE holds, and since ν is negligible as well, so is μ.

Finally, since $\mathcal{B}$ on input σ runs $\mathcal{A}$ on input $(G_p, [1], [x], \ldots, [x^q], [\alpha], [\alpha x], \ldots, [\alpha x^q])$ and with the same random tape as that with which $\mathcal{A}$ was run originally, we have $([u'], [v']) = ([u], [v])$. Observing that

$$k_{q+1}[r] + \sum_{i=0}^{q} k_i[x^i] = (k_0 + k_{q+1}r)[1] + \sum_{i=1}^{q} k_i[x^i]$$

completes the proof. □

4. Rational KEAs (RKEAs)

In this section, we propose a definition of a large class of assumptions, with the goal of capturing not only the KEAs that have appeared in the literature thus far, but also those that are likely to appear in the future. We then show that this large class is implied by a slightly smaller subclass.

4.1 Definition of RKEAs

Along the lines of the definition of target assumptions in [8], we generalise the PKE family of assumptions by allowing arbitrary rational functions of several variables instead of just powers of x. We call the resulting class of assumptions *rational knowledge-of-exponent assumptions* (RKEAs). Analogously to the target assumptions of [8], RKEAs are parameterised by three integers[†] d (the maximal degree of the polynomials involved), m (the number of variables) and n (the number of rational functions). We first define a very general notion of *non-interactive knowledge assumptions* (NIKAs) analogous to the non-interactive computational assumptions of [8]. (The intuitive meanings of the quoted terms should be clear from the previous examples of KEAs.)

Definition 4.1 (Non-interactive knowledge assumptions (NIKAs)). A *non-interactive knowledge assumption* consists of an instance generator $\mathcal{I}$, a verifier $\mathcal{V}$, and a knowledge verifier $\overline{\mathcal{V}}$, defined as follows.

- $(\mathsf{pub}, \mathsf{priv}) \leftarrow \mathcal{I}$: $\mathcal{I}$ is a uniform probabilistic algorithm which, on input 1^κ (where κ is a security parameter), outputs a pair of public/private information $(\mathsf{pub}, \mathsf{priv})$. We omit the input 1^κ as usual.
- $0/1 \leftarrow \mathcal{V}(\mathsf{pub}, \mathsf{priv}, \mathsf{sol})$: $\mathcal{V}$ is a uniform deterministic algorithm which, on input $(\mathsf{pub}, \mathsf{priv})$ and a purported solution sol, outputs 1 if the solution is "correct" and 0 otherwise.
- $0/1 \leftarrow \overline{\mathcal{V}}(\mathsf{pub}, \mathsf{priv}, \mathsf{sol}, \mathsf{sec})$: $\overline{\mathcal{V}}$ is a uniform deterministic algorithm which, on input $(\mathsf{pub}, \mathsf{priv}, \mathsf{sol})$ and a purported "secret" sec, outputs 1 if the secret is "correct" and 0 otherwise.

We say that the assumption holds if for any non-uniform probabilistic algorithm $\mathcal{A}$ (the *adversary*) there is a non-uniform probabilistic algorithm $\chi_{\mathcal{A}}$ (the *knowledge extractor*, or just the *extractor*) such that

$$\Pr[(\mathsf{pub}, \mathsf{priv}) \leftarrow \mathcal{I}; (\mathsf{sol}; \mathsf{sec}) \leftarrow (\mathcal{A} \| \chi_{\mathcal{A}})(\mathsf{pub}) :$$
$$\mathcal{V}(\mathsf{pub}, \mathsf{priv}, \mathsf{sol}) = 1 \wedge \overline{\mathcal{V}}(\mathsf{pub}, \mathsf{priv}, \mathsf{sol}, \mathsf{sec}) = 0] \leq \text{negl}.$$

We call the above probability the *error probability* of $\chi_{\mathcal{A}}$ relative to $\mathcal{A}$, and express it as a function of the security parameter κ; thus the assumption holds if for every adversary $\mathcal{A}$ there is an extractor $\chi_{\mathcal{A}}$ with negligible error probability relative to $\mathcal{A}$. We also say that $\chi_{\mathcal{A}}$ is *successful* (relative to $\mathcal{A}$) if the condition above does *not* hold, i.e., if $\chi_{\mathcal{A}}$ "successfully extracts" $\mathcal{A}$'s secret (hence the error probability is the probability that the extractor is not successful).

Definition 4.2 (Rational knowledge-of-exponent assumptions (RKEAs)). Given $d, m, n \in \mathbf{N}^*$ and a group generator $\mathcal{G}$, we say that an NIKA $(\mathcal{I}, \mathcal{V}, \overline{\mathcal{V}})$ is a (d, m, n)-RKEA if there is a uniform probabilistic algorithm $\mathcal{I}^{\text{core}}$ such that $\mathcal{I}$, $\mathcal{V}$ and $\overline{\mathcal{V}}$ are of the following forms.

- $(\mathsf{pub}, \mathsf{priv}) \leftarrow \mathcal{I}$:
  - $(G_p, [1]) \leftarrow \mathcal{G}$.
  - $\left(\left\{\frac{a_i(\mathbf{X})}{b_i(\mathbf{X})}\right\}_{i=1}^n, \mathsf{pub}', \mathsf{priv}'\right) \leftarrow \mathcal{I}^{\text{core}}(G_p)$, where the a_is and b_is are polynomials in m variables and of total degree at most d.
  - $\boldsymbol{x} \leftarrow \mathbf{F}_p^m$ conditioned on $b_i(\boldsymbol{x}) \neq 0$ for all i.
  - $\alpha \leftarrow \mathbf{F}_p$.
  - $\mathsf{pub} := \left(G_p, \left\{\left[\frac{a_i(\boldsymbol{x})}{b_i(\boldsymbol{x})}\right]\right\}_{i=1}^n, \left\{\left[\frac{\alpha \cdot a_i(\boldsymbol{x})}{b_i(\boldsymbol{x})}\right]\right\}_{i=1}^n, \left\{\frac{a_i(\mathbf{X})}{b_i(\mathbf{X})}\right\}_{i=1}^n, \mathsf{pub}'\right)$.
  - Return $(\mathsf{pub}, \mathsf{priv} := ([1], \boldsymbol{x}, \alpha, \mathsf{priv}'))$.
- $0/1 \leftarrow \mathcal{V}(\mathsf{pub}, \mathsf{priv}, \mathsf{sol} = ([u], [v]))$: if $[v] = \alpha[u]$, return 1; else, return 0.
- $0/1 \leftarrow \overline{\mathcal{V}}(\mathsf{pub}, \mathsf{priv}, \mathsf{sol}, \mathsf{sec} = (k_1, \ldots, k_n))$: if $\sum_{i=1}^n k_i\left[\frac{a_i(\boldsymbol{x})}{b_i(\boldsymbol{x})}\right] = [u]$, return 1; else, return 0.

Remark 4.3. We note that in an RKEA, the knowledge verifier $\overline{\mathcal{V}}$ does not use the private information priv, thus RKEAs would also satisfy an alternative definition of NIKAs where $\overline{\mathcal{V}}$ is not given priv.

[†] Again, we can allow d, m, n to be any functions of the security parameter κ, and assume that they are polynomial in κ.

Example 4.4 (q-PKE). q-PKE is a $(q, 1, q+1)$-RKEA, meaning that $\mathcal{I}^{\text{core}}$ generates $q+1$ rational functions consisting of polynomials in one variable and of degree at most q. Namely, we have $a_i(x) = x^{i-1}$ and $b_i(x) = 1$ for all $i = 1, \ldots, q+1$.

4.2 Simplifications of RKEAs

Definition 4.5 (Simple RKEAs)**.** We say that an RKEA is *simple* if $b_i(\mathbf{X}) = 1$ for all $i = 1, \ldots, n$, i.e., all the rational functions output by $\mathcal{I}^{\text{core}}$ are just polynomials.

Theorem 4.6. *For any* (d, m, n)*-RKEA* $\mathsf{A} = (\mathcal{I}_{\mathsf{A}}, \mathcal{V}_{\mathsf{A}}, \overline{\mathcal{V}}_{\mathsf{A}})$ *there is an* (nd, m, n)*-simple RKEA* $\mathsf{B} = (\mathcal{I}_{\mathsf{B}}, \mathcal{V}_{\mathsf{B}}, \overline{\mathcal{V}}_{\mathsf{B}})$ *such that* B *implies* A.

Proof. The algorithm $\mathcal{I}_{\mathsf{B}}^{\text{core}}$ of B proceeds as follows on input G_p.

- $\left(\left\{\frac{a_i(\mathbf{X})}{b_i(\mathbf{X})}\right\}_{i=1}^n, \mathsf{pub}'_{\mathsf{A}}, \mathsf{priv}'_{\mathsf{A}}\right) \leftarrow \mathcal{I}_{\mathsf{A}}^{\text{core}}(G_p)$.
- $c_i(\mathbf{X}) := a_i(\mathbf{X}) \cdot \prod_{j \neq i} b_j(\mathbf{X})$ for all $i = 1, \ldots, n$.
- $\mathsf{pub}'_{\mathsf{B}} := \left(\left\{\frac{a_i(\mathbf{X})}{b_i(\mathbf{X})}\right\}_{i=1}^n, \mathsf{pub}'_{\mathsf{A}}\right)$; $\mathsf{priv}'_{\mathsf{B}} := \mathsf{priv}'_{\mathsf{A}}$.
- Return $(\{c_i(\mathbf{X})\}_{i=1}^n, \mathsf{pub}'_{\mathsf{B}}, \mathsf{priv}'_{\mathsf{B}})$.

Let $\mathcal{A}$ be an adversary against A; we first construct an adversary $\mathcal{B}$ against B which uses $\mathcal{A}$ in a black-box manner. $\mathcal{B}$'s input is

$$\mathsf{pub}_{\mathsf{B}} = (G_p, \{[c_i(\boldsymbol{x})]\}_{i=1}^n, \{[\alpha \cdot c_i(\boldsymbol{x})]\}_{i=1}^n, \{c_i(\mathbf{X})\}_{i=1}^n, \mathsf{pub}'_{\mathsf{B}});$$

it runs $\mathcal{A}$ on input

$$\left(G_p, \{[c_i(\boldsymbol{x})]\}_{i=1}^n, \{[\alpha \cdot c_i(\boldsymbol{x})]\}_{i=1}^n, \left\{\frac{a_i(\mathbf{X})}{b_i(\mathbf{X})}\right\}_{i=1}^n, \mathsf{pub}'_{\mathsf{A}}\right)$$

and with its own random tape, and outputs the pair output by $\mathcal{A}$. Since we assume that B holds, there is an extractor $\chi_{\mathcal{B}}$ for $\mathcal{B}$ with negligible error probability ν; we construct an extractor $\chi_{\mathcal{A}}$ for $\mathcal{A}$ which uses $\chi_{\mathcal{B}}$ in a black-box manner. $\chi_{\mathcal{A}}$'s input is

$$\mathsf{pub}_{\mathsf{A}} = \left(G_p, \left\{\left[\frac{a_i(\boldsymbol{x})}{b_i(\boldsymbol{x})}\right]\right\}_{i=1}^n, \left\{\left[\frac{\alpha \cdot a_i(\boldsymbol{x})}{b_i(\boldsymbol{x})}\right]\right\}_{i=1}^n, \left\{\frac{a_i(\mathbf{X})}{b_i(\mathbf{X})}\right\}_{i=1}^n, \mathsf{pub}'_{\mathsf{A}}\right)$$

and its random tape is the same as that of $\mathcal{A}$. $\chi_{\mathcal{A}}$ runs $\chi_{\mathcal{B}}$ on input

$$\sigma_{\mathsf{B}} = \left(G_p, \left\{\left[\frac{a_i(\boldsymbol{x})}{b_i(\boldsymbol{x})}\right]\right\}_{i=1}^n, \left\{\left[\frac{\alpha \cdot a_i(\boldsymbol{x})}{b_i(\boldsymbol{x})}\right]\right\}_{i=1}^n, \{c_i(\mathbf{X})\}_{i=1}^n, \mathsf{pub}'_{\mathsf{B}}\right)$$

and with its own random tape, and outputs the values $(k_1, \ldots, k_n)$ output by $\chi_{\mathcal{B}}$. We claim that $\chi_{\mathcal{A}}$ is an extractor for $\mathcal{A}$ with (negligible) error probability at most $\nu + \frac{dn}{p}$[‡].

We run the NIKA experiment for A. Firstly, $\mathcal{A}$ is run on input $\mathsf{pub}_{\mathsf{A}}$, and we let $([u], [v])$ be its output. Then, $\chi_{\mathcal{A}}$ is run, again on input $\mathsf{pub}_{\mathsf{A}}$ and with the same random tape as $\mathcal{A}$, and it runs $\chi_{\mathcal{B}}$ on input σ_{B} and with its own random tape, outputting the output $(k_1, \ldots, k_n)$ of $\chi_{\mathcal{B}}$. We claim that σ_{B} is distributed identically to $\mathsf{pub}_{\mathsf{B}}$ except with negligible probability. To see this, observe that $\mathcal{I}_{\mathsf{A}}$ generates the polynomials $a_i(\mathbf{X})$ and $b_i(\mathbf{X})$ as well as the vector $\boldsymbol{x}$ independently of the generator $[1]$ output by $\mathcal{G}$. Further, assuming that $\prod_{i=1}^n b_i(\boldsymbol{x}) \neq 0$, the only difference between $\mathsf{pub}_{\mathsf{B}}$ and σ_{B} is the choice of generator; namely, if choosing the generator $[1]$ yields the input $\mathsf{pub}_{\mathsf{B}}$, then choosing the generator $[\prod_{i=1}^n b_i(\boldsymbol{x})]$ yields σ_{B}.

By the Schwartz–Zippel lemma, the probability that $\prod_{i=1}^n b_i(\boldsymbol{x}) = 0$ is at most $\frac{dn}{p}$. Thus, letting $([u'], [v'])$ be the pair output by $\mathcal{B}$ we have

$$([v'] = \alpha[u']) \wedge \left(\sum_{i=1}^n k_i \left[\frac{a_i(\boldsymbol{x})}{b_i(\boldsymbol{x})}\right] \neq [u']\right)$$

with probability at most $\nu + \frac{dn}{p}$. Observing that $\mathcal{B}$, when run on input σ_{B}, runs $\mathcal{A}$ on input $\mathsf{pub}_{\mathsf{A}}$ and with the same random tape as that with which $\mathcal{A}$ was run originally shows that $([u'], [v']) = ([u], [v])$, which completes the proof. □

5. Conclusions and Directions for Future Work

We have shown that, under a variant of the decisional Diffie-Hellman assumption, the q-PKE family of assumptions increases in strength as q grows. We have also introduced a more general class of KEAs than had previously appeared in the literature, and showed that it can be slightly simplified. All our results were obtained using the proof tech-

[‡] $\nu + \frac{dn}{p}$ is negligible (in κ) because ν is negligible by assumption, d and n are polynomial, and p is exponential since it is an integer of polynomial size.

nique from [2]. Many directions for future work remain open, which might require the introduction of new proof techniques:

- Can an analogue of Theorem 3.6 be proved in the bilinear setting, where in particular decisional assumptions do not hold?
- Is the q-PKE family *strictly* increasing? That is, can it be shown in some sense that q-PKE does *not* imply $(q+1)$-PKE?
- Can RKEAs be simplified further as in [8]? In particular, can Uber-assumptions be found?
- Can RKEAs be generalised further, for instance by allowing $\mathcal{V}$ and $\overline{\mathcal{V}}$ to be of a more general form?

REFERENCES

[1] Abe, M., and Fehr, S., Perfect NIZK with Adaptive Soundness, Vadhan, S. P. (Ed.), TCC 2007, Vol. 4392, pp. 118–136, Springer (2007) Lecture Notes in Computer Science. doi:10.1007/978-3-540-70936-7_7

[2] Bellare, M., and Palacio, A., The Knowledge-of-Exponent Assumptions and 3-Round Zero-Knowledge Protocols, Franklin, M. (Ed.), CRYPTO 2004, Vol. 3152, pp. 273–289, Springer (2004) Lecture Notes in Computer Science. doi:10.1007/978-3-540-28628-8_17

[3] Chateauneuf, M., Ling, A. C. H., and Stinson, D. R., Slope Packings and Coverings, and Generic Algorithms for the Discrete Logarithm Problem, Vol. 11, no. 1, pp. 36–50, Wiley (2003) Journal of Combinatorial Designs. doi:10.1002/jcd.10033

[4] Damgård, I., Towards Practical Public Key Systems Secure Against Chosen Ciphertext Attacks, Feigenbaum, J. (Ed.), CRYPTO '91, Vol. 576, pp. 445–456, Springer (1992) Lecture Notes in Computer Science. doi:10.1007/3-540-46766-1_36

[5] Diffie, W., and Hellman, M. E., New Directions in Cryptography, Vol. 22, no. 6, pp. 644–654, IEEE (1976) IEEE Transactions on Information Theory. doi:10.1109/TIT.1976.1055638

[6] Gennaro, R., Gentry, C., Parno, B., and Raykova, M., Quadratic Span Programs and Succinct NIZKs without PCPs, Johansson, T., and Nguyen, P. Q. (Eds.), EUROCRYPT 2013, Vol. 7881, pp. 626–645, Springer (2013) Lecture Notes in Computer Science. doi:10.1007/978-3-642-38348-9_37

[7] Gentry, C., and Wichs, D., Separating Succinct Non-Interactive Arguments from All Falsifiable Assumptions, STOC '11, pp. 99–108 (2011) Proc. of the Forty-third Annual ACM Symposium on Theory of Computing. doi:10.1145/1993636.1993651

[8] Ghadafi, E., and Groth, J., Towards a Classification of Non-interactive Computational Assumptions in Cyclic Groups, Takagi, T., and Peyrin, T. (Eds.), ASIACRYPT 2017, Part II, Vol. 10625, pp. 66–96, Springer (2017) Lecture Notes in Computer Science. doi:10.1007/978-3-319-70697-9_3

[9] Groth, J., Short Pairing-Based Non-interactive Zero-Knowledge Arguments, Abe, M. (Ed.), ASIACRYPT 2010, Vol. 6477, pp. 321–340, Springer (2010) Lecture Notes in Computer Science. doi:10.1007/978-3-642-17373-8_19

[10] Hada, S., and Tanaka, T., On the Existence of 3-round Zero-knowledge Protocols, Krawczyk, H. (Ed.), CRYPTO '98, Vol. 1462, pp. 408–423, Springer (1998) Lecture Notes in Computer Science. (Preliminary version of [11].) doi:10.1007/BFB0055744

[11] Hada, S., and Tanaka, T., On the Existence of 3-Round Zero-Knowledge Protocols, Cryptology ePrint Archive, Report 1999/009, 1999. (Full version of [10].) https://ia.cr/1999/009 (Accessed 14 March 2019.)

[12] Naor, M., On Cryptographic Assumptions and Challenges, Boneh, D. (Ed.), CRYPTO 2003, Vol. 2729, pp. 96–109, Springer (2003) Lecture Notes in Computer Science. doi:10.1007/978-3-540-45146-4_6

[13] Nechaev, V. I., Complexity of a Determinate Algorithm for the Discrete Logarithm, Vol. 55, no. 2, pp. 165–172, Plenum (1994) Mathematical Notes. doi:10.1007/BF02113297

[14] Parno, B., Howell, J., Gentry, C., and Raykova, M., Pinocchio: Nearly Practical Verifiable Computation, pp. 238–252 (2013) 2013 IEEE Symposium on Security and Privacy. doi:10.1109/SP.2013.47

[15] Shoup, V., Lower Bounds for Discrete Logarithms and Related Problems, Fumy, W. (Ed.), EUROCRYPT '97, Vol. 1233, pp. 256–266, Springer (1997) Lecture Notes in Computer Science. doi:10.1007/3-540-69053-0_18

[16] Zerocoin Electric Coin Company, What are zk-SNARKs?. https://z.cash/technology/zksnarks/ (Accessed 14 March 2019.)

[17] Zhang, F., Safavi-Naini, R., and Susilo, W., An Efficient Signature Scheme from Bilinear Pairings and Its Applications, Bao, F., Deng, R., and Zhou, J. (Eds.), PKC 2004, Vol. 2947, pp. 277–290, Springer (2004) Lecture Notes in Computer Science. doi:10.1007/978-3-540-24632-9_20