The Knowledge-of-Exponent Assumptions and 3-Round Zero-Knowledge Protocols

Mihir Bellare and Adriana Palacio

Dept. of Computer Science & Engineering, University of California, San Diego
9500 Gilman Drive, La Jolla, CA 92093, USA.
Email: {mihir,apalacio}@cs.ucsd.edu
URL: http://www-cse.ucsd.edu/users/{mihir,apalacio}

Abstract. Hada and Tanaka [11, 12] showed the existence of 3-round, negligible-error zero-knowledge arguments for NP based on a pair of non-standard assumptions, here called KEA1 and KEA2. In this paper we show that KEA2 is false. This renders vacuous the results of [11, 12]. We recover these results, however, under a suitably modified new assumption called KEA3. What we believe is most interesting is that we show that it is possible to "falsify" assumptions like KEA2 that, due to their nature and quantifier-structure, do not lend themselves easily to "efficient falsification" (Naor [15]).

1 Introduction

A classical question in the theory of zero knowledge (ZK) [10] is whether there exist 3-round, negligible-error ZK proofs or arguments for NP. The difficulty in answering this question stems from the fact that such protocols would have to be non-black-box simulation ZK [9], and there are few approaches or techniques to this end. A positive answer has, however, been provided, by Hada and Tanaka [11, 12]. Their result (a negligible-error, 3-round ZK argument for NP) requires a pair of non-standard assumptions that we will denote by KEA1 and KEA2.

The assumptions, roughly. Let q be a prime such that $2q + 1$ is also prime, and let g be a generator of the order q subgroup of Z^*_{2q+1}. Suppose we are given input q, g, g^a and want to output a pair (C, Y) such that $Y = C^a$. One way to do this is to pick some $c \in \mathbb{Z}_q$, let $C = g^c$, and let $Y = (g^a)^c$. Intuitively, KEA1 can be viewed as saying that this is the "only" way to produce such a pair. The assumption captures this by saying that any adversary outputting such a pair must "know" an exponent c such that $g^c = C$. The formalization asks that there be an "extractor" that can return c. Roughly:

KEA1: For any adversary $\mathbf{A}$ that takes input q, g, g^a and returns (C, Y) with $Y = C^a$, there exists an "extractor" $\bar{\mathbf{A}}$, which given the same inputs as $\mathbf{A}$ returns c such that $g^c = C$.

Suppose we are given input q, g, g^a, g^b, g^{ab} and want to output a pair (C, Y) such that $Y = C^b$. One way to do this is to pick some $c \in \mathbb{Z}_q$, let $C = g^c$, and let $Y = (g^b)^c$. Another way is to pick some $c \in \mathbb{Z}_q$, let $C = (g^a)^c$, and let $Y = (g^{ab})^c$. Intuitively, KEA2 can be viewed as saying that these are the "only" ways to produce such a pair. The assumption captures this by saying that any adversary outputting such a pair must "know" an exponent c such that either $g^c = C$ or $(g^a)^c = C$. The formalization asks that there be an "extractor" that can return c. Roughly:

KEA2: For any adversary $\mathbf{A}$ that takes input q, g, g^a, g^b, g^{ab} and returns (C, Y) with $Y = C^b$, there exists an "extractor" $\bar{\mathbf{A}}$, which given the same inputs as $\mathbf{A}$ returns c such that either $g^c = C$ or $(g^a)^c = C$.

As per [11, 12], adversaries and extractors are poly-size families of (deterministic) circuits. See Assumption 2 for a formalization of KEA2, and Assumption 4 for a formalization of KEA1.

History and nomenclature of the assumptions. KEA1 is due to Damgård [7], and is used by [11, 12] to prove their protocol is ZK. To prove soundness of their protocol, Hada and Tanaka [11, 12] introduce and use KEA2. (In addition, they make the Discrete Logarithm Assumption, DLA.) The preliminary version of their work [11] referred to the assumptions as SDHA1 and SDHA2 (**S**trong **D**iffie-**H**ellman **A**ssumptions 1 and 2), respectively. However, the full version [12] points out that the formalizations in the preliminary version are flawed, and provides corrected versions called non-uniform-DA1 and non-uniform-DA2. The latter are the assumptions considered in this paper, but we use the terminology of Naor [15] which we feel is more reflective of the content of the assumption: "KEA" stands for "**K**nowledge of **E**xponent **A**ssumption", the exponent being the value c above.

Falsifying KEA2. In this paper we show that KEA2 is false. What is interesting about this —besides the fact that it renders the results of [11, 12] vacuous— is that we are able to "falsify" an assumption whose nature, as pointed out by Naor [15], does not lend itself easily to "efficient falsification." Let us explain this issue before expanding more on the result itself.

The most standard format for an assumption is to ask that the probability that an adversary produces a certain output on certain inputs is negligible. For example, the Factoring assumption is of this type, asking that the probability that a polynomial-time adversary can output the prime factors of an integer (chosen by multipling a pair of random primes) is negligible. To show such an assumption is false, we can present an "attack," in the form of an adversary whose success probability is not negligible. (For example, a polynomial-time factoring algorithm.) KEA1 and KEA2 are not of this standard format. They involve a more complex quantification: "For every adversary there exists an extractor such that ...". To show KEA2 is false, we must show there is an adversary *for which there exists no extractor*. As we will see later, it is relatively simple to identify an adversary for which there does not *appear* to exist an extractor, but how can we actually show that none of the infinite number of possible extractors succeeds?

An analogy. The difficulty of falsifying an assumption with the quantifier format of KEA2 may be better appreciated via an analogy. The definition of ZK has a similar quantifier format: "For every (cheating) verifier there exists a simulator such that ...". This makes it hard to show a protocol is not ZK, for, even though we may be able to identify a cheating verifier strategy that appears hard to simulate, it is not clear how we can actually show no simulator exists. (For example, it is hard to imagine how one could find a simulator for the cheating verifier, for Blum's ZK proof of Hamiltonian Cycle [5], that produces its challenges by hashing the permuted graphs sent by the prover in the first step. But there is to date no proof that such a simulator does not exist). However it has been possible to show protocols are not black-box simulation ZK [9], taking advantage of the fact that the quantification in this definition is different from that of ZK itself. It has also been possible to show conditional results, for example that the parallel version of the Fiat-Shamir [8] protocol is not ZK, unless there is no hash function that, when applied to collapse this protocol, results in a secure signature scheme [16]. Our result too is conditional.

Falsification result. At an intuitive level, the weakness in KEA2 is easy to see, and indeed it is surprising this was not noted before. Namely, consider an adversary **A** that on input q, g, g^a, g^b, g^{ab} picks c_1, c_2 in some fashion, and outputs (C, Y) where $C = g^{c_1}(g^a)^{c_2}$ and $Y = (g^b)^{c_1}(g^{ab})^{c_2}$. Then $Y = C^b$ but this adversary does not appear to "know" c such that either $g^c = C$ or $(g^a)^c = C$. The difficulty, however, as indicated above, is to prove that there does not exist an extractor. We do this by first specifying a particular strategy for choosing c_1 and c_2 and then showing that if there exists an extractor for the resulting adversary, then this extractor can be used to solve the discrete logarithm problem (DLP). Thus, our result (cf. Theorem 1) is that if the DLP is hard then KEA2 is false. Note that if the DLP is easy, then KEA2 is true, for the extractor can simply compute a discrete logarithm of C and output it, and thus the assumption that it is hard is necessary to falsify KEA2.

Remark. We emphasize that we have not found any weaknesses in KEA1, an assumption used not only in [7, 11, 12] but also elsewhere.

KEA3. Providing a 3-round, negligible-error ZK protocol for NP is a challenging problem that has attracted considerable research effort. The fact that KEA2 is false means that we "lose" one of the only positive results [11, 12] that we had on this subject. Accordingly, we would like to "recover" it. To this end, we propose a modification of KEA2 that addresses the weakness we found. The new assumption is, roughly, as follows:

KEA3: For any adversary **A** that takes input q, g, g^a, g^b, g^{ab} and returns (C, Y) with $Y = C^b$, there exists an "extractor" $\bar{\mathbf{A}}$, which given the same inputs as **A** returns c_1, c_2 such that $g^{c_1}(g^a)^{c_2} = C$.

Before proceeding to use this assumption, we note a relation that we consider interesting, namely, that KEA3 implies KEA1 (cf. Proposition 2).[1] The relation means that KEA3 is a natural extension of KEA1. It also allows us to simplify result statements, assuming only KEA3 rather than both this assumption and KEA1.

Recovering the ZK result. Let HTP denote the 3-round protocol of Hada and Tanaka, which they claim to be sound (i.e., have negligible error) and ZK. The falsity of KEA2 invalidates their proof of soundness. However, this does not mean that HTP is not sound: perhaps it is and this could be proved under another assumption, such as KEA3. This turns out to be almost, but not quite, true. We identify a small bug in HTP based on which we can present a successful cheating prover strategy, showing that HTP is not sound. This is easily fixed, however, to yield a protocol we call pHTP (patched HTP). This protocol is close enough to HTP that the proof of ZK (based on KEA1) is unchanged. On the other hand, the proof of soundness of HTP provided in [12] extends with very minor modifications to prove soundness of pHTP based on KEA3 and DLA (cf. Theorem 2). In summary, assuming KEA3 and DLA, there exists a 3-round, negligible error ZK argument for NP.

Strength of the assumptions. The knowledge-of-exponent assumptions are strong and non-standard ones, and have been criticized for assuming that one can perform what some people call "reverse engineering" of an adversary. These critiques are certainly valid. Our falsification of KEA2 does not provide information on this aspect of the assumptions, uncovering, rather, other kinds of problems. However, by showing that such assumptions can be falsified, we open the door to further analyses.

We also stress that in recovering the result of [12] on 3-round ZK we have not succeeded in weakening the assumptions on which it is based, for KEA3 certainly remains a strong assumption of the same non-standard nature as KEA1.

Related Work. Since [11, 12] there has been more progress with regard to the design of non-black-box simulation ZK protocols [1]. However, this work does not provide a 3-round, negligible-error ZK protocol for NP. To date, there have been only two positive results. One is that of [11, 12], broken and recovered in this paper. The other, which builds a proof system rather than an argument, is reported in [14] and further documented in [13]. It also relies on non-standard assumptions, but different from the Knowledge of Exponent type ones. Roughly, they assume the existence of a hash function such that a certain discrete-log-based protocol, that uses this hash function and is related to the non-interactive OT of [3], is a proof of knowledge.

---

[1] KEA2 was not shown by [12] to imply KEA1. Our proof of Proposition 2 does extend to establish it, but the point is moot since KEA2 is false and hence of course implies everything anyway.

2 Preliminaries

If x is a binary string, then $|x|$ denotes its length, and if $n \geq 1$ is an integer, then $|n|$ denotes the length of its binary encoding, meaning the unique integer ℓ such that $2^{\ell-1} \leq n < 2^{\ell}$. The empty string is denoted ε. We let $\mathbb{N} = \{1, 2, 3, \ldots\}$ be the set of positive integers. If q is a prime number such that $2q+1$ is also prime, then we denote by G_q the subgroup of quadratic residues of $\mathbb{Z}^*_{2q+1}$. (Operations are modulo $2q+1$ but we will omit writing " $\bmod\ 2q+1$" for simplicity.) Recall this is a cyclic subgroup of order q. If g is a generator of G_q then we let $\mathrm{DLog}_{q,g}\colon\ G_q \to \mathbb{Z}_q$ denote the associated discrete logarithm function, meaning $\mathrm{DLog}_{q,g}(g^a) = a$ for any $a \in \mathbb{Z}_q$. We let

$$GL \;=\; \{\, (q, g) \,:\, q, 2q+1 \text{ are primes and } g \text{ is a generator of } G_q \,\}\ .$$

For any $n \in \mathbb{N}$ we let GL_n be the set of all $(q, g) \in GL$ such that the length of the binary representation of $2q+1$ is n bits, i.e.,

$$GL_n \;=\; \{\, (q, g) \in GL \,:\, |2q+1| = n \,\}\ .$$

Assumptions and problems in [11, 12] involve circuits. A family of circuits $\mathbf{C} = \{\mathbf{C}_n\}_{n \in \mathbb{N}}$ contains one circuit for each value of $n \in \mathbb{N}$. It is poly-size if there is a polynomial p such that the size of $\mathbf{C}_n$ is at most $p(n)$ for all $n \in \mathbb{N}$. Unless otherwise stated, *circuits are deterministic*. If they are randomized, we will say so explicitly. We now recall the DLA following [12].

Assumption 1. [**DLA**] Let $\mathbf{I} = \{\mathbf{I}_n\}_{n \in \mathbb{N}}$ be a family of randomized circuits, and $\nu\colon\ \mathbb{N} \to [0, 1]$ a function. We associate to any $n \in \mathbb{N}$ and any $(q, g) \in GL_n$ the following experiment:

Experiment $\mathbf{Exp}^{\mathrm{dl}}_{\mathbf{I}}(n, q, g)$

$a \stackrel{\$}{\leftarrow} \mathbb{Z}_q$; $A \leftarrow g^a$; $\bar{a} \stackrel{\$}{\leftarrow} \mathbf{I}_n(q, g, A)$; If $a = \bar{a}$ then return 1 else return 0

We let

$$\mathbf{Adv}^{\mathrm{dl}}_{\mathbf{I}}(n, q, g) \;=\; \Pr\left[\mathbf{Exp}^{\mathrm{dl}}_{\mathbf{I}}(n, q, g) = 1\right]$$

denote the *advantage* of $\mathbf{I}$ on inputs n, q, g, the probability being over the random choice of a and the coins of $\mathbf{I}_n$, if any. We say that $\mathbf{I}$ has *success bound* ν if

$$\forall n \in \mathbb{N}\ \forall (q, g) \in GL_n \;:\; \mathbf{Adv}^{\mathrm{dl}}_{\mathbf{I}}(n, q, g) \leq \nu(n)\ .$$

We say that the *Discrete Logarithm Assumption (DLA) holds* if for every poly-size family of circuits $\mathbf{I}$ there exists a negligible function ν such that $\mathbf{I}$ has success bound ν. ▌

The above formulation of the DLA, which, as we have indicated, follows [12], has some non-standard features that are important for their results. Let us discuss these briefly.

First, we note that the definition of the success bound is not with respect to (q, g) being chosen according to some distribution as is standard, but rather makes the stronger requirement that the advantage of $\mathbf{I}$ is small for all (q, g).

Second, we stress that the assumption only requires poly-size families of *deterministic* circuits to have a negligible success bound. However, in their proofs, which aim to contradict the DLA, Hada and Tanaka [11, 12] build adversaries that are poly-size families of randomized circuits, and then argue that these can be converted to related poly-size families of deterministic circuits that do not have a negligible success bound. We will also need to build such randomized adversaries, but, rather than using ad hoc conversion arguments repeated across proofs, we note the following more general Proposition, which simply says that DLA, as per Assumption 1, implies that poly-size families of randomized circuits also have a negligible success bound. We will appeal to this in several later places in this paper.

Proposition 1. *Assume the DLA, and let* $\mathbf{J} = \{\mathbf{J}_n\}_{n\in\mathbb{N}}$ *be a poly-size family of* randomized *circuits. Then there exists a negligible function* ν *such that* $\mathbf{J}$ *has success bound* ν. ▮

As is typical in such claims, the proof proceeds by showing that for every n there exists a "good" choice of coins for $\mathbf{J}_n$, and by embedding these coins we get a deterministic circuit. For completeness, we provide the proof in the full version of this paper [4].

3 KEA2 is false

We begin by recalling the assumption. Our presentation is slightly different from, but clearly equivalent to, that of [12]: we have merged the two separate conditions of their formalization into one. Recall that they refer to this assumption as "non-uniform-DA2," and it was referred to, under a different and incorrect formalization, as SDHA2 in [11].

Assumption 2. [**KEA2**] Let $\mathbf{A} = \{\mathbf{A}_n\}_{n\in\mathbb{N}}$ and $\bar{\mathbf{A}} = \{\bar{\mathbf{A}}_n\}_{n\in\mathbb{N}}$ be families of circuits, and ν: $\mathbb{N} \to [0,1]$ a function. We associate to any $n \in \mathbb{N}$, any $(q,g) \in GL_n$, and any $A \in G_q$ the following experiment:

Experiment $\mathbf{Exp}^{\mathrm{kea2}}_{\mathbf{A},\bar{\mathbf{A}}}(n,q,g,A)$

$b \stackrel{\$}{\leftarrow} \mathbb{Z}_q$; $B \leftarrow g^b$; $X \leftarrow A^b$

$(C,Y) \leftarrow \mathbf{A}_n(q,g,A,B,X)$; $c \leftarrow \bar{\mathbf{A}}_n(q,g,A,B,X)$

If ($Y = C^b$ AND $g^c \neq C$ AND $A^c \neq C$) then return 1 else return 0

We let

$$\mathbf{Adv}^{\mathrm{kea2}}_{\mathbf{A},\bar{\mathbf{A}}}(n,q,g,A) \;=\; \Pr\left[\mathbf{Exp}^{\mathrm{kea2}}_{\mathbf{A},\bar{\mathbf{A}}}(n,q,g,A) = 1\right]$$

denote the *advantage* of $\mathbf{A}$ relative to $\bar{\mathbf{A}}$ on inputs n,q,g,A. We say that $\bar{\mathbf{A}}$ is a *kea2-extractor for* $\mathbf{A}$ *with error bound* ν if

$$\forall n \in \mathbb{N}\ \forall (q,g) \in GL_n\ \forall A \in G_q \;:\; \mathbf{Adv}^{\mathrm{kea2}}_{\mathbf{A},\bar{\mathbf{A}}}(n,q,g,A) \leq \nu(n)\ .$$

We say that *KEA2 holds* if for every poly-size family of circuits $\mathbf{A}$ there exists a poly-size family of circuits $\bar{\mathbf{A}}$ and a negligible function ν such that $\bar{\mathbf{A}}$ is a kea2-extractor for $\mathbf{A}$ with error bound ν. ▮

We stress again that in the above formulations, following [12], both the adversary and the extractor are families of *deterministic* circuits. One can consider various variants of the assumptions, including an extension to families of randomized ciruits, and we discuss these variants following the theorem below.

Theorem 1. *If the DLA holds then KEA2 is false.* ▮

The basic idea behind the failure of the assumption, as sketched in Section 1, is simple. Consider an adversary given input q, g, A, B, X, where $A = g^a, B = g^b$ and $X = g^{ab}$. The assumption says that there are only two ways for the adversary to output a pair C, Y satisfying $Y = C^b$. One way is to pick some c, let $C = g^c$ and let $Y = B^c$. The other way is to pick some c, let $C = A^c$ and let $Y = X^c$. The assumption thus states that the adversary "knows" c such that either $C = g^c$ (i.e., $c = \mathrm{DLog}_{q,g}(C)$) or $C = A^c$ (i.e., $c = \mathrm{DLog}_{q,A}(C)$). This ignores the possibility of performing a linear combination of the two steps above. In other words, an adversary might pick c_1, c_2, let $C = g^{c_1}A^{c_2}$ and $Y = B^{c_1}X^{c_2}$. In this case, $Y = C^b$ but the adversary does not appear to necessarily know $\mathrm{DLog}_{q,g}(C) = c_1 + c_2\mathrm{DLog}_{q,g}(A)$ or $\mathrm{DLog}_{q,A}(C) = c_1\mathrm{DLog}_{q,A}(g) + c_2$.

However, going from this intuition to an actual proof that the assumption is false takes some work, for several reasons. The above may be intuition that there exists an adversary for which there would not exist an extractor, but we need to *prove* that there is no extractor. This cannot be done unconditionally, since certainly if the discrete logarithm problem (DLP) is easy, then in fact there is an extractor: it simply computes $\mathrm{DLog}_{q,g}(C)$ and returns it. Accordingly, our strategy will be to present an adversary $\mathbf{A}$ for which we can prove that if there exists an extractor $\bar{\mathbf{A}}$ then there is a method to efficiently compute the discrete logarithm of A.

An issue in implementing this is that the natural adversary $\mathbf{A}$ arising from the above intuition is randomized, picking c_1, c_2 at random and forming C, Y as indicated, but our adversaries must be deterministic. We resolve this by designing an adversary that makes certain specific choices of c_1, c_2. We now proceed to the formal proof.

Proof of Theorem 1. Assume to the contrary that KEA2 is true. We show that the DLP is easy.

The outline of the proof is as follows. We first construct an adversary $\mathbf{A}$ for the KEA2 problem. By assumption, there exists for it an extractor $\bar{\mathbf{A}}$ with negligible error bound. Using $\bar{\mathbf{A}}$, we then present a poly-size family of randomized circuits $\mathbf{J} = \{\mathbf{J}_n\}_{n\in\mathbb{N}}$ and show that it does not have a negligible success bound. By Proposition 1, this contradicts the DLA.

The poly-size family of circuits $\mathbf{A} = \{\mathbf{A}_n\}_{n\in\mathbb{N}}$ is presented in Figure 1. Now, under KEA2, there exists a poly-size family of circuits $\bar{\mathbf{A}} = \{\bar{\mathbf{A}}_n\}_{n\in\mathbb{N}}$ and a negligible function ν such that $\bar{\mathbf{A}}$ is an extractor for $\mathbf{A}$ with error bound ν. Using $\bar{\mathbf{A}}$, we define the poly-size family of circuits $\mathbf{J} = \{\mathbf{J}_n\}_{n\in\mathbb{N}}$ shown in Figure 1.

Claim 1. For all $n \in \mathbb{N}$, all $(q, g) \in GL_n$ and all $A \in G_q$

$$\Pr\left[\, \bar{a} \stackrel{\$}{\leftarrow} \mathbf{J}_n(q, g, A) \ :\ g^{\bar{a}} \neq A \,\right] \ \le\ \nu(n)\ . \qquad ▮$$

$\mathbf{A}_n(q, g, A, B, X)$	$\mathbf{J}_n(q, g, A)$
$C \leftarrow gA$	$b \xleftarrow{\$} \mathbb{Z}_q$; $B \leftarrow g^b$; $X \leftarrow A^b$
$Y \leftarrow BX$	$c \leftarrow \bar{\mathbf{A}}_n(q, g, A, B, X)$
Return (C, Y)	$C \leftarrow gA$
	If $g^c = C$ then $\bar{a} \leftarrow (c-1) \bmod q$ EndIf
	If $A^c = C$ then $\bar{a} \leftarrow (c-1)^{-1} \bmod q$ EndIf
	Return $\bar{a}$

Fig. 1. Adversary $\mathbf{A} = \{\mathbf{A}_n\}_{n \in \mathbb{N}}$ for the KEA2 problem and adversary $\mathbf{J} = \{\mathbf{J}_n\}_{n \in \mathbb{N}}$ for the DLP, for the proof of Theorem 1.

Note the claim shows much more than we need. Namely, $\mathbf{J}$ does not merely have a success bound that is not negligible. In fact, it succeeds with probability almost one.

Proof (Claim 1). We let $\Pr[\cdot]$ denote the probability in the experiment of executing $\mathbf{J}_n(q, g, A)$. We first write some inequalities leading to the claim and then justify them:

$$\Pr\left[g^{\bar{a}} \neq A\right] \leq \Pr\left[g^c \neq C \wedge A^c \neq C\right] \tag{1}$$

$$\leq \mathbf{Adv}^{\mathrm{kea2}}_{\mathbf{A},\bar{\mathbf{A}}}(n, q, g, A) \tag{2}$$

$$\leq \nu(n) . \tag{3}$$

We justify Equation (1) by showing that if $g^c = C$ or $A^c = C$ then $g^{\bar{a}} = A$. First assume $g^c = C$. Since $C = gA$, we have $g^c = gA$, whence $A = g^{c-1}$. Since we set $\bar{a} = (c-1) \bmod q$, we have $A = g^{\bar{a}}$. Next assume $A^c = C$. Since $C = gA$, we have $A^c = gA$, whence $A^{c-1} = g$. Now observe that $c \neq 1$, because otherwise $A^c = A \neq gA$. (Since g is a generator, it is not equal to 1). Since $c \neq 1$ and q is prime, $c-1$ has an inverse modulo q which we have denoted by $\bar{a}$. Raising both sides of the equation "$A^{c-1} = g$" to the power $\bar{a}$ we get $A = g^{\bar{a}}$.

$\mathbf{Exp}^{\mathrm{kea2}}_{\mathbf{A},\bar{\mathbf{A}}}(n, q, g, A)$ returns 1 exactly when $Y = C^b$ and $g^c \neq C$ and $A^c \neq C$. By construction of $\mathbf{A}$, we have $C = gA$ and $Y = BX$, and thus $Y = C^b$, so $\mathbf{Exp}^{\mathrm{kea2}}_{\mathbf{A},\bar{\mathbf{A}}}(n, q, g, A)$ returns 1 exactly when $g^c \neq C$ and $A^c \neq C$. This justifies Equation (2).

Equation (3) is justified by the assumption that $\bar{\mathbf{A}}$ is an extractor for $\mathbf{A}$ with error bound ν. ▮

Claim 1 implies that $\mathbf{J}$ does not have a negligible success bound, which, by Proposition 1, shows that the DLP is not hard, contradicting the assumption made in this Theorem. This completes the proof of Theorem 1.

Extensions and variants. There are many ways in which the formalization of Assumption 2 can be varied to capture the same basic intuition. However, Theorem 1 extends to these variants as well. Let us discuss this briefly.

As mentioned above, we might want to allow the adversary to be randomized. (In that case, it is important that the extractor get the coins of the adversary as an additional input, since otherwise the assumption is clearly false.) Theorem 1 remains true for the resulting assumption, in particular because it is stronger than the original assumption. (Note however that the proof of the theorem would be easier for this stronger assumption.)

Another variant is that adversaries and extractors are uniform, namely standard algorithms, not circuits. (In this case we should certainly allow both to be randomized, and should again give the extractor the coins of the adversary.) Again, it is easy to see that Theorem 1 extends to show that the assumption remains false.

4 The KEA3 assumption

The obvious fix to KEA2 is to take into account the possibility of linear combinations by saying this is the only thing the adversary can do. This leads to the following.

Assumption 3. **[KEA3]** Let $\mathbf{A} = \{\mathbf{A}_n\}_{n\in\mathbb{N}}$ and $\bar{\mathbf{A}} = \{\bar{\mathbf{A}}_n\}_{n\in\mathbb{N}}$ be families of circuits, and $\nu\colon \mathbb{N} \to [0, 1]$ a function. We associate to any $n \in \mathbb{N}$, any $(q, g) \in GL_n$, and any $A \in G_q$ the following experiment:

Experiment $\mathbf{Exp}^{\mathrm{kea3}}_{\mathbf{A},\bar{\mathbf{A}}}(n, q, g, A)$

$b \xleftarrow{\$} \mathbb{Z}_q$; $B \leftarrow g^b$; $X \leftarrow A^b$

$(C, Y) \leftarrow \mathbf{A}_n(q, g, A, B, X)$; $(c_1, c_2) \leftarrow \bar{\mathbf{A}}_n(q, g, A, B, X)$

If ($Y = C^b$ AND $g^{c_1}A^{c_2} \neq C$) then return 1 else return 0

We let

$$\mathbf{Adv}^{\mathrm{kea3}}_{\mathbf{A},\bar{\mathbf{A}}}(n, q, g, A) \;=\; \Pr\left[\mathbf{Exp}^{\mathrm{kea3}}_{\mathbf{A},\bar{\mathbf{A}}}(n, q, g, A) = 1\right]$$

denote the *advantage* of $\mathbf{A}$ relative to $\bar{\mathbf{A}}$ on inputs n, q, g, A. We say that $\bar{\mathbf{A}}$ is a *kea3-extractor for* $\mathbf{A}$ *with error bound* ν if

$$\forall n \in \mathbb{N}\ \forall (q, g) \in GL_n\ \forall A \in G_q \;:\; \mathbf{Adv}^{\mathrm{kea3}}_{\mathbf{A},\bar{\mathbf{A}}}(n, q, g, A) \leq \nu(n)\ .$$

We say that *KEA3 holds* if for every poly-size family of circuits $\mathbf{A}$ there exists a poly-size family of circuits $\bar{\mathbf{A}}$ and a negligible function ν such that $\bar{\mathbf{A}}$ is a kea3-extractor for $\mathbf{A}$ with error bound ν. ▮

We have formulated this assumption in the style of the formalization of KEA2 of [12] given in Assumption 2. Naturally, variants such as discussed above are possible. Namely, we could strengthen the assumption to allow the adversary to be a family of randomized circuits, of course then giving the extractor the adversary's coins as an additional input. We do not do this because we do not need it for what follows. We could also formulate a uniform-complexity version of the assumption. We do not do this because it does not suffice to prove the results that follow. However, these extensions or variations might be useful in other contexts.

In Appendix A we recall the formalization of KEA1 and prove the following:

Proposition 2. *KEA3 implies KEA1.* ▮

This indicates that KEA3 is a natural extension of KEA1.

5 Three-round zero knowledge

The falsity of KEA2 renders vacuous the result of [11, 12] saying that there exists a negligible-error, 3-round ZK argument for NP. In this section we look at recovering this result.

We first consider the protocol of [11, 12], here called HTP. What has been lost is the proof of soundness (i.e., of negligible error). The simplest thing one could hope for is to re-prove soundness of HTP under KEA3 without modifying the protocol. However, we identify a bug in HTP that renders it unsound. This bug has nothing to do with the assumptions on which the proof of soundness was or can be based.

The bug is, however, small and easily fixed. We consider a modified protocol which we call pHTP. We are able to show it is sound (i.e., has negligible error) under KEA3. Since we have modified the protocol we need to re-establish ZK under KEA1 as well, but this is easily done.

Arguments. We begin by recalling some definitions. An argument for an NP language L [6] is a two-party protocol in which a polynomial-time prover tries to "convince" a polynomial-time verifier that their common input x belongs to L. (A party is said to be polynomial time if its running time is polynomial in the length of the common input.) In addition to x, the prover has an auxiliary input a. The protocol is a message exchange at the end of which the verifier outputs a bit indicating its decision to accept or reject. The probability (over the coin tosses of both parties) that the verifier accepts is denoted $\mathbf{Acc}_V^{P,a}(x)$. The formal definition follows.

Definition 1. A two-party protocol (P, V), where P and V are both polynomial time, is an *argument for L with error probability* $\delta : \mathbb{N} \to [0, 1]$, if the following conditions are satisfied:

Completeness: For all $x \in L$ there exists $w \in \{0,1\}^*$ such that $\mathbf{Acc}_V^{P,w}(x) = 1$.

Soundness: For all probabilistic polynomial-time algorithms $\widehat{P}$, all sufficiently long $x \notin L$, and all $a \in \{0,1\}^*$, $\mathbf{Acc}_V^{\widehat{P},a}(x) \le \delta(|x|)$.

We say (P, V) is a *negligible-error argument for L* if there exists a negligible function $\delta : \mathbb{N} \to [0, 1]$ such that (P, V) is an argument for L with error probability δ. ▮

Canonical protocols. The 3-round protocol proposed by [11, 12], which we call HTP, is based on a 3-round argument $(\bar{P}, \bar{V})$ for an NP-complete language L with the following properties:

Prover $\bar{P}$		Verifier $\bar{V}$
Initial State $St = (x, w, R)$		
$((\textsc{Cmt}, q, g), St) \leftarrow \bar{P}(\varepsilon; St)$		$d \leftarrow 1$
	$\xrightarrow{(\textsc{Cmt}, q, g)}$	
		$n \leftarrow \lvert x \rvert$ If $(q, g) \notin GL_n$ then $d \leftarrow 0$ EndIf $r \xleftarrow{\$} \mathbb{Z}_q^*$; $\textsc{Ch} \leftarrow g^r$
	$\xleftarrow{\textsc{Ch}}$	
$(\textsc{Rsp}, St) \leftarrow \bar{P}(\textsc{Ch}; St)$		
	$\xrightarrow{\textsc{Rsp}}$	
		If $\mathrm{DEC}_x((\textsc{Cmt}, q, g), \textsc{Ch}, \textsc{Rsp}) = 0$ then $d \leftarrow 0$ EndIf

Fig. 2. A 3-round argument. The common input is x. Prover $\bar{P}$ has auxiliary input w and random tape R, and maintains state St. Verifier $\bar{V}$ returns boolean decision d.

(1) The protocol is of the form depicted in Figure 2. The prover is identified with a function $\bar{P}$ that given an incoming message M_{in} (this is ε when the prover is initiating the protocol) and its current state St, returns an outgoing message M_{out} and an updated state. The initial state of the prover is (x, w, R), where x is the common input, w is an auxiliary input and R is a random tape. The prover's first message is called its *commitment*. This is a tuple consisting of a string $\textsc{Cmt}$, a prime number q and an element g, where $(q, g) \in GL_{|x|}$. The verifier selects a *challenge* $\textsc{Ch}$ uniformly at random from G_q, and, upon receiving a *response* $\textsc{Rsp}$ from the prover, applies a deterministic *decision predicate* $\mathrm{DEC}_x((\textsc{Cmt}, q, g), \textsc{Ch}, \textsc{Rsp})$ to compute a boolean decision.

(2) For any $x \notin L$ and any commitment $(\textsc{Cmt}, q, g)$, where $(q, g) \in GL_{|x|}$, there is at most one challenge $\textsc{Ch} \in G_q$ for which there exists a response $\textsc{Rsp} \in \{0,1\}^*$ such that $\mathrm{DEC}_x((\textsc{Cmt}, q, g), \textsc{Ch}, \textsc{Rsp}) = 1$. This property is called *strong soundness*.

(3) The protocol is honest-verifier zero knowledge (HVZK),meaning there exists a probabilistic polynomial-time *simulator* S such that the following two ensembles are computationally indistinguishable:

$$\{S(x)\}_{x \in L} \text{ and } \{\mathbf{View}_{\bar{V}}^{\bar{P}, W(x)}(x)\}_{x \in L} ,$$

where W is any function that given an input in L returns a witness to its membership in L, and $\mathbf{View}_{\bar{V}}^{\bar{P}, W(x)}(x)$, is a random variable taking value $\bar{V}$'s internal coin tosses and the sequence of messages it receives during an interaction between prover $\bar{P}$ (with auxiliary input $W(x)$) and verifier $\bar{V}$ on common input x.

Prover P		Verifiers V, V'
Initial State $St = (x, w, R)$		
$((\text{Cmt}, q, g), St) \leftarrow \bar{P}(\varepsilon; St)$		
$a \xleftarrow{\$} \mathbb{Z}_q\,;\ A \leftarrow g^a$		$d \leftarrow 1$
	$\xrightarrow{(\text{Cmt}, q, g, A)}$	
		$n \leftarrow \lvert x \rvert$
		If $(q, g) \notin GL_n$ then $d \leftarrow 0$ EndIf
		$b \xleftarrow{\$} \mathbb{Z}_q^*\,;\ B \leftarrow g^b\,;\ X \leftarrow A^b$
	$\xleftarrow{(B, X)}$	
If $X \neq B^a$ then abort		
else $c \xleftarrow{\$} \mathbb{Z}_q^*\,;\ C \leftarrow g^c\,;\ \text{Ch} \leftarrow B^c$		
$(\text{Rsp}, St) \leftarrow \bar{P}(\text{Ch}; St)$ EndIf		
	$\xrightarrow{(\text{Rsp}, C, \text{Ch})}$	
		If $\text{Ch} \neq C^b\ \vee$ $\text{Ch} = 1\ \vee$
		$\text{DEC}_x((\text{Cmt}, q, g), \text{Ch}, \text{Rsp}) = 0$
		then $d \leftarrow 0$ EndIf

Fig. 3. HTP and pHTP. Verifier V of protocol HTP $= (P, V)$ does not include the highlighted portion. Verifier V' of protocol pHTP $= (P, V')$ does.

If $(\bar{P}, \bar{V})$ is a 3-round argument for an NP-complete language, meeting the three conditions above, then we refer to $(\bar{P}, \bar{V})$ as a *canonical argument*. In what follows, we assume that we have such canonical arguments. They can be constructed in various ways. For example, a canonical argument can be constructed by modifying the parallel composition of Blum's zero-knowledge protocol for the Hamiltonian circuit problem [5], as described in [11, 12].

The Hada-Tanaka protocol. Let $(\bar{P}, \bar{V})$ be a canonical argument for an NP-complete language L, and let DEC be the verifier's decision predicate. The Hada-Tanaka protocol HTP $= (P, V)$ is described in Figure 3. Note V's decision predicate does not include the highlighted portion of its code.

We now observe that the HTP protocol is unsound. More precisely, there exist canonical arguments such that the HTP protocol based on them does not have negligible error. This is true for any canonical argument $(\bar{P}, \bar{V})$ satisfying the extra condition that for infinitely many $x \notin L$ there exists a commitment (Cmt_x, q_x, g_x) for which there is a response Rsp_x to challenge 1 that will make the verifier accept. There are many such canonical arguments. For instance, a canonical argument satisfying this condition results from using an appropriate encoding of group elements in Hada and Tanaka's modification of the parallel composition of Blum's zero-knowledge protocol for the Hamiltonian circuit problem.

Proposition 3. *Let HTP be the Hada-Tanaka protocol based on a canonical argument satisfying the condition stated above. Then there exists a polynomial-time prover for HTP that can make the verifier accept with probability one for infinitely many common inputs not in L.* ▮

Proof (Proposition 3). Let $(\bar{P}, \bar{V})$ be the canonical argument and let V be the verifier of the corresponding protocol HTP. Consider a cheating prover $\widehat{P}$ that on initial state $(x, ((\textsc{Cmt}_x, q_x, g_x), \textsc{Rsp}_x), \varepsilon)$ selects an exponent $a \in \mathbb{Z}_{q_x}$ uniformly at random, and sends $(\textsc{Cmt}_x, q_x, g_x, g_x^a)$ as its commitment to verifier V. Upon receiving a challenge (B, X), it checks if $X = B^a$. If not, it aborts. Otherwise, it sends $(\textsc{Rsp}_x, 1, 1)$ as its response to V. By the assumption about protocol $(\bar{P}, \bar{V})$, for infinitely many $x \notin L$ there exists an auxiliary input $y = ((\textsc{Cmt}_x, q_x, g_x), \textsc{Rsp}_x) \in \{0,1\}^*$ such that $\mathbf{Acc}_V^{\widehat{P},y}(x) = 1$. ▮

Protocol pHTP. The above attack can be avoided by modifying the verifier to include the highlighted portion of the code in Figure 3. We call the resulting verifier V'. The following guarantees that the protocol pHTP $= (P, V')$ is sound under KEA3, if the DLP is hard.

Theorem 2. *If KEA3 holds, the DLA holds, and $(\bar{P}, \bar{V})$ is a canonical 3-round argument for an NP-complete language L, then pHTP $= (P, V')$ as defined in Figure 3 is a negligible-error argument for L.*

Proof of Theorem 2. The proof is almost identical to that of [12]. For completeness, however, we provide it.

Completeness follows directly from the completeness of protocol $(\bar{P}, \bar{V})$. To prove soundness, we proceed by contradiction. Assume that pHTP is not sound, i.e., there is no negligible function δ such that the soundness condition in Definition 1 holds with respect to δ. We show that the DLP is easy under KEA3.

By the assumption that pHTP is not sound and a result of [2], there exists a probabilistic polynomial-time algorithm $\widehat{P}$ such that the function

$$\mathbf{Err}_{\widehat{P}}(n) \;=\; \max\{\, \mathbf{Acc}_{V'}^{\widehat{P},a}(x) \;:\; x \in \{0,1\}^n \wedge x \notin L \wedge a \in \{0,1\}^* \,\}^2$$

is not negligible. Hence there exists a probabilistic polynomial-time algorithm $\widehat{P}$, a polynomial p, and an infinite set $S = \{\,(x,a) : x \in \{0,1\}^* \setminus L \wedge a \in \{0,1\}^*\}$ such that for every $(x,a) \in S$

$$\mathbf{Acc}_{V'}^{\widehat{P},a}(x) \;>\; 1/p(|x|)\;, \tag{4}$$

and $\{\, x \in \{0,1\}^* \;:\; \exists a \in \{0,1\}^* \text{ such that } (x,a) \in S \,\}$ is infinite.

Since $\widehat{P}$ takes an auxiliary input a, we may assume, without loss of generality, that $\widehat{P}$ is deterministic. We also assume that, if $(\textsc{Cmt}, q', g', A')$ is $\widehat{P}$'s commitment on input ε when the initial state is (x, a, ε), for some $x, a \in \{0,1\}^*$ with $|x| = n$, then $(q', g') \in GL_n$. (There exists a prover $\widehat{P}'$ for which $\mathbf{Acc}_{V'}^{\widehat{P}',a}(x) =$

[2] We note that this set is finite since $\widehat{P}$ is a polynomial-time algorithm and $\mathbf{Acc}_{V'}^{\widehat{P},a}(x)$ depends only on the first $t_{\widehat{P}}(|x|)$ bits of a, where $t_{\widehat{P}}(\cdot)$ is the running time of $\widehat{P}$.

$\mathbf{A}_n(q, g, A, B, X) \quad // n \in K$
$\quad St \leftarrow (x, a, \varepsilon)$; $((\text{CMT}, q', g', A'), St) \leftarrow \widehat{P}(\varepsilon; St)$
$\quad$ If $q' \neq q \vee g' \neq g \vee A' \neq A$ then return $(1, 1)$
$\quad$ else $((\text{RSP}, C, \text{CH}), St) \leftarrow \widehat{P}((B, X); St)$; return (C, CH) EndIf

$\mathbf{A}_n(q, g, A, B, X) \quad // n \notin K$
$\quad$ Return $(1, 1)$

$\mathbf{J}_n(q, g, A) \quad // n \in K$
$\quad St \leftarrow (x, a, \varepsilon)$; $((\text{CMT}, q', g', A'), St) \leftarrow \widehat{P}(\varepsilon; St)$
$\quad$ If $q' \neq q \vee g' \neq g$ then return $\perp$ EndIf
$\quad b \xleftarrow{\$} \mathbb{Z}_q$; $B \leftarrow A \cdot g^b$; $X \leftarrow B^{a'}$
$\quad ((\text{RSP}, C, \text{CH}), St_1) \leftarrow \widehat{P}((B, X); St)$; $(c_1, c_2) \leftarrow \bar{\mathbf{A}}_n(q, g, A', B, X)$
$\quad$ If $\text{DEC}_x((\text{CMT}, q, g), \text{CH}, \text{RSP}) = 0 \vee \text{CH} \neq B^{c_1} X^{c_2}$ then return $\perp$ EndIf
$\quad b' \xleftarrow{\$} \mathbb{Z}_q$; $B' \leftarrow g^{b'}$; $X' \leftarrow B'^{a'}$
$\quad$ If $B = B'$ then $\bar{a} \leftarrow b' - b \bmod q$; return $\bar{a}$ EndIf
$\quad ((\text{RSP}', C', \text{CH}'), St_1') \leftarrow \widehat{P}((B', X'); St)$; $(c_1', c_2') \leftarrow \bar{\mathbf{A}}_n(q, g, A', B', X')$
$\quad$ If $\text{DEC}_x((\text{CMT}, q, g), \text{CH}', \text{RSP}') = 0 \vee \text{CH}' \neq B'^{c_1'} X'^{c_2'}$ then return $\perp$ EndIf
$\quad$ If $c_1 + a'c_2 \not\equiv 0 \pmod q$ then
$\quad\quad \bar{a} \leftarrow (b'c_1' + b'a'c_2' - bc_1 - ba'c_2) \cdot (c_1 + a'c_2)^{-1} \bmod q$; return $\bar{a}$
$\quad$ else return $\perp$ EndIf

$\mathbf{J}_n(q, g, A) \quad // n \notin K$
$\quad$ Return $\perp$

Fig. 4. Adversary $\mathbf{A} = \{\mathbf{A}_n\}_{n\in\mathbb{N}}$ for the KEA3 problem and adversary $\mathbf{J} = \{\mathbf{J}_n\}_{n\in\mathbb{N}}$ for the DLP, for the proof of Theorem 2.

$\mathbf{Acc}_{V'}^{\widehat{P},a}(x)$ for every $x, a \in \{0,1\}^*$ and this assumption holds.) We will use $\widehat{P}$ to construct an adversary $\mathbf{A}$ for the KEA3 problem. By assumption, there exists for it an extractor $\bar{\mathbf{A}}$ with negligible error bound. Using $\bar{\mathbf{A}}$ and $\widehat{P}$, we then present a poly-size family of randomized circuits $\mathbf{J} = \{\mathbf{J}_n\}_{n\in\mathbb{N}}$ and show that it does not have a negligible success bound. By Proposition 1, this shows that the DLP is not hard.

Let $K = \{\, n \in \mathbb{N} : \exists\, (x, a) \in S$ such that $|x| = n \,\}$. We observe that K is an infinite set. For each $n \in K$, fix $(x, a) \in S$ such that $|x| = n$. The poly-size family of circuits $\mathbf{A} = \{\mathbf{A}_n\}_{n\in\mathbb{N}}$ is presented in Figure 4. Now, under KEA3, there exists a poly-size family of circuits $\bar{\mathbf{A}} = \{\bar{\mathbf{A}}_n\}_{n\in\mathbb{N}}$ and a negligible function ν such that $\bar{\mathbf{A}}$ is an extractor for $\mathbf{A}$ with error bound ν. For each $n \in K$, let $a' = \text{DLog}_{q',g'}(A')$, where (CMT, q', g', A') is $\widehat{P}$'s commitment on input ε when the initial state is (x, a, ε). Using $\bar{\mathbf{A}}$, we define the poly-size family of circuits $\mathbf{J} = \{\mathbf{J}_n\}_{n\in\mathbb{N}}$ shown in Figure 4. The proof of the following is in [4].

Claim 2. For infinitely many $n \in \mathbb{N}$ there exists $(q, g) \in GL_n$ such that for every $A \in G_q$

$$\Pr\left[\, \bar{a} \xleftarrow{\$} \mathbf{J}_n(q, g, A) \;:\; g^{\bar{a}} = A \,\right] \;>\; \frac{1}{p(n)^2} - \frac{8}{2^n p(n)} - 2\nu(n)\;. \;\blacksquare$$

Claim 2 implies that $\mathbf{J}$ does not have a negligible success bound, which, by Proposition 1, shows that the DLP is not hard, contradicting the assumption made in this Theorem.

Zero knowledge of pHTP. Having modified HTP, we need to revisit the zero knowledge. Hada and Tanaka proved that if the canonical argument is HVZK (property (3) above) then HTP is zero knowledge under KEA1. However, we observe that pHTP modifies only the verifier, not the prover. Furthermore, only the decision predicate of the verifier is modified, not the messages it sends. This means that the view (i.e., the internal coin tosses and the sequence of messages received during an interaction with a prover P) of verifier V' of pHTP is identical to that of verifier V of HTP. Thus, zero knowledge of pHTP follows from zero knowledge of HTP, and in particular is true under the same assumptions, namely KEA1.

Summary. In summary, pHTP is a 3-round protocol that we have shown is a negligible-error argument for NP assuming DLA and KEA3, and is ZK assuming KEA1. Given Proposition 2, this means we have shown that assuming DLA and KEA3 there exists a 3-round negligible-error ZK argument for NP.

Acknowledgments

Mihir Bellare is supported in part by NSF grants CCR-0098123, ANR-0129617 and CCR-0208842, and by an IBM Faculty Partnership Development Award. Adriana Palacio is supported in part by a NSF Graduate Research Fellowship.

Proposition 2 is due to Shai Halevi and we thank him for permission to include it. We thank the Crypto 2004 referees for their comments on the paper.

References

1. B. Barak. How to go beyond the black-box simulation barrier. *Proceedings of the 42nd Symposium on Foundations of Computer Science*, IEEE, 2001.
2. M. Bellare. A note on negligible functions. *Journal of Cryptology*, Vol. 15, No. 4, pp. 271–284, June 2002.
3. M. Bellare and S. Micali. Non-interactive oblivious transfer and applications. *Advances in Cryptology – CRYPTO '89*, Lecture Notes in Computer Science Vol. 435, G. Brassard ed., Springer-Verlag, 1989.
4. M. Bellare and A. Palacio. The Knowledge-of-Exponent assumptions and 3-round zero-knowledge protocols. Full version of the current paper, available via `http://www-cse.ucsd.edu/users/mihir`.
5. M. Blum. How to prove a theorem so no one else can claim it. *Proceedings of the International Congress of Mathematicians*, pp. 1444–1451, 1986.

6. G. Brassard, D. Chaum and C. Crépeau. Minimum disclosure proofs of knowledge. *J. Computer and System Sciences*, Vol. 37, No. 2, pp. 156–189, October 1988.
7. I. Damgård. Towards practical public-key cryptosystems provably-secure against chosen-ciphertext attacks. *Advances in Cryptology – CRYPTO '91*, Lecture Notes in Computer Science Vol. 576, J. Feigenbaum ed., Springer-Verlag, 1991.
8. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. *Advances in Cryptology – CRYPTO '86*, Lecture Notes in Computer Science Vol. 263, A. Odlyzko ed., Springer-Verlag, 1986.
9. O. Goldreich and H. Krawczyk. On the Composition of Zero Knowledge Proof Systems. *SIAM J. on Computing*, Vol. 25, No. 1, pp. 169–192, 1996.
10. S. Goldwasser, S. Micali and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal of Computing*, Vol. 18, No. 1, pp. 186–208, February 1989.
11. S. Hada and T. Tanaka. On the existence of 3-round zero-knowledge protocols. *Advances in Cryptology – CRYPTO '98*, Lecture Notes in Computer Science Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998. [Preliminary version of [12].]
12. S. Hada and T. Tanaka. On the existence of 3-round zero-knowledge protocols. Cryptology ePrint Archive: Report 1999/009, March 1999. `http://eprint.iacr.org/1999/009/`. [Final version of [11].]
13. M. Lepinski. On the existence of 3-round zero-knowledge proofs. SM Thesis, MIT, June 2002. `http://theory.lcs.mit.edu/~cis/theses/lepinski-masters.ps`.
14. M. Lepinski and S. Micali. On the existence of 3-round zero-knowledge proof systems. MIT LCS Technical Memo. 616, April 2001. `http://www.lcs.mit.edu/publications/pubs/pdf/MIT-LCS-TM-616.pdf`.
15. M. Naor. On cryptographic assumptions and challenges. Invited paper and talk, *Advances in Cryptology – CRYPTO '03*, Lecture Notes in Computer Science Vol. 2729, D. Boneh ed., Springer-Verlag, 2003.
16. K. Sakurai and T. Itoh. On the discrepancy between serial and parallel of zero-knowledge protocols. *Advances in Cryptology – CRYPTO '92*, Lecture Notes in Computer Science Vol. 740, E. Brickell ed., Springer-Verlag, 1992.

A KEA3 implies KEA1

We recall KEA1, following [12], but applying the same simplifications as we did for KEA2 so as to merge their two conditions into one:

Assumption 4. **[KEA1]** Let $\mathbf{A} = \{\mathbf{A}_n\}_{n\in\mathbb{N}}$ and $\bar{\mathbf{A}} = \{\bar{\mathbf{A}}_n\}_{n\in\mathbb{N}}$ be families of circuits, and $\nu\colon \mathbb{N} \to [0,1]$ a function. We associate to any $n \in \mathbb{N}$, any $(q,g) \in GL_n$, and any $A \in G_q$ the following experiment:

Experiment $\mathbf{Exp}^{\mathrm{kea1}}_{\mathbf{A},\bar{\mathbf{A}}}(n,q,g)$
  $b \stackrel{\$}{\leftarrow} \mathbb{Z}_q$; $B \leftarrow g^b$
  $(C,Y) \leftarrow \mathbf{A}_n(q,g,B)$; $c \leftarrow \bar{\mathbf{A}}_n(q,g,B)$
  If ($Y = C^b$ AND $g^c \neq C$) then return 1 else return 0

We let

$$\mathbf{Adv}^{\mathrm{kea1}}_{\mathbf{A},\bar{\mathbf{A}}}(n,q,g) \;=\; \Pr\left[\,\mathbf{Exp}^{\mathrm{kea1}}_{\mathbf{A},\bar{\mathbf{A}}}(n,q,g)=1\,\right]$$

denote the *advantage* of $\mathbf{A}$ relative to $\bar{\mathbf{A}}$ on inputs n,q,g. We say that $\bar{\mathbf{A}}$ is a *kea1-extractor for* $\mathbf{A}$ *with error bound* ν if

$$\forall n\in\mathbb{N}\ \forall(q,g)\in GL_n \ : \ \mathbf{Adv}^{\mathrm{kea1}}_{\mathbf{A},\bar{\mathbf{A}}}(n,q,g)\le\nu(n)\ .$$

We say that *KEA1 holds* if for every poly-size family of circuits $\mathbf{A}$ there exists a poly-size family of circuits $\bar{\mathbf{A}}$ and a negligible function ν such that $\bar{\mathbf{A}}$ is a kea1-extractor for $\mathbf{A}$ with error bound ν. ∎

Proof (Proposition 2). Let $\mathbf{A}$ be an adversary (poly-size family of circuits) for KEA1. We need to show there exists a negligible function ν and a poly-size family of circuits $\bar{\mathbf{A}}$ such that $\bar{\mathbf{A}}$ is a kea1-extractor for $\mathbf{A}$ with error-bound ν.

We begin by constructing from $\mathbf{A}$ the following adversary $\mathbf{A}'$ for KEA3:

Adversary $\mathbf{A}'_n(q,g,A,B,X)$
  $(C,Y)\leftarrow\mathbf{A}_n(q,g,B)$
  Return (C,Y)

We have assumed KEA3. Thus there exists a negligible function ν and an extractor $\bar{\mathbf{A}}'$ such that $\bar{\mathbf{A}}'$ is a kea3-extractor for $\mathbf{A}'$ with error bound ν. Now we define an extractor $\bar{\mathbf{A}}$ for $\mathbf{A}$ as follows:

Extractor $\bar{\mathbf{A}}_n(q,g,B)$
  $a\stackrel{\$}{\leftarrow}\mathbb{Z}_q\,;\ A\leftarrow g^a\ ;\ X\leftarrow B^a$
  $(c_1,c_2)\leftarrow\bar{\mathbf{A}}'(q,g,A,B,X)$
  $c\leftarrow c_1+ac_2 \bmod q$
  Return c

We claim that $\bar{\mathbf{A}}$ is a kea1-extractor for $\mathbf{A}$ with error bound ν. To see this, assume $\bar{\mathbf{A}}'_n(q,g,A,B,X)$ is successful, meaning $g^{c_1}A^{c_2}=C$. Then $g^c=g^{c_1+ac_2}=g^{c_1}A^{c_2}=C$ so $\bar{\mathbf{A}}_n(q,g,B)$ is successful as well. ∎