Perfect NIZK with Adaptive Soundness

Masayuki Abe[1] and Serge Fehr[2]

[1] Information Sharing Platform Laboratories
NTT Corporation, Japan
abe.masayuki@lab.ntt.co.jp
[2] CWI Amsterdam, The Netherlands
fehr@cwi.nl

Abstract. This paper presents a very simple and efficient adaptively-sound perfect NIZK argument system for any NP-language. In contrast to recently proposed schemes by Groth, Ostrovsky and Sahai, our scheme does not pose any restriction on the statements to be proven. Besides, it enjoys a number of desirable properties: it allows to re-use the common reference string (CRS), it can handle *arithmetic* circuits, and the CRS can be set-up very efficiently without the need for an honest party. We then show an application of our techniques in constructing *efficient* NIZK schemes for proving arithmetic relations among committed secrets, whereas previous methods required expensive generic NP-reductions.
The security of the proposed schemes is based on a strong non-standard assumption, an extended version of the so-called Knowledge-of-Exponent Assumption (KEA) over bilinear groups. We give some justification for using such an assumption by showing that the commonly-used approach for proving NIZK *arguments* sound does not allow for adaptively-sound statistical NIZK arguments (unless NP $\subset$ P/poly). Furthermore, we show that the assumption used in our construction holds with respect to generic adversaries that do not exploit the specific representation of the group elements. We also discuss how to avoid the non-standard assumption in a pre-processing model.

1 Introduction

1.1 Background

Non-Interactive Zero-Knowledge (NIZK). The notion of NIZK captures the problem of proving a statement by just sending one message and without revealing any additional information besides the validity of the statement, provided that a common reference string (CRS) has been properly set up. Since its introduction by Blum, Feldman and Micali in 1988 [7], NIZK has been a fundamental cryptographic primitive used throughout modern cryptography in essential ways.

There is a considerable amount of literature dedicated to NIZK, in particular to the study of which languages allow for what flavor of NIZK proof. As in case of interactive ZK it is well known that there cannot be statistical NIZK proofs (i.e.,

both ZK and soundness are unconditional) for NP-complete languages unless the polynomial hierarchy collapses [24, 3, 32]. Hence, when considering general NP-languages, this only leaves room for a NIZK proof with *computational* ZK or *computational* soundness (where the proof is also called an *argument*), or both. However, in contrast to interactive ZK where it has long been known that both flavors can exist [9, 8, 25], all proposed NIZK proofs or arguments for general NP-languages have computational ZK (see e.g. [7, 22, 6, 29, 17]). Hence the construction of a statistically NIZK (NISZK) argument has remained an open problem (until very recently, see below). The question of the existence of NISZK arguments is in particular interesting in combination with a result by De Santis *et al.* [17], where they observe that for a strong notion of NIZK, called *same-string* NIZK, soundness can only be computational when considering NP-complete languages (assuming that one-way functions exist).

STATISTICAL NIZK ARGUMENTS. Recently, Groth, Ostrovsky and Sahai proposed an elegant construction for a perfect NIZK (NIPZK) argument for circuit-SAT [26] by using bilinear groups. This shows NIZK *can* come with perfect ZK for any NP-language. However, the scheme only provides security against a non-adaptive dishonest prover who chooses the target instance $x^* \notin L$ (for which it wants to fake a proof) *independent* of the CRS. In an application though, it is likely that the adversary first sees the CRS and then chooses the false statement on which he wants to cheat. Using a counting argument, they argue that under some strengthened assumption their scheme is secure against an adaptive dishonest prover if the size of the circuit to be proven is a-priori limited. However, the bound on the size of the circuit is so restrictive that the circuit must be smaller than sublinear in the bit size of the CRS (as discussed in Section 1.3). Groth *et al.* also proposed a perfect NIZK argument for SAT which is provably secure in Canetti's Universal Composability (UC) framework [10]. However, besides being much less efficient than their first construction, the scheme still does not guarantee unrestricted security against an adaptive dishonest prover who chooses the target instance $x^* \notin L$ depending on the CRS. For instance, the UC security does not exclude the possibility that a dishonest prover comes up with an accepting proof for the statement "the CRS is invalid or S is true" for an arbitrary false statement S. Since in a real-life execution the CRS is assumed to be valid, this is a convincing argument of the false statement S. Accordingly, the existence of an unrestricted statistical or perfect NIZK argument, which does not pose any restriction on the instances to be proven, is still an open problem.

THE KNOWLEDGE-OF-EXPONENT ASSUMPTION. Informally, the Knowledge-of-Exponent Assumption (KEA) says that for certain groups, given a pair g and $\hat{g} = g^x$ of group elements with unknown discrete-log x, the only way to efficiently come up with another pair A and $\hat{A}$ such that $\hat{A} = A^x$ (for the same x) is by raising g and $\hat{g}$ to some power a: $A = g^a$ and $\hat{A} = \hat{g}^a$. KEA was first introduced and used by Damgård in 1991 [14], and later, together with an extended version (KEA2), by Hada and Tanaka [27]. Recently, Bellare and Palacio [5] showed that KEA2 does not hold, and proposed a new extended version called KEA3 in order to save Hada and Tanaka's results. KEA3, which we call XKEA for eXtended KEA,

says that given *two* pairs $(g, \hat{g})$ and $(h, \hat{h})$ with the same unknown discrete-log x, the only way to efficiently come up with another pair A and $\hat{A}$ such that $\hat{A} = A^x$ is by computing $A = g^a h^\alpha$ and $\hat{A} = \hat{g}^a \hat{h}^\alpha$. Assumptions like KEA and XKEA are widely criticized in particular because they do not appear to be "efficiently falsifiable", as Naor put it [30], though Bellare and Palacio showed that this is not necessarily the case.

1.2 Our Result

Based on XKEA over bilinear groups, we construct an adaptively-sound NIPZK argument for circuit-SAT without any restrictions on the instances to be proven. Besides being the first unrestricted adaptively-sound NISZK argument for any NP-language, the proposed scheme enjoys a number of additional desirable properties: It is *same-string* NIZK, which allows to re-use the CRS. It is very efficient: the CRS essentially consists of a few group elements, and a proof consists of a few group elements per multiplication gate; this is comparable (if not better) to the first scheme by Groth *et al.*, which is the most efficient general-purpose NIZK scheme known up to date (see the comparison in [26]). Furthermore, our scheme can also be applied to arithmetic circuits over $\mathbb{Z}_q$ for a large prime q whereas known schemes are tailored to binary circuits; this often allows a more compact representation of the statement to be proven. Finally, the CRS does not need to be set-up by a trusted party. It can efficiently be set-up jointly by the prover and the verifier. Furthermore, it can even be provided solely from a (possibly dishonest) verifier without any correctness proof if we view the proof system as a *zap* [21] rather than a NIZK. We are not aware of any other NIZK arguments or proofs that enjoy all these desirable properties.

Based on the techniques developed for the perfect NIZK argument for SAT, we also construct an *efficient* NIPZK argument for arithmetic relations among committed secrets over $\mathbb{Z}_q$ with large prime q. To the best of our knowledge, all known schemes only work for secrets from restricted domains such as $\mathbb{Z}_2$ and have to rely on generic inefficient reductions to NP-complete problems to handle larger secrets. Our approach in particular allows for additive and multiplicative relations among secrets committed to by standard Pedersen commitments.

We give two justifications for using such a strong non-standard assumption like XKEA. First, we prove that KEA and XKEA hold in the generic group model (even over bilinear groups). This suggests that if there exists an algorithm that breaks, say, KEA in a certain group, then this algorithm must use the specific representation of the elements of that group, and it is likely to fail when some other group (representation) is used. A similar result was independently developed by Dent [20] for non-bilinear groups. Second, we give some indication that a non-standard assumption is unavoidable for adaptively-sound NISZK arguments. We prove that the common approach for proving computational soundness, which has been used for all NIZK arguments (we are aware of), does not allow for statistical ZK unless NP $\subset$ P/poly (i.e. unless any NP-problem can be solved by an efficient non-uniform algorithm). Due to lack of space, this result is moved to the full version of this paper [1].

Finally, we discuss how to avoid XKEA in our NIZK arguments by allowing a pre-processing phase. Our scheme allows very efficient pre-processing where the prover only needs to make commitments for random values and prove their knowledge using efficient off-the-shelf zero-knowledge schemes.

1.3 Related Work

In order to make it easier for the reader to position our results, we would like to give a brief discussion about recently proposed NIPZK arguments. In [26] Groth *et al.* presented two schemes for proving circuit satisfiability, where the first one comes in two flavors. Let us name the resulting three schemes by the *non-adaptive*, the *adaptive* and the *UC* GOS scheme. These are the first (and so far only) NISZK arguments proposed in the literature. The non-adaptive GOS scheme is admitted by the authors to be *not* adaptively sound. The adaptive GOS scheme *is* adaptively sound, but it only allows for circuits that are limited in size, and the underlying computational assumption is somewhat non-standard in that it requires that some problem can only be solved with "sub-negligible" probability, like $2^{-\epsilon\kappa^{\epsilon}\log\kappa}negl(\kappa)$ where κ is the bit size of the problem instance. The more one relaxes the bound on the size of the circuits, the stronger the underlying assumption gets in terms of the assumed bound on the success probability of solving the problem; but in any case the size of the circuits are doomed to be sub-linear in the size of the CRS.

Concerning the UC GOS scheme, we first would like to point out that it is of theoretical interest, but it is very inefficient (though poly-time). Furthermore, it has some tricky weak soundness property in that if a dishonest prover should succeed in proving a false statement, then the statement cannot be distinguished from a true one. It is therefore claimed in [26] that the scheme "achieves a weaker, but sufficient, form of adaptive security." This is true but only if some care is taken with the kind of statements that the (dishonest) prover is allowed to prove; in particular, soundness is only guaranteed if the statement to be proven does not incorporate the CRS. Indeed, the same example that the authors use to reason that their first scheme is not adaptively sound can also be applied to the UC secure scheme: Consider a dishonest prover that comes up with an accepting proof for the statement "the CRS is invalid", or for a statement like "the CRS is invalid or S is true" where S is an arbitrary false statement. In real-life, where the CRS is guaranteed to be correct, this convinces the verifier of the truth of the false statement S. would expect such a dishonest prover to be ruled out. However, such a prover is *not* ruled out by the UC security: the simulator given in [26] does generate an *invalid* CRS so that the statement in fact becomes true; and thus the proof can obviously be simulated in the ideal-world (when given a corresponding witness, which the simulator has in case of the UC GOS scheme). We stress that this is not a flaw in the UC GOS scheme but it is the UC security definition that does not provide any security guarantees for statements that incorporate the CRS, essentially because in the ideal-life model there is *no*

(guaranteed-to-be-correct) CRS.[3] This issue is addressed in a recent extension of the UC framework [11].

In conclusion, UC NIZK security provides good enough security under the condition that the statements to be proven do not incorporate the CRS. This is automatically guaranteed in a UC setting, where the statements to be proven must make sense in the ideal-world model, but not necessarily in other settings.

2 Preliminaries

2.1 Notation

We consider uniform probabilistic algorithms (i.e. Turing machines) which take as input (the unary encoding of) a security parameter $\kappa \in \mathbb{N}$ and possibly other inputs and run in deterministic poly-time in κ. We thus always implicitly require the size of the input to be bounded by some polynomial in κ. Adversarial behavior is modeled by *non-uniform* poly-time probabilistic algorithms, i.e., by algorithms which together with the security parameter κ also get some poly-size auxiliary input aux_κ. In order to simplify notation, we usually leave the dependency on κ (and on aux_κ) implicit. By $y \leftarrow \mathcal{A}(x)$, we mean that algorithm $\mathcal{A}$ is executed on input x (and the security parameter κ and, in the non-uniform case, aux_κ) and the output is assigned to y. Similarly, for any finite set S, we use the notation $y \leftarrow S$ to denote that y is sampled uniformly from S, and $y \leftarrow x$ means that the value x is assigned to y.

For two algorithms $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A}\|\mathcal{B}$ denotes the joint execution of $\mathcal{A}$ and $\mathcal{B}$ on the same input and the same random tape, and we write $(x;y) \leftarrow (\mathcal{A}\|\mathcal{B})(w)$ to express that in the joint execution on input w (and the same random tape) $\mathcal{A}$'s output is assigned to x and $\mathcal{B}$'s to y. Furthermore, $P\big[\, y = \mathcal{A}(x) \,\big]$ denotes the probability (taken over the uniformly distributed random tape) that $\mathcal{A}$ outputs y on input x, and we write $P\big[\, x \leftarrow \mathcal{B} : \mathcal{A}(x) = y \,\big]$ for the (average) probability that $\mathcal{A}$ outputs y on input x when x is output by $\mathcal{B}$: $P\big[\, x \leftarrow \mathcal{B} : \mathcal{A}(x) = y \,\big] = \sum_x P\big[\, y = \mathcal{A}(x) \,\big] P\big[\, x = \mathcal{B} \,\big]$. We also use natural self-explanatory extensions of this notation.

An *oracle* algorithm $\mathcal{A}$ is an algorithm in the above sense connected to an oracle in that it can write on its own tape an input for the oracle and tell the oracle to execute, and then, in a single step, the oracle processes its input in a prescribed way, and writes its output to the tape. We write $\mathcal{A}^{\mathcal{O}}$ when we consider $\mathcal{A}$ to be connected to the particular oracle $\mathcal{O}$.

[3] A minor flaw regarding the UC GOS scheme though is that Groth *et al.* claim the scheme to be *non-malleable*, and their UC NIZK functionality indeed does guarantee non-malleability in that a proof cannot be transformed into a different proof for the same instance without knowing a witness. But it is easy to see that the UC GOS scheme is *not* non-malleable, because the NIZK proof π generated at step 6. in Figure 4 (by using the non-adaptive GOS scheme) is malleable: it uses the NIZK proof from Figure 1 (with h of order n though) which is malleable by raising π_1 and π_3 to some power $s \in \mathbb{Z}_n^*$ and π_2 to power $s^{-1} \pmod n$.

As is common practice, a value $\nu(\kappa) \in \mathbb{R}$, which depends on the security parameter κ, is called *negligible*, denoted by $\nu(\kappa) \leq negl(\kappa)$ or $\nu \leq negl$, if $\forall\, c > 0\ \exists \kappa_\circ \in \mathbb{N}\ \forall \kappa \geq \kappa_\circ : \nu(\kappa) < 1/\kappa^c$. Furthermore, $\nu(\kappa) \in \mathbb{R}$ is called *noticeable* if $\exists\, c > 0$, $\kappa_\circ \in \mathbb{N}\, \forall\, \kappa \geq \kappa_\circ : \nu(\kappa) \geq 1/\kappa^c$.

2.2 Definition

Let $L \subseteq \{0,1\}^*$ be an NP-language.

Definition 1. *Consider poly-time algorithms $\mathcal{G}$, $\mathcal{P}$ and $\mathcal{V}$ of the following form: $\mathcal{G}$ takes the security parameter κ (implicitly treated hereafter) and outputs a* common reference string (CRS) *Σ together with a* trapdoor *τ. $\mathcal{P}$ takes as input a CRS Σ and an instance $x \in L$ together with an* NP*-witness w and outputs a* proof *π. $\mathcal{V}$ takes as input a CRS Σ, an instance x and a proof π and outputs* 1 *or* 0*. The triple $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ is a* statistical/perfect NIZK argument *for L if the following properties hold.*

Completeness: For any $x \in L$ with corresponding NP*-witness w*

$$P\big[\,(\Sigma,\tau) \leftarrow \mathcal{G}, \pi \leftarrow \mathcal{P}(\Sigma, x, w) :\ \mathcal{V}(\Sigma, x, \pi) = 0\,\big] \leq negl\,.$$

Soundness: For any non-uniform poly-time adversary $\mathcal{P}^*$

$$P\big[\,(\Sigma,\tau) \leftarrow \mathcal{G}, (x^*, \pi^*) \leftarrow \mathcal{P}^*(\Sigma) :\ x^* \notin L \wedge \mathcal{V}(\Sigma, x^*, \pi^*) = 1\,\big] \leq negl\,.$$

Statistical/Perfect Zero-Knowledge (ZK): There exists a poly-time simulator *$\mathcal{S}$ such that for any instance $x \in L$ with* NP*-witness w, and for $(\Sigma,\tau) \leftarrow \mathcal{G}$, $\pi \leftarrow \mathcal{P}(\Sigma, x, w)$ and $\pi_{sim} \leftarrow \mathcal{S}(\Sigma, \tau, x)$, the joint distributions of (Σ, π) and (Σ, π_{sim}) are statistically/perfectly close.*

Remark 2. The notion of soundness we use here guarantees security against an *adaptive* attacker, which may choose the instance x^* depending on the CRS. We sometimes emphasize this issue by using the term *adaptively-sound*. Note that this is a strictly stronger notion than when the adversary must choose x^* independent of the CRS.

Remark 3. In the notion of ZK we use here, $\mathcal{P}$ and $\mathcal{S}$ use the *same* CRS string. In [17], this is called *same-string* ZK. In the context of *statistical* ZK, this notion is *equivalent* (and not only sufficient) to *unbounded* ZK,which captures that the same CRS can be used an unbounded number of times. This is obviously much more desirable compared to the original notion of NIZK, where every proof requires a fresh CRS. In [17], it is shown that there cannot be a *same-string* NIZK *proof* with statistical soundness for a NP-complete language unless there exist no one-way functions. This makes it even more interesting to find out whether there exists a same-string NIZK argument with statistical security on at least one side, namely the ZK side.

2.3 Bilinear Groups and the Hardness Assumptions

We use the standard setting of bilinear groups. Let $\mathcal{BGG}$ be a *bilinear-group generator* that (takes as input the security parameter κ and) outputs $(\mathbb{G}, \mathbb{H}, q, g, e)$ where $\mathbb{G}$ and $\mathbb{H}$ is a pair of groups of prime order q, g is a generator of $\mathbb{G}$, and e is a non-degenerate bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{H}$, meaning that $e(g^a, g^b) = e(g, g)^{ab}$ for any $a, b \in \mathbb{Z}_q$ and $e(g, g) \neq 1_{\mathbb{H}}$.

We assume the Discrete-Log Assumption, DLA, that for a random $h \in \mathbb{G}$ it is hard to compute $w \in \mathbb{Z}_q$ with $h = g^w$. In some cases, we also assume the Diffie-Hellman Inversion Assumption, DHIA, which states that, for a random $h = g^w \in \mathbb{G}$, it is hard to compute $g^{1/w}$. Formally, these assumptions for a bilinear-group generator $\mathcal{BGG}$ are stated as follows. In order to simplify notation, we abbreviate the output $(\mathbb{G}, \mathbb{H}, q, g, e)$ of $\mathcal{BGG}$ by pub (for "public parameters").

Assumption 4 (DLA). For every non-uniform poly-time algorithm $\mathcal{A}$

$$P\left[\, pub \leftarrow \mathcal{BGG},\ h \leftarrow \mathbb{G},\ w \leftarrow \mathcal{A}(pub, h) :\ g^w = h \,\right] \leq negl\,.$$

Assumption 5 (DHIA). For every non-uniform poly-time algorithm $\mathcal{A}$

$$P\left[\, pub \leftarrow \mathcal{BGG},\ h \leftarrow \mathbb{G},\ g^{1/w} \leftarrow \mathcal{A}(pub, h) :\ g^w = h \,\right] \leq negl\,.$$

Furthermore, we assume XKEA, a variant of the Knowledge-of-Exponent Assumption KEA, (referred to as KEA3 respectively KEA1 in [5]). KEA informally states that given $\hat{g} = g^x \in \mathbb{G}$ with unknown discrete-log x, the only way to efficiently come up with a pair $A, \hat{A} \in \mathbb{G}$ such that $\hat{A} = A^x$ for the same x is by choosing some $a \in \mathbb{Z}_q$ and computing $A = g^a$ and $\hat{A} = \hat{g}^a$. XKEA states that given $\hat{g} = g^x \in \mathbb{G}$ as well as another pair h and $\hat{h} = h^x$ with the same unknown discrete-log x, the only way to efficiently come up with a pair $A, \hat{A}$ such that $\hat{A} = A^x$ is by choosing $a, \alpha \in \mathbb{Z}_q$ and computing $A = g^a h^\alpha$ and $\hat{A} = \hat{g}^a \hat{h}^\alpha$. Formally, KEA and XKEA are phrased by assuming that for every algorithm which outputs A and $\hat{A}$ as required, there exists an *extractor* which outputs a (and α in case of XKEA) when given the same input and randomness.

Assumption 6 (KEA). For every non-uniform poly-time algorithm $\mathcal{A}$ there exists a non-uniform poly-time algorithm $\mathcal{X}_{\mathcal{A}}$, the *extractor*, such that

$$P\left[\, pub \leftarrow \mathcal{BGG},\ x \leftarrow \mathbb{Z}_q,\ (A, \hat{A}; a) \leftarrow (\mathcal{A} \| \mathcal{X}_{\mathcal{A}})(pub, g^x) :\ \hat{A} = A^x \wedge A \neq g^a \,\right] \leq negl\,.$$

Recall that $(A, \hat{A}; a) \leftarrow (\mathcal{A} \| \mathcal{X}_{\mathcal{A}})(pub, g^x)$ means that $\mathcal{A}$ and $\mathcal{X}_{\mathcal{A}}$ are executed on the same input (pub, g^x) and the *same random tape*, and $\mathcal{A}$ outputs $(A, \hat{A})$ whereas $\mathcal{X}_{\mathcal{A}}$ outputs a.

Assumption 7 (XKEA). For every non-uniform poly-time algorithm $\mathcal{A}$ there exists a non-uniform poly-time algorithm $\mathcal{X}_{\mathcal{A}}$, the *extractor*, such that

$$P\left[\begin{array}{l} pub \leftarrow \mathcal{BGG},\ x \leftarrow \mathbb{Z}_q,\ h \leftarrow \mathbb{G}, \\ (A, \hat{A}; a, \alpha) \leftarrow (\mathcal{A} \| \mathcal{X}_{\mathcal{A}})(pub, g^x, h, h^x) \end{array} :\ \hat{A} = A^x \wedge A \neq g^a h^\alpha \right] \leq negl\,.$$

It is well known that DLA holds provably with respect to generic algorithms (see e.g. [34]), which operate on the group elements only by applying the group operations (multiplication and inversion), but do not make use of the specific representation of the group elements. It is not so hard to see that this result extends to groups $\mathbb{G}$ that come with a bilinear pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{H}$, i.e., to generic algorithms that are additionally allowed to apply the pairing and the group operations in $\mathbb{H}$. We prove in Section 5 that also KEA and XKEA hold with respect to generic algorithms.

We would also like to point out that we only depend on XKEA for "proof-technical" reasons: our perfect NIZK argument still *appears* to be secure even if XKEA should turn out to be false (for the particular generator $\mathcal{BGG}$ used), but we cannot prove it anymore formally. This is in contrast to how KEA and XKEA are used in [27] respectively [5] for 3-round ZK, where there seems to be no simulator anymore as soon as KEA is false.

3 A Perfect NIZK Argument for SAT

3.1 Handling Multiplication Gates

Let $(\mathbb{G}, \mathbb{H}, q, g, e)$ be generated by $\mathcal{BGG}$, as described in Section 2.3 above. Furthermore, let $h = g^w$ for a random $w \in \mathbb{Z}_q$ which is unknown to anybody. Consider a prover who announces an arithmetic circuit over $\mathbb{Z}_q$ and who wants to prove in NIZK that there is a satisfying input for it. Following a standard design principle, where the prover commits to every input value using Pedersen's commitment scheme with "basis" g and h as well as to every intermediate value of the circuit when evaluating it on the considered input, the problem reduces to proving the consistency of the multiplication gates in NIZK (whereas the addition gates come for free due to the homomorphic property of Pedersen's commitment scheme).

Concretely, though slightly informally, given commitments $A = g^a h^\alpha$, $B = g^b h^\beta$ and $C = g^c h^\gamma$ for values a, b and $c \in \mathbb{Z}_q$, respectively, the prover needs to prove in NIZK that $c = a \cdot b$. Note that

$$e(A, B) = e(g^a h^\alpha, g^b h^\beta) = e(g, g)^{ab}\, e(g, h)^{a\beta + \alpha b}\, e(h, h)^{\alpha\beta} \qquad \text{and}$$
$$e(C, g) = e(g^c h^\gamma, g) = e(g, g)^c\, e(g, h)^\gamma$$

and hence, if indeed $c = a \cdot b$, then

$$e(A, B)/e(C, g) = e(g, h)^{a\beta + \alpha b - \gamma}\, e(h, h)^{\alpha\beta} = e(g^{a\beta + \alpha b - \gamma} h^{\alpha\beta}, h)\,. \qquad (1)$$

Say that, in order to prove that $c = a \cdot b$, the prover announces $P = g^{a\beta + \alpha b - \gamma} h^{\alpha\beta}$ and the verifier accepts if and only if P is *satisfying* in that

$$e(A, B)/e(C, g) = e(P, h)\,.$$

Then, by the above observations it is immediate that an honest verifier accepts the correct proof of an honest prover. Also, it is quite obvious that a

simulator which knows w can "enforce" $c = a \cdot b$ by "cheating" with the commitments, and thus perfectly simulate a satisfying P for the multiplication gate. Note that the simulator needs to know *some* opening of the commitments in order to simulate P; this though is good enough for our purpose. For completeness, though, we address this issue again in Section 4 and show a version which allows a full-fledged simulation. Finally, it appears to be hard to come up with a satisfying P unless one can indeed open A, B and C to a, b and c such that $c = a \cdot b$. Concretely, the following holds.

Lemma 8. *Given openings of A, B and C to a, b and c, respectively, with $c \neq a \cdot b$, and given an opening of a satisfying P, one can efficiently compute w.*

Proof. Let $P = g^{\rho} h^{\varpi}$ be the given opening of P. Then, inheriting the notation from above,

$$e(A,B)/e(C,g) = e(g^a h^{\alpha}, g^b h^{\beta})/e(g^c h^{\gamma}, g) = e(g,g)^{ab-c}\, e(g,h)^{a\beta+\alpha b-\gamma}\, e(h,h)^{\alpha\beta}\,.$$

and

$$e(A,B)/e(C,g) = e(P,h) = e(g^{\rho} h^{\varpi}, h) = e(g,h)^{\rho}\, e(h,h)^{\varpi}$$

are two different representations of the same element in $\mathbb{H}$ with respect to the "basis" $e(g,g)$, $e(g,h) = e(g,g)^w$, $e(h,h) = e(g,g)^{w^2}$. This allows to compute w by solving a quadratic equation in $\mathbb{Z}_q$. □

The need for an opening of P can be circumvented by basing security on DHIA rather than DLA as stated in the following lemma.

Lemma 9. *Given openings of A, B and C to a, b and c, respectively, with $c \neq a \cdot b$, and given a satisfying P, one can efficiently compute $g^{1/w}$.*

Proof. For a satisfying P it holds that

$$e(P,h) = e(A,B)/e(C,g) = e(g,g)^{ab-c}\, e(g,h)^{a\beta+b\alpha-\gamma}\, e(h,h)^{\alpha\beta}$$

and thus, when $c \neq a \cdot b$ as assumed, the following equalities follow one after the other.

$$\begin{aligned} e(g,g) &= e\big((P\, g^{-a\beta-b\alpha+\gamma} h^{-\alpha\beta})^{1/(ab-c)}, h\big) \\ e(g^{1/w}, g) &= e\big((P\, g^{-a\beta-b\alpha+\gamma} h^{-\alpha\beta})^{1/(ab-c)}, g\big) \\ g^{1/w} &= (P\, g^{-a\beta-b\alpha+\gamma} h^{-\alpha\beta})^{1/(ab-c)} \end{aligned}$$

Thus, $g^{1/w}$ can be computed from the available information. □

It remains to argue that a (successful) prover can indeed open all the necessary commitments. This can be enforced as follows. Instead of committing to every value s by $S = g^s h^{\sigma}$, the prover has to commit to s by $S = g^s h^{\sigma}$ and $\hat{S} = \hat{g}^s \hat{h}^{\sigma}$, where $\hat{g} = g^x$ for a random $x \in \mathbb{Z}_q$ and $\hat{h} = h^x$ (with the same x). Note that the same randomness σ is used for computing S and $\hat{S}$, such that

$\hat{S} = S^x$; this can be verified using the bilinear map: $e(\hat{S}, g) = e(S, \hat{g})$. XKEA now guarantees that for every correct double commitment $(S, \hat{S})$ produced by the prover, he knows (respectively there exists an algorithm that outputs) s and σ such that $S = g^s h^\sigma$.

Based on the above observations, we construct and prove secure an adaptively-sound perfect NIZK argument for circuit-SAT in the next section.

3.2 The Perfect NIZK Scheme

The NIZK scheme for circuit-SAT is given in Figure 1. Note that we assume an arithmetic circuit C over $\mathbb{Z}_q$ (rather than a binary circuit), but of course it is standard to "emulate" a binary circuit by an arithmetic one.

CRS Generator $\mathcal{G}(1^\kappa)$:

G-1. $(\mathbb{G}, \mathbb{H}, q, g, e) \leftarrow \mathcal{BGG}(1^\kappa)$, $w \leftarrow \mathbb{Z}_q$, $\hat{g} \leftarrow \mathbb{G}$, $h \leftarrow g^w$, $\hat{h} \leftarrow \hat{g}^w$,
G-2. output $\Sigma \leftarrow (\mathbb{G}, \mathbb{H}, q, g, h, \hat{g}, \hat{h}, e)$ and $\tau \leftarrow w$.

Prover $\mathcal{P}(\Sigma, \mathsf{C}, x = (x_1, \ldots, x_n))$:

P-1. Compute commitments for every input x_i by $X_i = g^{x_i} h^{\xi_i}$ and $\hat{X}_i = \hat{g}^{x_i} \hat{h}^{\xi_i}$.
P-2. Inductively, for every multiplication gate in C for which the two input values a and b are committed upon (either directly or indirectly via the homomorphic property) by $A = g^a h^\alpha$ and $\hat{A} = \hat{g}^a \hat{h}^\alpha$ respectively $B = g^b h^\beta$ and $\hat{B} = \hat{g}^b \hat{h}^\beta$, do the following. Compute a (double) commitment $C = g^c h^\gamma$ and $\hat{C} = \hat{g}^c \hat{h}^\gamma$ for the corresponding output value $c = a \cdot b$, and compute the (double) commitment $P = g^{a\beta + \alpha b - \gamma} h^{\alpha\beta}$ and $\hat{P} = \hat{g}^{a\beta + \alpha b - \gamma} \hat{h}^{\alpha\beta}$.
P-3. As proof π output all the commitments as well as the randomness η for the commitment $Y = g^{\mathsf{C}(x)} h^\eta$ for the output value $\mathsf{C}(x) = 1$.

Verifier $\mathcal{V}(\Sigma, \mathsf{C}, \pi)$:

Output 1 (i.e. "accept") if all of the following holds, otherwise output 0.

V-1. Every double commitment $(S; \hat{S})$ satisfies $e(\hat{S}, g) = e(S, \hat{g})$.
V-2. Every multiplication gate in C, with associated (double) commitments $(A, \hat{A})$, $(B, \hat{B})$, $(C, \hat{C})$ and $(P, \hat{P})$ for the two input values, the output value and the "multiplication proof", satisfies $e(A, B)/e(C, g) = e(P, h)$.
V-3. The commitment Y for the output value satisfies $Y = g^1 h^\eta$.

Fig. 1. Perfect NIZK argument for circuit-SAT

Theorem 10. *$(\mathcal{G}, \mathcal{P}, \mathcal{V})$ from Fig. 1 is an adaptively-sound perfect NIZK argument for circuit-SAT, assuming* XKEA *and* DLA.

Proof. Completeness is straightforward using observation (1). Also, perfect ZK is easy to see. Indeed, the simulator $\mathcal{S}$ can run $\mathcal{P}$ with a default input for x, say

$o = (0, \ldots, 0)$, and then simply open the commitment Y for the output value $y = \mathsf{C}(o)$ (which is likely to be different from 1) to 1 using the trapdoor w. Since Pedersen's commitment scheme is perfectly hiding, and since P and $\hat{P}$ computed in step P-2. for every multiplication gate are uniquely determined by A, B, and C, it is clear that this simulation is perfectly indistinguishable from a real execution of $\mathcal{P}$.

It remains to argue soundness. Assume there exists a dishonest poly-time prover $\mathcal{P}^*$, which on input the CRS Σ outputs a circuit C^* together with a proof π^* such that with non-negligible probability, C^* is not satisfiable but $\mathcal{V}(\Sigma, \mathsf{C}^*, \pi^*)$ outputs 1. By XKEA, there exists a poly-time extractor $\mathcal{X}_{\mathcal{P}^*}$ such that when run on the same CRS and the same random tape as $\mathcal{P}^*$, the extractor $\mathcal{X}_{\mathcal{P}^*}$ outputs the opening information for all (double) commitments in the proof with non-negligible probability.[4] Concretely, for every multiplication gate and the corresponding commitments A, B, C and P, the extractor $\mathcal{X}_{\mathcal{P}^*}$ outputs $a, \alpha, b, \beta, c, \gamma, \rho, \varpi$ such that $A = g^a h^\alpha$, $B = g^b h^\beta$, $C = g^c h^\gamma$ and $P = g^\rho h^\varpi$. If $\mathcal{P}^*$ succeeds in forging a proof for an unsatisfiable circuit, then there obviously must be an inconsistent multiplication gate with inputs a and b and output $c \neq a \cdot b$. (Note that since addition gates are processed using the homomorphic property, there cannot be an inconsistency in an addition gate.) But this contradicts DLA by Lemma 8. □

Remark 11. The NIZK argument from Fig. 1 actually provides *adaptive ZK*, which is a stronger flavor of ZK than guaranteed by Definition 1. It guarantees that $\mathcal{S}$ cannot only perfectly simulate a proof π for any circuit C, but when later given a satisfying input x for C, it can also provide the randomness that explains how π could have been generated by running $\mathcal{P}$ on witness x.

Remark 12. It is reasonable to assume that one can efficiently verify that, for given $(\mathbb{G}, \mathbb{H}, q, g, e)$, $\mathbb{G}$ and $\mathbb{H}$ are groups of order q, g generates $\mathbb{G}$, and e is a non-degenerate bilinear map. Then, the CRS $\Sigma = (\mathbb{G}, \mathbb{H}, q, g, h, \hat{g}, \hat{h}, e)$ may be generated by the (possibly dishonest) verifier, together with an (interactive) ZK proof of the knowledge of w with $g^w = h$, which can be done very efficiently by using the 4-round ZK proof from [13] for instance. The prover additionally needs to check if $e(\hat{g}, h) = e(g, \hat{h})$. Hence, the set-up of the CRS requires no honest party nor any expensive 2-party (or multi-party) computation. If the proof of knowledge of w is omitted, so that the verifier only publishes the CRS Σ, then the argument is still *witness indistinguishable*. Thus, our scheme can also be appreciated as a (computationally sound) *zap* [21].

Remark 13. By omitting $\hat{P}$ (and the corresponding verifications), one can obtain a slightly more efficient protocol based on the possibly stronger assumption DHIA

[4] As a matter of fact, XKEA guarantees that for every double commitment there exists an extractor that outputs the opening *for that particular commitment* with non-negligible probability; but of course running all these extractors together allows to extract for all commitments simultaneously with non-negligible probability (since the size of C must be polynomially bounded).

instead of DLA. The security can be proven in exactly the same way based on Lemma 9 instead of Lemma 8. Furthermore, if one is willing to trade XKEA by a new assumption (though of similar flavor, but which can also be proven in the generic model) that the only way to come up with A, B, C and P such that $e(A,B)/e(C,g) = e(P,h)$ is by choosing A, B and C as commitments of a, b and $c = a \cdot b$, respectively, then one can get a NIZK scheme where (not counting the unavoidable commitments A, B and C) the proof for each multiplication gate consists of only 1 group element, P. Note that this requires less communication than using standard interactive ZK techniques in combination with the Fiat-Shamir heuristic [23].

4 Efficient Proof for Relations Among Commitments

We again consider the problem of proving that a Pedersen commitment $C = g^c h^\gamma$ "contains" the product $c = a \cdot b$ of a and b committed to by $A = g^a h^\alpha$ respectively $B = g^b h^\beta$. Recall that the multiplication proof discussed in Section 3.1, consisting of P such that $e(A,B)/e(C,g) = e(P,h)$ (and maybe the corresponding $\hat{P}$), can only be simulated if the simulator knows *some* openings of A, B and C. This was good enough for the application to NIZK for SAT, as in this case all commitments may be prepared by the simulator. However, for other applications, it might be desirable to have a similarly efficient non-interactive multiplication proof which allows a fully-fledged simulation, i.e., which can be simulated for any *given* A, B and C. We show in this section a modification of the multiplication proof of Section 3.1 which has this property.

The setting is the same as in the previous section; We assume that a CRS $\Sigma = (\mathbb{G}, \mathbb{H}, q, g, h, \hat{g}, \hat{h}, e)$ has been properly set up and is publicly available. Per default, the prover is required to provide $\hat{A}$, $\hat{B}$ and $\hat{C}$ for the commitments A, B and C in question, and the verifier should check if $e(\hat{g}, A) = e(g, \hat{A})$ etc., so that the opening of A, B and C can be extracted via XKEA. Note that such $\hat{A}$, $\hat{B}$ and $\hat{C}$ can be computed from A, B and C and $x = \log_g \hat{g} \in \mathbb{Z}_q$. Thus, the ZK simulator who knows x can simulate $\hat{A}$, $\hat{B}$ and $\hat{C}$ without knowing the openings of the original commitments. For ease of description, these "hatted" commitments and corresponding verifications are treated implicitly hereafter. We begin with a simple relation for proving that a commitment A can be opened to zero.

Open to Zero ($a = 0$): For $A = g^0 h^\alpha$, the prover publishes $P = g^\alpha$. The verifier accepts if $e(g, A) = e(h, P)$.

It is obvious that an honest verifier accepts the correct proof of an honest prover. ZK is straightforward: the simulator who knows w can compute $P = A^{1/w}$ (without knowing the opening of A). Finally, given an opening (a, α) of A and a satisfying P, i.e., such that $e(g, A) = e(h, P)$, if $a \neq 0$ then one can efficiently compute $g^{1/w}$. This follows from the following equalities:

$$e(g, g^a h^\alpha) = e(h, P)$$

$$e(g, g^a) = e(g, Pg^{-\alpha})^w$$
$$e(g, g^{1/w}) = e(g, (Pg^{-\alpha})^{1/a})$$

and thus $g^{1/w} = (Pg^{-\alpha})^{1/a}$.

The above protocol for opening to zero can be easily applied to show equality ($a = b$) and addition ($a + b = c$) by replacing A in the above protocol with A/B and AB/C, respectively.

We next show a proof system for multiplicative relation $a \cdot b = c$. Recall that the goal is to have a multiplication proof which allows a simulation for any A, B and C given to the simulator.

Multiplication ($ab = c$): The prover publishes $P = (R, S, T)$ such that $R = h^r$, $S = g^r$ for random r and $T = g^{a\beta+\alpha b-\gamma-ar} h^{\alpha\beta-\alpha r}$. The verifier accepts if $e(g, R) = e(h, S)$ and $e(A, B)/e(g, C) = e(A, R)e(h, T)$.

Completeness is verified by seeing that the first verification equation follows from $e(g, R) = e(g, h^r) = e(g^r, h) = e(h, S)$, and the second from $e(A, B)/e(g, C) = e(g, g)^{ab-c}\, e(g, h)^{a\beta+\alpha b-\gamma}\, e(h, h)^{\alpha\beta}$ in combination with

$$\begin{aligned} e(A, R)\, e(h, T) &= e(g^a h^\alpha, h^r)\, e(h, g^{a\beta+\alpha b-\gamma-ar} h^{\alpha\beta-\alpha r}) \\ &= e(g^a, h^r)\, e(h^\alpha, h^r)\, e(h, g^{a\beta+\alpha b-\gamma-ar})\, e(h, h^{\alpha\beta-\alpha r}) \\ &= e(g, h)^{a\beta+\alpha b-\gamma}\, e(h, h)^{\alpha\beta}\,, \end{aligned}$$

which gives the desired equality if indeed $ab - c = 0$.

Fully-fledged ZK and soundness are captured by following Lemma 14 and 15, respectively.

Lemma 14. *Given A, B and C, one can efficiently simulate random R, S and T such that $e(g, R) = e(h, S)$ and $e(A, B)/e(g, C) = e(A, R)e(h, T)$.*

Proof. Given the trapdoor w, the simulator picks random u and sets $R = Bh^u$, $S = R^{1/w}$, and $T = A^{-u}C^{-1/w}$. As in the real proof, (S, R, T) is uniformly distributed subject to the verification equations:

$$e(h, S) = e(h, R^{1/w}) = e(g, R)$$

and

$$\begin{aligned} e(A, R)\, e(h, T) &= e(A, Bh^u)\, e(h, A^{-u}C^{-1/w}) \\ &= e(A, B)\, e(A^u, h)\, e(h, A^{-u})\, e(g, C^{-1}) \\ &= e(A, B)/e(g, C)\,. \end{aligned}$$

Thus, the simulation is perfect. □

Lemma 15. *Given openings of A, B and C to a, b and c, respectively, with $c \neq a \cdot b$, and given a satisfying $P = (R, S, T)$, one can efficiently compute $g^{1/w}$.*

Proof. We first observe that R and S constitute a proof of zero-opening. Hence we can say that $R = h^r$ for some r. Furthermore, we can extract such r by applying XKEA to R and S since they are in correct relation verified by $e(g,R) = e(h,S)$. Now, for a, α, b, β, c, γ and r, we see the following holds from the second verification equation:

$$\begin{aligned} e(A,B)/e(g,C) &= e(A,R)\, e(h,T) \\ e(g^a h^\alpha, g^b h^\beta)\, e(g, g^{-c} h^{-\gamma}) &= e(g^a h^\alpha, R)\, e(h,T) \\ e(g, g^{ab-c}) &= e(h, g^{-a\beta-\alpha b+ra+\gamma} h^{-\alpha\beta} R^\alpha T) \\ e(g, g^{1/w}) &= e(g, g^{-a\beta-\alpha b+ra+\gamma} h^{-\alpha\beta} R^\alpha T)^{1/(ab-c)} \end{aligned}$$

Therefore, if $ab \neq c$, one can compute $g^{1/w} = (g^{-a\beta-\alpha b+ra+\gamma} h^{-\alpha\beta} R^\alpha T)^{1/(ab-c)}$, which contradicts to DHIA. □

Now, we need to discuss what kind of NIZK arguments these protocols formally are. The crucial issue stems from the fact that Pedersen's commitment scheme is unconditionally hiding and thus the language of all triples (A,B,C) which allow an opening with $a \cdot b = c$ is trivial. Therefore, proving a statement among these commitments only makes sense in terms of *proof of knowledge*. By Lemma 15, the "knowledge soundness" can be proven by using the extractor of XKEA as knowledge extractor. Accordingly, the quality of the extractor of XKEA immediately affects to the quality of the knowledge extractor. Since XKEA only provides a non-black-box extractor, the best the protocol can achieve is a proof of knowledge characterized by a non-black-box knowledge extractor.

Let $\mathcal{R}$ be a binary poly-time relation, which we allow to depend on (κ and) Σ in order to capture schemes that prove something about commitments with "basis" g and h, which are part of the CRS. Let $L_{\mathcal{R}} = \{x \mid \exists w : (x,w) \in \mathcal{R}\}$ be the language characterized by $\mathcal{R}$.

Definition 16. *A NIZK* proof of knowledge *for $\mathcal{R}$ is a NIZK proof (or argument) for $L_{\mathcal{R}}$ such that additionally for every non-uniform poly-time prover $\mathcal{P}^*$ there exists a non-uniform poly-time* extractor *$\mathcal{E}_{\mathcal{P}^*}$ such that*

$$P\left[\begin{matrix}(\Sigma,\tau)\leftarrow\mathcal{G},\\ (x^*,\pi^*;w^*)\leftarrow(\mathcal{P}^*\|\mathcal{E}_{\mathcal{P}^*})(\Sigma)\end{matrix} : (x^*,w^*)\notin\mathcal{R}\wedge\mathcal{V}(\Sigma,x^*,\pi^*)=1\right] \leq negl\,.$$

Such NIZK proof of knowledge with non-black-box extractor might be weaker than the one with universal black-box extractor originally defined in [19]. This issue is analogue to black-box vs non-black-box ZK where both definitions are widely accepted. Although a stronger definition is in general favorable, a weaker definition has potential to capture nicer schemes with weaker assumptions or even schemes that are impossible otherwise, but still guarantees sufficient security.

The following now follows immediately from the above lemmas.

Theorem 17. *The above scheme gives a perfect NIZK proof of knowledge for* $\mathcal{R}_{\text{mult}} = \{((A,B,C),(a,\alpha,b,\beta,\gamma)) \in \mathbb{G}^3 \times \mathbb{Z}_q^5 \mid A = g^a h^\alpha, B = g^b h^\beta, C = g^{ab} h^\gamma\}$ *under* XKEA *and* DHIA.

Finally, we note that all the statements in this section can be strengthened to be based on DLA rather than DHIA by additionally providing $\hat{P} = (\hat{R}, \hat{S}, \hat{T})$, similarly as for the proof of SAT in Section 3.

5 The Security of (X)KEA against Generic Attacks

The notion of a generic algorithm is due to Nechaev [31] and Shoup [34], where it was shown that the discrete-log problem is hard for generic algorithms. Informally, a *generic algorithm* for trying to solve some DL-related problem in a group $\mathbb{G}$ is one that does not exploit and thus does not depend on the actual encoding of the group elements, but only relies on the group structure (and that the encoding is injective). In our context, where $\mathbb{G}$ allows a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{H}$, we also allow the algorithm to make use of the bilinear map and the group structure in $\mathbb{H}$.

Formally, a generic algorithm for a bilinear group is an oracle algorithm $\mathcal{A}$ which takes as input a prime q, encodings of elements of $\mathbb{Z}_q$ with respect to a *random* injective encoding function $\sigma : \mathbb{Z}_q \to S$, and possibly encodings of elements of $\mathbb{Z}_q$ with respect to another random encoding function $\tau : \mathbb{Z}_q \to T$ (with finite $S, T \subset \{0,1\}^*$). Furthermore, $\mathcal{A}$ is allowed to make oracle queries in order to compute on encoded group elements: upon a query $\big(\texttt{add_in_G}, sign, \sigma(y), \sigma(z)\big)$ the oracle $\mathcal{O}$ replies by $\sigma\big(y + (-1)^{sign} z\big)$ and upon $\big(\texttt{add_in_H}, sign, \tau(y), \tau(z)\big)$ by $\tau\big(y + (-1)^{sign} z\big)$, and upon $\big(\texttt{pair}, \sigma(y), \sigma(z)\big)$ the oracle replies by $\tau(y \cdot z)$. Note that the `add`-queries model the group operations in $\mathbb{G}$ and $\mathbb{H}$, and the `pair`-query models the pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{H}$.

Interestingly, in the literature a generic algorithm $\mathcal{A}$ is typically only allowed to query the oracle on encodings that it has received either as input or as a reply to one of the previous queries, but it is not allowed to take such an encoding and, say, flip the last bit and query the oracle on that encoding. Sometimes (but not always), this is argued by letting the set of encodings (here S and T) be so large that essentially any such query would be invalid anyway. But this also implies that $\mathcal{A}$ cannot sample random group elements without "knowing" their discrete-log. We do not want to make such a restriction, in particular in the context of KEA; even though such a step does not appear to be beneficial, we still feel it should be taken care off in a rigorous analysis. In order to avoid having to deal with invalid encodings, we assume that $S = T = \mathbb{Z}_q$ (actually, the natural representation of $\mathbb{Z}_q$ as strings) and that $\mathcal{A}$ queries $\mathcal{O}$ only on valid encodings, meaning strings in $\mathbb{Z}_q$. In some sense this models groups whose elements can be efficiently recognized.

Theorem 18. *The assumptions* KEA *and* XKEA *over bilinear groups hold with respect to generic algorithms (as long as* $1/q$ *is negligible).*

Note that the generic security of KEA in the standard (rather than the bilinear) group setting was concurrently and independently shown by Dent [20].

Proof. Let us first consider KEA. A generic algorithm $\mathcal{A}$ takes as input $\sigma(1)$ and $\sigma(x)$ for a random $x \in \mathbb{Z}_q$, and it should output $\sigma(a)$ and $\sigma(ax)$ for some $a \in \mathbb{Z}_q$. Let m be the (polynomial) number of oracle queries $\mathcal{A}$ makes. It is easy to see that any encoding that $\mathcal{A}$ might use (or receive) in an oracle query or that $\mathcal{A}$ might output is of the form $\sigma\big(P_k(x, u_1, \ldots, u_{2m})\big)$ respectively $\tau\big(Q_k(x, u_1, \ldots, u_{2m})\big)$ for multi-variate polynomials $P_k \in \mathbb{Z}_q[X, U_1, \ldots, U_{2m}]$ of total degree at most 1 respectively $Q_k \in \mathbb{Z}_q[X, U_1, \ldots, U_{2m}]$ of total degree at most 2, and for random (but fixed once and for all P_k and Q_k) pairwise-different $u_1, \ldots, u_{2m} \in \mathbb{Z}_q \setminus \{x\}$. Indeed, $\sigma(1)$ and $\sigma(x)$ correspond to the polynomials 1 and X, every encoding that $\mathcal{A}$ uses in a query which is fresh in that it has not been given to $\mathcal{A}$ in a reply (or as input) corresponds to a new variable U_i, and any reply given by the oracle corresponds to a P_k respectively Q_k that is inductively computed as $P_k = P_i \pm P_j$ respectively as $Q_k = Q_i \pm Q_j$ or $Q_k = P_i \cdot P_j$. In particular, it is easy to see that by observing $\mathcal{A}$'s oracle queries, one can keep track of these polynomials.

We now define the extractor $\mathcal{X}_\mathcal{A}$ as follows. $\mathcal{X}_\mathcal{A}$ runs $\mathcal{A}$ but keeps track of these polynomials P_k and Q_k; and if the two polynomials $P_{\text{out}_\circ}, P_{\text{out}_1} \in \{P_k\}_{k=1\ldots m}$ that correspond to the two encodings that $\mathcal{A}$ outputs are of the form $P_{\text{out}_\circ} = a$ and $P_{\text{out}_1} = aX$, then it outputs a and otherwise 0.

Let us analyze $\mathcal{X}_\mathcal{A}$. Obviously, if $\mathcal{X}_\mathcal{A}$ fails (in that $\mathcal{A}$ outputs $\sigma(a)$ and $\sigma(ax)$ but $\mathcal{X}_\mathcal{A}$ does not output a) then, by the restriction on the degree, $P_{\text{out}_1} \neq X \cdot P_{\text{out}_\circ}$, whereas P_{out_1} and $X \cdot P_{\text{out}_\circ}$ coincide when evaluated at $(x, u_1, \ldots, u_{2m})$. The event that $\mathcal{X}_\mathcal{A}$ fails is thus contained in the event $\mathcal{E}$ that at least two distinct polynomials in $\{P_{\text{out}_1}, X \cdot P_{\text{out}_\circ}\}$, in $\{P_k\}_{k=1\ldots m}$ or in $\{Q_k\}_{k=1\ldots m}$ evaluate to the same value when applied to $(x, u_1, \ldots, u_{2m})$. The standard argument for analyzing generic algorithms, using Schwartz' Lemma below, guarantees that the probability of $\mathcal{E}$ is upper bounded by $O(m^2/q)$; since m is polynomial in κ, this proves the claim (for KEA).[5]

The proof for XKEA uses exactly the same reasoning, the only difference is that $\mathcal{A}$ gets four inputs, encodings of $1, x, w, xw \in \mathbb{Z}_q$, which are associated with the polynomials $1, X, W, XW \in \mathbb{Z}_q[X, W, U_1, \ldots, U_{2m}]$, and $\mathcal{X}_\mathcal{A}$ outputs a, α if $P_{\text{out}_\circ}$ is of the form $P_{\text{out}_\circ} = a + \alpha W$ (which is the only $P_{\text{out}_\circ}$ which allows $P_{\text{out}_1} = X \cdot P_{\text{out}_\circ}$). As above we can argue that if $\mathcal{X}_\mathcal{A}$ fails then $P_{out_1}(x, w, u_1, \ldots, u_{2m}) = x \cdot P_{\text{out}_\circ}(x, w, u_1, \ldots, u_{2m})$ but $P_{\text{out}_1} \neq X \cdot P_{\text{out}_\circ}$. Rea-

[5] To make the argument rigorous, one has to consider a modified "game" where $\mathcal{A}$ is provided with random encodings as long as the corresponding polynomial (rather than its evaluation) is new, and

then observe that one can define a joint probability distribution for the original and the modified game which leaves the individual (marginal) distributions intact, and such that $\mathcal{E}$ occurs in the original game if and only if it occurs in the modified one (and thus has the same probability in both cases). In the modified game, however, the polynomials are chosen completely independent of $(x, u_1, \ldots, u_{2m})$ and thus we can apply Schwartz' Lemma.

soning as above, the probability of this to happen can again be upper bounded by $O(m^2/q)$. □

Lemma 19 (Schwartz [33]). *Let $q \in \mathbb{Z}$ be a prime. For any polynomial P in $\mathbb{Z}_q[X_1, \ldots, X_n]$ of total degree at most d, the probability that P vanishes on a uniformly distributed tuple $(x_1, \ldots, x_n) \in \mathbb{Z}_q^n$ is at most d/q.*

6 Eliminating XKEA by Pre-Processing

In this section, we briefly discuss the possibility of circumventing XKEA, and to solely rely on standard assumptions, by allowing *pre-processing*. Note that in all of the above results, XKEA is only needed in order to *extract* openings of commitments that are prepared by the possibly dishonest player. Therefore, a possible way to circumvent XKEA is to have all players prepare in a pre-processing phase *random* commitments $U = g^u h^\nu$ in such a way that one can extract the openings (u, ν) of these commitments in the security proof: for instance in the 2-player setting by a standard interactive ZK proof of knowledge (e.g. the 4-round ZK scheme from [13]), or in the multi-player setting with dishonest minority by a simple Pedersen VSS sharing. Then, when the actual NIZK argument needs to be executed, instead of providing for every commitment $S = g^s h^\sigma$ its hatted version $\hat{S} = \hat{g}^s \hat{h}^\sigma$, for every commitment $S = g^s h^\sigma$ the opening $(s + u, \sigma + \nu)$ of SU is provided, where U is a fresh unused random commitment from the pre-processing phase. This then obviously also allows to extract s in the security proof as required. There are some feasibility results about statistical NIZK arguments in the pre-processing model, cf. [18, 28, 15], which rely only on general assumptions but require a complicated pre-processing stage.

Beaver's pre-processing techniques [4] can be applied in a straightforward way to yield similarly efficient schemes as we do. However, this approach requires the generation of random commitments *with multiplicative relations* in the pre-processing phase, whereas with our techniques purely random commitments, which are potentially easier to prepare, suffice. For instance in the multi-player setting, this is known as the *linear* pre-processing model [16], and when the number of players is small, using the techniques of [12], one can have a once-and-for-all pre-processing stage that allows to produce an *unbounded* number of pseudo-random commitments on the fly.

Acknowledgments

We would like to thank Alexander Dent for useful and interesting remarks regarding our proof of the generic security of KEA and XKEA in the bilinear group setting (Section 5). Thanks also to the anonymous reviewers of TCC'07 for their invaluable comments.

References

1. M. Abe and S. Fehr. Perfect NIZK with adaptive soundness. Cryptology ePrint Archive, Report 2006/423, 2006. `http://eprint.iacr.org`.
2. L. M. Adleman. Two theorems on random polynomial time. In *19th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 1978.
3. W. Aiello and J. Håstad. Perfect zero-knowledge languages can be recognized in two rounds. In *28th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 1987.
4. D. Beaver. Efficient multiparty protocols using circuit randomization. In *Advances in Cryptology—CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*. Springer, 1991.
5. M. Bellare and A. Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In *Advances in Cryptology—CRYPTO '04*, volume 3152 of *Lecture Notes in Computer Science*. Springer, 2004.
6. M. Blum, A. De Santis, S. Micali, and G. Persiano. Non-interactive zero-knowledge proof systems. *SIAM Journal on Computing*, 20(6), 1991.
7. M. Blum, P. Feldman, and S. Micali. Non-interactive zero-knowledge and its applications. In *20th Annual ACM Symposium on Theory of Computing (STOC)*, 1988.
8. G. Brassard, D. Chaum, and C. Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Science*, 37(2), 1988.
9. G. Brassard and C. Crépeau. Non-transitive transfer of confidence: A perfect zero-knowledge interactive protocol for SAT and beyond. In *28th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 1987.
10. R. Canetti. Universally composable security: a new paradigm for cryptographic protocols. In *42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2001. Full version available from `http://eprint.iacr.org/2000/067`.
11. R. Canetti, Y. Dodis, R. Pass, and S. Walfish. Universally composable security with global setup. In *Theory of Cryptography Conference (TCC)*, *Lecture Notes in Computer Science*. Springer, 2007.
12. R. Cramer, I. B. Damgård, and Y. Ishai. Share conversion, pseudorandom secret-sharing and applications to secure computation. In *Theory of Cryptography Conference (TCC)*, volume 3378 of *Lecture Notes in Computer Science*. Springer, 2005.
13. R. Cramer, I. B. Damgård, and P. MacKenzie. Efficient zero-knowledge proofs of knowledge without intractability assumptions. In *Practice and Theory in Public Key Cryptography (PKC)*, volume 1751 of *Lecture Notes in Computer Science*. Springer, 2000.
14. I. B. Damgård. Towards practical public-key cryptosystems provably-secure against chosen ciphertext attacks. In *Advances in Cryptology—CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*. Springer, 1991.
15. I. B. Damgård. Non-interactive circuit based proofs and non-interactive perfect zero-knowledge with preprocessing. In *Advances in Cryptology—EUROCRYPT '92*, volume 658 of *Lecture Notes in Computer Science*. Springer, 1992.
16. I. B. Damgård and Y. Ishai. Constant-round multiparty computation using a black-box pseudorandom generator. In *Advances in Cryptology—CRYPTO '05*, volume 3621 of *Lecture Notes in Computer Science*. Springer, 2005.
17. A. De Santis, G. Di Crescenzo, R. Ostrovsky, G. Persiano, and A. Sahai. Robust non-interactive zero knowledge. In *Advances in Cryptology—CRYPTO '01*, volume 2139 of *Lecture Notes in Computer Science*. Springer, 2001.

18. A. De Santis, S. Micali, and G. Persiano. Non-interactive zero-knowledge with preprocessing. In *Advances in Cryptology—CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*. Springer, 1988.
19. A. De Santis and G. Persiano. Zero-knowledge proofs of knowledge without interaction. In *33rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 1992.
20. A. W. Dent. The hardness of the DHK problem in the generic group model. Cryptology ePrint Archive, Report 2006/156, 2006. `http://eprint.iacr.org`.
21. C. Dwork and M. Naor. Zaps and their applications. In *41st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2000.
22. U. Feige, D. Lapidot, and A. Shamir. Multiple non-interactive zero-knowledge proofs based on a single random string. In *31st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 1990.
23. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology—CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*. Springer, 1986.
24. L. Fortnow. The complexity of perfect zero-knowledge. In *19th Annual ACM Symposium on Theory of Computing (STOC)*, 1987.
25. O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(3), 1991.
26. J. Groth, R. Ostrovsky, and A. Sahai. Perfect non-interactive zero knowledge for NP. In *Advances in Cryptology—EUROCRYPT '06*, volume 4004 of *Lecture Notes in Computer Science*. Springer, 2006.
27. S. Hada and T. Tanaka. On the existence of 3-round zero-knowledge protocols. In *Advances in Cryptology—CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*. Springer, 1998. Full version available from `http://eprint.iacr.org/1999/009`.
28. J. Kilian, S. Micali, and C. Rackoff. Minimum resource zero-knowledge proofs. In *Advances in Cryptology—CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*. Springer, 1989.
29. J. Kilian and E. Petrank. An efficient noninteractive zero-knowledge proof system for NP with general assumptions. *Journal of Cryptology*, 11(1), 1998.
30. M. Naor. On cryptographic assumptions and challenges. In *Advances in Cryptology—CRYPTO '03*, volume 2729 of *Lecture Notes in Computer Science*. Springer, 2003.
31. V. I. Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55(2), 1994.
32. R. Pass and A. Shelat. Unconditional characterizations of non-interactive zero-knowledge. In *Advances in Cryptology—CRYPTO '05*, volume 3621 of *Lecture Notes in Computer Science*. Springer, 2005.
33. J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27(4), 1980.
34. V. Shoup. Lower bounds for discrete logarithms and related problems. In *Advances in Cryptology—EUROCRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*. Springer, 1997.