ADDITIVE PROOFS OF KNOWLEDGE
A New Notion for Non-Interactive Proofs

Amitabh Saxena[*]
Department of Information and Communication Technology
University of Trento, TN, Italy

Keywords: Non-interactive zero-knowledge proofs of knowledge, proofs of decision power, additive proofs, identification.

Abstract: This paper has two contributions. Firstly, we describe an efficient Non-Interactive Zero-Knowledge (NIZK) Proof of Knowledge (PoK) protocol using bilinear pairings. The protocol assumes the hardness of the Computational Diffie-Hellman (CDH) problem. The prover does not perform any pairing computations while the verifier performs 3 pairing computations. The protocol can be used for identification (eg. in smart-cards).
Secondly, we extend the idea to multiple proofs and propose the notion of efficient Additive Non-Interactive Witness-Indistinguishable (A-NIWI) proofs. Intuitively an A-NIWI proof can be considered as a PoK of another A-NIWI proof. Our ideas are based on the aggregate signature scheme of Boneh et al. (proposed in Eurocrypt 2003).

1 INTRODUCTION

We study the aggregate signatures of (Boneh et al., 2003) in more detail. Many schemes derived from aggregate signatures such as Verifiably Encrypted signatures (VES) (Boneh et al., 2003) and Chain Signatures (CS) (Saxena and Soh, 2005) require that although some given aggregate signature can be verified, no useful information about the individual signatures is leaked. However, the very fact that the aggregate signature can be verified leaks certain information - that the individual signatures are indeed well-formed. Apart from this, is there any other information leaked? We show that there is absolutely no other information leaked about the individual signatures when the aggregation contains only two signatures.

Another observation is that the aggregate signatures are extensible. This leads to an interesting construction of Non-Interactive Witness Indistinguishable (NIWI) proofs - a given NIWI proof π_1 of statement m_1 can be combined with another NIWI proof π_2 of m_2 to yield a new NIWI proof $\pi_{(1,2)}$ of $m_1 \wedge m_2$ such that given $\pi_{(1,2)}, m_1, m_2$, it is no longer possible to obtain π_1 (or π_2). This process can be continued using another proof π_3 of m_3. We term this property *additiveness* and formally give a construction of a NIWI proof that satisfies this property. We call any NIWI proof system satisfying this property an Additive NIWI (A-NIWI) proof system. Although this property can be achieved using conventional constructions of NIZK proofs (using an NP reduction), such constructions are extremely inefficient and therefore useless in practice.

The rest of this paper is organized as follows. In Section 2, we give some background on zero-knowledge. In Section 3, we give an informal description of our idea by showing that aggregate signatures (of two users) are zero-knowledge. We then give a formal construction of our zero-knowledge proofs of knowledge in Section 4. Finally, in Section 5, we present our example of additive NIWI proofs.

2 PRELIMINARIES

Zero-Knowledge (ZK). Zero Knowledge proofs are proofs which convince a verifier that a given statement (eg. $x \in L$ for some $L \in$ NP) is indeed true without giving any information as to *why* it is true (Goldwasser et al., 1989). This concept can be intuitively captured by saying that whatever the verifier knows after seeing the proof was already known to the verifier before seeing the

[*] The author is funded by the European Commission (contract No. 021186-2 for the RE-TRUST project).

Saxena A. (2007).
ADDITIVE PROOFS OF KNOWLEDGE - A New Notion for Non-Interactive Proofs.
In *Proceedings of the Second International Conference on Security and Cryptography*, pages 239-244
DOI: 10.5220/0002117102390244

proof. More formally, we require that there exist a PPT simulator outputting a transcript that is indistinguishable to the transcript produced by the real prover.

Witness Indistinguishability (WI). Another intuitive way to restrict knowledge leakage is using *witness indistinguishable* proofs (Feige and Shamir, 1990; Dwork and Naor, 2000). However, unlike ZK proofs, a WI proof cannot be simulated. Informally, a WI proof can be defined as follows. Let $x \in L$ for some $L \in$ NP such that x has two or more witness for L. A proof is WI if it convinces a verifier that indeed $x \in L$ but does not reveal which witness was used to construct the proof (even if the verifier knows all witnesses).

Proofs of Knowledge (PoKs). Till now we restricted ourselves to proofs of statements of the type $x \in L$ for some $L \in$ NP. These are called *proofs of membership* (PoMs). However, a more useful notion is of proofs of statements of the type *I know the witness of* $x \in L$. That is, the prover not only proves that $x \in L$ but also proves *knowledge* of a witness to the fact. Such proofs are called *proofs of knowledge* (PoKs) and are formally defined in (Bellare and Goldreich, 1993). Informally, a PoK requires that there be a *knowledge extractor* that uses the prover in a black-box manner and extracts the witness for the statement to be proved (Bellare and Goldreich, 1993).

Proofs of Decision Power (PoDPs). Let $L \in$ NP $\cap$ co-NP. A ZK (or WI) *proof of decision power* (PoDP) is a PoK for some $x \in L \cup$ co-L that convinces a verifier about the knowledge of a witness for x but does not reveal whether $x \in L$ or $x \in$ co-L. See (Crescenzo et al., 1997; Crescenzo et al., 2000) for a discussion on this concept. All our proofs presented in this paper (whether WI or ZK) will be PoDPs.

Non-Interactive (NI) ZK and WI Proofs. ZK (and WI) proofs come in two flavors: *interactive* and *non-interactive* (NI). In the interactive variants, there are many exchanges of messages (called rounds) before the proof is completed. On the other hand, in the non-interactive variants, the verifier's role is played by a hash function or some other random source of information (such as a random oracle) (Blum et al., 1988; Rackoff and Simon, 1992; Goldreich, 2001; Groth et al., 2006). Depending on whether the proof is ZK of WI, we call it a NIZK or NIWI proof. Similar to interactive proofs, NI proofs can also be classified as PoMs or PoKs.

3 ZERO KNOWLEDGE IN AGGREGATE SIGNATURES

We give the motivation behind our The aggregate signatures of (Boneh et al., 2003) can be briefly described (with some simplifications) as follows. The construction requires a bilinear map between prime order groups, which we describe first.

3.1 Bilinear Maps

Let G_1 and G_2 be two cyclic multiplicative groups both of prime order q such that computing discrete logarithms in G_1 and G_2 is intractable. A bilinear pairing is a map $\hat{e} : G_1 \times G_1 \mapsto G_2$ that satisfies the following properties (Boneh et al., 2004; Boneh et al., 2003).

1. *Bilinearity*: $\hat{e}(a^x, b^y) = \hat{e}(a,b)^{xy}$ $\forall a,b \in G_1$ and $x,y \in \mathbb{Z}_q$.
2. *Non-degeneracy*: If g is a generator of G_1 then $\hat{e}(g,g)$ is a generator of G_2.
3. *Computability*: The map $\hat{e}$ is efficiently computable.

For the rest of this paper we will assume that $g \in G_1$ is some fixed generator.

Security of aggregate signatures is based on the hardness of the following problem.

Definition 3.1. ***Computational Diffie-Hellman (CDH) problem:*** *Given $(X,Y) \in {G_1}^2$, compute $Z \in G_1$ satisfying $\hat{e}(X,Y) = \hat{e}(Z,g)$.*

3.2 Aggregate Signatures

In the aggregate signature scheme of (Boneh et al., 2003), the private keys of two users are $x_1, x_2 \in \mathbb{Z}_q$, while the public keys are $X_1 = g^{x_1}, X_2 = g^{x_2}$ respectively. The scheme also requires a cryptographic hash function $\mathcal{H} : \{0,1\} \mapsto G_1$. Let the hashes of the messages to be signed be $Y_1 = g^{y_1}$ and $Y_2 = g^{y_2}$ respectively (for unknown y_1, y_2). Then the the aggregate signature under public keys X_1, X_2 corresponds to the value $Z_2 = g^{x_1 y_1 + x_2 y_2}$ (to verify the signature we check if the equality $\hat{e}(X_1,Y_1) \cdot \hat{e}(X_2,Y_2) \stackrel{?}{=} \hat{e}(Z_2,g)$ holds). Additionally, the corresponding individual signature under the public key $X_1 = g^{x_1}$ turns out to be $g^{x_1 y_1}$, the extraction of which will correspond to the solution of the CDH instance $(X_1,Y_1) = (g^{x_1}, g^{y_1})$. Call this the signature extraction problem for the tuple (X_1,Y_1,X_2,Y_2,Z_2). Without the extra inputs X_2,Y_2,Z_2, this reduces to the ordinary CDH problem for (X_1,Y_1). We show next that these extra inputs leak no information about the solution of the CDH instance (X_1,Y_1).

Observe that given just the CDH instance $(X_1,Y_1)=(g^{x_1},g^{y_1})$, we can straightaway transform it into an instance of the signature extraction problem without knowing either x_1 or y_1 as follows. Generate two random integers r,u. Then compute $X_2 = X_1\cdot g^r = g^{r+x_1}$, $Y_2 = g^u/Y_1 = g^{u-y_1}$, $Z_2 = {X_1}^u\cdot {Y_2}^r = g^{x_1u+ru-ry_1}$. The tuple (X_1,Y_1,X_2,Y_2,Z_2) forms a valid instance of the signature extraction problem.[2] In other words, the aggregate signature leaks absolutely no knowledge about the individual signature![3] This motivates the following application.

3.3 An Identification Protocol

Assume that Alice has public key X_1 in the example of Section 3.2 and is authorized to issue credentials that can be used for identification. Bob would like to identify using his credential and at the same time ensure that the verifier cannot impersonate him later.

1. Alice uses the signature scheme of Section 3.2 to sign the message m_1 =*"The holder of this card is Bob"* such that the resulting hash of m_1 is Y_1. She gives the resulting signature Z_1 to Bob.
2. Suppose Bob wants to identify to Carol using Alice's card. Both Bob and Carol agree on a common random string (CRS) $Y_2 \in G_1$.[4] Then Bob generates random $x_2 \overset{R}{\leftarrow} \mathbb{Z}_q^*$ and computes $(X_2,Z_2) = (g^{x_2}, Z_1\cdot Y_2^{x_2}) \in {G_1}^2$. He gives (Z_2,X_2) to Carol.

We show in Section 4.1 that Z_2 proves (to Carol) the knowledge of the credential Z_1 without leaking any information. We do this by proving that the non-interactive variant of the above protocol (where the CRS is decided beforehand) is a NIZK-PoK of Z_1.

4 NIZK PROOFS OF KNOWLEDGE

We now give a formal discussion of the above zero-knowledge property. We use the common-random-string model (Blum et al., 1988) - both prover (P) and verifier (V) share a common random string (CRS). Our notion of NIZK-PoKs is similar to that of (Santis and Persiano, 1992). Let $L \in \text{NP} \cap \text{co-NP}$ be some language. For any $x \in L \cup \text{co-}L$, let $\mathcal{W}_x$ be the set of witnesses for either $x \in L$ or $x \notin L$. For simplicity, we will assume that all strings in $\{0,1\}^*$ correspond to either "yes" or "no" instances of L. Let k be a security parameter. Define the following protocol.

Protocol (P, V)

1. **Common Random String:** P and V agree on a common random string (crs) $r \overset{R}{\leftarrow} \{0,1\}^k$.
2. **Common Input:** Some string $x \overset{R}{\leftarrow} \{0,1\}^k$ (possibly chosen by P) is common input to P and V.
3. **Prover's Auxiliary Input:** P is given as auxiliary input $w \overset{R}{\leftarrow} \mathcal{W}_x$.
4. **Proof Generation:** P uses (r,w,x) to compute and output a proof π.
5. **Proof Verification:** V uses a deterministic procedure on input (x,r,π) and outputs either 0 or 1.

Definition 4.1. *(P,V) is a NIZK-PoK (and a PoDP) for $L \in NP \cap co\text{-}NP$ if the following hold.*

1. *Completeness*: For all $x \in \Sigma^*$ and honest provers P
$$\Pr\left[\, V(x,r,\pi)=1 \;\middle|\; \begin{array}{l} r,x \overset{R}{\leftarrow} \{0,1\}^k, w \overset{R}{\leftarrow} \mathcal{W}_x, \\ \pi \leftarrow P(x,w,r) \end{array} \right] = 1$$
2. *Zero-Knowledge*: There is a universal PPT simulator M that on input some random string x (the problem instance) outputs a tuple (r_m,π_m) such that $V(x,r_m,\pi_m)=1$ and the distributions $\{x,r,\pi\}$ and $\{x,r_m,\pi_m\}$ below are indistinguishable.
$$\{x,r,\pi\} \overset{\text{def}}{=} \left[\, V(x,r,\pi)=1 \;\middle|\; \begin{array}{l} r,x \overset{R}{\leftarrow} \{0,1\}^k, \\ w \overset{R}{\leftarrow} \mathcal{W}_x, \\ \pi \leftarrow P(x,w,r) \end{array} \right]$$
$$\{x,r_m,\pi_m\} \overset{\text{def}}{=} \left[\, V(x,r_m,\pi_m)=1 \;\middle|\; \begin{array}{l} x \overset{R}{\leftarrow} \{0,1\}^k, \\ (r_m,\pi_m) \leftarrow M \end{array} \right]$$
3. *Proof-of-Knowledge*: There is a universal PPT extractor E that functions as follows. E gives a "random looking" string r_e to the prover P^*, who outputs a pair (x,π). If $V(x,r_e,\pi)=1$ then E takes in as input (x,r_e,π) and outputs a string w_e. We require that for all P^*, the strings r_e are indistinguishable from truly random strings, and
$$\Pr\left[\, w_e \in \mathcal{W}_x \;\middle|\; \begin{array}{l} r_e \leftarrow E(x), (x,\pi) \leftarrow P^*(r_e), \\ V(x,r_e,\pi)=1, \\ w_e \leftarrow E(x,r_e,\pi) \end{array} \right] \approx 1$$

Note that our NIZK-PoKs are *adaptive* - the prover can choose the statement x after seeing the CRS r.

[2] This was proved in (Coron and Naccache, 2003).

[3] There is a subtility here. The resulting value Y_2 needs to be the output of the hash function $\mathcal{H}$. However, if we consider $\mathcal{H}$ to be a random oracle then we can ignore this subtility in our context.

[4] The CRS could be decided before Z_1 is computed. However, it is necessary for both Bob and Carol to ensure that the CRS is indeed random. For the purpose of this paper, we will assume that there is a trusted authority that is responsible for generating the CRS. Also note that a CRS can be used only once. Hence both parties must ensure that the CRS is *fresh*.

4.1 NIZK-PoK for a CDH Solution

Let $\hat{e} : G_1 \times G_1 \mapsto G_2$ be a bilinear map as defined in Section 3.1 such that $|G_1| = |G_2| = q$ (prime). Let g be some fixed generator of G_1, which we will use as the base for our problem instances. Assume that the computational Diffie-Hellman (CDH) problem is hard in G_1. Therefore, due to the Goldreich-Levin Theorem (Goldreich and Levin, 1989), there must exist a hard-core predicate (say $\delta()$) for the solution of the CDH instance.[5] Consider the language consisting of pairs of the form $(g^x, g^y) \in G^2$:

$$L = \{(g^x, g^y)|\text{hard-core predicate } \delta(g^{xy}) = 1\}$$

Clearly, $L \in$ NP $\cap$ co-NP and the element g^{xy}, the solution to the CDH instance (g^x, g^y) forms the witness to both the "yes" and "no" instances. We describe a NIZK-PoK for the knowledge of this witness. First we define the following problem.

Definition 4.2. ***Decision Class-Diffie-Hellman (DCDH) problem****. Given $X, Y \in G_1$, output 1 if $(X, Y) \in L$, otherwise output 0.*

Define the following protocol between P and V.

Protocol (P, V).

1. **Common random string (CRS):** An element $Y_2 \overset{R}{\leftarrow} G_1$. Let $Y_2 = g^{y_2}$ for unknown y_2.
2. **Common input:** A DCDH instance $(X_1, Y_1) = (g^{x_1}, g^{y_1}) \in {G_1}^2$.
3. **Provers auxiliary input:** Witness $W = g^{x_1 y_1} \in G_1$ for the DCDH instance (X_1, Y_1).
4. **Proof generation:** P generates $x_2 \overset{R}{\leftarrow} \mathbb{Z}_q^*$ and computes $(X_2, Z_2) = (g^{x_2}, W \cdot {Y_2}^{x_2}) \in {G_1}^2$. It outputs (X_2, Z_2) as its proof.
5. **Proof verification:** V accepts the above proof if the following holds:

$$\hat{e}(X_1, Y_1) \cdot \hat{e}(X_2, Y_2) \overset{?}{=} \hat{e}(Z_2, g) \tag{1}$$

Theorem 4.3. *The above non-interactive protocol (P, V) is a NIZK proof of knowledge of the witness to the DCDH decision problem instance (X_1, Y_1).*

Proof. Completeness is trivial:
$LHS = \hat{e}(X_1, Y_1) \cdot \hat{e}(X_2, Y_2) = \hat{e}(g^{x_1}, g^{y_1}) \cdot \hat{e}(g^{x_2}, g^{y_2})$
$= \hat{e}(g^{x_1 y_1 + x_2 y_2}, g) = RHS$

Zero Knowledge: The input is some DCDH instance (X_1, Y_1). Simulator M generates two random elements $r, u \overset{R}{\leftarrow} \mathbb{Z}_q^*$. It then computes $X_2 = X_1 \cdot g^r$, $Y_2 = g^u / Y_1$ and $Z_2 = {X_1}^u \cdot {Y_2}^r$. It outputs X_2, Y_2, Z_2 as part of the simulated transcript. The tuple $(X_1, Y_1, X_2, Y_2, Z_2)$ is indistinguishable from a real transcript.

Proof of Knowledge: We construct an extractor E as follows. E generates a random element $y_2 \overset{R}{\leftarrow} \mathbb{Z}_q^*$ and sets $Y_2 = g^{y_2}$. It gives Y_2 as the random string to the prover P, who outputs (X_1, Y_1, X_2, Z_2) such that $(X_1, Y_1, X_2, Y_2, Z_2, g)$ satisfies Equation 1. Then E computes and outputs $W = Z_2 / (X_2)^{y_2}$, the witness to the DCDH instance (X_1, Y_1). □

5 ADDITIVE NON-INTERACTIVE PROOFS OF KNOWLEDGE

Observe that in the protocol of Section 4.1, given the transcript $(X_1, Y_1, X_2, Y_2, Z_2, g)$, we can generate a new DCDH instance $(X_3, Y_3) = (g^{x_3}, g^{y_3})$ and form the tuple $(X_1, Y_1, X_2, Y_2, X_3, Y_3, Z_3, g)$, such that $Z_3 = Z_2 \cdot g^{x_3 y_3}$ behaves like a PoK of Z_2. We call this property "additiveness" - whenever a non-interactive PoK Z_i can be converted into a new non-interactive PoK Z_{i+1} of Z_i. First we define the following problem.

5.1 The Composite-CDH Problem

Let $S_i = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_i, Y_i)\}$ be a set containing i DCDH instances. Define $Z_i \in G_1$ to be the value such that

$$\prod_{(X_j, Y_j) \in S_i} \hat{e}(X_j, Y_j) = \hat{e}(Z_i, g)$$

Definition 5.1. ***Composite Computational Diffie Hellman (CCDH) problem****. Given S_i, compute Z_i.*

We say that Z_i is the CCDH solution of the set S_i. The CCDH problem is as hard as the CDH problem.

Lemma 5.1. *The CCDH problem is hard if and only if the CDH problem is hard.*

Proof. The "only if" part is trivial to prove. For the "if" part, consider an adversary $\mathcal{A}$ who can always output the CCDH solution of any set S_i. We can use $\mathcal{A}$ to solve any CDH instance (X, Y) as follows. Generate random $x', y' \overset{R}{\leftarrow} \mathbb{Z}_q^*$ and compute $X' = g^{x'}$; $Y' = g^{y'}$. The set $S_i = \{(X, Y), (X', Y')\}$ is given to $\mathcal{A}$, who outputs the CCDH solution Z_i of S_i. In this case $Z_i / g^{x'y'}$ is the solution of our CDH instance. □

[5] To apply the Goldreich-Levin result, we must exhibit a one-way function that takes as input any CDH solution (say $H \in G_1$) and outputs a corresponding CDH instance $(H_1, H_2) \in {G_1}^2$. To do this, generate $\alpha \overset{R}{\leftarrow} \mathbb{Z}_q^*$ and compute $(H_1, H_2) = (H^{1/\alpha}, g^{\alpha}) \in {G_1}^2$. Then H is the solution (to base g) of the CDH instance (H_1, H_2).

5.2 Additive NIWI Proofs

We now present a construction of an *Additive Non-Interactive Witness-Indistinguishable Proof of Knowledge* (A-NIWI-PoK). An A-NIWI-PoK can be instantly transferred into another another A-NIWI-PoK such that the new proof behaves like a PoK of the older PoK. Define the following protocol between P and V.

Protocol (P,V)

1. **Common Random String:** A random element $Y_{n+1} \overset{R}{\leftarrow} G_1$.
2. **Common Input:** The common input is a set $S_n = \{(X_1,Y_1),(X_2,Y_2),\ldots,(X_n,Y_n)\}$ containing n DCDH instances w.r.t. a common generator g.
3. **Prover's Auxiliary Input:** Z_n, the CCDH solution of S_n. P will prove knowledge of Z_n.
4. **Proof Generation:** P generates $x_{n+1} \overset{R}{\leftarrow} \mathbb{Z}_q^*$ and sets $(X_{n+1},Z_{n+1}) \leftarrow (g^{x_{n+1}}, Z_n \cdot Y_{n+1}^{x_{n+1}}) \in {G_1}^2$. It outputs (X_{n+1},Z_{n+1}). Observe that Z_{n+1} is the CCDH solution of $S_{n+1} = S_n \cup \{(X_{n+1},Y_{n+1})\}$.
5. **Proof Verification:** V verifies that Z_{n+1} is indeed the CCDH solution of S_{n+1}.

Theorem 5.2. *The pair (Z_{n+1},S_{n+1}) is a NIWI-PoK of the CCDH solution Z_n of S_n for all $n \geq 1$.*

Proof. Similar to ZK proofs, a WI proof has completeness, witness-indistinguishability and knowledge extractor requirements (Feige and Shamir, 1990; Dwork and Naor, 2000). Completeness is trivial.

Witness-Indistinguishability: The claim is true for $n = 1$ (because ZK implies WI). For any $n > 1$, given the set S_n and random string Y_{n+1}, we can construct a pair (X_{n+1},Z_{n+1}) such that Z_{n+1} is the CCDH solution of $S_{n+1} = S_n \cup \{(X_{n+1},Y_{n+1})\}$. This can be done in at least two different ways: (1) Using the CCDH solution Z_n of S_n and the witness for the DCDH instance (X_{n+1},Y_{n+1}). (2) Using the CCDH solution of $S_{n+1} \backslash \{(X_1,Y_1)\}$ and the witness for the DCDH instance (X_1,Y_1). Clearly, it is infeasible to distinguish which strategy was used.

Proof of Knowledge: We must exhibit an extractor E_{n+1} that works as follows. First E_{n+1} outputs a random string Y_{n+1}, which is given to the prover. The prover then outputs a tuple (S_n,X_{n+1},Z_{n+1}) such that S_n is a set containing n DCDH instances and Z_{n+1} is the CCDH solution of $S_n \cup \{(X_{n+1},Y_{n+1})\}$. Finally, E_{n+1} takes as input (S_n,X_{n+1},Z_{n+1}) and outputs Z_n, the CCDH solution of S_n.

E_{n+1} generates $y_{n+1} \overset{R}{\leftarrow} \mathbb{Z}_q^*$ and computes $Y_{n+1} = g^{y_{n+1}} \in G_1$. E_{n+1} gives Y_{n+1} to some prover P who outputs a tuple (S_n,X_{n+1},Z_{n+1}) such that S_n contains n DCDH instances and Z_{n+1} is the CCDH solution of $S_{n+1} = S_n \cup \{(X_{n+1},Y_{n+1})\}$. From this E_{n+1} computes $Z_n = Z_{n+1} \cdot X_{n+1}^{-y_{n+1}}$ and outputs Z_n as the CCDH solution of S_n. □

5.2.1 Additiveness

Observe that any given Niwi-PoK (Z_n,S_n) can be instantly transferred into a new Niwi-PoK (Z_{n+1},S_{n+1}) of (Z_n,S_n) (in other words, (Z_{n+1},S_{n+1}) proves knowledge of (Z_n,S_n)). We call this property *additiveness* and any Niwi-PoK exhibiting this property an Additive Niwi-PoK (A-Niwi-PoK).

5.2.2 Is it Zero-knowledge?

The witness indistinguishability property of above NIWI-PoK ensures that Z_{n+1} does not leak any "useful" information about the secret Z_n. However, we have been unable to construct a simulator and it is quite likely that the above protocol is not zero-knowledge.

To see why it may not be zero-knowledge (and still be witness hiding), observe that given the pair (Z_3,S_3) with $|S_3| = 3$, an adversary may be able to obtain some information about all the CCDH solutions Z_2^* for the 3 sets $S_2^* \subsetneq S_3$ with $|S_2^*| = 2$ without getting any information about the witnesses of the individual DCDH instances of S_3.

6 SUMMARY

In this paper we presented an efficient Non-Interactive Zero-Knowledge (NIZK) protocol that is a Proof of Knowledge (PoK) for the solution of some given Diffie-Hellman problem instance in bilinear groups. Our protocol is based on the aggregate signatures of (Boneh et al., 2003) and its interactive variant (where the CRS is generated "on-the-fly") can be used for efficient identification (eg. in smart-cards).

We also proposed the notion of *Additive Non-Interactive Witness Indistinguishable Proofs of Knowledge* (A-NIWI-PoKs). An A-NIWI proof can be considered as a PoK of another A-NIWI proof. However, we have unable to construct a simulator to achieve zero-knowledge. We can use the simulator of the proof of Theorem 4.3 and achieve additive NIZK property at the cost of increasing the size of the proof to 2^n at n levels. As an open question, we would like to ask if constant-size additive NIZK PoKs exist.

In summary, we feel that the proposed paradigm of A-NIWI-PoKs can be used in a vast majority of e-commerce applications, more specifically in the core of protocols for smart cards and secure web purchases but even more so in the context of auctions (due to the inherent non-interactive nature of the scheme).

ACKNOWLEDGEMENTS

We would like to thank Ioannis Atsonios and the anonymous referees for valuable comments.

REFERENCES

Bellare, M. and Goldreich, O. (1993). On defining proofs of knowledge. *Lecture Notes in Computer Science*, 740:390–420.

Blum, M., Feldman, P., and Micali, S. (1988). Non-interactive zero-knowledge and its applications. In *STOC ’88: Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 103–112. ACM Press.

Boneh, D., Gentry, C., Lynn, B., and Shacham, H. (2003). Aggregate and verifiably encrypted signatures from bilinear maps. In Biham, E., editor, *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 416–432. Springer.

Boneh, D., Lynn, B., and Shacham, H. (2004). Short signatures from the weil pairing. *J. Cryptology*, 17(4):297–319.

Coron, J.-S. and Naccache, D. (2003). Boneh et al.’s k-element aggregate extraction assumption is equivalent to the Diffie-Hellman assumption. In Laih, C.-S., editor, *ASIACRYPT*, volume 2894 of *Lecture Notes in Computer Science*, pages 392–397. Springer.

Crescenzo, G. D., Sakurai, K., and Yung, M. (1997). Zero-knowledge proofs of decision power: new protocols and optimal round-complexity. In *ICICS ’97: Proceedings of the First International Conference on Information and Communication Security*, pages 17–27, London, UK. Springer-Verlag.

Crescenzo, G. D., Sakurai, K., and Yung, M. (2000). On zero-knowledge proofs (extended abstract): “from membership to decision”. In *STOC ’00: Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 255–264, New York, NY, USA. ACM Press.

Dwork, C. and Naor, M. (2000). Zaps and their applications. In *FOCS ’00: Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, pages 283–293, Washington, DC, USA. IEEE Computer Society.

Feige, U. and Shamir, A. (1990). Witness indistinguishable and witness hiding protocols. In *STOC ’90: Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 416–426, New York, NY, USA. ACM Press.

Goldreich, O. (2001). *Foundations of Cryptography I*, volume Basic Tools. Cambridge University Press.

Goldreich, O. and Levin, L. A. (1989). A hard-core predicate for all one-way functions. In *STOC ’89: Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 25–32, New York, NY, USA. ACM Press.

Goldwasser, S., Micali, S., and Rackoff, C. (1989). The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208.

Groth, J., Ostrovsky, R., and Sahai, A. (2006). Perfect non-interactive zero knowledge for np. In Vaudenay, S., editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 339–358. Springer.

Rackoff, C. and Simon, D. R. (1992). Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *CRYPTO ’91: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, pages 433–444, London, UK. Springer-Verlag.

Santis, A. D. and Persiano, G. (1992). Zero-knowledge proofs of knowledge without interaction. In *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science*, pages 427–436.

Saxena, A. and Soh, B. (2005). One-way signature chaining: A new paradigm for group cryptosystems. Cryptology ePrint Archive, Report 2005/335.