A GENERAL ZERO-KNOWLEDGE SCHEME *

Mike V. D. Burmester
Dept. of Mathematics
RHBNC - University of London
Egham, Surrey TW20 OEX
U.K.

Yvo Desmedt†
Dept. EE & CS
Univ. of Wisconsin – Milwaukee
P.O. Box 784
WI 53201 Milwaukee
U.S.A.

Fred Piper
Dept. of Mathematics
RHBNC - University of London
Egham, Surrey TW20 OEX
U.K.

Michael Walker
Racl Research Ltd.
Worton Grange Industrial Estate
Reading, Berks RG2 0SB
U.K.

Extended Abstract

Abstract

There is a great similarity between the Fiat-Shamir zero-knowledge scheme [8],
the Chaum-Evertse-van de Graaf [4], the Beth [1] and the Guillou-Quisquater [12]
look alike. This suggests that a generalization is overdue. We present a general
zero-knowledge proof which encompasses all these schemes.

I. Introduction

An interactive proof-system, or simply a proof, is an interactive protocol by which,
on input I, a prover A(lice) attempts to convince a verifier B(ob) that either (a)
I ∈ L, a language (proof of membership), or (b) that she "knows" a witness S for
which (I, S) satisfies a polynomial-time predicate P(·, ·) (proof of knowledge). A
proof is zero-knowledge if it reveals no more than is strictly necessary (for a formal
definition of a proof of membership see [11]; for proofs of knowledge see [7]). Many
zero-knowledge proofs have been described in the literature and various definitions
of a proof-system have been suggested. The property of zero-knowledge has also
been analyzed and refined (e.g., [7]). One might wonder why so many different
zero-knowledge proofs have been proposed. One reason is that schemes which are

*Some of the results in this paper have been briefly announced at the rump session of Crypto'88.
†Research partly done when visiting RHBNC and sponsored by SERC.
based on zero-knowledge protocols must be easy to implement. Another is the complexity of protocols: practical considerations make it necessary to increase the speed of a protocol [8], to reduce its storage requirements [1,12] and to reduce the number of its iterations [2]. Finally the theoretical approach to zero-knowledge is closely related to the theory of computational complexity [11].

The purpose of this paper is to provide a general setting for these zero-knowledge protocols and to show that many known protocols fit into this setting. The advantages of having such a generalization are that:

- it illustrates the essential features of the protocol,
- it provides a proof that a general class of protocols are zero-knowledge, thereby establishing a straightforward set of criteria to determine whether or not a given protocol is zero-knowledge.

In this paper we consider an algebraic framework which includes the systems of Fiat-Shamir [8], Feige-Fiat-Shamir [7], Chaum-Evertse-van de Graaf [4], Beth [1], Desmedt [6] and Guillou-Quisquater [12]. We shall not discuss non-interactive zero-knowledge protocols [2].

The Fiat-Shamir scheme

To start with we briefly describe the set up of the Fiat-Shamir scheme [8]. This will help the reader to appreciate the setting for our scheme and to understand the details. In the Fiat-Shamir scheme we have:

- a set of secret numbers $S_1, S_2, \ldots, S_m$ which are chosen from the group of units $Z_n^*$ of the ring of integers modulo $n$.
- a set of public numbers $I_1, I_2, \ldots, I_m \in QR_n$, the set of quadratic residues.
- a predicate $P(I, S) \equiv (I = S^2 (mod n))$, satisfied by all the pairs $(I_j, S_j)$.

The protocol repeats $t = O(|n|)$ times:

Step 1 $A$, the prover, selects a random integer $X$ modulo $n$ and sends $B$, the verifier, the number $Z = X^2 (mod n)$.

Step 2 $B$ sends $A$ the random bits $q_1, q_2, \ldots, q_m$ as a query.

Step 3 $A$ sends $B$: $Y = X \cdot \prod\limits_{j} S_j^{q_j} (mod n)$, when all $q_i \in \{0, 1\}$.

Step 4 $B$ verifies that $Y \in Z_n^*$ and that $Y^2 = Z \cdot \prod\limits_{j} I_j^{q_j} (mod n)$.
B accepts A's proof only if for all \( t \) iterations the verifications in Step 4 are successful.

**Remark:** If \( Y \not\in Z_n^* \) were allowed (as in the Fiat-Shamir protocol) then a crooked prover \( A' \) could convince the verifier \( B \) (who must adhere to the protocol) that some quadratic non-residues \( \tilde{I} \) belong to \( QR_n \). *E.g.*, if \( A' \) chooses \( X \equiv 0 \pmod{n} \), then \( B \) will always accept.\(^1\)

We will describe a protocol which generalizes this scheme and we will show that all the protocols in \([1,4,6,7,8,12]\) are particular cases of this protocol. In Section III, we will prove that our protocol is a zero-knowledge proof of membership or a zero-knowledge proof of knowledge, depending on the setting.

**II. A framework for a zero-knowledge proof**

In our general scheme the "public numbers" \( I_1, I_2, \ldots, I_m \) are taken from a set \( \mathcal{H} \) and the "secret numbers" belong to a set \( \mathcal{G} \). These numbers are related by a predicate \( P(\cdot, \cdot) \), that is \( P(I_j, S_j) \) for all \( j \). We assume that \( \mathcal{H}, \mathcal{G} \) have some algebraic structure and we take \( P(I, S) \) to be the predicate \( (I = f(S)) \), where \( f \) is a homomorphism. Such predicates are a common feature of all the protocols we consider. We remark that the notion of group homomorphisms has also been used in [13] but in a different context. In our protocol we use the following:

- a monoid \( \mathcal{G}'' \), with subsets \( \mathcal{G}, \mathcal{G}' \) such that \( \mathcal{G} \subset \mathcal{G}' \subset \mathcal{G}'' \). All the secret numbers \( S_i \) belong to \( \mathcal{G} \). \( \mathcal{G}' \) contains the identity and all the elements of \( \mathcal{G} \) are units (it means invertible elements).

- a semigroup \( \mathcal{H}'' \), with subsets \( \mathcal{H}, \mathcal{H}' \) such that \( \mathcal{H} \subset \mathcal{H}' \subset \mathcal{H}'' \). \( \mathcal{H}' \) has an identity and its elements are units.

- a (possibly one-way) homomorphism \( f : \mathcal{G}'' \rightarrow \mathcal{H}'' \) with \( f(\mathcal{G}) = \mathcal{H} \).

The security parameter is \( |n| = O(\log n) \), where \( n = |\mathcal{H}| \). We shall regard this framework as being a particular instance of a general framework which is defined for all (sufficiently large) integers \( n \). We therefore are tacitly assuming that \( \mathcal{G} = \mathcal{G}_n \), \( \mathcal{H} = \mathcal{H}_n \), etc. In this setting we have a framework for (a) a proof of membership for the language \( L = \bigcup_n \mathcal{H}_n : \) the prover wants to prove that all the public numbers \( I_j \) belong to \( L \); (b) a proof of knowledge for the predicate \( P(I, S) : \) the prover wants to prove that she "knows" secret numbers \( S_j \) such that \( P(I_j, S_j) \) for all \( j \). Let us now describe the protocol.

\(^1\)An interesting case occurs when \( I_1 \) is a quadratic non-residue of \( p, I_1 \equiv 1 \pmod{p}q, n = pq \), and \( m = 1 \). If \( A' \) sends \( Z = p^2 \) in Step 1 and \( Y = p \) in Step 2 then \( B \) will always accept \((p = 5, q = 7, I_1 = 8 \) is worth exploring).
Protocol

First the verifier checks that all the \( I_j \in \mathcal{H} \). Then the protocol starts. Repeat \( t \) times:

**Step 1** A selects a random \( X \in \mathcal{G}'' \) and sends \( B: Z = f(X) \) (A's cover).

**Step 2** B sends A a random \( q = (q_1, \ldots, q_m) \in Q^m \) (B's query).

**Step 3** When all \( q_i \in Q \), A sends \( B: Y = X \cdot \prod_j S_j^{q_j} \) (A's answer).

**Step 4** B verifies that \( Y \in \mathcal{G}' \) and that \( f(Y) = Z \cdot \prod_j I_j^{q_j} \) (B's verification).

If the precondition is satisfied, and if for all iterations the conditions in Step 4 are satisfied then B accepts A's proof.

**Remark:** An important feature of this protocol is the inbuilt probability \( |(\mathcal{G}' \setminus \mathcal{G}')/|\mathcal{G}''|) \) that an honest prover fails to convince the verifier.

II.1. A group based framework

We now state conditions that make the protocol a zero-knowledge proof. First consider the case when \( \mathcal{G} = \mathcal{G}' = \mathcal{G}'' \) is a group. We assume that:

1. **Conditions for computational boundedness of B:**
   1.a) We can check if \( I \in \mathcal{H} \) in polynomial time.
   1.b) We can check if \( Y \in \mathcal{G}' \) in polynomial time.
   1.c) Multiplication in \( \mathcal{H}'' \) can be executed in polynomial time.
   1.d) \( f \) is a polynomial time mapping.

2. **Completeness condition:** none.

3. **Soundness conditions:**
   3.a) The set of exponents is \( Q = \{0, 1\} \).

4. **Zero-knowledge condition:**
   4.a) We can choose at random with uniform distribution an element \( X \in \mathcal{G}'' \).
   4.b) \( m \) is \( O(\log |n|) \).

5. **Conditions for Proofs of knowledge:**
   5.a) \( \mathcal{H}' = \mathcal{H} \).  

Copyright (c) 1998, Springer-Verlag
5.b) Multiplication in $G'$ and taking inverses in $G'$ are polynomial time operations.

We show in Section III, that the conditions above are sufficient to make the protocol a zero-knowledge proof. However these conditions are rather restrictive and we only get the Chaum-Evertse-van de Graaf protocols [4]. In the following section we relax these conditions and show that the $[1,6,7,8,12]$ are also particular cases of our protocol.

The Chaum-Evertse-van de Graaf protocols

Many protocols related to the discrete logarithm problem in a general sense were presented by Chaum-Evertse-van de Graaf [4]. The first one, called the multiple discrete logarithm, proves existence (and knowledge) of $S_j$, such that $\alpha^S = I_j$, where $\alpha$ is an element of a group $H''$. Examples of $H''$ are $\mathbb{Z}_N^*$, where $N$ is a prime or composite number. This is a particular case of our protocol for which

- $G = \mathbb{Z}_n(\cdot)$, $n$ is a multiple of the order of $\alpha$,

- $H'' = H$ is a group, $H = \langle \alpha \rangle$ is the group generated by $\alpha$,

- $Q = \{0,1\}$, $m = 1$, and $f$ is the group homomorphism $f : \mathbb{Z}_n \rightarrow H ; x \rightarrow \alpha^x$.

We assume that the verifier knows an upper bound for $n$. Let us check the above conditions. Conditions 1.b and 5.b are satisfied even if one does not know what $n$ is. Conditions 1.a and 1.c must be satisfied by $H''$, which is automatically the case when $H'' = \mathbb{Z}_N^*$. All the other conditions are trivially satisfied.

Next let us consider the Chaum-Evertse-van de Graaf protocol for the relaxed discrete log and show that it is also a particular case. This proves existence (and knowledge) of $S = (s_1, s_2, \ldots, s_k)$ such that $\alpha_1^{s_1} \alpha_2^{s_2} \cdots \alpha_k^{s_k} = I$, where $\alpha_1, \alpha_2, \ldots, \alpha_k, I$ are elements of a group $H''$. To relate this scheme to our protocol we use “direct product groups”. We take:

- $G = \mathbb{Z}_{n_1}(\cdot) \times \mathbb{Z}_{n_2}(\cdot) \times \cdots \times \mathbb{Z}_{n_k}(\cdot)$, where $n_i$ is a multiple of the order of $\alpha_i$ ($1 \leq i \leq k$),

- $H'' = H$ is a group, $H = \langle \alpha_1, \alpha_2, \ldots, \alpha_k \rangle$,

- $Q = \{0,1\}$, $f : G \rightarrow H ; (x_1, x_2, \ldots, x_k) \rightarrow \alpha_1^{x_1} \alpha_2^{x_2} \cdots \alpha_k^{x_k}$.

As in Chaum-Evertse-van de Graaf, $H''$ has to be commutative, ($G$ is commutative). There is one difference between the Chaum-Evertse-van de Graaf scheme and our description of it. In the former, $A$ sends $\alpha_1^{x_1}, \alpha_2^{x_2}, \ldots, \alpha_k^{x_k}$ in Step 1,
whilst in ours $A$ sends $f(X) = \alpha_1^a_1\alpha_2^a_2 \cdots \alpha_k^a_k$. This means that the prover makes more multiplications. the verifier makes fewer multiplications, and less is communicated.

Chaum-Evertse-van de Graaf take $m$ to be 1, which is not necessary. Indeed when $m > 1$ the protocol proves knowledge of the multiple relaxed discrete log. It proves knowledge of $S_1 = (s_{11}, \ldots, s_{1k}), S_2 = (s_{21}, \ldots, s_{2k}), \ldots, S_m = (s_{m1}, \ldots, s_{mk})$, such that $\alpha_1^s_1 \cdots \alpha_k^s_k = I_1, \alpha_1^s_2 \cdots \alpha_k^s_k = I_2, \ldots, \alpha_1^s_m \cdots \alpha_k^s_m = I_k$.

Chaum-Evertse-van de Graaf also discussed a protocol for the simultaneous discrete log. This proves knowledge of $S$ such that $\alpha_1^s = I_1, \alpha_2^s = I_2, \ldots, \alpha_k^s = I_k$. For this protocol we have $G = Z_n(+), H = \langle \alpha_1 \rangle \times \langle \alpha_2 \rangle \times \cdots \langle \alpha_k \rangle$, and $f : G \rightarrow H : z \rightarrow (\alpha_1^z, \alpha_2^z, \ldots, \alpha_k^z)$. The other sets an the remarks about the conditions are similar to those for the multiple discrete logarithm.

II.2. A monoid based framework

We relax the conditions of the group based framework by allowing the sets $G, G', G''$ to be distinct, by taking the set of exponents $Q$ to be any set of integers, and by introducing some new conditions and modifying others. We use the same numbering and list only those conditions which are new or modified.

2. Completeness conditions:

2.a) $|G'|/|G''| \geq 1 - |n|^{-c}, c$ any constant.

2.b) $G'.G \subseteq G'$.

3. Soundness conditions:

3.a) There is an $a$ such that: (i) $|Q \pm a) \cap Q| \geq \psi |Q|$, where $(Q \pm a) = (Q + a) \cup (Q - a)$ and $\psi \in (0, 1]$ is a constant, and (ii) if $f(Y') = f(Y) \cdot I^a$ for some $Y, Y' \in G'$ and $I \in H$ then there exists an element $S \in G$ such that $P(I, S)$.

4. Zero-knowledge condition:

4.b) $m \log |Q|$ is $O(\log |n|)$.

5. Condition for Proofs of knowledge:

5.b) (replaces 3.a (ii)) Given $Y, Y' \in G'$ and $I \in H$ with $f(Y') = f(Y) \cdot I^a$, we can obtain in polynomial time an element $S \in G$ such that $P(I, S)$.

Remark: In most cases $Q$ is of the form $[0:m]$ or $[1:m], a = 1$ and $\psi = 1$. If $Y$ is a unit and $1 \in Q$ then Condition 3.a is trivially satisfied for $a = 1$ and $S = Y^{-1}Y'$. 

Copyright (c) 1998, Springer-Verlag
The Fiat-Shamir scheme

This protocol was discussed earlier. We take, $G'' = H'' = Z_n(\cdot)$, $n$ a product of two distinct primes, $G' = G = H' = Z_n^*(\cdot)$, $H = QR_n$, $Q = \{0, 1\}$, $a = 1$ and $f : Z_n \rightarrow Z_n; x \rightarrow x^2$, which is a homomorphism of the monoid $Z_n$. The reader can easily check that all conditions of Section II.2. are satisfied.

The Feige-Fiat-Shamir scheme

For this scheme $I_j = \pm s_j^2$ [7] (to be consistent with our general presentation we have modified slightly the notation), so that the secrets $S_j$ consists of two parts: the sign part and the $s_j$. To make the relation of the Feige-Fiat-Shamir scheme with our protocol we use direct products of monoids. Let $n = pq$, $p, q$ distinct primes with $p \equiv q \equiv 3 \pmod{4}$. Take

- $G'' = \{-1, +1\}(\cdot) \times Z_n(\cdot)$, $G' = \{-1, +1\} \times Z_n^0$, $Z_n^0 = Z_n \setminus \{0\}$, $G = \{-1, +1\} \times Z_n^*$,

- $H = H' = Z_n(\cdot)$, $H = Z_n^{+1} = \{y \in Z_n^* \mid (y \mid n) = 1\}$, where $(y \mid n)$ is the Jacobi symbol,

- $Q = \{0, 1\}$, $a = 1$ and $f : \{-1, +1\} \times Z_n \rightarrow Z_n; (g, x) \rightarrow gx^2$.

This scheme is essentially the same as the Feige-Fiat-Shamir scheme except that in Step 3 of the protocol the prover sends $Y = X \prod_j S_j^{y_j}$, where $Y$ is a pair with a sign part $y_1 \in \{-1, 1\}$ and a number part $y_2 \in Z_n$, whereas in Feige-Fiat-Shamir only a number is sent. However in the latter the verifier must check if $Y^2 = Z \cdot \prod_j I_j^{y_j} \pmod{n}$ or if $Y^2 = -Z \cdot \prod_j I_j^{y_j} \pmod{n}$. By doing this he knows exactly what the sign $y_1$ is. Therefore, for us the prover sends one extra bit in Step 3 whereas in Feige-Fiat-Shamir the verifier has to check one more equation. The two schemes are essentially the same, only the actual implementation is slightly different. Observe that the remark about the Fiat-Shamir protocol in the introduction applies to this protocol as well: if $Y \not\in Z_n^0$ were allowed then we do not have a proof system.

The Desmedt scheme

For this scheme [6] take the same parameters as we discussed for the Feige-Fiat-Shamir scheme, except that $f : \{-1, 1\} \times Z_n \rightarrow Z_n; (h, x) \rightarrow hx^{20j}$. Take $I_j = R_j/g_i(1)(\pmod{n})$, where $g_i(x) = g_{i_1}(g_{i_{i-1}}(\cdots (g_{i_1}(g_{i_0}(x))) \cdots))$, with $g_0(x) = x^2(\pmod{n})$ and $g_1(x) = 4x^2(\pmod{n})$. 
The Guillou-Quisquater scheme

Take

- \( G'' = H'' = Z_n(\cdot) \), \( n \) a product of two different primes, \( G' = G = H' = Z_n^* \),
- \( H = \{ y \in Z_n^* \mid y = x^a, x \in Z_n^* \} \), \( v \) a prime, \( Q = [0 : v - 1] \), \( a = 1 \)
- \( f : Z_n \to Z_n; x \to x^v \).

For \( m = 1 \) we get the Guillou-Quisquater scheme [12]. We observe that:

1. When \( v^m = O(|n|^c) \), \( c \) a constant, this scheme is insecure (since then "guessing the query" is a convincing strategy). So we must have \( mt \log v \gg \log |n| \). In Section III. we shall see that this scheme is sound when \( t \gg \log |n| \).

2. The zero-knowledge proof in Section III. requires that \( tu^m = O(|n|^c) \), \( c \) a constant. This proof cannot be used when either \( t \gg |n|^c \), or \( v^m \gg |n|^c \).

The Beth scheme

In this scheme [1], a centre possesses the security numbers \( x_1 \ldots x_m \in Z_{q-1} \) and makes public \( \alpha \), a primitive root of \( GF(q) \) and the values \( y_j = \alpha^{x_j} \) for all \( j \).

For each user the centre chooses a random \( k \in Z_{q-1} \) and gives the user \( r = \alpha^k \) as one part of her public number. The other part consists of the numbers \( ID_1, \ldots, ID_m \in Z_{q-1} \). The centre determines the secret numbers \( S_1, \ldots, S_m \) by solving the congruence

\[
x_j r + kS_j \equiv ID_j \pmod{(q - 1)}, \quad j = 1, \ldots, m.
\]

In Step 1 of the protocol the prover sends \( z = r^{-t} (t \) random in \( Z_{q-1} \)) to the verifier. In Step 2 the verifier replies with \( b = (b_1 \ldots b_m), b_i \in Q \subset Z_{q-1} \), and finally in Step 3 the prover sends \( u = t + \sum_j b_j S_j \in Z_{q-1} \). The verification is

\[
\prod_j y_j^{rb_j} r^{-z} = \alpha^{\sum_j b_j ID_j}.
\]

Let us now make the relation with our protocol. Take

- \( G = G' = G'' = Z_{q-1}(\cdot), \quad Q \subset Z_{q-1} \), \( H'' = H' = GF(q)^*(\cdot) \),
- \( H = \langle r \rangle, \quad r \in GF(q)^*, \) and \( f : Z_{q-1} \to GF(q)^*; x \to r^x \).

\footnote{This means that \( \log |n| (mt \log v)^{-1} \to 0 \) as \( |n| \to \infty \).}
Clearly $f$ is a homomorphism of $G$ onto $H$. This is a discrete logarithm proof which looks very similar to the Beth scheme, except for the relation between the public and secret keys of $A$ and the consequences in Step 4. Let us discuss this difference. We have,

$$I_j = f(S_j) = r^S_j = \alpha^{kS_j} = \alpha^{ID_j} = \alpha^{-z_r} = \alpha^{ID_j} y_j^{-r},$$

using (1), so that we can rewrite (2) in the form

$$f(u) = r^u = z^{-1} \alpha^{\sum_j ID_j y_j} \prod_j y_j^{-r} = z^{-1} \prod_j (\alpha^{ID_j} y_j^{-r}) = z^{-1} \prod_j I_j^y.$$  

This is the same as the verification in our protocol for $Y = u$, $Z = z^{-1}$ and $q = b$. So the Beth scheme is essentially a particular case of our protocol. Observe that the verifier can use the $I_j$'s instead of the $\alpha^{ID_j} y_j^{-r}$, which simplifies the computations (if $0, 1 \in Q$ then the verifier can obtain $I_j$ by sending the query $q = q_1 \cdots q_m$ with all entries zero except the $j$-th entry which is 1). The difference between the Beth scheme and our scheme is that in the former it is hard for the user to make her own $ID_j$'s, whereas in the latter it is trivial to make the $I_j$'s. This is exactly the same difference as exists between the Fiat-Shamir versions in [8] and the Fiat-Shamir scheme of [7,9].

III. Fundamentals of the scheme

**Theorem 1** If the conditions of Section II.1. are satisfied with $G = G' = G''$, then the conditions in Section II.2. are also satisfied.

**Proof.** Trivial (take $a = 1$, $\psi = 1$ and $S = Y^{-1} Y''$). $\square$

**Theorem 2** If the Conditions 1-4 of Section II.2. are satisfied, if $m \log |Q| \leq \log |n|$ and if $t$ is bounded by $\log |n| \leq t \leq |n|^c$, $c$ any constant, then the protocol in Section II. is a (perfect) zero-knowledge proof of membership for the language $L = \bigcup_n H_n$. If, furthermore, Conditions 5 are satisfied$^3$ then the protocol is a (perfect) zero-knowledge proof of knowledge for the predicate $P(I, S)$.

**Proof.** (sketch) We remark that we do not rely on unproven assumptions.

**Completeness:** (If $A$ is genuine then $B$ accepts the proof of $A$ with overwhelming probability)

This is obvious since the mapping $f$ is an operation preserving mapping.

---

$^3$We can relax the condition $n = |H|$ to $n = |G|$ in this case.
Soundness: (If A' is crooked then the probability that B accepts the proof of A' is negligible)

The proof is an extension of the one in Feige-Fiš-Shamir [7]. Suppose that A' convinces B with non-negligible probability. We consider the execution tree T of (A', B): this is a truncated tree which describes the responses of A' to the requests of B. A vertex of T is super heavy if it has more than \( \omega = 1 - \frac{1}{4}\psi \) sons (\( \psi \) is the constant in Condition 3.a of Section II.2.; in [7] we have heavy vertices with \( \omega = \frac{1}{2} \)). In the final paper we will show that the condition \( \log |n| < t \) guarantees that T has at least one super heavy vertex. The following Lemma makes it possible to show that there exist \( S_j \) such that \( P(I_j, S_j) \) for all \( j \).

**Lemma 1**: At a super heavy vertex, for each \( j \in [1:m] \) there exists at least one pair of queries \( q = (q_i), q' = (q'_i) \) with \( q'_i = q_i \) for all \( i \neq j \) and \( q'_j = q_j + a \), which A' answers correctly.

**Proof**: Will be given in the full paper.

Apply this Lemma to a super heavy vertex. For each pair of sons we have:

\[
\begin{align*}
  f(Y) &= f(X) I_1^{m-1} I_m^m \\
  f(Y') &= f(X') I_1^{m-1} I_m^m
\end{align*}
\]

with \( f(X) = f(X') \). To find the \( S_j \) we use a recursive procedure: first we find \( S_m \) and then we use it to calculate \( S_{m-1} \) and continue in the same way until we find all the \( S_j \). Suppose that \( q \) and \( q' \) differ in the last place. Since \( I_m^m \) and \( I_m^{m+a} = I_m^{m+a} \) are units the equations above can be written in the form,

\[
\begin{align*}
  f(Y) I_m^{-q_m} &= f(X) I_1^{m-1} I_m^{m-1} \\
  f(Y') I_m^{-q'_m-a} &= f(X') I_1^{m-1} I_m^{m-1},
\end{align*}
\]

so that \( f(Y') = f(Y) I_m^a \). Then using Condition 3.a we obtain an \( S_m \) such that \( P(I_m, S_m) \). This solution is not necessarily the \( S_m \), but it is a good substitute.

This procedure is repeated to find \( S_{m-1}, S_{m-2}, \ldots, S_1 \). This completes the proof, for proofs of membership. For proofs of knowledge we have to show that there exists a polynomial time Turing machine, the interrogator \( M \), that will extract the secrets from A'. \( M \) is allowed to reset A' to any previous state: this means that it can "obtain" all the sons from a super heavy vertex and hence all the \( S_j \) in the manner described earlier, this time using Condition 5.b. It remains to show how the interrogator can find a super heavy vertex in polynomial time. In the extended proof we will show that:

**Lemma 2**: At a suitable level i of the execution tree the fraction of super heavy vertices is at least \( \gamma \), where \( \gamma \in (0,1] \) is a constant.
Proof: Will be given in the full paper.

In the final paper we prove that $M$ will find a super heavy vertex (with overwhelming probability) in polynomial time.

Zero-knowledge: (For each $B'$ there exists a probabilistic expected polynomial time Turing Machine $M_{B'}$ which can simulate the communication of $A$ and $B'$)

The simulator proceeds as follows:

Step 1 $M_{B'}$ chooses a random $X$ from $G''$ (using Condition 4.a) and a random vector $q$ from $Q^m$ and sends to $B'$: $Z = f(X)(\prod_i I_i^{(i)})^{-1}$.

Step 2 $M_{B'}$ reads the answer of $B'$, $q'$. If $q' = q$ then it sends $X$ to $B'$. If $q' \neq q$ then it rewinds $B'$ to its configuration at the beginning of the current iteration and repeats Step 1 and Step 2 with new random choices.

When all the iterations are completed, $M_{B'}$ outputs its record. The expected number of probes for a complete run is $t |Q|^n = O(n^r)$. Observe that the probability distribution output by $M_{B'}$ is identical to that of the transcript set of $(A, B')$. So this scheme is a perfect zero-knowledge scheme [11].

\[ \square \]

IV. Conclusion

In this paper we have shown that the schemes described in [1,4,6,7,8,9,12] are all particular cases of one protocol. This protocol has been further generalized to include the Goldreich-Micali-Wigderson graph isomorphism scheme [10], the Chaum-Evertse-van de Graaf-Peralta scheme [5], and schemes based on encryption functions, such as the Brassard-Chauam-Crepeau [3] scheme and the Goldreich-Micali-Wigderson proof of 3-colourability [10]. However this is not in the scope of the monoid based framework.

REFERENCES


