Efficient Zero-Knowledge Identification Schemes for Smart Cards

M. BURMESTER*[1], Y. DESMEDT*[2] AND T. BETH[3]

[1] *Department of Mathematics, University of London – RHBNC, Egham, Surrey TW20 0EX*

[2] *Department of EE & CS, University of Wisconsin – Milwaukee, P.O. Box 784, WI 53201, USA*

[3] *European Institute for System Security, Universität Karlsruhe, Fakultät für Informatik, D-7500 Karlsruhe, Germany*

Secure identification is an important security issue to avoid computer fraud due to masquerading. This can be achieved with zero-knowledge based smart cards. We present very efficient new zero-knowledge schemes in a general algebraic setting. Particular cases of our scheme improve the performance of the Guillou–Quisquater and the Chaum–Evertse–van de Graaf schemes. Our scheme is formally proven and, overall, is more efficient than currently available schemes including the Fiat–Shamir scheme. As an application we discuss how our scheme can be used for identification, in particular as an electronic passport scheme.

Received May 1991, revised September 1991

1. INTRODUCTION

To guarantee computer security one needs a *secure login*. Login methods used nowadays are insecure. When typing in a password on a terminal bystanders can easily overlook and see what is being entered. Worse, often one is logging in at a remote machine, thousands of miles away. The password travels electronically from a terminal or a personal computer to the remote machine and meanwhile it is easy to eavesdrop. Due to the use of workstations and the progress in networking, ethernet based Local Area Networks (LANs) have become common. For such networks the password of each user passes through all local workstations, making the task of eavesdropping much easier. In many cases of computer break-in the attacker does not have to eavesdrop to get the password, but just consults a dictionary of frequently used passwords. So there is no doubt that more secure login procedures must be developed. In our paper we discuss techniques to achieve this.

Before addressing this problem we remark that entering a PIN in an Automatic Teller Machine (ATM), or at a Point of Sales (POS) terminal, is a special case of login. The same applies for phone cards. Although the amount involved in such transactions usually is small, criminal organisations are sometimes targeting ATMs and POSs. Indeed due to the very large number of customers the total amount involved in such transactions could be enormous. To reduce such criminal activities some credit card companies are using cards which are harder to forge, for example, by displaying a hologram. In some countries, such as France,[27] smart cards are already in use today to further reduce the risk.

The problem of secure login is a special case of the problem of *secure identification*. Identification is the process in which an individual (the prover) identifies himself to another individual or a machine. The last party is called the verifier. Tokens for identification purposes such as, drivers' licences, identification cards, passports and visas, are often used in our society. Furthermore, to an increasing degree credit cards are accepted for identification purposes (although without pictures). Fake passports and fake drivers' licenses are handy tools for criminals and terrorists.

A brief survey of identification schemes

Let us briefly overview the history of identification mechanisms. One of the oldest methods of identification has been the password. This mechanism is based on secret knowledge. As already mentioned this solution is insecure because the prover *reveals* his secret. Indeed the prover does not know if he identifies himself to an adversary who could later on use the password fraudulently.

Other techniques rely on the assumption that it is hard (expensive) to produce certain types of cards. However old-fashioned cards 'can be manufactured, without excessive difficulty, to resemble true cards sufficiently well as to deceive most people having to deal with them'.[15]

Recently systems relying on the physical description of the individual have been proposed. Such systems have a questionable security and in many circumstances this approach is either too expensive, or too impractical or unacceptable for the users. Davies and Price have stressed this last aspect,[14] giving examples, for example measuring lips for identification purposes which would be rejected on hygienic grounds. The idea of measuring fingerprints creates the fear that some machines could cut an individual's finger.

Today, a major research topic in data security is the design of secure identification mechanisms.[35,41,21] Recently great attention has been directed to identification based on zero-knowledge.[22] The prover does *not* have to reveal his secret password. Instead, he *proves* (using interactive challenges) to a verifier the knowledge of a secret without revealing it. Since this proof requires a computation, a chip card which contains the secret of the prover is used. At first glance, it appears that the verifier cannot masquerade as the prover, because the secret is not revealed. However, it has been pointed out[3,18] that the verifier could perform an on-line fraud by forwarding

* Research partly done while visiting the EISS, University of Karlsruhe, West Germany.

the bits which are being exchanged. Techniques to avoid this, and similar frauds, have been proposed[3,5] and make zero-knowledge identification attractive when implemented carefully. One of these is that a trusted centre keeps a black list of stolen or lost tokens. Another is that the prover's secret is hidden from him, to prevent the prover from helping others impersonate him. This can be achieved by having the trusted centre encapsulate the secret in a tamper-proof device. All so far known problems with zero-knowledge based identification methods can be easily overcome, *except* the rental of smart cards. If no physical description is used then it is hard to detect this fraud because the card is identified and *not* the individual. However, one could argue that it is unlikely that one will rent one's own credit card!

This paper is organised as follows. In the remainder of this section we give some definitions. In Section 2 we discuss a particular case of our protocol and introduce zero-knowledge informally. In Section 3 we describe the general protocol, and in Section 4 we discuss its efficiency. In Sections 5 and 6 we prove that our protocol is a zero-knowledge proof. Many informally presented zero-knowledge schemes have collapsed, or require radical modifications.[10] For this reason we have opted for a formal approach. At a first reading the proofs could be omitted. In Section 7 we consider an application of our scheme to electronic passports, and we conclude in Section 8.

Definitions

In this paper we focus on the development of new and faster zero-knowledge protocols. A characteristic property of our protocols is that the number of messages exchanged, or *rounds*, is 'almost-constant'. This property to a large extent accounts for their efficiency. A function $t = t(n)$ of the natural numbers is *almost-constant* if it is an unbounded function which grows very slowly, for example significantly slower than $\log n$. For our purpose it is not necessary to specify such functions any further, but if it helps to have a particular function in mind then we could take $t(n) = \log * n$ (so $t(n)$ grows slower than any iterated logarithm).[7]

We note that if f, g are functions of the natural numbers then $f(n) = O(g(n))$ if there is a constant c such that $f(n) \leqslant cg(n)$ for (almost) all n, and $f(n) = \Theta(g(n))$ if there are positive constants c, d such that $cg(n) \leqslant f(n) \leqslant dg(n)$ for (almost) all n.

Finally a set L belongs to the class **BPP** if it can be recognised by a polynomially bounded probabilistic Turing machine with bounded error probability.[23]

2. AN ALMOST-CONSTANT ROUND PROTOCOL

In this Section we discuss intuitively the concept of zero-knowledge schemes, by considering a concrete new example. We use this to make a practical identification scheme for passports and remote login (see Section 7). This scheme will be utilised as the building block for a zero-knowledge proof (of membership) for the discrete logarithm problem, which is the most efficient so far presented (see Corollary 2 of Section 5).

The discrete logarithm is an important one-way function which is often used in modern cryptography. Many key exchange protocols[19,33,34] and the El Gamal signature scheme[20] have been based on the discrete logarithm. As an illustrative example, when we perform our calculations mod 7, then the discrete logarithm of 2 base 5 is 4, because $5^4 \equiv 2 \pmod 7$. When 7 is replaced by a very large prime number then finding the discrete logarithm, or even finding whether it exists, is considered to be difficult.[37]

Let p be a prime number and Z_p be the ring of integers modulo p. Define k_p to be the largest divisor of $p-1$ whose prime factors are at least as large as $v = \lceil \log_2 p \rceil$. We note that it is easy to compute k_p.

We shall consider a language (set) L whose elements are the strings $x = (I; \beta, p)$, where β is an element of $Z_p^* = Z_p \backslash \{0\}$ for which $\beta^{k_p} \equiv 1 (\operatorname{mod} p)$ and $I \in Z_p^*$ is such that $I \cdot \beta^s \equiv 1 (\operatorname{mod} p)$ for some $s \in Z_{p-1}$. That is,

$$L = \{(I; \beta, p) \mid p \text{ is a prime}, \beta, I \in Z_p^*;\ \ \beta^{k_p} \equiv 1 (\operatorname{mod} p);\ \exists s \in Z_{p-1} : I \cdot \beta^s \equiv 1 (\operatorname{mod} p)\}.$$

We call I the 'public' number and s the 'secret' number. Let H be the subgroup of $Z_p^*(\cdot)$ which has order k_p. Then the public numbers are elements of H. Observe that any $\beta \in H, \beta \neq 1$, has order at least $v = \lceil \log_2 p \rceil$.

In the following protocol, $|p|$ is the binary length of p and $a \in_R A$ indicates that the element a is selected randomly with uniform distribution from the set A.

Protocol (P, V): Input $x = (I; \beta, p)$.

V checks that p is a prime,[42] then computes k_p and checks that $I \in H$ and that $\beta^{k_p} \equiv 1 (\operatorname{mod} p)$. If any one of these conditions fails then V halts and rejects. Otherwise the following Steps are repeated t times independently, where $t = t(|p|)$ is almost-constant.

Step 1. P sends to V: $z = \beta^r (\operatorname{mod} p)$, where $r \in_R Z_{p-1}$.

Step 2. V sends to P: q, where $q \in_R Z_{|p|}$.

Step 3. P verifies that $q \in Z_{|p|}$, and if so, sends to V:

$$y \equiv r + qs (\operatorname{mod} (p-1)), \text{ where the } s \text{ is such that } \beta^s \cdot I \equiv 1 (\operatorname{mod} p).$$

Step 4. V verifies that $y \in Z_{p-1}$ and that $z \equiv \beta^y \cdot I^q (\operatorname{mod} p)$. If this check fails then the protocol is halted.

After t complete rounds V accepts (V is convinced).

This protocol is a *proof of membership in* the language L. That is, when $x \in L$ then the verifier will (almost) always accept, whereas when $x \notin L$ the verifier will (almost) always reject. Indeed, when $x \in L$ then $\beta^y \cdot I^q \equiv \beta^r \cdot \beta^{qs} \cdot I^q \equiv z \cdot (\beta^s \cdot I)^q \equiv z (\operatorname{mod} p)$, so that the verification in Step 4 checks and V always accepts. Next let $x \notin L$. When $I \notin H$ or $\beta^{k_p} \not\equiv 1 (\operatorname{mod} p)$, V always rejects. When $I \in H$ and $\beta^{k_p} \equiv 1 (\operatorname{mod} p)$ and when there is no s such that $\beta^s \cdot I \equiv 1 (\operatorname{mod} p)$, then it can be shown (Lemma 1 of Section 5) that it is not possible, even for an infinitely powerful dishonest prover, to answer more than one of the queries q in Step 3 of any particular round. So the probability that V will accept after t consecutive rounds when there is no s such that $\beta^s \cdot I \equiv 1 (\operatorname{mod} p)$ is negligible (at most $|p|^{-t}$).

The protocol is also a *zero-knowledge* proof (a formal definition and proof are given in Section 5). Intuitively, the prover does not reveal any new knowledge to the verifier because the verifier can simulate by himself the communications of the protocol. The verifier can guess his own question q and obtain by himself the z and y which the prover would send him. That is he can produce strings (z, q, y) which satisfy the conditions in Step 4, with the same distribution as in the actual protocol. A crucial aspect of this simulation is that given any q and any y it is easy to obtain an appropriate z. Of course a simulated proof will not convince the verifier that $x \in L$.

This protocol with suitable preprocessing can be used to recognise the elements of any subgroup $\langle \beta \rangle$ generated by β of Z_p^*, with *no* restriction on the order of β or I (Corollary 2 of Section 5).

If the factors of $(p-1)$ are not known then it is generally believed that checking membership in L is a hard problem.[1] So a prover P who has unlimited computational power can use this protocol to prove that $x \in L$ to a verifier with limited resources.

Next suppose that the verifier V is given the factors of $(p-1)$, say as an additional part of the common input x. Then V can easily check by himself membership in L. Indeed from the factors of $(p-1)$ the verifier can obtain the order of β, say d. Then when the initial test is satisfied, $x \in L$ if and only if $I^d \equiv 1 (\bmod p)$, since Z_p^* is a cyclic group (from elementary group theory). However, even though V can confirm the existence of an s such that $\beta^s \cdot I \equiv 1 (\bmod p)$, it is generally believed[1] that it is hard to find such an s. This is called the discrete logarithm problem. In this case the protocol can be used as a *proof of knowledge*. The prover convinces the verifier that he 'knows' a secret s such that $\beta^s \cdot I \equiv 1 (\bmod p)$, and not merely that there exists such an s (as in the case of proofs of membership).

3. A GENERAL PROTOCOL

The protocol above can be generalised. One obvious generalisation is to replace Z_p by any finite field $GF(q)$. There are many other possible settings. One such setting is based on elliptic curves whose q-rational points form an Abelian group. Discrete logarithms on elliptic curves have recently been extensively studied because in our present state of knowledge they implement much faster algorithms[36, 32, 6] (for a given level of security). To allow for such settings, and for others which cannot be foreseen, we shall consider a general protocol.

The protocol in Section 2 is based on the homomorphism $Z_{p-1}(+) \to Z_p^*(\cdot) : r \to \beta^r$. We can generalise this by taking a group homomorphism $f: G \to H$. However, to make a complexity theoretic treatment possible we have to consider families $f_n : G_n \to H_n$ of such homomorphisms. Here G_n, H_n are groups, and $f_n(ab) = f_n(a) f_n(b)$ for all $a, b \in G_n$. These families are indexed by a string $n \in J$, where J is an infinite subset of $\{0, 1\}^*$. To have cryptographic value the functions f_n must be one-way. A number of things are specified by n. In particular, the triple (f_n, G_n, H_n) is described in some uniform way. Also n specifies a number $v = v(n)$ which will be used in our protocol. For convenience, and when there is no ambiguity, we will omit the subscript n from G_n and H_n (but not from f_n).

The input x of our protocol consists of the 'public number' $I \in H_n$ and the string n. We use the abbreviation $x = (I; n)$.

General Protocol (P, V): Input $x = (I; n)$.

V checks that $n \in J$ and $I \in H$, calculates $v = v(n)$, and then executes a probabilistic polynomial time test T to be specified later (T is always true when unspecified). If any of these conditions fails then V halts the protocol and rejects (the proof of P). Otherwise the following steps are repeated t times independently, where $t = t(|n|)$ is almost-constant.

Step 1. P sends to V: $z = f_n(r)$, where $r \in_R G$.

Step 2. V sends to P: q, where $q \in_R Z_v$.

Step 3. P verifies that $q \in Z_v$ and if so, sends to V:

$y = r \cdot s^q \in G$, where the s is such that $I \cdot f_n(s) = 1$.

Step 4. V verifies that $y \in G$ and that $z = f_n(y) \cdot I^q$. If this check fails then the protocol is halted.

After t successful rounds V accepts (the proof of P).

In order that the verifier can execute efficiently his part of the protocol we have to restrict the setting as follows:

- the mappings f_n and the operation of H_n can be executed in polynomial time (in $|n|$),
- $L_G = \cup_{n \in J} G_n$ and $L_H = \cup_{n \in J} H_n$ are in **BPP**.
- I can be efficiently represented, that is $|I|$ is $O(|n|)$.
- it is easy to recognise the strings n of J (that is, J belongs to **BPP**).

From now on we shall assume that these conditions are always satisfied.

In Section 5 we will prove that this protocol is a proof of membership for the language $L = \{(I; n) \mid n \in J; I \in f_n(G_n)\}$, if appropriate conditions are satisfied. In Section 6 we consider the case when the protocol is a proof of knowledge.

4. EFFICIENCY

In this section we discuss the performance of our protocol and compare it with existing schemes. We shall assume that v (the number of 'queries' in Step 2) is polynomially bounded in $|n|$, and hence that $|v| = O(\log |n|)$. For the communication and computation complexity we only consider the case when the elements of the groups G, H are modular numbers and when the group operations can be expressed in terms of modular operations. For convenience we assume that $H = Z_m^*$ (the set of numbers modulo m which are relatively prime to m) and that the length of the input is $\Theta(|m|)$. The performance of our protocol will be measured in terms of $|m|$.

The number of iterations

An interactive protocol is of Arthur–Merlin (A–M) type if the strings sent by the honest verifier consist of random bits.[2] It has been shown[24] that a constant round A–M proof for a language L cannot be proven zero-knowledge (using black-box simulation) unless L belongs to **BPP**. This result has been extended to proofs of knowledge.[30] Our protocol is A–M (for us v is not necessarily a power of 2: however this makes little difference). Consequently, in the cases when our protocol is a zero-knowledge proof, our choice of almost-constant rounds is optimal (provided that the corresponding problem is hard).

To achieve almost-constant rounds we have had to use a relatively large v. The idea of using such vs is related to 'deep coin tosses'.[28] However, in earlier work no bounds were set on v, resulting in informal systems which are not zero-knowledge (provided that the corresponding problem is hard).

The number of bits communicated

With each round of the protocol $|m|$ bits are communicated in Step 1 and Step 3 and $|v|$ bits in Step 2. For t iterations we have $t(2|m|+|v|)$ bits. Since $|v| = O(\log |m|)$, when $t \approx \log^{\varepsilon}|m|$, the number of bits communicated is $|m| \log^{\varepsilon}|m|$.

The number of operations

We shall assume, for convenience, that the cost of multiplication in G is no more than in H and that the cost of the exponentiation r^q in H is (on average) $3/2 \log q$ multiplications. Suppose that the cost of computing $f_n(r) \in H, r \in G$, is equivalent to $l(r)$ multiplications in H. Then we have $l(r)$ multiplications in Step 1, $1+3/2\log q$ multiplications in Step 3, and $1+l(y)+3/2\log q$ multiplications in the verification step. For $t \approx \log^{\varepsilon}|m|$ iterations we get approximately $(l(r)+l(y)+O(\log |m|))$ $\log^{\varepsilon}|m|$ multiplications, since $q < v$ and $|v| = O(\log |m|)$. If we assume that $l(r) = \Theta(|v|)$, for example when $f_n(r) = r^v$ (as in many schemes[28, 38, 8]), then the computational complexity is approximately $\log^{1+\varepsilon}|m|$ multiplications. When $l(r) = \Theta(|r|)$, for example when $f(r) = \beta^r$ (as in many schemes),[11, 4, 40] the computational complexity is approximately $|m| \log^{\varepsilon}|m|$ multiplications (in the general case).

Comparison with other schemes

In Table 1 we compare the performance of many schemes.

For the Fiat–Shamir scheme,[22] $G = H = Z_m(\cdot)$, where $m = pq, p, q$ distinct primes, and $f: r \to r^2$. The input $x = (I_1, \ldots, I_k; m)$ has k public numbers, $k = O(\log |m|)$. We can take the number of rounds to be $t \approx \log^{1+\varepsilon}|m|$ so that the number bits communicated is $t(2|m|+k) \approx |m| \log^{1+\varepsilon}|m|$. The (average) number of multiplications is $t(k+2) \approx k \log^{1+\varepsilon}|m|$. For the Guillou–Quisquater[28] and Ohta–Okamoto[38] schemes we have $G = H = Z_m^*(\cdot)$ and $f: r \to r^v$, and we get the same bounds. However if our conditions for Corollary 1 of Section 5 apply, then we can take $t \approx \log^{\varepsilon}|m|$ and the communication complexity reduces to $|m| \log^{\varepsilon}|m|$ bits. For the Chaum–Evertse–van de Graaf scheme[11] and the Beth scheme[4] we have $G = Z_{p-1}(+)$, p a prime, $H = Z_p^*(\cdot)$ and $f: r \to \beta^r, \beta \in Z_p^*$, and we can take $t \approx \log^{1+\varepsilon}|p|$. The number of bits communicated is $t(2|p|+k) \approx |p| \log^{1+\varepsilon}|p|$, where $k = O(\log |p|)$ is the number of secrets, and the number of multiplications is $k|p| \log^{\varepsilon}|p|$. If we use our protocol in Section 2 then we can reduce the communication complexity to $|p| \log^{\varepsilon}|p|$ (see Section 5, Corollary 2, and Section 6, Corollary 3). The Schnorr identification scheme[40] uses a one round protocol and therefore is not zero-knowledge[24] (unless the discrete logarithm problem is easy). This scheme is not necessarily sound.[9] However, if the Schnorr protocol is modified so that Corollary 3 of Section 6 applies, then we get an almost-constant round zero-knowledge proof. We note that it is not necessary for the order of β to be a prime number (as in Ref. 40).

Reducing the computational complexity

We consider the protocol in Section 2 when $G = Z_{p-1}(+)$ and $f_n: G \to H; r \to \beta^r, \beta \in H$. In this case the computational complexity can be as much as $|p| \log^{\varepsilon}|p|$ multiplications. We can reduce this to $\log^{1+\varepsilon}|p|$ by modifying the language in such a way that H is replaced by H', a subgroup of H with superpolynomial order, G by $G' = Z_{ord(H')}(+) \subset Z_{p-1}(+)$, and $f_n: G' \to H'; r \to \beta^r, \beta \in H'$.

5. A PROOF OF MEMBERSHIP

Definitions

We present an overview of the concept of zero-knowledge interactive proofs.[25] Informally a *proof* should convince a reader, i.e. a verifier, that a theorem is *true*. It should not be possible for the verifier to be convinced of an untruth. With *interactive* proofs[25] the verifier is allowed to ask questions. For us both the prover and the verifier will use *randomness* (toss coins). The randomness is obtained from private random tapes (that is bit strings with uniform distribution). As a consequence the verifier is only convinced with a certain probability.

Formally the prover P and the verifier V are probabilistic interactive Turing machines. P and V share a common input tape, private work tapes and private random tapes. The interaction is done by using communication tapes.

Definition 1. Let $L \subset \{0,1\}^*$ be a language and let (P, V) be a pair of interactive probabilistic Turing machines. (P, V) is an *interactive proof for L* if for common input x,

- *Completeness*: when $x \in L$ then (P, V) accepts with *overwhelming* probability ($> 1-(1/|x|^a$, any a)
- *Soundness*: when $x \notin L$, for any prover P', (P', V) accepts only with *negligible* probability ($< (1/|x|^a)$, any a).

P' is a prover who may deviate from the protocol (for example, in an attempt to prove to V an untruth). With this setting it is customary to assume that P has unlimited

Table 1. The complexity of some zero-knowledge identification schemes. $|x|$ is the length of the input

Zero-knowledge schemes	Number of rounds	Bits communicated	Multiplications
Fiat–Shamir[22]	$\log^{1+\varepsilon}\|x\|$	$\|x\| \log^{1+\varepsilon}\|x\|$	$k \log^{1+\varepsilon}\|x\|, k = O(\log \|x\|)$
Guillou–Quisquater,[28] Ohta–Okamoto[38]	$\log^{1+\varepsilon}\|x\|$	$\|x\| \log^{1+\varepsilon}\|x\|$	$\log^{1+\varepsilon}\|x\|$
Chaum–Evertse–van de Graaf,[11] Beth[4]	$\log^{1+\varepsilon}\|x\|$	$\|x\| \log^{1+\varepsilon}\|x\|$	$k\|x\| \log^{1+\varepsilon}\|x\|, k = O(\log \|x\|)$
Schnorr[40] (not zero-knowledge)	One	$\Theta(\|x\|)$	$O(\|x\|)$ (without preprocessing)
Our scheme (Particular cases)	$\log^{\varepsilon}\|x\|$	$\|x\| \log^{\varepsilon}\|x\|$	$\log^{1+\varepsilon}\|x\|$

computational resources (and can therefore 'recognise' the elements of L), whereas V is polynomially bounded (and cannot check membership by himself, if this is a 'hard' problem). P and V are the 'honest' parties.

We shall now consider the impact of a possibly dishonest verifier V' on a proof (P, V). V' may deviate from the protocol in an attempt to gain some additional knowledge, for example, about the input x. Informally a proof is *zero-knowledge*[25] if it guarantees that the prover reveals no more knowledge than that $x \in L$ (one bit), to *any* verifier V'. V' may wish to use previously obtained knowledge. To allow for this we shall assume that the verifier has an additional tape, called the *history tape*, on which such knowledge can be written. During the execution of a proof, P and V' exchange messages. We define the *view* of V' to be everything that V' sees. That is, the messages exchanged and the portion of the random tape that V' uses. Clearly V' will gain no further knowledge if this view can be simulated *without any help from* the prover. We define $\mathrm{View}_{\langle P,V'\rangle}(x,h)$ to be the random variable whose value is the view of V'. Here h is the contents of the history tape of V'. Formally we have,

Definition 2. An interactive proof (P, V) for L is *perfect zero-knowledge*[25] if, for every V' there is an expected polynomial time Turing machine $M_{V'}$, the *simulator*, which given $x \in L$ and h will generate a distribution which is identical to that generated by $\mathrm{View}_{\langle P,V'\rangle}(x,h)$.

We also have[25] *statistical* and *computational* zero-knowledge. These, however, will not concern us here.

Our zero-knowledge proof of membership

Theorem 1. *Suppose that the following condition is satisfied*:

C1 $\forall n \in J : v = \lceil \log n \rceil : \forall I \in H_n : \forall q \in Z_v^0 :$
$$I^q \in f_n(G_n) \Rightarrow I \in f_n(G_n).$$

Then protocol (P, V) *is a perfect zero-knowledge proof for the language* $L = \{(I,n) \mid n \in J; I \in f_n(G_n)\}$.

To prove this theorem we use the following,

Lemma 1. *Suppose that condition* C1 *is satisfied. If there is a* $z \in H_n$ *for which there is more than one* $q \in Z_v$ *such that* $z = f_n(y) \cdot I^q$, *some* $y \in G_n$, *then* $I \in f_n(G_n)$.

Proof. If $z = f_n(y') \cdot I^{q'} = f_n(y'') \cdot I^{q''}$ for y', $y'' \in G$, and $q', q'' \in Z_v, q' > q''$, then $f_n(y) \cdot I^q = 1$ with $y = (y'')^{-1} y'$, $q = q' - q''$. So $I^q = f_n(y^{-1}) \in f_n(G)$, and by condition C1 we have $I \in f_n(G)$. ■

Proof of Theorem. Completeness follows directly from the fact that f_n is a homomorphism. For soundness we have to show that if $x \notin L$ then the probability with which the verifier will accept is negligible. From the Lemma there cannot be more than one query q such that $z = f_n(y) \cdot I^q$. Since the verifier V chooses the $q \in Z_v$ randomly with a uniform probability distribution, the probability that the verification in Step 4 is successful when $x \notin L$ is bounded by $1/v$. This is the *probability of error per iteration*. So the probability that V will accept after t independent iterations is bounded by v^{-t}. This is negligible. Indeed we are assuming that $v = \Theta(|n|)$ and that $|n| = \Theta(|x|)$. So $v^{-t} = (1/\Theta(|x|)^t)$, which is negligible when t is unbounded.

For perfect zero-knowledge we have to show that when $x \in L$, then the view of any verifier V' can be simulated by a probabilistic expected polynomial time Turing machine $M_{V'}$, with an identical distribution. We use the technique of 'probing and resetting' the verifier.[25] In our case $M_{V'}$ first prepares 'transcripts' of the form $(z = f_n(r) \cdot I^q, q, y = r)$, where $r \in_R G$ and $q \in_R Z_v$, and then obtains the appropriate distribution by probing and resetting V'. The simulator can do this because the expected number of probings is $O(tv)$, which is polynomial in $|x|$. ■

We remark that if t is linear in $|n|$ then we can do without condition C1 in Theorem 1.

Corollary 1. *Suppose that* v *is as in Theorem 1, and that there exists an integer* $k = k(n)$ *such that*

C2 $H^k \subset f_n(G)$: *that is, for any* $I \in H$ *there exists a* $w \in G$ *such that* $I^k = f_n(w)$,

C3 *The smallest prime factor of* k *is at least* v.

Then protocol (P, V) *is a perfect zero-knowledge interactive proof for* L *provided that test* T *checks these conditions.*

Proof. We only need to show that conditions C2 and C3 imply C1 of Theorem 1. Suppose that $I^q = f_n(u), I \in H, q \in Z_v^0, u \in G$. Then since $0 < q < v$ and since the smallest prime factor of k is at least v, we must have $gcd(q,k) = 1$. So there exist integers a, d with $aq + dk = 1$. Consequently $I = I^{aq+dk} = I^{aq} \cdot I^{dk} = f_n(u)^a \cdot f_n(w)^d = f_n(u^a w^d) \in f_n(G)$, since $I^k = f_n(w)$ for some $w \in G$ by condition C2. So $I \in f_n(G)$ and we get condition C1. ■

Examples. We will consider two particular cases for which condition C2 is trivially satisfied. In the first case, $G_n = H_n = Z_m^*$ and $f_n(r) : r \rightarrow r^k$, with $m = m(n), k = k(n)$. Then $H_n^k = (Z_m^*)^k = f_n(G_n)$. This case has been discussed elsewhere.[8] The second case is the one considered in Section 2. For this protocol we have $H_n^k = \{1\}$.

Corollary 2. *The discrete logarithm decision problem is polynomial-time reducible* (*efficiently*) *to the language in Section 2. This result can be generalised to the* (*relaxed*) *discrete logarithm over any Abelian group with known order. This implies that after suitable preprocessing*[16] *the zero-knowledge proof of Section 2 recognises the elements of any subgroup of* Z_p^*.

Proof. Our argument is based on smooth numbers[12] and the Pohlig-Hellmann algorithm.[39] Let the input be $(I; \beta, p)$. The verifier first checks that $I^{k_p} \in \langle \beta^{k_p} \rangle$. If this is successful, the prover and verifier use the protocol of Section 2 with input $(I^{(p-1)/k_p}; \beta^{(p-1)/k_p}, p)$. ■

6. PROOFS OF KNOWLEDGE

Definitions

We refer the reader to the discussion at the end of Section 2 for an example of a proof of knowledge. The only difference from a proof of membership is that with proofs of knowledge a dishonest prover who does not know the secret (or an equivalent) should fail to convince the verifier. The problem is to define the meaning of 'knowledge', allowing for knowledge one is not aware of. The solution[43,21] is to employ an 'outsider', the *extractor*, who succeeds in extracting from the prover (a machine) the secret. The extractor will fail if the prover does not know the secret.

The formal setting for proofs of knowledge[21] requires that both the prover P and the verifier V are polynomially bounded Turing machines, and the prover has access to a restricted oracle (from which he obtains the secret).

Definition 3. (P, V) is an *interactive proof of knowledge*[21] *of the relation* R if for common input x,

- *Completeness*: (P, V) will accept $x = (I; n)$ with overwhelming probability when $(x, s) \in R$.
- *Soundness*: there is a probabilistic polynomial time Turing machine M (the *extractor*) which is such that, given any P' and any string h: if (P', V) accepts x with non-negligible probability then M will output an $s' = M(P', h, x)$ which is such that $(x, s') \in R$ with overwhelming probability.

Definition 4. Zero-knowledge is as for proofs of languages. We have *unrestricted input* zero-knowledge[21] if the simulator can simulate the view of V' *even when* $x \notin L$, where $L = \{x | \exists s : (x, s) \in R\}$.

The zero-knowledge proof of knowledge

For proofs of knowledge the prover P is polynomially bounded. So we have to assume, additionally, that

- the operation in G_n can be executed in polynomial time (otherwise P cannot compute y in Step 3),
- for all $y \in G_n, y^{-1}$ can be computed in polynomial time.

With such proofs, P proves that it knows an s such that $I \cdot f_n(s) = 1$ (and not merely that such an s exists). So condition C1 of Theorem 1 must be strengthened.

Theorem 2. *Suppose that* $x = (I; n)$, *that* v *is as in Theorem 1 and that,*

C4 *There exists a probabilistic polynomial time algorithm which for any* $n \in J$, *when given as input* n, $I \in H_n, q \in Z_v^0, u \in G_n$ *with* $I^q = f_n(u)$, *will output an* $w \in G_n$ *such that* $I = f_n(w)$.

Then the protocol (P, V) *is a perfect zero-knowledge proof of knowledge of the relation*

$$R = \{(x, s) | n \in J; s \in G_n; I \cdot f_n(s) = 1\}.$$

Furthermore (P, V) *is unrestricted input zero-knowledge if test* T *checks that* $I \in f_n(G_n)$.

Proof. The proof of completeness and zero-knowledge are the same as for Theorem 1. The proof of soundness is similar to that in Feige–Fiat–Shamir when one defines vertices of degree 2, or more, as 'heavy'.[21] We assume that (a portion of) the random tape of a (possibly cheating) prover P' is such that the verifier V accepts with non-negligible probability when the input is x, and show that there is a probabilistic polynomial time machine M which will obtain a witness $s \in G$ such that $I \cdot f_n(s) = 1$. As in Feige–Fiat–Shamir,[21] by probing and resetting the prover P', it is possible for M to find in expected polynomial time a 'heavy' vertex (of degree 2 or more), and hence to obtain a $z \in H, y', y'' \in G$, and numbers $q', q'' \in Z_v, q' > q''$, such that $z = f_n(y') I^{q'} = f_n(y'') I^{q''}$. So M can obtain $q = q' - q''$ and $u = (y')^{-1} y''$ such that $I^q = f_n(u)$. Then by using condition C4, M can compute a $w \in G$ such that $I = f_n(w)$, and thus an $s = w^{-1}$ such that $I \cdot f_n(s) = 1$. ■

When G_n and H_n are cyclic groups then condition C4 can be simplified.

Corollary 3. *Suppose that* G_n *and* H_n *are cyclic groups, that* $f_n : G_n \to H_n ; r \to \beta^r$, *that* β *is given, and that*

C5 *The order of* $f_n(G_n)$, *say* m, *is known, and test* T *checks that* $I^m = 1$.

Then condition C4 *is satisfied.*

Proof. Test T is equivalent to that in Theorem 2. Observe[31] that for any divisor d of the order of a cyclic group there is only one subgroup of order d. So for any $I \in H : I \in f_n(G)$, if and only if, $I^m = 1$. We shall now show that condition C5 implies condition C4 of Theorem 2. Suppose that $I^q = f_n(u) = \beta^u, I \in H, q \in Z_v^0, u \in G$. Since $I^m = 1$, we must have $I = f_n(w) = \beta^w$ for some $w \in G$. So $\beta^{qw} = \beta^u$. Let $m_0 = \gcd(q, m)$ with $m = m' m_0, q = q' m_0$, $u = u' m_0$. Then there exists[29] an integer a such that,

$$w = au' \bmod m' + cm', \quad \text{for some} \quad c \in Z_{m_0}. \tag{1}$$

Now $m_0 \leqslant q < v$, so c is polynomially bounded in $|x|$. Consequently it is possible to compute in polynomial time all the values of w in (1) and thus obtain the appropriate w. So we get condition C4. ■

Examples. Consider the case when $G_n = H_n = Z_N^*$ and $f_n(r) : r \to r^k$, with $N = N(n), k = k(n)$, and $\gcd(k, \phi(N)) = 1$ (a trusted centre which may know the factorisation of N must guarantee this). Then we only need condition C3 to get C4 of Theorem 2. So we have a zero-knowledge proof of knowledge if the smallest prime factor of k is at least v. Indeed suppose that we are given $I \in H, q \in Z_v^0, u \in G$ with $I^q \equiv u^k (\bmod N)$. Then by C3, $\gcd(q, k) = 1$, so that $aq + bk = 1$ for suitable a, b. So $I = I^{aq+bk} \equiv u^{ak} \cdot I^{bk} \equiv (u^a I^b)^k (\bmod N)$. Thus $I \equiv w^k (\bmod N)$ for $w \equiv u^a I^b (\bmod N)$, and we have condition C4.

For the second example we take $G_n = Z_{p-1}, H_n = Z_p^*$ and $f_n : r \to \beta^r, \beta \in Z_p^*$. The input is $(I; \beta, \gamma, p)$, were γ is the list of all prime factors of $(p-1)$. Note that it is easy to compute the order m of β, given γ. So we apply Corollary 3 to get a zero-knowledge proof of knowledge.

Remark. Let us compare Corollary 1 to Corollary 3. When H is cyclic, condition C2 of Corollary 1 is satisfied for $k = h/m$ (the *index* of $f_n(G)$ in H). Indeed in this case $H^k = f_n(G)$. However, in Corollary 3 we make no assumption on the size or the factors of k.

Corollary 3 can be extended to Abelian groups, with minor adjustments. We recall that finite Abelian groups have an independent set of generators[31] with invariant properties.

Corollary 4. *Suppose that* H_n *is an Abelian group, that a set of independent generators* $\beta_i, i = 1, \ldots, b$, *of* $f_n(G_n)$ *is given, that the orders* m_i *of* β_i *are given, that* $G_n = Z_{m_1}(+) \times \ldots \times Z_{m_b}(+)$(*direct product*), *and that* v *is as in Theorem 1. Let* $f_n : (u_1, \ldots, u_b) \to \Pi_{i-1}^b \beta_i^{u_i}$. *Suppose that the order of* H_n, *say* h, *is given and that* $m = \Pi_{i-1}^b m_i$ *is relative prime to* $k = h/m$. *Then protocol* (A, B) *is a perfect unrestricted input zero-knowledge proof of knowledge of the relation*

$$R = \{((I; n), (s_1, \ldots, s_b)) | n \in J; (s_1, \ldots, s_b) \in G_n; \\ I \cdot \beta_l^{s_1} \cdots \beta_b^{s_b} = 1\},$$

provided that test T *checks that* $I^m = 1$.

Proof. We show that $f_n(G) \in \mathbf{BPP}$ and that condition C4 of Theorem 2 applies. Observe[31] that H has only one subgroup which has order m since $\gcd(m,k) = 1$. Thus $I \in f_n(G)$, if and only if, $I^m = 1$. The rest is similar to Corollary 3. ■

This result holds for any set of independent generators β_i of $f_n(G_n)$. Corollary 4 can be modified to allow for the case when $\gcd(m,k) \neq 1$, provided that we impose a condition on the index k. We have

Corollary 5. *We get the same conclusion as in Corollary 4 if we replace the condition $k = h/m$ by the assumption that k satisfies conditions* C2 *and* C3 *of Corollary 1.*

Proof. We show that condition C4 is valid. If $I \in H$, $u = (u_1, \ldots, u_b) \in G$ and $q \in Z_v^0$ are such that,

$$I^q = f_n(u) = \beta_1^{u_1} \cdots \beta_b^{u_b}, \tag{2}$$

and if condition C2 holds, then it is possible to compute in polynomial time a $w = (w_1, \ldots, w_b) \in G$ such that $I = f_n(w) = \beta_1^{w_1} \cdots \beta_b^{w_b}$. Indeed by C2 there is a $(\bar{w}_1, \ldots, \bar{w}_b) \in G$ such that $I^k = \beta_1^{\bar{w}_1} \cdots \beta_b^{\bar{w}_b}$. Then by (2), $I^{qk} = \beta_1^{ku_1} \cdots \beta_p^{ku_q} = \beta_1^{q\bar{w}_1} \cdots \beta_p^{q\bar{w}_q}$. Since the β_i are independent, we must have $q\bar{w}_i \equiv ku_i (\operatorname{mod} m_i), i = 1, \ldots, b$. As in Corollary 3, we can compute in polynomial time the $\bar{w}_i$. We now proceed as in Corollary 1. In this case we have $I^{aq} = \beta_1^{au_1} \cdots \beta_b^{au_b}$, and hence $I = \beta_1^{au_1 + d\bar{w}_1} \cdots \beta_b^{au_b + d\bar{w}_b}$, where a, d are such that $aq + dk = 1$. ■

7. APPLICATIONS

Passports are an important means of identification. Compared with other identification tokens (or tools) such as credit cards, there are many centres, each one distrusting the other, that issue tokens.

An electronic passport scheme

An international body, say the U.N. (or I.S.O.), organises a setting in which various countries can issue unforgeable electronic passports.[13] It is assumed that the countries will not trust the U.N., or each other. Furthermore it should not be possible for citizens to make passports, or for one country to issue passports for another country.

Initiation. The U.N. chooses a large prime number p and a primitive element $\alpha \in Z_p^*$ (it is not necessary for α to be primitive, but its order must be superpolynomial in $|p|$). The U.N. guarantees that p, α are correct and makes these numbers public.

Registration of a country. Each participating country chooses a random $u \in Z_{p-1}$ and computes $\psi = \alpha^u$. The public number of the country is ψ, and the secret number is u.

Making a passport. For each citizen who applies for a passport, the country chooses a random $k \in Z_{p-1}^*$, such that the smallest prime factor of k is at least $v, v = \lceil \log p \rceil$. Then it computes $\beta = \alpha^k \operatorname{mod} p$, and the secret number s by solving the congruence[20]

$$u\beta + ks \equiv J \quad (\operatorname{mod}(p-1)), \tag{3}$$

where J is the value of a string which identifies the citizen (name, etc.). The numbers $J, \beta = \alpha^k \operatorname{mod} p$ and $I = \beta^{-s} \operatorname{mod} p$, are the public numbers of the passport. These can be read using bar codes or may be given by the electronic passport. The country encapsules k and s in a *tamper-proof* electronic passport (modern smart cards are not very tamper-proof). k, s are the secret numbers of the passport.

Verification of a passport. To verify a passport $x = (I; J; \beta; \psi; \alpha; p)$, a verifier must first check that ψ, α, and p are as published, that J is not on the black list, and that

$$\alpha^J \cdot I \equiv \psi^\beta \quad (\operatorname{mod} p). \tag{4}$$

Then he must check that the secret numbers of the passport are k, s. For this purpose our protocol is used twice as a proof of existence (or knowledge) of a k *and* an s such that $\beta \equiv \alpha^k (\operatorname{mod} p)$ and $I \equiv \beta^{-s} (\operatorname{mod} p)$. Then $\alpha^J \cdot I \equiv \alpha^{J-ks} (\operatorname{mod} p)$ and $\psi^\beta \equiv \alpha^{u\beta} (\operatorname{mod} p)$, so that from (4) we see that k, s must satisfy (3).

A secure remote login scheme

In our introduction we mentioned that login was a special case of identification. The aspect of remoteness of login makes it possible for an active eavesdropper to divert the zero-knowledge proof to another verifier (remote machine). The scheme above can be extended to cover this case. To solve[17] this problem the prover signs[20] the name of the remote machine J, and keeps the resulting signature k, s secret. This time, u is the secret key of the prover and ψ is his public key. To login he proves knowledge of k, s to the remote machine.

8. CONCLUSION

Modern cryptography is being used to securely authenticate and sign messages. Many schemes rely on the difficulty of factoring or on the discrete logarithm. The use of cryptography has led to the design of more secure identification tokens. Zero-knowledge identification schemes have the advantage that the secret used to identify is guaranteed not to leak. A lot of research has concentrated on finding more efficient zero-knowledge schemes. In this paper we present new schemes for which the efficiency is far better than that of existing schemes. Although our examples have focused on schemes which are based on the discrete logarithm and on factoring, our setting is more general.

We have presented two practical identification schemes: a cryptographic passport scheme and a secure remote login scheme. These can be further enhanced by using smart cards.

Generalisations

In this paper we have assumed that $v = \lceil \log n \rceil$. We can extend all our results to the case when $|n|^a \leqslant v \leqslant |n|^b$, where a, b are constants, $0 < a < b$.

The input for our protocol $x = (I; n)$ has only one I. We can easily extend all our results to the case when there are l inputs $I_1, \ldots, I_l$, where l is a constant. The proof is then a proof of existence (or knowledge) of secrets s_j for the $I_j, j = 1, \ldots, l$.

Acknowledgement

The authors wish to thank an anonymous referee for helpful suggestions.

REFERENCES

1. L. M. Adleman and K. S. McCurley, Open problems in number theoretic complexity. *Discrete Algorithms and Complexity, Proceedings of the Japan–US Joint Seminar (Perspective in Computing series, Vol. 15)*, edited by D. Johnson, T. Nishizeki, A. Nozaki and H. Wilf, pp. 263–286. Academic Press, Orlando, Florida (1986).
2. L. Babai, Trading group theory for randomness. *Proceedings of the seventeenth annual ACM Symp. Theory of Computing, STOC.* pp. 421–429 (1985).
3. S. Bengio, G. Brassard, Y. Desmedt, C. Goutier and J.-J. Quisquater, Secure implementations of identification systems. Accepted for publication in the *Journal of Cryptology.*
4. T. Beth, A Fiat-Shamir-like authentication protocol for the El-Gamal-scheme. In *Advances in Cryptology – Eurocrypt 88, Lecture Notes in Computer Science 330*, edited by C. G. Günther, pp. 77–84. Springer-Verlag, Berlin (1988).
5. T. Beth and Y. Desmedt, Identification tokens – or: Solving the chess grandmaster problem. Presented at Crypto 90, August 12–15, 1990, Sant Barbara, California, U.S.A. To appear in: Proc. of Crypto 90, Lecture Notes in Computer Science, Springer-Verlag.
6. T. Beth and F. Schaefer, Non-Supersingular Elliptic Curves for Public Key Cryptosystems. Presented at Eurocrypt 91, April 8–11, 1991, Brighton, England. To appear in: Proc. of Eurocrypt 91, Lecture Notes in Computer Science, Springer-Verlag.
7. G. Brassard and P. Bratley, *Algorithmics – Theory & Practice.* Prentice Hall (1988).
8. M. V. D. Burmester, An optimal class of interactive zero-knowledge proofs. Submitted to Information Processing Letters, under revision.
9. M. V. D. Burmester, A remark on the efficiency of identification schemes. *Advances in Cryptology – Eurocrypt 90, Lecture Notes in Computer Science 473*, edited by I. B. Damgard, pp. 493–495. Springer-Verlag, Berlin (1991).
10. M. V. D. Burmester and Y. G. Desmedt, Remarks on the soundness of proofs. *Electronics Letters*, **25** (22), pp. 1509–1511 (1989).
11. D. Chaum, J.-H. Evertse and J. van de Graaf, An improved protocol for demonstrating possession of discrete logarithms and some generalizations. *Advances in Cryptology – Eurocrypt 87, Lecture Notes in Computer Science 304*, edited by D. Chaum and W. L. Price, pp. 127–141. Springer-Verlag, Berlin (1988).
12. D. Coppersmith, A. Odlyzko and R. Schroeppel, Discrete logarithms in *GF*(*p*). *Algorithmica*, pp. 1–15 (1986).
13. G. Davida and Y. Desmedt, Passports and visas versus IDs. *Advances in Cryptology – Eurocrypt 88, Lecture Notes in Computer Science 330*, edited by C. G. Günther, pp. 183–188. Springer-Verlag, Berlin (1988).
14. D. W. Davies and W. L. Price. *Security for Computer Networks.* John Wiley and Sons, New York (1984).
15. D. W. Davies and W. L. Price. *Security for Computer Networks.* John Wiley and Sons, New York, second edition (1989).
16. Y. Desmedt and M. Burmester, An efficient zero-knowledge scheme for the discrete logarithm based on smooth numbers. To be presented at Asiacrypt 91, November 11–14, Fujiyoshida, Yamanashi, Japan. To appear in: Proc. of Asiacrypt 91, Lecture Notes in Computer Science, Springer-Verlag.
17. Y. Desmedt, Major security problems with the "unforgeable" (Feige–)Fiat–Shamir proofs of identity and how to overcome them. *Securicom 88, 6th worldwide congress on computer and communications security and protection*, SEDEP Paris, France, pp. 147–159 (1988).
18. Y. Desmedt, C. Goutier and S. Bengio, Special uses and abuses of the Fiat–Shamir passport protocol. *Advances in Cryptology – Crypto 87, Lecture Notes in Computer Science 293*, edited by C. Pomerance, Springer-Verlag, Berlin, pp. 21-39 (1988).
19. W. Diffie and M. E. Hellman, New directions in cryptography. *IEEE Trans. Inform. Theory* **IT-22** (6), 644–654 (1976).
20. T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inform. Theory* **31**, 469–472 (1985).
21. U. Feige, A. Fiat and A. Shamir, Zero knowledge proofs of identity. *Journal of Cryptology*, **1**(2), 77–94 (1988).
22. A. Fiat and A. Shamir, How to prove yourself: Practical solutions to identification and signature problems. *Advances in Cryptology – Crypto 86, Lecture Notes in Computer Science 263*, edited by A. Odlyzko, Springer-Verlag, Berlin, pp. 186–194 (1987).
23. J. Gill, Computational complexity of probabilistic Turing machines. *Siam J. Comput.*, **6** (4), 675–695 (1977).
24. O. Goldreich and H. Krawczyk. On the composition of zero-knowledge proof systems. *Proceedings of 17th International Colloquium on Automata, Languages and Programming (ICALP), Lecture Notes in Computer Science 443*, Springer-Verlag, Berlin, pp. 268–282 (1990).
25. S. Goldwasser, S. Micali and C. Rackoff, The knowledge complexity of interactive proof systems. *Siam J. Comput.*, **18** (1), 186–208 (1989).
26. R. Graham, D. E. Knuth and O. Patashnik, *Concrete Mathematics – A foundation for computer science.* Addison-Wesley, Reading, MA, (1989).
27. L. Guillou, Data encipherement: Techniques, standards and applications, *Proc. la Carte à Mémoire*, pp. 17–23. Paris, France (1983).
28. L. C. Guillou and J.-J. Quisquater, A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. *Advances in Cryptology – Eurocrypt 88, Lecture Notes in Computer Science 330*, edited by C. G. Günther, pp. 123–128. Springer-Verlag, Berlin (1988).
29. Hua, *Introduction to Number Theory.* Springer, New York (1982).
30. T. Itoh and K. Sakurai, On the complexity of constant round zkip of possession of information. Manuscript (1990).
31. N. Jacobson, *Basic Algebra I.* W. H. Freeman and Company, New York (1985).
32. N. Koblitz, Elliptic curve cryptosystems. *Mathematics of Computation*, **48**, 203–209 (1987).
33. A. Konheim, *Cryptography: A Primer.* John Wiley, Toronto (1981).
34. J. L. Massey and J. K. Omura, A New Multiplicative Algorithm over Finite Fields and its Applicability in Public-Key Cryptography. Presented at Eurocrypt 83, Udine, Italy.
35. P. D. Merillat, Secure stand-alone positive personnel identity verification system (ssa-ppiv). Technical Report SAND79-0070, Sandia National Laboratories (1979).
36. V. Miller, Use of elliptic curves in cryptography. *Advances in Cryptology – Crypto 85, Lecture Notes in Computer Science 218*, edited by Hugh C. Williams, pp. 417–426. Springer-Verlag (1986).
37. A. M. Odlyzko, Discrete logs in a finite field and their cryptographic significance. *Advances in Cryptology – Eurocrypt 84, Lecture Notes in Computer Science 209*, edited by T. Beth, N. Cot and I. Ingemarsson, pp. 224–314. Springer-Verlag, Berlin (1984).
38. K. Ohta and T. Okamoto, A modification of the Fiat–Shamir scheme. *Advances in Cryptology – Crypto 88,*

Lecture Notes in Computer Science 403, edited by S. Goldwasser, pp. 232–243. Springer-Verlag (1990).

39. S. C. Pohlig and M. E. Hellman, An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance. *IEEE Trans. Inform. Theory*, **IT-24** (1), 106–110 (1978).
40. C. P. Schnorr, Efficient identification and signatures for smart cards. *Advances in Cryptology – Crypto 89, Lecture Notes in Computer Science 435*, edited by G. Brassard, pp. 239–252. Springer-Verlag (1990).
41. G. J. Simmons, A system for verifying user identity and authorization at the point-of-sale or access. *Cryptologia*, **8** (1), 1–21 (1984).
42. R. Solovay and V. Strassen, A fast Monte-Carlo test for primality. *SIAM Journal on Computing*, **6** (1), 84–85, erratum (1978), ibid, **7**, 118 (1977).
43. M. Tompa and H. Woll, Random self-reducibility and zero-knowledge interactive proofs of possession of information. *The Computer Society of IEEE, 28th Annual Symp. on Foundations of Computer Science (FOCS)*, pp. 472–482, IEEE Computer Society Press (1987).

Book Review

BRIAN SHACKEL and SIMON RICHARDSON (eds), *Human Factors for Informatics Usability*, Cambridge University Press, 1991. £35, ISBN 0-521-36570-8.

This book is the outcome of an Advanced Study Course held in December 1986. It is divided into five parts: Informatics Usability – introduction, scope and importance; System Design – orientation and approaches; Special topics in depth; Organisational aspects and design in large systems; and Design and Evaluation – some specific methods. In the preface Shackel explains the main objectives: to review the knowledge and methods available from the field of human factors to help improve the usability of informatics systems; to present recent theoretical and methodological developments in this field; to stimulate increased application of this knowledge and these methods.

In the first part Brian Shackel and Simon Richardson give a background to human factors and usability, and in the following chapters Shackel goes into more detail about usability, its design and evaluation. In chapter 3, Alphonse Chapanis reviews many studies of human factors issues in real world situations and attempts to evaluate the cost savings involved in good human factors designs.

Part 1 commences with Ken Eason and Susan Harker discussing the needs of a broad range of users and how human factors can help if they are incorporated in the design process. Tom Stewart's chapter again looks at these issues but from a consultant's perspective, and William Newman's lends yet another voice from the perspective of the system designer. Arthur Gardner describes an attempt to classify different sources of existing advice from the literature on human factors into an overall framework, and locate them within the system life cycle.

Phil Barnard starts the section on special topics with a discussion of the contribution of cognitive psychology. This is another attempt to establish a much-needed framework within which to place aspects of usability. Formal models are discussed by Jurgen Ziegler and Hans-Jorg Bullinger; and they finish with a table showing where each method has been used, and conclude that no one method is universally applicable to all situations. The following chapter by Brian Gaines applies familiar human–computer interface design principles to expert systems which generate new modes of interaction between people and computers.

Part 4, with chapters by Greif, Mumford and Damodaran, emphasises organisational aspects of design. In particular Siegfried Greif discusses the dimensions which must be considered: size, centralisation versus decentralisation, the reward and feedback system, and division of labour or specialisation. Enid Mumford looks at participation in design and describes the ETHICS methodology which deals with organisational factors in design; and the human factors strategy, a holistic approach to IT from the users organisation's perspective, is elaborated in the chapter by Leela Damodaran.

The last part discusses some of the more basic interface issues. Ben Shneiderman shows the various styles of interface dialogues and Patricia Wright explains the reasons for the prevalence of poor documentation, and methods by which it can be improved. In the final chapter, Chapanis argues that lack of evaluation is why some computers, computer programs and manuals are hard to use. He gives guidelines for evaluation and cites numerous relevant examples.

Overall, the book brings together an interesting, well-written and worthwhile collection of papers, and an extensive collection of references; my only criticism is the length of time taken to achieve this status. As Shackel points out, much of the material is as timely now as it was five years ago when the papers were written. However, it is clear, especially from the screens in some chapters, that the material is somewhat dated. Indeed, much has happened on the human factors front in the last few years, as is evident from CHI and similar conferences.

This is a solid and useful book for anyone interested in interface design issues including undergraduate and postgraduate students. Although more recent material is missing, what is in this book is valuable, and inevitably something had to be omitted.

MILDRED L. G. SHAW
University of Calgary