CRYPTOGRAPHIC METHODS FOR DEMONSTRATING SATISFIABLE FORMULAS FROM PROPOSITIONAL LOGIC

INVENTOR

Stefanus Alfonsus Brands.

BACKGROUND OF THE INVENTION

1. Field of the invention.

This invention relates to secure cryptographic methods for demonstrating characteristics of privately held data, and more specifically to methods for demonstrating formulas from propositional logic without revealing additional information.

2. Description of the prior art.

US Patent application Ser. No. 08/203,231, filed February 28, 1994, by the present applicant, describes and claims apparatus for demonstrating characteristics of privately held data without revealing additional information. In particular it describes how a prover party, holding one or more secrets, can securely demonstrate a relation that is linear in the secrets. A special case of demonstrating a linear relation pertains to demonstrating that a secret has a certain value.

In one application the privately held data represents a set of credentials that have been issued by some authorized party, and the prover party demonstrates that its credentials have a particular value or that a linear relation applies to the credentials, without revealing additional information. By denoting the absence of a credential by a specially appointed value, typically zero, and its presence by another, typically one, the prover party can demonstrate in this manner that it has or does not have that credential, by demonstrating that the secret representing the credential is one or zero, respectively. Similarly, it can demonstrate possession of exactly one of two such credentials, by demonstrating that the sum of the respective secrets equals 1. In general, in this manner possession of exactly $l \geq 0$ out of $k \geq l$ credentials can be demonstrated.

While the credential issuing and updating methods described in patent application Ser. No. 08/203,231 are quite powerful, the method for demonstrating linear relations is seriously limited in functionality.

To demonstrate a system consisting of more than just a single linear relation, it would seem that the method described in patent application Ser. No. 08/203,231 must be applied repeatedly, once for each relation. It would be desirable to have a method that allows any system of linear relations to be demonstrated by the prover party, without revealing additional information, and without significantly degrading efficiency.

The methods of patent application Ser. No. 08/203,231 also cannot be applied to demonstrate that a credential that can take on many values does not have a certain value, without revealing information about its actual value.

The method for demonstrating possession of exactly l out of k credentials only applies to yes/no credentials, and does not allow one to demonstrate that one has at least l out of k credentials.

More generally, the methods in patent application Ser. No. 08/203,231 do not allow to demonstrate arbitrary satisfiable formulas from so-called propositional logic for linear relations; these are formulas that connect atomic propositions by the logical connectives "AND," "OR," and "NOT," the atomic propositions being linear relations over a finite field. "Satisfiable" in this context means that there exists an assignment to the variables in the atomic propositions, such that the formula becomes true. An example of such a formula is "(S_1 AND S_2) OR (S_3 AND (NOT S_4))," where each of $S_1, \ldots, S_4$ denotes one linear relation involving numbers $x_1, \ldots, x_k$; the formula is satisfiable if there is an assignment to $(x_1, \ldots, x_k)$ such that the formula becomes true. In this interpretation, the methods in patent application Ser. No. 08/203,231 only allow to demonstrate formulas that connect linear relations by zero logical connectives; it does not allow demonstration of formulas that have one or more logical connectives.

A method is known in the art (see, Cramer, R., Damgard, I., and Schoenmakers, B., "Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols," Advances in Cryptology - CRYPTO '94, Lecture Notes in Computer Science, Springer-Verlag, 1995) that allows the prover party to demonstrate, for example, that it knows secret keys corresponding to a second and a third public key, or to a fifth public key, without revealing more than that. This method, however, only allows to demonstrate monotone formulas, which are formulas with only "AND" and

"OR" connectives: the "NOT" connective cannot be dealt with. More importantly, the atomic propositions for this method are of the form "The prover party knows a secret key corresponding to the i-th public key," instead of linear relations between secrets.

US Patent application Ser. No. 08/203,231 also describes a method for enabling the set of credentials, or part thereof, to be computed by the verifier party in case the prover party demonstrates the exact same linear relation more than once (or, more generally, more times than a predetermined limit). This method requires the prover party to know beforehand which particular linear relation it will have to demonstrate, which is often not desirable. Furthermore it would be desirable to have a method whereby the verifier party is able to compute the set of credentials, or a part thereof, in case the prover party demonstrates any two formulas from propositional logic (i.e., not necessarily the same) for the set of credentials, or any two formulas out of a special category of all such formulas.

Furthermore, the methods in US Patent application Ser. No. 08/203,231 do not make any other distinction than between zero-knowledge proofs and signed proofs of formulas. They do not distinguish between how much of the verifier party's conviction can be passed on in case a signed proof was provided, in other words between different flavors of signed proof. In particular, in some cases it may be desirable that the verifier party can pass on its conviction that the prover party knows a set of secrets, but not any information about the formula that has been demonstrated by the prover party; in other cases the verifier party may need to be able to pass on its conviction that the prover party knows a set of secrets, and that the prover party has demonstrated one formula out of a subclass of formulas, without convincing of which one; and in still other cases the verifier party may need to be able to pass on its conviction that the prover party knows a set of secrets, as well as its conviction of the particular formula that the prover party has demonstrated.

The method described in patent application Ser. No. 08/203,231 for demonstrating linear relations moreover requires the prover party to demonstrate knowledge of a representation with respect to a set of numbers. This set of numbers depends on the particular relation that has to be demonstrated. It would be desirable to have a method for which the set of numbers is always the same, regardless of the formula that has to be demonstrated. This would allow efficient use of the well-known

simultaneous repeated squaring technique (see. for example, exercise 27 on page 465 of Knuth, D., "The Art of Computer Programming," Volume 2 / Seminumerical Algorithms, second edition, Addison-Wesley Publishing Company) because only a single table would be needed, containing for each non-empty subset of the set of numbers the product of the numbers in the subset, which could easily be pre-computed once and then stored.

OBJECTS OF THE INVENTION

Accordingly, it is an object of the present invention to:

allow a prover party to demonstrate any system of linear relations pertaining to a set of privately held secrets, without revealing additional information about the secrets;

allow a prover party to demonstrate that a linear relation does not pertain to a set of privately held secrets, without revealing additional information about the secrets;

allow a prover party to demonstrate that a set of privately held secrets satisfies at least some of a specified set of linear relations, without revealing additional information about the secrets;

allow a prover party to demonstrate arbitrary satisfiable formulas from propositional logic, atomic propositions being linear relations pertaining to a set of privately held secrets of the prover party, without revealing additional information about the secrets;

enable a verifier party to compute the set of privately held secrets of a prover party, or some of these secrets, if and only if the prover party engages more than a predetermined number of times for its set of secrets in the demonstration of a formula from a predetermined category of satisfiable formulas from propositional logic, atomic propositions being linear relations pertaining to the secrets;

allow, in any one of the above objects, the verifier party to restrictively blind the demonstration performed by the prover;

allow, in any one of the above objects, the verifier party to fully blind the demonstration performed by the prover;

allow, in any of the above objects, the prover party to perform its demonstration as a zero-knowledge proof;

allow, in any one of the first seven of the above objects, the prover party to perform its demonstration as a signed proof, in such a manner that the verifier party can subsequently pass on its conviction that the prover party

knows a set of secrets, and its conviction of the formula that has been demonstrated by the prover party;

allow, in any one of the first seven of the above objects, the prover party to perform its demonstration as a signed proof, in such a manner that the verifier party can subsequently pass on its conviction that the prover party knows a set of privately held secrets, but not of the formula that has been demonstrated by the prover party;

allow, in any one of the first seven of the above objects, the prover party to perform its demonstration as a signed proof, in such a manner that the verifier party can subsequently pass on its conviction that the prover party knows a set of privately held secrets and that the prover party has demonstrated one formula out of a restricted class of formulas, without convincing of which particular formula;

allow, in any one of the three foregoing objects, the prover party to perform its signed proof without interacting with the verifier party;

allow, in any of the above objects, the prover party to be implemented by two sub-parties that have mutually conflicting interests, in such a manner that each sub-party knows only some of the secrets of the prover party;

allow efficient computation for the prover party and the verifier party in the above objects, by guaranteeing that simultaneous repeated squaring can be applied using a single pre-computed table;

enable use of the above objects of the invention in privacy-protecting credential mechanisms;

allow efficient, economical, and practical apparatus and methods fulfilling the other objects of the invention.

Other features, objects, and advantages of this invention will be appreciated when the description and appended claims are read in conjunction with the figures.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a flowchart of a first protocol for demonstrating that a system of linear relations pertains to a set of secrets, without revealing additional information about the secrets, in accordance with the teachings of the present invention.

FIG. 2 shows a flowchart of a second protocol for demonstrating that a system of linear relations pertains to a set of secrets, without revealing

additional information about the secrets, in accordance with the teachings of the present invention.

FIG. 3 shows a flowchart of a protocol for demonstrating that a linear relation does not hold for a set of secrets, without revealing additional information about the secrets, in accordance with the teachings of the present invention.

FIG. 4 shows a flowchart of a protocol for demonstrating that at least one of two linear relations pertains to a set of secrets, without revealing additional information about the secrets, in accordance with the teachings of the present invention.

FIG. 5 shows a flowchart of a protocol for demonstrating that a system of linear relations pertains to a set of secrets, revealing additional information about the secrets if and only if performed more than once, in accordance with the teachings of the present invention.

FIG. 6 shows a flowchart of a protocol for demonstrating a satisfiable formula that connects linear relations, pertaining to a set of secrets, by zero or more "AND" connectives and at most one "NOT" connective, revealing additional information about the secrets if and only if performed more than once, in accordance with the teachings of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

While it is believed that the notation of FIGS. 1 to 6 would be clear to those of ordinary skill in the art, it is first reviewed here for definiteness.

The figures describe protocols between a prover party and a verifier party, denoted in the figures and in the accompanying descriptions by $\mathcal{P}$ and $\mathcal{V}$, respectively. The actions performed by the parties are grouped together into flowchart boxes. The party performing the actions described in a flowchart box is indicated at the top of the column that the box is in. Each line in a box describes a step, such as a computation, a transmittal or a storage action. Flowcharts must be read box by box, in accordance with the attached numbers, and, within boxes, line by line. The term "party" indicates an entity, and might sometimes be regarded as an agent who performs a step or a collection of steps in a protocol. It might also be regarded as a means for performing those steps, and might be comprised of any suitable configuration of digital logic circuitry. For example, any box or collection of boxes from the figures could be realized by hard-wired and dedicated combinatorial logic, or by some sort of suitably programmed machine, a microprocessor for instance, such as are well-known in the art,

just as long as it is able to perform the storage, input/output and transformational steps (possibly apart from the random source functions) described by the corresponding box or boxes.

The symbol "$\leftarrow$" denotes assignment, meaning that the variable or symbol on its left-hand side is assigned the value or values on its right-hand side to. Assignments do not necessarily imply the reservation of storage space, but can refer to intermediate computations performed in random-access memory.

A test for equality is indicated by the symbol "$\stackrel{?}{=}$." Unless specified otherwise, the present invention is not concerned with what should happen in case an indicated equality does not hold. For definiteness the protocol can be assumed to halt in such a case.

The symbol "$\in_R$" indicates that the number or numbers on its left-hand side are chosen from the set on its right-hand side according to a probability distribution that is substantially uniform and independent. In practice a physical random number generator can be used, possibly in conjunction with additional post-processing, or a deterministic pseudo-random number generator. In the latter case all random numbers generated by a device may be derived from a single secret, for example by diversifying the secret by a sequence number and/or type specifier indicative of the random number to be generated.

The word "Send," followed by a colon and at least one number, indicates that the at least one number is sent by the party performing the actions, described in the corresponding box, to the other party, indicated by the connections between the flowchart boxes.

For any integer, l, the symbol $\mathbb{Z}_l$ denotes the set of of numbers $\{0, \ldots, l-1\}$, called the ring of integers modulo l. (Of course, any other set of representatives can be taken instead, as will be clear to those of ordinary skill in the art.) Addition and multiplication of elements in $\mathbb{Z}_l$ are defined modulo l.

As in standard mathematical notation, the symbols "$\prod$" and "$\sum$" denote product and sum, respectively, and they are always indexed. For example, $\prod_{i=2}^{4} x_i$ denotes the product $x_2 x_3 x_4$.

The standard (horizontal, vertical or diagonal) ellipsis notation from mathematics is used in cases where one or more numbers are formed according to the same rule, which can be logically inferred from the information preceding and succeeding the ellipsis. For example, the notation in the first three lines of Box **13** is to be interpreted as follows: if

$k-l$ equals 1, then only a single relation is defined, namely that shown for r_1; if $k-l$ is greater than 1, then r_2 is computed as $r_2 \leftarrow cx_{\pi(2)} + w_2 \bmod q$ and so on; and in the special case that $k-l$ is zero, no relation is defined. Likewise, the notation in the fourth line of Box **13** is to be interpreted as follows: if $k-l$ equals 1, then only a single number is sent, namely r_1; if $k-l$ is greater than 1, the numbers r_1 up to and including r_{k-l} are sent; and in the special case that $k-l$ is zero, nothing is sent.

FIGS. 1 to 6 describe preferred embodiments based on the so-called Discrete Log assumption, well-known in the art. Initially a group is determined, denoted here by G_q, that contains q numbers, where q is a large prime number. Without loss of generality, the group operation in G_q is assumed to be multiplication. Efficient algorithms should be available for recognizing, generating, testing equivalence of, and multiplying numbers in G_q, but not for computing discrete logarithms. An expression involving a number in G_q and no explicit modulo operator indicates a computation in G_q. For computations in the ring $\mathbb{Z}_q$, the modulo operator is denoted explicitly.

Additionally, $k \geq 2$ so-called generators, $g_1, \ldots, g_k$, all in G_q and unequal to 1, are selected. Using the terminology of US patent application Ser. No. 08/203,231, a representation of a number, h, in G_q, with respect to $(g_1, \ldots, g_k)$ is a vector of numbers, $(x_1, \ldots, x_k)$, all in $\mathbb{Z}_q$, such that h is equal to $\prod_{i=1}^{k} g_i^{x_i}$. It is important that at least $\mathcal{P}$ cannot feasibly compute a representation of 1 with respect to $(g_1, \ldots, g_k)$ that differs from the all-zero representation $(0, \ldots, 0)$. Hereto the generators, $g_1, \ldots, g_k$, can, for example, be generated at random by a party trusted by $\mathcal{V}$, or by $\mathcal{V}$ itself; this party may also specify the group G_q. As will be clear to those of ordinary skill in the art, it may not be needed to choose all generators in a substantially random manner; one of the generators may be chosen from a small set, for example amongst the first hundred prime numbers, as long as it is believed infeasible to compute discrete logarithms with respect to that generator.

The vector $(x_1, \ldots, x_k)$ is from now on referred to as the set of secrets of $\mathcal{P}$. The inventive methods enable the $\mathcal{P}$ to demonstrate properties of its set of secrets, the properties being described by satisfiable formulas from propositional logic, atomic propositions being linear equations in the secrets.

The inventive methods require $\mathcal{P}$ to demonstrate knowledge of a

representation of a number with respect to a vector of generators, the number and each generator being a product of powers of generators in the set $\{h, g_1, \ldots, g_k\}$, and the particular definition of products specifying the formula to be demonstrated. To demonstrate knowledge of a representation without revealing (useful) information about it, the protocol described by Figure 18 of US patent application Ser. No. 08/203,231 is used throughout; from now on it is referred to as "the basic proof of knowledge." This particular choice for proving knowledge of a representation is believed to be preferred in most applications. However, as will be clear to those of ordinary skill in the art, the inventive techniques can be implemented based on any other proof of knowledge of a representation that has the well-known characteristics of being complete and sound. and that is (believed to be) witness hiding; some such alternatives may be simple variations, perhaps arrived at by interchanging some or all of the exponents (such as the position of the challenges and the responses) in the verification relation for the verifier party, while others may depend on entirely different approaches.

It is well-known in the art that one can rapidly evaluate a product of powers, such as $g_1^{x_1} \cdots g_k^{x_k}$, by simultaneous repeated squaring. For efficiency it is preferable that, for all $2^k - 1$ non-empty subsets of $\{g_1, \ldots, g_k\}$, the product of the g_i's in the subset is pre-computed and stored in a table. For example if k equals 14, and g_i's are represented by 512-bit numbers, this table takes up almost one megabyte of storage space; well within reach of currently available personal computers (that can load the entire table into RAM). Regardless of the choice of k, the number of modular multiplications in G_q needed to evaluate the product of powers then never needs to exceed twice the binary length of q. Also known in the art are ways to trade off between the storage space for the table and the number of required modular multiplications (for example by considering groups of exponent bits instead of single bits), and ways to further improve computational efficiency.

The choice of describing embodiments related to the difficulty of computing discrete logarithms is only intended to be suggestive, and not limiting in any way. The inventive techniques can also be implemented based on the (presumed) difficulty of computing so-called RSA-roots. Hereto a set of secrets, $(x_1, \ldots, x_k)$, consists of numbers in the ring $\mathbb{Z}_v$, where v is a large prime number. Computations are performed in a group $\mathbb{Z}_n^*$, where n is the product of (at least) two large distinct primes that have been generated at random and are unknown at least to $\mathcal{P}$, and $k \geq 1$

numbers, $Y_1, \ldots, Y_k$, all in $\mathbb{Z}_n^*$ and preferably of large order, are decided on. As in US patent application Ser. No. 08/203.231, a vector $(x_1, \ldots, x_k, x_{k+1})$, with $x_1, \ldots, x_k \in \mathbb{Z}_v$ and $x_{n+1} \in \mathbb{Z}_n^*$, is called a representation of a number h in $\mathbb{Z}_n^*$, with respect to $(Y_1, \ldots, Y_k, v)$, if h is equal to $\prod_{i=1}^{k} Y_i^{x_i} x_{k+1}^v \bmod n$. The choice of n and $(Y_1, \ldots, Y_k, v)$ must be such that it is believed infeasible for at least $\mathcal{P}$ to compute a representation of 1, with respect to $(Y_1, \ldots, Y_k, v)$, that differs from the "trivial" representation $(0, \ldots, 0, 1)$. This requirement is easy to meet: a party that knows the factorization of n can easily generate at random numbers of order $\lambda(n)$ (the highest possible order in $\mathbb{Z}_n^*$). Alternatively, if n is the product of two hard primes, well-known in the art, then a random choice in Z_n^* has high order with overwhelming probability. Of course, any other suitable method may be used as well. To demonstrate knowledge of a representation without revealing (useful) information about it, the basic proof of knowledge described by Figure 5 of US patent application Ser. No. 08/203,231 can be used. Each detailed description of a figure is followed by a description of how to embody the inventive technique based on the difficulty of computing RSA roots. The descriptions for the RSA case detail only the changes that need to be made to the Discrete Log-based protocols, rather then spelling out the entire RSA-based protocols; those of ordinary skill in the art are believed to be able to straightforwardly do so themselves, by studying the provided descriptions.

Turning now to FIG. 1, a flowchart of a first protocol for demonstrating that a system of linear relations applies to a set of secrets, without revealing additional information about the secrets, will now be described in detail.

At the outset, $\mathcal{P}$ knows a representation $(x_1, \ldots, x_k)$ of a number h in G_q with respect to $(g_1, \ldots, g_k)$; this representation is its set of secrets. The number, h, is a commitment of $\mathcal{P}$ on its set of secrets. Without loss of generality, it is assumed that $\mathcal{P}$ has to demonstrate to $\mathcal{V}$ that its secrets satisfy the following $l \geq 1$ independent linear relations:

$$\begin{aligned}
b_1 &= \alpha_{11} x_{\pi(1)} + \cdots + \alpha_{1,k-l} x_{\pi(k-l)} + x_{\pi(k-l+1)} \bmod q \\
b_2 &= \alpha_{21} x_{\pi(1)} + \cdots + \alpha_{2,k-l} x_{\pi(k-l)} + x_{\pi(k-l+2)} \bmod q \\
&\vdots \quad\quad\quad\quad \vdots \\
b_l &= \alpha_{l1} x_{\pi(1)} + \cdots + \alpha_{l,k-l} x_{\pi(k-l)} + x_{\pi(k)} \bmod q,
\end{aligned}$$

or, in matrix form,

$$\begin{pmatrix} \alpha_{11} & \dots & \alpha_{1,k-l} & 1 & 0 & \dots & 0 \\ \vdots & & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & & \vdots & \vdots & \ddots & \ddots & 0 \\ \alpha_{l1} & \dots & \alpha_{l,k-l} & 0 & \dots & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{\pi(1)} \\ \vdots \\ \vdots \\ x_{\pi(k)} \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_l \end{pmatrix} \bmod q.$$

For further reference, the formula that is specified by this system of linear relations is denoted by F. The numbers α_{ij}, for $1 \le i \le l$ and $1 \le j \le k-l$, and the numbers b_i, for $1 \le i \le l$, are elements of $\mathbb{Z}_q$, and $\pi(\cdot)$ is a permutation of $\{1,\dots,k\}$. It will be clear to those of ordinary skill in the art that any system of linear relations in $x_1,\dots,x_k$ can be described in this form. Namely, if a system of linear relations contains dependent relations, it can be reduced to a system of independent linear relations, denoted in number by l here; then the matrix of coefficients can be brought into so-called row canonical form (also known as row echelon form); and finally, by applying a suitable permutation $\pi(\cdot)$, the columns of the matrix of coefficients can be interchanged, arriving at the system displayed above.

Clearly, any predetermined arrangement of the numbers α_{ij}, for $1 \le i \le l$ and $1 \le j \le k-l$, and the numbers b_i, for $1 \le i \le l$, can be used to describe the formula F, possibly in combination with additional bits that specify that F is a formula consisting of a system of linear relations (i.e., F has only "AND" connectives). Of course, how much additional bits are needed to specify F depends on the set of formulas from which $\mathcal{P}$ is allowed to pick an F for demonstration.

The inventive technique consists of $\mathcal{P}$ demonstrating that it knows a representation of the number, $h\,(\prod_{i=1}^{l} g_{\pi(k-l+i)}^{b_i})^{-1}$, with respect to

$$(g_{\pi(1)} \prod_{i=1}^{l} g_{\pi(k-l+i)}^{-\alpha_{i1}}, \dots, g_{\pi(k-l)} \prod_{i=1}^{l} g_{\pi(k-l+i)}^{-\alpha_{i,k-l}}).$$

$\mathcal{P}$ indeed knows such a representation in case the system of linear relations holds, namely $(x_{\pi(1)},\dots,x_{\pi(k-l)})$. By applying the basic proof of knowledge, and expanding the resulting expressions, $\mathcal{P}$ and $\mathcal{V}$ can evaluate all products of powers in the proof by evaluating products of powers of the numbers in the set $\{h, g_1,\dots,g_k\}$. $\mathcal{P}$ and $\mathcal{V}$ can hence efficiently apply, say, the simultaneous repeated squaring technique, by making use of one pre-computed table whose $2^{k+1}-1$ entries consist of the products of the numbers in the non-empty subsets of $\{h, g_1,\dots,g_k\}$. The protocol steps are as follows.

Box 11 first shows $\mathcal{P}$ generating $k-l$ random numbers, $w_1, \ldots, w_{k-l}$, all in $\mathbb{Z}_q$. The second line shows $\mathcal{P}$ computing a number a in G_q, in such a manner that $\mathcal{P}$ knows a randomly generated representation of a with respect to

$$(g_{\pi(1)} \prod_{i=1}^{l} g_{\pi(k-l+i)}^{-\alpha_{i1}}, \ldots, g_{\pi(k-l)} \prod_{i=1}^{l} g_{\pi(k-l+i)}^{-\alpha_{i,k-l}}),$$

namely, $(w_1, \ldots, w_{k-l})$. The number a may be called an initial witness, in line with terminology commonly used in the art for three-move proofs of knowledge. The third line shows $\mathcal{P}$ sending h, a digital certificate (denoted by cert(h)) of a certification authority on h, and a to $\mathcal{V}$. The certificate may also be on such information as an identity description of $\mathcal{P}$ and an expiration date, but such information is not explicitly shown because it is unrelated to the inventive technique. Alternatively, depending on the application, $\mathcal{V}$ may be able to retrieve h and the certificate from a public-key directory or the like, or a certificate is not needed at all because $\mathcal{V}$ can check the validity of h in another way. In any case, it will be clear that if the protocol is to be repeated several times, h and the certificate need only be sent at most once to $\mathcal{V}$.

The first line of Box 12 shows $\mathcal{V}$ generating a challenge number c in $\mathbb{Z}_{2^t}$, and the second line shows $\mathcal{V}$ sending c to $\mathcal{P}$. There are several fundamentally different manners in which c can be chosen, each of which makes an important distinction for the inventive method: depending on the manner in which c is generated, the protocol is zero-knowledge, witness hiding or a signed proof (the latter can come in many flavors). This is described in detail shortly, after the description of Box 13 and Box 14.

The first three lines of Box 13 show $\mathcal{P}$ computing $k-l$ response numbers, all in $\mathbb{Z}_q$, and denoted by $r_1, \ldots, r_{k-l}$. The fourth line shows $\mathcal{P}$ sending the $k-l$ response numbers to $\mathcal{V}$.

The first three lines of Box 14 show $\mathcal{V}$ computing l numbers, $r_{k-l+1}, \ldots, r_k$. The fourth line shows $\mathcal{V}$ verifying the $k-l$ responses provided by $\mathcal{P}$. If the verification holds, then $\mathcal{V}$ accepts the proof. If $\mathcal{V}$ accepts then the probability that $\mathcal{P}$ has been cheating successfully is $1/2^t$ for truly random c, and should be close to it when c is computed as a one-way hash of at least a.

If one represents atomic propositions by linear relations over $\mathbb{Z}_q$, then the inventive technique of Figure 1 can be used to demonstrate any satisfiable formula from so-called propositional logic that has only "AND" logical connectives. An example follows.

Example 1: To demonstrate the formula "($x_1 = 2x_3 + 3x_4 + 5 \bmod q$) AND ($x_2 = 6x_4 - 7 \bmod q$)," for a situation with $k = 5$, $\mathcal{P}$ proves to $\mathcal{V}$ knowledge of a representation of $h/(g_1^5 g_2^{-7})$ with respect to $(g_1^2 g_3, g_1^3 g_2^6 g_4, g_5)$. $\mathcal{P}$ hereto computes a as $g_1^{2w_1+3w_2} g_2^{6w_2} g_3^{w_1} g_4^{w_2} g_5^{w_3}$, for random w_1, w_2 and w_3 in $\mathbb{Z}_q$, and provides responses $r_1 \leftarrow cx_3 + w_1 \bmod q$, $r_2 \leftarrow cx_4 + w_2 \bmod q$ and $r_3 \leftarrow cx_5 + w_3 \bmod q$, responsive to a challenge c. $\mathcal{V}$ verifies the responses by checking whether a equals $h^{-c} g_1^{5c+2r_1+3r_2} g_2^{-7c+6r_2} g_3^{r_1} g_4^{r_2} g_5^{r_3}$. [End of example 1]

Returning to the description of Box 12, the first manner in which c can be determined is for $\mathcal{V}$ to choose it from a small set (i.e., t must be small). The four protocol steps must then be repeated several times, in order for $\mathcal{V}$ to be convinced that $\mathcal{P}$ knows a set of secrets (a representation) of h, and that the formula F applies to the secrets of $\mathcal{P}$. This results in a zero-knowledge demonstration.

The second manner is to chose c at random from a large set (i.e., t is large). In this case no repetition of the protocol steps is needed to convince $\mathcal{V}$ with high probability. Although the protocol is believed not to be zero-knowledge, it is believed to be infeasible for $\mathcal{V}$ to learn additional information about the secrets of $\mathcal{P}$. (Indeed, in case k is at least $l + 2$, which can always be ensured by $\mathcal{P}$ by extending its set of secrets with an additional "dummy" secret, the protocol can be proved to be witness hiding.) Alternatively, the protocol can be made zero-knowledge by having $\mathcal{V}$ start the protocol, by sending to $\mathcal{P}$ a commitment on its challenge c; $\mathcal{V}$ must then open the commitment in Box 12 (this ensures that $\mathcal{V}$ cannot compute c depending on the information provided by $\mathcal{P}$ in box 11). An example of a suitable commitment is $g_i^c h^s$, for a random s in $\mathbb{Z}_q$ and a generators g_i from the set of generators used by $\mathcal{P}$ and $\mathcal{V}$.

Thirdly, the demonstration can be performed as a signed proof, meaning that $\mathcal{V}$ can subsequently demonstrate to other parties that it has performed the protocol with $\mathcal{P}$, by showing the transcript of the protocol execution. In this case the challenge c must be computed as a hash (implying that t is large) of certain information; the hash-function should preferably be collision-intractable, and to prevent combining of old signed proofs into new ones several algebraic relations in terms of the hashfunction should be infeasible to compute. The information that is hashed in order to compute c should comprise at least a. Furthermore some or all of the following may be hashed along: h; a digital certificate on h (in particular that shown in the third line of Box 11; in case h can be verified in another way to correspond

to $\mathcal{P}$, the certificate may be omitted); a message for $\mathcal{V}$ (the signed proof then serves as a digital signature of $\mathcal{P}$ on that message); additional information such as an expiration date; and, a description of the formula F that is demonstrated by $\mathcal{P}$. The following information suffices to constitute the signed proof: h, $(r_1, \ldots, r_{k-l})$, c, and any (other) information hashed in order to compute c, with the possible exception of a (it suffices to include only one of c and a; the former results in a more compact signed proof).

Whether a description of F is hashed along or not has important implications for what exactly the signed proof demonstrates. Namely, if a description of F is hashed along, then the transcript of the protocol execution subsequently can be used to convince others that $\mathcal{P}$ knows a set of secrets corresponding to h, and that the formula F applies to the secrets of $\mathcal{P}$. If, on the other hand, F is not hashed along, then in general the transcript of the protocol execution only demonstrates that $\mathcal{P}$ knows a set of secrets corresponding to h; it is not a convincing demonstration of the fact that the formula F applies to the secrets of $\mathcal{P}$. However, if $\mathcal{P}$ is not allowed to demonstrate any system of linear relations for its secrets, but only systems of linear relations for which at least some of the coefficients α_{ij} must be in a predetermined negligible subset of $\mathbb{Z}_q$, then it may not be necessary to hash along the entire description of F: it may suffice to hash along only a part of the description of F (such as some of the α_{ij}'s, or a description of linear relations that apply to the α_{ij}'s, or more generally some information about F), or perhaps no information about F at all, in order for the signed proof to convince others of the fact that $\mathcal{P}$ knows a set of secrets corresponding to h, and that the formula F applies to the secrets of $\mathcal{P}$. Or, when only some of the coefficients α_{ij} and/or b_i are hashed along in the computation of c, or more generally only some information about the coefficients α_{ij} and/or b_i (in other words, a partial description of F), the signed proof may be convincing that one out of a restricted set of formulas applies to the set of secrets of $\mathcal{P}$, without revealing which one. From the above description it is believed to be clear to those of ordinary skill in the art how these various forms of signed proofs can be constructed, and that different forms of signed proof may each have their own merits, depending on the application in which a signed proof is to be used.

As a simple example, consider $h = g_1^{x_1} g_2^{x_2}$, and suppose that $\mathcal{P}$ has to demonstrate a formula F taken from the formulas described by relations $x_2 = A \bmod q$, for $A \in \mathbb{Z}_q$. It hereto provides a response r such that $a = h^{-c} g_1^r g_2^{Ac}$. If c is a hash of h and a only, then the signed proof

(consisting of h, c, A and r) does not demonstrate that $x_2 = A \bmod q$; it merely shows that $\mathcal{P}$ knows a representation of h with respect to (g_1, g_2). Namely, $\mathcal{P}$ and $\mathcal{V}$ could have taken $a \leftarrow g_1^{w_1} g_2^{w_2}$, for random w_1 and w_2, and $A \leftarrow x_2 + w_2/c \bmod q$. However, A is not under control of $\mathcal{P}$ and $\mathcal{V}$ in this case (assuming the hashfunction is sufficiently secure), and so the signed proof does demonstrate that $x_2 = A \bmod q$ in the special case that A must be in a subset of $\mathbb{Z}_q$ that is negligible (which is for example the case if A may only be zero or one). By hashing not only h and a, but also a description of F (if A parameterizes all possible formulas F, then A can be used to describe F), it is ensured that the signed proof also shows that F applies to the secrets of $\mathcal{P}$, no matter whether or not a restriction is imposed on the set of formulas that may be demonstrated.

It will also be clear to those of ordinary skill in the art that $\mathcal{P}$ may be able to compute c by itself in case of a signed proof (and it then does not need to supply a to $\mathcal{V}$); this depends on whether the additional information to be included in the hashvalue is available to $\mathcal{P}$, and on whether $\mathcal{P}$ is allowed to know c. Namely, if $\mathcal{V}$ provides $\mathcal{P}$ with the information that it wants to be hashed along in the computation of c, then $\mathcal{P}$ can compute c by itself; in particular, if $\mathcal{P}$ can determine by itself all the information that needs to be hashed, no interaction with $\mathcal{V}$ is needed. If, on the other hand, $\mathcal{V}$ wants to hash along a message that must remain unknown to $\mathcal{P}$, or $\mathcal{V}$ wants to blind the signed proof, then $\mathcal{V}$ should compute c by itself.

In order to completely blind the signed proof, $\mathcal{V}$ can compute c in Box 12 as follows: it generates random $k - l + 1$ blinding factors $\gamma_0, \ldots, \gamma_{k-l}$; it uses these to compute a blinded form, a', of a, according to

$$a' \leftarrow a\,(h(\prod_{i=1}^{l} g_{\pi(k-l+i)}^{b_i})^{-1})^{\gamma_0} \prod_{i=1}^{k-l} g_i^{\gamma_i};$$

it computes a number c' by hashing in any one of the manners described above for obtaining a signed proof, only this time hashing along a' instead of a; and, it computes the challenge c according to $c \leftarrow c' + \gamma_0 \bmod q$. In Box 14, $\mathcal{V}$ computes blinded forms of $r'_1, \ldots, r'_{k-l}$ of $r_1, \ldots, r_{k-l}$, according to

$$r'_i \leftarrow r_i + \gamma_i \bmod q,$$

for all $1 \le i \le k - l$. The signed proof is then formed by: h, $(r'_1, \ldots, r'_{k-l})$, c', and any information hashed in order to compute c', with the possible exception of a' (it suffices to include only one of c' and a'; the former results in a more compact signed proof). It will also be clear to those of ordinary

skill in the art how $\mathcal{V}$ must compute c, and adjust the responses of $\mathcal{P}$ in Box 13, in order to obtain a restrictively blinded signed proof: hereto the inventive techniques for restrictive blinding, detailed in US patent application Ser. No. 08/203,231 and in US patent application Ser. No. 08/321,855, filed October 14, 1994, both by the present applicant, can be combined straightforwardly with the inventive technique of Figure 1.

Furthermore, the protocol for demonstrating F can be made non-transferable, meaning that $\mathcal{V}$ cannot in real-time divert the protocol execution to another party in order to convince that party (as is well-known, zero-knowledge does not suffice to prevent this so-called man-in-the-middle attack). Hereto the inventive technique can be combined with a technique known in the art (See, Jakobsson, M., Sako, K. and Impagliazzo, R., "Designated verifier proofs and their applications," Advances in Cryptology – EUROCRYPT '96, Lecture Notes in Computer Science, Springer-Verlag, 1996). In effect, $\mathcal{P}$ must prove either that F applies to its own representation of h, or that it knows a representation of a number in G_q that is associated with $\mathcal{V}$. For this it must prove that it knows a secret key (representation) of $h\,(\prod_{i=1}^{l} g_{\pi(k-l+i)}^{b_i})^{-1}$ with respect to

$$(g_{\pi(1)} \prod_{i=1}^{l} g_{\pi(k-l+i)}^{-\alpha_{i1}}, \cdots, g_{\pi(k-l)} \prod_{i=1}^{l} g_{\pi(k-l+i)}^{-\alpha_{i,k-l}}).$$

(as detailed already above), or that it knows a secret key of a public key of $\mathcal{V}$. To accomplish this, another technique known in the art can be used (See, Cramer, R., Damgard, I., and Schoenmakers, B., "Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols," Advance in Cryptology – CRYPTO '94, Lecture Notes in Computer Science, Springer-Verlag, 1995); this technique enables $\mathcal{P}$ to demonstrate that it knows a secret key of a first public key or a secret key of a second public key.

Using techniques described in US patent application Ser. No. 08/203,231, $\mathcal{P}$ can consist of two (or more) parties, not necessarily acting in the same interest. A typical example is for $\mathcal{P}$ to be computer equipment held by a person, with $\mathcal{P}$ consisting of a tamper-resistant computing device $\mathcal{T}$ (typically a smart card or the like) acting in the interest of an issuer of $\mathcal{T}$, and a user-controlled personal computer $\mathcal{U}$ acting in the interest of the person, and $\mathcal{T}$ and $\mathcal{U}$ configured such that all flow of information to and from $\mathcal{T}$ must pass through $\mathcal{U}$. By ensuring that part of the representation $(x_1, \ldots, x_k)$ of h is known only to $\mathcal{T}$, which has been programmed on behalf

of its issuer or some other party, the demonstration in general can only be performed when $\mathcal{T}$ cooperates. It hereto suffices that $\mathcal{U}$ knows all but one of the exponents, say x_k, which is known only to $\mathcal{T}$; $g_k^{x_k}$ can be viewed as a public key of $\mathcal{T}$, and x_k as its secret key. When the linear relations do not involve x_k, the proof can be performed in such a way that x_k is not revealed by $\mathcal{T}$.

To illustrate this, consider again example 1. In this case, $\mathcal{T}$ knows x_5 and $\mathcal{U}$ knows $(x_1, \ldots, x_4)$. $\mathcal{T}$ should generate w_3 and provide $\mathcal{U}$ with $a_1 \leftarrow g_5^{w_3}$, and $\mathcal{U}$ should generate w_1 and w_2 and multiply a_1 by $g_1^{2w_1+3w_2} g_2^{6w_2} g_3^{w_1} g_4^{w_2}$ to obtain a; $\mathcal{U}$ can then compute r_1 and r_2, while $\mathcal{T}$ has to provide r_3.

As detailed in US patent application Ser. No. 08/203,231, $\mathcal{U}$ can furthermore prevent subliminal channels from $\mathcal{T}$ to $\mathcal{V}$ and vice versa, and in fact even the development of random numbers known to both $\mathcal{T}$ and $\mathcal{V}$. In example 1, $\mathcal{U}$ can hereto add a random number t in $\mathbb{Z}_q$ to the challenge, c, before passing it on to $\mathcal{T}$ (to prevent inflow), and add a random number u in $\mathbb{Z}_q$ to r_3 before passing it on to $\mathcal{V}$ (to prevent outflow); correspondingly, $\mathcal{U}$ can then multiply a in addition by $g_5^u (g_5^{x_5})^t$, before passing it to $\mathcal{V}$.

It will be clear to those of ordinary skill in the art that the above manners of computing c in Box 12, the techniques for blinding and restrictive blinding, the technique for making the demonstration protocol non-transferable, and the technique for dividing $\mathcal{P}$ into parties with conflicting interests, apply equally well to the other inventive methods that will be described hereafter. For this reason in later descriptions only a reference to these techniques will be provided (whenever appropriate).

The inventive technique of FIG. 1 can also be based on the difficulty of computing RSA-roots. Consider $\mathcal{P}$ having to prove the same system of linear relations, but then with all equations modulo v instead of modulo q, and having committed to a set of secrets $(x_1, \ldots, x_k)$ by using $h \leftarrow Y_1^{x_1} \cdots Y_k^{x_k} x_{k+1}^v \bmod n$ for a number x_{k+1} in $\mathbb{Z}_n^*$. (Depending on the application, x_{k+1} may, or may not have been chosen secret and at random; in the latter case it can be taken equal to 1, for computational efficiency.) The inventive technique then consists of letting $\mathcal{P}$ demonstrate that it knows a representation of the number, $h \left(\prod_{i=1}^{l} Y_{\pi(k-l+i)}^{b_i}\right)^{-1} \bmod n$, with respect to

$$\left(Y_{\pi(1)} \prod_{i=1}^{l} Y_{\pi(k-l+i)}^{-\alpha_{i1}} \bmod n, \ldots, Y_{\pi(k-l)} \prod_{i=1}^{l} Y_{\pi(k-l+i)}^{-\alpha_{i,k-l}} \bmod n, v\right).$$

By using the basic proof of knowledge described by Figure 5 of US patent

application Ser. No. 08/203,231, expanding the resulting expressions, a single pre-computed table can be used for efficient simultaneous repeated squaring. A detailed description is omitted here, since those of ordinary skill in the art are believed to be capable of straightforwardly applying the inventive method of FIG. 1 by studying the drawing in conjunction with the corresponding description. Likewise, it is believed to be straightforward for those of ordinary skill in the art to apply the techniques described above for computing the challenge, for blinding and restrictive blinding, for making the demonstration protocol non-transferable, and for dividing $\mathcal{P}$ into parties with conflicting interests.

Turning now to FIG. 2, a flowchart of a second protocol for demonstrating that a system of linear relations applies to a set of secrets, without revealing additional information about the secrets, will now be described in detail.

As in FIG. 1, at the outset $\mathcal{P}$ knows a representation $(x_1, \ldots, x_k)$ of a number h in G_q with respect to $(g_1, \ldots, g_k)$. $\mathcal{P}$ has to demonstrate a system consisting of l independent linear relations.

$$\begin{pmatrix} \alpha_{11} & \ldots & \alpha_{1k} \\ \vdots & & \vdots \\ \alpha_{l1} & \ldots & \alpha_{lk} \end{pmatrix} \begin{pmatrix} x_{\pi(1)} \\ \vdots \\ x_{\pi(k)} \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_l \end{pmatrix} \bmod q.$$

Contrary to the method of FIG. 1, the matrix of coefficients α_{ij}, for $1 \leq i \leq l$ and $1 \leq j \leq k$, does not need to be in row canonical form. The protocol steps for demonstrating this formula F are as follows.

Box 21 first shows $\mathcal{P}$ generating k random numbers, $w_1, \ldots, w_k$, all in $\mathbb{Z}_q$. The second line shows $\mathcal{P}$ computing a number a in G_q, in such a manner that $(w_1, \ldots, w_k)$ is a representation of a with respect to $(g_1, \ldots, g_k)$. The number a may be called an initial witness. The next three lines show $\mathcal{P}$ computing l numbers, $e_1, \ldots, e_l$, all in $\mathbb{Z}_q$, in accordance with the system of linear relations. The sixth line shows $\mathcal{P}$ sending h, a digital certificate on h, and a to $\mathcal{V}$; the remarks made for Box 11 apply here also.

The first line of Box 22 shows $\mathcal{V}$ generating a challenge number c in $\mathbb{Z}_{2^t}$, and the second line shows $\mathcal{V}$ sending c to $\mathcal{P}$.

The first three lines of Box 23 show $\mathcal{P}$ computing k response numbers, all in $\mathbb{Z}_q$, and denoted by $r_1, \ldots, r_k$. The fourth line shows $\mathcal{P}$ sending the k response numbers to $\mathcal{V}$.

The first four lines of Box 24 show $\mathcal{V}$ verifying the k responses,

$r_1, \ldots, r_k$, by checking $l + 1$ verification relations. If all verifications hold, then $\mathcal{V}$ accepts.

Compared to the method of FIG. 1 this method requires more information to be transmitted from $\mathcal{P}$ to $\mathcal{V}$. On the other hand, an advantage is that the computation of a does not depend on the particular system of linear relations that is to be demonstrated. Hence $\mathcal{P}$ can compute a in advance, before knowing the system of relations that is to demonstrated; only the computation of $e_1, \ldots, e_l$ is dependent of the particular system to be demonstrated, but the required computation time for this is negligible in comparison to that for a.

As will be clear to those of ordinary skill in the art, all the techniques described in relation to Figure 1 also apply to the inventive technique of FIG. 2. Moreover, in case of a signed proof, the issue of whether a description of F must be hashed along or not when computing c, or whether perhaps only some information about F needs to be hashed along, now also applies to $(e_1, \ldots, e_l)$. For example, if $\mathcal{P}$ is allowed to demonstrate any formula F, and the signed proof should convince that F applies to the secrets of $\mathcal{P}$, then a description of F as well as $(e_1, \ldots, e_l)$ should be hashed along. If it is sufficient that the signed proof only convinces that some formula out of a subset of all formulas applies to the secrets of $\mathcal{P}$, then only some information about F and/or some information about $(e_1, \ldots, e_l)$ needs to be hashed along (with the choice of which information depending on the particular subset of formulas that is considered). If no information about $(e_1, \ldots, e_l)$ is hashed along, then the signed proof only demonstrates that $\mathcal{P}$ knows a representation of h with respect to $(g_1, \ldots, g_k)$, even in case $\mathcal{P}$ is allowed to demonstrate only a few formulas.

Variations of the inventive method of FIG. 2 are possible. For example, the inventive method of FIG. 5, described later on, is also an adaptation of the method of FIG. 1. It has the same advantage as FIG. 2, while having somewhat greater communication efficiency. (The inventive technique of FIG. 5 serves an additional goal, namely to prevent $\mathcal{P}$ from performing the demonstration more than once, but this extra feature is not of relevance here; for the present discussion, $\mathcal{P}$ in FIG. 5 could just as well be allowed to choose a freely each time the proof with respect to h is performed.) The following changes to Box 23 and Box 24 ensure that the method of FIG. 2 becomes equally efficient to that of FIG. 5: in Box 23, $\mathcal{P}$ only computes $k - l$ of the response numbers, and sends these to $\mathcal{V}$ (the $k - l$ positions must be agreed upon with $\mathcal{V}$ in advance); instead of verifying the linear

relations as specified in the last three lines of Box 24, $\mathcal{V}$ computes the remaining l response numbers in accordance with these l relations; and finally, $\mathcal{V}$ verifies the response numbers, k in total, as specified by the first line of Box 24.

More generally, $\mathcal{V}$ can restrict the set of formulas that $\mathcal{P}$ will be able to perform to a subclass, by requiring the numbers $(e_1, \ldots, e_l)$ sent by $\mathcal{P}$ in Box 21 to satisfy certain restrictions. Namely, the numbers $(e_1, \ldots, e_l)$ can be thought of as correction factors for adjusting a, in order to ensure that the adjusted form of a can be used to demonstrate a particular formula. By allowing $\mathcal{P}$ in Box 21 to send only up to z correction factors, where $z < l$, and/or by allowing $\mathcal{P}$ to send only correction factors that satisfy a system of (linear or otherwise) relations, $\mathcal{P}$'s computation of a will be able to anticipate only a subset of possible formulas. The protocol of FIG. 1 can be thought of as the extreme case where $\mathcal{V}$ does not allow $\mathcal{P}$ to use any correction factors: $\mathcal{P}$ hence must compute a in direct correspondence to the formula F.

It will be clear to those of ordinary skill in the art how to base the inventive technique of FIG. 2, as well as all other described variations and related techniques, on the difficulty of computing RSA-roots; for this reason a detailed description is omitted.

Turning now to FIG. 3, a flowchart of a protocol for demonstrating that a linear relation does not apply to a set of secrets, without revealing additional information about the secrets, will now be described in detail.

The situation at the outset is as described for FIG. 1. $\mathcal{P}$ has to demonstrate to $\mathcal{V}$ that its set of secrets (the representation it knows of h) satisfies the formula $x_{\pi(1)} \neq \alpha_1 + \alpha_2 x_{\pi(2)} \cdots + \alpha_k x_{\pi(k)} \bmod q$. The coefficients α_i, for $1 \leq i \leq k$, are elements of $\mathbb{Z}_q$, and they form a description of the formula F that $\mathcal{P}$ has to demonstrate (again, depending on the space of all formulas that may be demonstrated by $\mathcal{P}$, additional bits may be needed to describe F). As will be apparent to those of ordinary skill in the art, any linear inequality can be written in this form, for a suitable permutation, $\pi(\cdot)$, of $\{1, \ldots, k\}$. Moreover $\pi(\cdot)$ can always be defined to interchange at most two elements and leave the rest unchanged.

If inequality holds, then $x_{\pi(1)}$ is equal to $(\alpha_1 + \alpha_2 x_{\pi(2)} + \cdots + \alpha_k x_{\pi(k)}) - \epsilon \bmod q$, for a nonzero ϵ in $\mathbb{Z}_q$. The inventive technique consists of $\mathcal{P}$ demonstrating that it knows a

representation of the number, $g_{\pi(1)}$, with respect to

$$(g_{\pi(1)}^{\alpha_1}h^{-1}, g_{\pi(1)}^{\alpha_2}g_{\pi(2)}, \ldots, g_{\pi(1)}^{\alpha_k}g_{\pi(k)}).$$

$\mathcal{P}$ indeed knows such a representation in case the inequality holds, namely $(\delta \bmod q, x_{\pi(2)}\delta \bmod q, \ldots, x_{\pi(k)}\delta \bmod q)$, where δ denotes $\epsilon^{-1} \bmod q$. Applying the basic proof of knowledge, and expanding the resulting expressions into products of powers of numbers in the set $\{h, g_1, \ldots, g_k\}$, the following protocol results (again, any other proof of knowledge can be used instead).

Box **31** first shows $\mathcal{P}$ generating k random numbers, $w_1, \ldots, w_k$, all in $\mathbb{Z}_q$. The second line shows $\mathcal{P}$ computing a number a in G_q, in such a manner that $\mathcal{P}$ knows a randomly generated representation of a with respect to

$$(g_{\pi(1)}^{\alpha_1}h^{-1}, g_{\pi(1)}^{\alpha_2}g_{\pi(2)}, \ldots, g_{\pi(1)}^{\alpha_k}g_{\pi(k)}),$$

namely, $(w_1, \ldots, w_k)$. The number a may be called an initial witness. The third line shows $\mathcal{P}$ sending h, a digital certificate on h, and a to $\mathcal{V}$; the remarks made for Box **11** apply here also.

The first line of Box **32** shows $\mathcal{V}$ generating a challenge number c in $\mathbb{Z}_{2^t}$, and the second line shows $\mathcal{V}$ sending c to $\mathcal{P}$.

Box **33** first shows $\mathcal{P}$ computing a difference number ϵ, which is non-zero if the inequality holds. The second line shows $\mathcal{P}$ computing the inverse modulo q of ϵ, denoted by δ. The next four lines show $\mathcal{P}$ computing k response numbers, all in $\mathbb{Z}_q$, and denoted by $r_1, \ldots, r_k$. The seventh line shows $\mathcal{P}$ sending the k response numbers to $\mathcal{V}$.

Box **34** shows $\mathcal{V}$ verifying the k responses, $r_1, \ldots, r_k$. If the verification holds, then $\mathcal{V}$ accepts the proof.

Representing atomic propositions by linear relations over $\mathbb{Z}_q$, this inventive technique can be used to demonstrate any satisfiable formula from propositional logic that has one "NOT" logical connective and no other logical connectives. The protocol described by FIG. **3** demonstrates to $\mathcal{V}$ the formula "NOT $(x_{\pi(1)} = \alpha_1 + \alpha_2 x_{\pi(2)} + \cdots + \alpha_k x_{\pi(k)} \bmod q)$."

As will be clear to those of ordinary skill in the art, all the techniques described in relation to Figure **1** also apply to the inventive technique of FIG. **3**. Also, the technique of using correction factors, demonstrated by the inventive technique of FIG. **2**, can be used, to ensure that $\mathcal{P}$ can compute a independent of the formula it has to demonstrate; and $\mathcal{V}$ can restrict the number of correction factors, and/or the relations they need to

satisfy, in order to restrict the set of formulas that can be anticipated by $\mathcal{P}$ with the same a, as described for FIG. 2.

Moreover, to demonstrate that none of l independent linear relations holds, where $l > 1$, $\mathcal{P}$ has to perform the protocol l times, once for each inequality. All l demonstrations can be performed in parallel, and can furthermore be responsive to the same challenge c. In the latter case, all the data of each individual demonstration must be hashed into c in order to obtain a signed proof. The various flavors of signed proofs, described in relation to FIG. 1, can also be obtained in this case, as will be clear to those of ordinary skill in the art.

Combination with the inventive technique of FIG. 1 allows $\mathcal{P}$ to demonstrate any satisfiable formulas from propositional logic that has both "AND" and "NOT" logical connectives, atomic propositions being linear relations over $\mathbb{Z}_q$. For a system consisting of one or more linear relations and one linear inequality, the inequality is first written as a linear relation by introducing a difference term (denotes by ϵ in the description above). By appropriate substitution, the system can then be represented by the following matrix relation

$$\begin{pmatrix} \alpha_{11} & \dots & \alpha_{1,k-l} & 1 & 0 & \dots & 0 \\ \vdots & & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & & \vdots & \vdots & \ddots & \ddots & 0 \\ \alpha_{l1} & \dots & \alpha_{l,k-l} & 0 & \dots & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{\pi(1)} \\ \vdots \\ \vdots \\ x_{\pi(k)} \end{pmatrix} = \begin{pmatrix} b_1 - f_1\epsilon \\ \vdots \\ \vdots \\ b_l - f_l\epsilon \end{pmatrix} \bmod q,$$

where $f_1, \dots, f_l$ are numbers in $\mathbb{Z}_q$. (Clearly f_1, say, can always be made equal to 1.) $\mathcal{P}$ then has to demonstrate knowledge of a representation of the number, $\prod_{i=1}^{l} g_{\pi(k-l+i)}^{f_i}$, with respect to

$$(h^{-1} \prod_{i=1}^{l} g_{\pi(k-l+i)}^{b_i}, g_{\pi(1)} \prod_{i=1}^{l} g_{\pi(k-l+i)}^{-\alpha_{i1}}, \dots, g_{\pi(k-l)} \prod_{i=1}^{l} g_{\pi(k-l+i)}^{-\alpha_{i,k-l}}).$$

$\mathcal{P}$ indeed knows such a representation in case the system holds, namely $(\delta, x_{\pi(1)}\delta \bmod q, \dots, x_{\pi(k-l)}\delta \bmod q)$, where δ denotes $\epsilon^{-1} \bmod q$. An example, based on example 1 and augmented with one "NOT" relation, follows.

Example 2: To demonstrate the formula "$(x_1 = 2x_3 + 3x_4 + 5 \bmod q)$ AND $(x_2 = 6x_4 - 7 \bmod q)$ AND (NOT $(x_3 = 4x_4 + 8 \bmod q)$)," for a situation with $k = 5$, $\mathcal{P}$ needs to show that $x_1 = 2x_3 + 3x_4 + 5 \bmod q$, $x_2 = 6x_4 - 7 \bmod q$ and $x_3 = 4x_4 + 8 - \epsilon \bmod q$, for some ϵ that must be

demonstrated to be non-zero. Substitution gives that $\mathcal{P}$ must prove that $x_1 = 11x_4 + 21 - 2\epsilon \bmod q$, $x_2 = 6x_4 - 7 \bmod q$ and $x_3 = 4x_4 + 8 - \epsilon$. $\mathcal{P}$ can perform this proof, since $(\delta \bmod q, x_4\delta \bmod q, x_5\delta \bmod q)$ is a representation of $g_1^2 g_3$ with respect to $(h^{-1}g_1^{21}g_2^{-7}g_3^8, g_1^{11}g_2^6 g_3^4 g_4, g_5)$. where $\delta = \epsilon^{-1} \bmod q$. More specifically, $\mathcal{P}$ computes a as $h^{-w_1}g_1^{21w_1+11w_2}g_2^{-7w_1+6w_2}g_3^{8w_1+4w_2}g_4^{w_2}g_5^{w_3}$, for random w_1, w_2 and w_3 in $\mathbb{Z}_q$, and provides responses $r_1 \leftarrow c\delta + w_1 \bmod q$, $r_2 \leftarrow c\delta x_4 + w_2 \bmod q$ and $r_3 \leftarrow c\delta x_5 + w_3 \bmod q$ responsive to a challenge c. $\mathcal{V}$ verifies the responses by checking whether a is equal to $h^{-r_1}g_1^{-2c+21r_1+11r_2}g_2^{-7r_1+6r_2}g_3^{-c+8r_1+4r_2}g_4^{r_2}g_5^{r_3}$.

As with the inventive technique of FIG. 1. $\mathcal{P}$ can consist of $\mathcal{U}$ and $\mathcal{T}$, with $\mathcal{T}$ knowing x_5 and $\mathcal{U}$ knowing (x_1, x_2, x_3, x_4): in the computation of a, w_3 must then be generated by $\mathcal{T}$, and $g_5^{w_3}$ must be provided to $\mathcal{U}$ by $\mathcal{T}$ so that $\mathcal{U}$ can compute a. To prevent $\mathcal{T}$ from learning ϵ, $\mathcal{P}$ can additionally raise $g_5^{w_3}$ to the power δ before multiplying it with $h^{-w_1}g_1^{21w_1+11w_2}g_2^{-7w_1+6w_2}g_3^{8w_1+4w_2}g_4^{w_2}$; when computing the responses, $\mathcal{T}$ then provides $\mathcal{U}$ with $r_3 = cx_5 + w_3 \bmod q$, which $\mathcal{U}$ multiplies by δ before passing it on to $\mathcal{V}$. More generally, to prevent inflow $\mathcal{P}$ can add a random number t in $\mathbb{Z}_q$ to the challenge c of $\mathcal{V}$ before passing it on to $\mathcal{T}$, and to prevent outflow $\mathcal{T}$ can add a number u in $\mathbb{Z}_q$ to r_3 before multiplying it by δ and passing it on to $\mathcal{V}$; correspondingly, the contribution $g_5^{w_3}$ of $\mathcal{T}$ in the computation of a can then be transformed by $\mathcal{U}$ to $g_5^{u\delta}(g_5^{x_5})^{t\delta}$ before multiplying it with $h^{-w_1}g_1^{21w_1+11w_2}g_2^{-7w_1+6w_2}g_3^{8w_1+4w_2}g_4^{w_2}$ to obtain a. [End of example 2]

More generally, the inventive techniques of FIG. 1 and FIG. 3 allow $\mathcal{P}$ to demonstrate any satisfiable formula that consists of an arbitrary number of "AND" and "NOT" logical connectives. When more than one "NOT" logical connective is present, the demonstration can be divided into several sub-formula proofs, which may be performed in parallel and responsive to the same challenge. For example, to demonstrate the formula "S_1 AND S_2 AND S_3 AND S_4 AND (NOT S_5) AND (NOT S_6)," where $S_1, \ldots, S_6$ are linear relations such that the formula is satisfiable, two sub-formula proofs can be performed. The division can be done in many ways: $\mathcal{P}$ can demonstrate "S_1 AND S_2 AND S_3 AND S_4 AND (NOT S_5)" and "NOT S_6;" or it can demonstrate "S_1 AND S_2 AND (NOT S_5)" and "S_3 AND S_4 AND (NOT S_6);" and so on.

The inventive technique for demonstrating any satisfiable formula from propositional logic that has "AND" and "NOT" logical connectives can also be based on the difficulty of computing RSA-roots. Assume that $\mathcal{P}$ has to

prove the same inequality as for FIG. 3, but then modulo v instead of modulo q, and has committed to $(x_1, \ldots, x_k)$ by using $h \leftarrow Y_1^{x_1} \cdots Y_k^{x_k} x_{k+1}^v \bmod n$ for a number x_{k+1} in $\mathbb{Z}_n^*$. The inventive technique for demonstrating one inequality then consists of letting $\mathcal{P}$ demonstrate that it knows a representation of the number, $Y_{\pi(1)}$, with respect to

$$(Y_{\pi(1)}^{a_1} h^{-1} \bmod n, Y_{\pi(1)}^{a_2} Y_{\pi(2)} \bmod n, \ldots, Y_{\pi(1)}^{a_k} Y_{\pi(k)} \bmod n, v).$$

If the inequality holds, $\epsilon \not\equiv 0 \bmod v$ and $\mathcal{P}$ can compute numbers d and f in $\mathbb{Z}_v$ such that $\epsilon d = 1 + fv$, by using the well-known Extended Euclidean Algorithm: a satisfying representation known to $\mathcal{P}$ is

$$(d, dx_{\pi(2)} \bmod v, \ldots, dx_{\pi(k)} \bmod v, x_{k+1}^d Y_{\pi(1)}^{-f} \prod_{i=2}^{k} (Y_{\pi(1)}^{a_i} Y_{\pi(i)})^{(dx_{\pi(i)}) \operatorname{div} v} \bmod n).$$

Here, a div v denotes the maximum number of times that v can be subtracted from a such that the remainder is positive, that is, $a = v(a \text{ div } v) + a \bmod v$. As will be clear to those of ordinary skill in the art, to demonstrate the system

$$\begin{pmatrix} \alpha_{11} & \ldots & \alpha_{1,k-l} & 1 & 0 & \ldots & 0 \\ \vdots & & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & & \vdots & \vdots & \ddots & \ddots & 0 \\ \alpha_{l1} & \ldots & \alpha_{l,k-l} & 0 & \ldots & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{\pi(1)} \\ \vdots \\ \vdots \\ x_{\pi(k)} \end{pmatrix} = \begin{pmatrix} b_1 - f_1 \epsilon \\ \vdots \\ \vdots \\ b_l - f_l \epsilon \end{pmatrix} \bmod v,$$

where $f_1, \ldots, f_l$ are numbers in $\mathbb{Z}_v$, $\mathcal{P}$ can demonstrate knowledge of a representation of the number, $\prod_{i=1}^{l} Y_{\pi(k-l+i)}^{f_i}$, with respect to

$$(h^{-1} \prod_{i=1}^{l} Y_{\pi(k-l+i)}^{b_i}, Y_{\pi(1)} \prod_{i=1}^{l} Y_{\pi(k-l+i)}^{-\alpha_{i1}}, \ldots, Y_{\pi(k-l)} \prod_{i=1}^{l} Y_{\pi(k-l+i)}^{-\alpha_{i,k-l}}, v).$$

$\mathcal{P}$ indeed knows such a representation in case the system holds, namely $(\delta, x_{\pi(1)}\delta \bmod v, \ldots, x_{\pi(k-l)}\delta \bmod v, x_{k+1})$, where δ denotes $\epsilon^{-1} \bmod v$. Adaptation to proving general satisfiable formulas with "AND" and "NOT" logical connectives is believed to be straightforward for those of ordinary skill in the art when studying the described method for the discrete log-based approach, and hence is omitted here; the same holds for all other previously detailed techniques and variations.

Turning now to FIG. 4, a flowchart of a protocol for demonstrating that at least one of two linear relations applies to a set of secrets, without

revealing additional information about the secrets, will now be described in detail.

The situation at the outset is again as in FIG. 1. $\mathcal{P}$ has to demonstrate to $\mathcal{V}$ that the representation known to it satisfies either the linear relation $x_{\pi(1)} = \alpha_1 + \alpha_2 x_{\pi(2)} \cdots + \alpha_k x_{\pi(k)} \bmod q$, or the linear relation $x_{\rho(1)} = \beta_1 + \beta_2 x_{\rho(2)} \cdots + \beta_k x_{\rho(k)} \bmod q$, or both. The coefficients α_i and β_i, for $1 \leq i \leq k$, are elements of $\mathbb{Z}_q$, and $\pi(\cdot)$ and $\rho(\cdot)$ are suitable permutations of $\{1, \ldots, k\}$ that can always be defined to interchange at most two elements each.

If (and only if) the first linear relation holds, then $\mathcal{P}$ can compute, for any challenge c_1, responses $(r_2, \ldots, r_k)$ such that

$$a_1 = h^{-c_1} g_{\pi(1)}^{c_1\alpha_1 + \sum_{i=2}^{k} \alpha_i r_i} g_{\pi(2)}^{r_2} \cdots g_{\pi(k)}^{r_k},$$

where a_1 has been generated by $\mathcal{P}$ of the form $g_{\pi(1)}^{\sum_{i=2}^{k} \alpha_i w_i} \prod_{i=2}^{k} g_{\pi(i)}^{w_i}$ for random $w_2, \ldots, w_k$ in $\mathbb{Z}_q$. Likewise, if (and only if) the second linear relation holds, then $\mathcal{P}$ can compute, for any challenge c_2, responses $(s_2, \ldots, s_k)$ such that

$$a_2 = h^{-c_2} g_{\rho(1)}^{c_2\beta_1 + \sum_{i=2}^{k} \beta_i s_i} g_{\rho(2)}^{s_2} \cdots g_{\rho(k)}^{s_k},$$

where a_2 has been generated by $\mathcal{P}$ of the form $g_{\rho(1)}^{\sum_{i=2}^{k} \beta_i v_i} \prod_{i=2}^{k} g_{\rho(i)}^{v_i}$ for random $v_2, \ldots, v_k$ in $\mathbb{Z}_q$. To perform the proof, $\mathcal{P}$ chooses one of the two challenges, c_1 or c_2, at random by itself, so that it can anticipate that challenge by calculating a suitable a_i from the self-chosen c_i and a set of randomly self-chosen "responses;" this is referred to as a simulated proof. The other challenge cannot be anticipated by $\mathcal{P}$, thereby forcing $\mathcal{P}$ to perform a genuine proof. However, $\mathcal{P}$ can hide which challenge (and response set) it has chosen by itself.

More specifically, assuming for concreteness that the first linear relation holds and the second does not, $\mathcal{P}$ computes a_2 by generating at random a challenge c_2 and "responses" $s_2, \ldots, s_k$, and computing

$$a_2 \leftarrow h^{-c_2} g_{\rho(1)}^{c_2\beta_1 + \sum_{i=2}^{k} \beta_i s_i} g_{\rho(2)}^{s_2} \cdots g_{\rho(k)}^{s_k}.$$

For the first relation $\mathcal{P}$ performs the basic proof of knowledge. To ensure that $\mathcal{P}$ can choose (and hence anticipate) at most one challenge correctly, without revealing which one, $\mathcal{V}$ in the protocol sends to $\mathcal{P}$ a random

challenge c, and $\mathcal{P}$ uses challenges c_1 and c_2 such that, say, the bitwise exclusive-or of c_1 and c_2 is equal to c.

The first two lines of Box **41** show $\mathcal{P}$ computing a number a_1, as specified above. The third and fourth lines show $\mathcal{P}$ generating at random a self-chosen challenge c_2, and $k-1$ self-chosen responses, $s_2, \ldots, s_k$. As shown in the fifth line, $\mathcal{P}$ then computes a number a_2, based on c_2 and $s_2, \ldots, s_k$. Finally, as shown in the sixth line, $\mathcal{P}$ sends h, a digital certificate on h, and a_1 and a_2 to $\mathcal{V}$; the remarks made for Box **11** apply here also. The numbers a_1 and a_2 may be called initial witnesses.

The first line of Box **42** shows $\mathcal{V}$ generating a challenge number c in $\mathbb{Z}_{2^t}$, and the second line shows $\mathcal{V}$ sending c to $\mathcal{P}$.

Box **43** first shows $\mathcal{P}$ computing challenge c_1, from the self-chosen challenge c_2 and the challenge c, by taking their bitwise exclusive-or (denoted by $\ominus$). For the resulting challenge, c_1, $\mathcal{P}$ then computes $k-1$ responses, $r_2, \ldots, r_k$, as shown in the next three lines. As shown in the fifth line, $\mathcal{P}$ then sends to $\mathcal{V}$ the two challenges, the response set for the first linear relation, and the self-chosen response set for the second linear relation.

Box **44** first shows $\mathcal{V}$ verifying that $\mathcal{P}$ used two challenges, c_1 and c_2, whose bitwise exclusive-or equals c; this guarantees to $\mathcal{V}$ that $\mathcal{P}$ could have generated only one of c_1, c_2, by itself. The second line shows $\mathcal{V}$ verifying the response set for the first linear relation, with respect to challenge c_1. The third line shows $\mathcal{V}$ verifying the response set for the second linear relation, with respect to challenge c_2. If all three verifications hold, then $\mathcal{V}$ accepts the proof.

Representing atomic propositions by linear relations over $\mathbb{Z}_q$, this inventive technique can be used to demonstrate any satisfiable formula from propositional logic that has only "OR" logical connectives. The protocol described by FIG. **4** demonstrates to $\mathcal{V}$ the formula
"$(x_{\pi(1)} = \alpha_1 + \sum_{i=2}^{k} \alpha_i x_{\pi(i)} \bmod q)$ OR $(x_{\rho(1)} = \beta_1 + \sum_{i=1}^{k} \beta_i x_{\rho(i)} \bmod q)$."

Clearly one of c_1, c_2 need not be transmitted by $\mathcal{P}$ in Box **43**, as $\mathcal{V}$ can compute it by itself. Furthermore, instead of using in Box **43** the bitwise exclusive-or operator to compute the remaining challenge values from the self-chosen challenge and the challenge of $\mathcal{V}$, other functions can be used. For example, it can be required that c_1 and c_2 satisfy $c_1 + c_2 = c \bmod 2^t$.

As will be appreciated, the inventive methods of FIGS. **1**, **3** and **4** can be combined in order to demonstrate arbitrary satisfiable formulas from propositional logic, where the atomic propositions are linear relations over

$\mathbb{Z}_q$. Hereto such a formula F can be expressed in the format "Q_1 AND Q_2 AND $\cdots$ AND Q_l," where sub-formula Q_i has the format "R_{1i} OR R_{2i} OR $\cdots$ OR R_{li};" and each of subsub-formulas R_{1i}, R_{2i}, $\ldots$, R_{li} is a formula from propositional logic that connects linear relations over $\mathbb{Z}_q$ by "AND" connectives, at most one "NOT" connective, and no other logical connectives. By using the inventive technique of FIG. 4, to demonstrate one out of l representations, $\mathcal{P}$ can then demonstrate the sub-formula Q_i. To prove the formula F, $\mathcal{P}$ needs to demonstrate separately each of the sub-formulas Q_1, Q_2, $\ldots$, Q_l. These l sub-formula proofs can be performed in parallel, and responsive to the same challenge. An example, based on example 2, follows.

Example 3: Consider the linear relations S_1 = "$(x_1 = 2x_3 + 3x_4 + 5 \bmod q)$," S_2 = "$(x_2 = 6x_4 - 7 \bmod q)$," S_3 = "$(x_3 = 4x_4 + 8 \bmod q)$," S_4 = "$(x_1 = 5x_2 + 7 \bmod q)$." S_5 = "$(x_2 = 3x_4 + 2 \bmod q)$," S_6 = "$(x_1 = x_2 + x_4 + 3 \bmod q)$," and S_7 = "$(x_2 = -3x_3 + 7 \bmod q)$," and $h = \prod_{i=1}^{5} g_i^{x_i}$. Suppose that $\mathcal{P}$ has to demonstrate the formula F = "(S_1 AND S_2 AND (NOT (S_3 OR S_7)) AND S_6) OR ((NOT (S_4 OR S_7)) AND S_5 AND S_6)." By applying the rules of propositional logic, F can be re-written in the format "((S_1 AND S_2 AND (NOT S_3)) OR ((NOT S_4) AND S_5)) AND (S_6 AND (NOT S_7))." Let now R_{11} = "S_1 AND S_2 AND (NOT S_3)," R_{12} = "(NOT S_4) AND S_5," and R_{22} = "S_6 AND (NOT S_7)," and Q_1 = "R_{11} OR R_{12}," and $Q_2 = R_{22}$. $\mathcal{P}$ can then demonstrate the formula F to $\mathcal{V}$ by separately proving sub-formula Q_1 and sub-formula Q_2. To demonstrate Q_1, $\mathcal{P}$ can demonstrate knowledge of a representation of $g_1^2 g_3$ with respect to $(h^{-1} g_1^{21} g_2^{-7} g_3^8, g_1^{11} g_2^6 g_3^4 g_4, g_5)$ (cf. example 2), or knowledge of a representation of g_1 with respect to $(h^{-1} g_1^{17} g_2^2, g_1^{15} g_2^3 g_4, g_3, g_5)$, or both. Hereto the inventive technique of FIG. 4 can be used straightforwardly: $\mathcal{P}$ first provides two numbers, a_1 and a_2, both in G_q, to $\mathcal{V}$; responsive to a challenge, c, $\mathcal{P}$ provides to $\mathcal{V}$ a first set of responses, (r_1, r_2, r_3) responsive to a challenge c_1, and a second set of responses, (s_1, s_2, s_3, s_4), responsive to a challenge c_2; and $\mathcal{V}$ verifies that $a_1 = h^{-r_1} g_1^{-2c_1+21r_1+11r_2} g_2^{-7r_1+6r_2} g_3^{-c_1+8r_1+4r_2} g_4^{r_2} g_5^{r_3}$ and $a_2 = h^{-s_1} g_1^{-c_2+17s_1+15s_2} g_2^{2s_1+3s_2} g_3^{s_3} g_4^{s_2} g_5^{s_4}$, and accepts if both verifications hold. Here $\mathcal{P}$ is allowed to choose one of c_1, c_2, freely by itself, subject to $c_1 \oplus c_2 = c$, it correspondingly chooses the appropriate response set at random by itself, and computes one of the two numbers, a_1, a_2, in such a manner from the self-chosen challenge and response set that the verification by $\mathcal{V}$ holds. Similarly, to demonstrate Q2, $\mathcal{P}$ must demonstrate knowledge

of a representation of $g_1 g_4$ with respect to $(h^{-1} g_1^{10} g_4^7, g_1 g_2, g_1^{-3} g_3 g_4^{-3}, g_5)$. $\mathcal{P}$ hereto provides a number, a_3 in G_q, to $\mathcal{V}$; responsive to a challenge, c_3, $\mathcal{P}$ computes a response set such that $a_3 = h^{-t_1} g_1^{-c_3+10t_1+t_2-3t_3} g_2^{t_2} g_3^{7t_1+3t_3} g_4^{-c_3+7t_1-3t_3} g_5^{t_4}$; and $\mathcal{V}$ verifies the response set. By performing the two proofs in parallel, c can be taken equal to c_3, so that only one challenge is needed from $\mathcal{V}$. Furthermore, a signed proof of F consists of (a_1, a_2, a_3), challenge $c = c_3$ computed as a one-way hashvalue of at least $(h, (a_1, a_2, a_3))$, (one of) c_1 and c_2, and the three response sets. Depending on how much conviction the signed proof should convey, the hash should or need not operate on a description of F, or perhaps only on partial information about F. [End of example 3]

The inventive technique can more generally be used to demonstrate knowledge of at least m out of $l \geq m$ representations. Hereto $\mathcal{P}$ simulates the proof of knowledge for the at most $l - m$ linear relations that do not hold, by generating self-chosen challenge values and response sets; and $\mathcal{P}$ performs a genuine proof of knowledge for the linear relations that do hold. (Alternatively, $\mathcal{P}$ can perform genuine proofs of knowledge for exactly m relations that hold, and simulate the proof for the $l - m$ remaining relations, regardless of whether one of these holds as well.) The l challenge values must then be forced to be such that $\mathcal{P}$ can choose any $l - m$ freely, but not more, without revealing which it has chosen freely. Hereto the l challenge values can be required, for example, to satisfy m pre-specified independent linear relations. Note that this generalized method need never be applied for m greater than 1: any satisfiable formula can be demonstrated by applying the inventive technique for demonstrating a formula consisting of zero or more "AND" connectives and at most one "NOT" connective, and the inventive technique for demonstrating knowledge of 1 out of l representations. However, the technique for proving knowledge of $m \geq 2$ representations out of l can be used to improve efficiency. For example, to demonstrate the formula "(R_1 AND R_2) OR (R_1 AND R_3) OR (R_2 AND R_3)," $\mathcal{P}$ can use the $m = 2$, $l = 3$ situation; application of the $m = 1$, $l = 3$ case might result in a less efficient proof (for example when both R_1 and R_2 contain a "NOT" connective).

For greater efficiency, $\mathcal{V}$ can process the l verifications simultaneously, similar a batch-processing technique known in the art for verifying DSA signatures (See, Naccache, D., M'Raïhi, D., Vaudenay, S. and Raphaeli, D., "Can D.S.A. be Improved? - Complexity Trade-Offs with the Digital Signature Standard -," Advances in Cryptology –EUROCRYPT '94,

Lecture Notes in Computer Science. Springer-Verlag, 1995, pp. 77–85). To batch-process $l \geq 2$ verification relations, $a_i \stackrel{?}{=} h^{u_i} \prod_{j=1}^{k} g_j^{v_{ij}}$ for $1 \leq i \leq l$, where the u_i's and the v_{ij}'s are expressions in terms of the responses of $\mathcal{P}$, $\mathcal{V}$ generates random numbers $p_1, \ldots, p_k$ and verifies the single relation $\prod_{i=1}^{l} a_i^{p_i} \stackrel{?}{=} h^{\sum_{i=1}^{l} p_i u_i} \prod_{j=1}^{k} g_j^{\sum_{i=1}^{l} p_i v_{ij}}$. The size of the set from which the p_i's are chosen determines the security level, and a small optimization is to take p_1 always equal to 1.

The inventive method of FIG. 4 is related to a method known in the art (See, Cramer, R., Damgard, I., and Schoenmakers, B., "Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols," Advance in Cryptology – CRYPTO '94, Lecture Notes in Computer Science, Springer-Verlag, 1995). As will be appreciated, the methods differ in two important respects. The reference method applies only to situations in which $\mathcal{P}$ demonstrates knowledge of m out of $l \geq m$ subsets of k secret keys, each of which corresponds to one of k public keys. In other words, it allows $\mathcal{P}$ to prove monotone formulas from propositional logic (meaning that only "AND" and "OR" connectives are present), the formulas always being satisfiable because the atomic propositions are of the form "$\mathcal{P}$ knows the secret key corresponding to the i-th public key," for $1 \leq i \leq k$. In contrast, with the inventive techniques only one public key is present, to which a multitude of secret keys correspond, and any satisfiable formulas from propositional logic can be demonstrated, the atomic propositions being linear relations over $\mathbb{Z}_q$. However, as will be appreciated clear, the reference method can be used to optimize the inventive method for demonstrating arbitrary satisfiable formulas from propositional logic for linear relations. Namely, one can interpret an atomic proposition for the reference method to correspond to a sub-formula that is a system consisting of linear relations and at most one linear inequality; as has been detailed in the description of FIG. 3, one such sub-formula requires $\mathcal{P}$ to demonstrate knowledge of a representation of a number with respect to a tuple consisting of k base numbers, in other words to demonstrate knowledge of a secret key corresponding to a distorted public key. For example, a formula F in conjunctive normal form "(R_{11} AND R_{21} AND $\cdots$ AND R_{l1}) OR $\cdots$ OR (R_{12} AND R_{22} AND $\cdots$ AND R_{m2})," where each R_{ij} is a sub-formula from propositional logic that connects linear relations over $\mathbb{Z}_q$ by zero or more "AND" connectives and at most one "NOT" connective, can be demonstrated efficiently by applying a construction known in the art (see, Benaloh, J., and Leichter, J., "Generalized Secret Sharing and Monotone

Functions. Advance in Cryptology – CRYPTO '88, Lecture Notes in Computer Science, Springer-Verlag, pp. 25–35) in order to restrict the manner in which $\mathcal{P}$ can generate self-chosen challenges from the challenge that is supplied by $\mathcal{V}$. An example follows.

Example 4: Consider the linear relations of example 3, $S_1, \ldots, S_7$, with $h = \prod_{i=1}^{5} g_i^{x_i}$. Suppose now that $\mathcal{P}$ has to demonstrate the formula F = "(S_1 AND (NOT S_2) AND (NOT S_3)) OR (S_4 AND S_5 AND (NOT S_6) AND (NOT S_7))." F can be re-written in the format "(R_{11} AND R_{21}) OR (R_{12} AND R_{22})," where R_{11} = "S_1 AND (NOT S_2)," R_{21} = "NOT S_3," R_{12} = "S_4 AND (NOT S_6)," and R_{22} = "S_5 AND (NOT S_7)" are all sub-formulas of F. Note that if, say, sub-formula R_{12} would not contain a "NOT" connective, then sub-formulas R_{12} and R_{22} could have been lumped together into one sub-formula. Now, in terms of the reference method this means that $\mathcal{P}$ has to demonstrate knowledge of a first and a second secret key, or a third and a fourth secret key. Hence $\mathcal{P}$, after having computed four numbers $a_1, \ldots, a_4$, corresponding to R_{11}, R_{12}, R_{22}, R_{22}, respectively (cf. example 2), can perform the demonstration by deriving four challenges, $c_1, \ldots, c_4$ from a challenge c of $\mathcal{V}$. The four challenges must be such that $\mathcal{P}$ can either choose c_1, c_2 by itself but not c_3, c_4, or can choose c_3, c_4 by itself but not c_1, c_2. For the part of the formula that is true, say "R_{12} AND R_{22}," $\mathcal{P}$ can then perform a genuine proof of knowledge, and the "R_{11} AND R_{21}" part, which may not be true, can be simulated by using self-chosen responses and self-chosen challenges c_1, c_2. According to the construction of Benaloh and Leichter in the abovementioned reference, $c_1, \ldots, c_4$ can be taken to satisfy the restrictions $c_1 = c_2$, $c_3 = c_4$, and $c_1 \oplus c_3 = c \bmod 2^t$, which must be checked by $\mathcal{V}$ as part of the verification process. [End of example 4]

As will be clear to those of ordinary skill in the art, all the techniques described in relation to Figure 1 also apply to the inventive technique of FIG. 4, and to its generalizations. Also, the technique of using correction factors, demonstrated by the inventive technique of FIG. 2, can be used. $\mathcal{V}$ can also restrict the number of correction factors, and/or the relations they need to satisfy, in order to restrict the set of formulas that can be anticipated by $\mathcal{P}$, as described for FIG. 2.

Application of the inventive techniques, as well as the described generalizations, variations and related techniques, based on the difficulty of computing RSA-roots is believed to be straightforward for those of ordinary skill in the art, when studying the drawings in conjunction with the

detailed descriptions: a detailed description is therefore omitted.

Turning now to FIG. 5, a flowchart of a protocol for demonstrating that a system of linear relations applies to a set of secrets, without revealing additional information about the secrets unless performed more than once, will now be described in detail.

As in FIG. 1, $\mathcal{P}$ at the outset knows a representation $(x_1, \ldots, x_k)$ of a number h in G_q with respect to $(g_1, \ldots, g_k)$. The inventive technique enables $\mathcal{P}$ to demonstrate to $\mathcal{V}$ that its representation satisfies the $l \geq 1$ independent linear relations that have been described with respect to FIG. 1, such that the demonstration cannot be performed responsive to two different challenges without revealing the entire representation of $\mathcal{P}$. The technique is as follows.

$\mathcal{P}$ initially commits to a number a in G_q, computed according to $\prod_{i=1}^{k} g_i^{w_i}$ for randomly generated numbers $w_1, \ldots, w_k$ in $\mathbb{Z}_q$. Hereto a party authorized to issue certificates can issue a digital certificate on at least (h, a), which must be revealed by $\mathcal{P}$ when performing the proof. For concreteness, but without loss of generality, it will be assumed that $\mathcal{P}$ does not need to know the particular system of relations to be demonstrated at the time of computing a. Applying the inventive technique of FIG. 2, when performing its demonstration $\mathcal{P}$ also sends a set of l correction factors, all in $\mathbb{Z}_q$, to $\mathcal{V}$; these correction factors are used to adjust a such that the representation known to $\mathcal{P}$ of the "new" a is suitable to perform the demonstration. The adjustment consist of dividing a by the product $\prod_{i=1}^{l} g_{\pi(k-l+i)}^{e_i}$. By performing the protocol of FIG. 1, with the adjusted number playing the role of a, and expanding the resulting expressions, the protocol of FIG. 5 results.

The first three lines of Box 51 show $\mathcal{P}$ computing computing l correction factors, $e_1, \ldots, e_l$, all in $\mathbb{Z}_q$. As shown in the fourth line, $\mathcal{P}$ then sends (h, a), a digital certificate on (h, a), and the l correction factors to $\mathcal{V}$. Depending on the application, $\mathcal{V}$ may be able to retrieve (h, a) and the certificate from a public-key directory or the like, or a certificate is not needed because $\mathcal{V}$ can check the correctness of (h, a) in some other way.

The first line of Box 52 shows $\mathcal{V}$ generating a challenge number c in $\mathbb{Z}_{2^t}$, and the second line shows $\mathcal{V}$ sending c to $\mathcal{P}$.

The first three lines of Box 53 show $\mathcal{P}$ computing $k - l$ response numbers, $(r_1, \ldots, r_{k-l})$, and the fourth line shows $\mathcal{P}$ sending these to $\mathcal{V}$.

The first three lines of Box 54 show $\mathcal{V}$ computing l numbers,

$r_{k-l+1}, \ldots, r_k$. The fourth line shows $\mathcal{V}$ verifying the $k-l$ responses provided by $\mathcal{P}$. If the verification holds, then $\mathcal{V}$ accepts the proof. Not shown is $\mathcal{V}$ storing (h, a), c, and $(r_1, \ldots, r_k)$ in a database. (Alternatively, only a hash of (h, a) is stored, instead of (h, a). Furthermore, depending on what information of $\mathcal{P}$'s representation should be computable when $\mathcal{P}$ performs two demonstrations using the same (h, a), it may suffice or be more efficient to store information derived from c and $(r_1, \ldots, r_k)$. These variations are discussed in detail shortly.)

As will be clear to those of ordinary skill in the art, $\mathcal{V}$ could access the database directly in Box 52, to check whether $\mathcal{P}$ has already used the same (h, a) before; in that case only (h, a) (or only h or a hash of it, depending on the application) needs to be stored in the database. The inventive technique of FIG. 5 is especially concerned with situations in which such on-line checking may not (always) be feasible. This is plausible when $\mathcal{V}$ is distributed, and many demonstrations are performed by prover parties. In that case it is more practical for a central party (typically representing the issuer of the certificate on (h, a)) to maintain the database, and for $\mathcal{V}$ to contact the central party only occasionally, in order to send to it the transcripts of protocol executions (and possibly also a description of the formula demonstrated, to inform the central party of the particular system that has been demonstrated). Exemplary applications are off-line electronic cash systems and, more generally, credential mechanisms, as detailed in US patent application Ser. No. 08/203,231.

In case $\mathcal{P}$ performs the demonstration a second time, using the same (h, a), and responsive to a challenge c' differing modulo 2^t from c, but not necessarily applying to the same system of linear relations, the corresponding numbers, $(r'_1, \ldots, r'_k)$, allow the representation $(x_1, \ldots, x_k)$ to be computed afterwards by the party controlling the database. Specifically, x_i can be computed as $(r_i - r'_i)(c - c')^{-1} \bmod q$, for $1 \leq i \leq k$. In case it suffices for only one particular x_i to be computable when $\mathcal{P}$ performs the demonstration more than once, it suffices to store only r_i instead of the entire vector $(r_1, \ldots, r_k)$. Still greater database storage efficiency may be achieved by storing only part of (h, a), or a one-way hash of h or (h, a), in order to check for collisions.

Especially in case the interests of the central party, which maintains the database, are not the same as the interests of $\mathcal{V}$, the central party may prefer to require $\mathcal{V}$ to perform a signed proof with $\mathcal{P}$. To ensure that the signed proof irrefutably shows which particular formula has been

demonstrated, the challenge c in Box 52 can be determined as a one-way hashvalue of at least a, a description of the formula demonstrated, the vector $(e_1, \ldots, e_l)$, and possibly also h and additional information (such as time and date of the protocol execution, the identity of the holder of $\mathcal{V}$, and so on). In this cased, when $\mathcal{V}$ sends the transcript of the protocol execution to the central party, it must also send along the description of the formula that has been demonstrated.

As will be clear to those of ordinary skill in the art, a in Box 51 may be computed by $\mathcal{P}$ to be of forms other than $\prod_{i=1}^{k} g_i^{w_i}$; $\mathcal{P}$ must then compute the l correction factors correspondingly, in Box 51. The protocol in FIG. 5 enables $\mathcal{P}$ to perform any system of linear relations, without needing to know the system in advance. Clearly, if $\mathcal{P}$ should only be allowed to demonstrate one particular system of linear relations, such that when performed twice the representation is revealed, no correction factors should be allowed in the protocol; this requires $\mathcal{P}$ to compute a of the appropriate form right from the start (cf. Box 11). More generally, as already detailed in the description of FIG. 2, $\mathcal{V}$ may impose restrictions on the number and form of the correction factors that $\mathcal{P}$ may use; in this manner it can be ensured that $\mathcal{P}$ can demonstrate any one of a restricted set of formulas without revealing additional information, while the set of secrets of $\mathcal{P}$ can be computed if it performs any two formulas from the restricted set for the same (h, a). The certifying party can encode the number and the form of correction factors that $\mathcal{P}$ is allowed to use into the certificate, or into one of the x_i's.

As will be clear to those of ordinary skill in the art, all the techniques described in relation to Figure 1 also apply to the inventive technique of FIG. 4, and to its generalizations. Furthermore, the technique of using correction factors, demonstrated by the inventive technique of FIG. 2, can be used. Of special interest here is the technique for signed proofs to not hash $(e_1, \ldots, e_l)$ along when determining c, as already described with respect to FIG. 2. This may be desirable in applications in which $\mathcal{V}$ wants to hide from the central party the formula that has been demonstrated by $\mathcal{P}$, while enabling the central party to compute (part of) the representation of $\mathcal{P}$ in case $\mathcal{P}$ uses (h, a) more than once. An exemplary such application is an off-line electronic cash system, as described in US patent application Ser. No. 08/203,231, where (h, a) represents a digital coin. The numbers $x_1, \ldots, x_k$ may represent characteristics of $\mathcal{P}$, such as income category etcetera, some of which $\mathcal{P}$ can choose to reveal to $\mathcal{V}$ (the merchant) in order

to get a discount; $\mathcal{V}$ may prefer not to reveal the income category of $\mathcal{P}$ to the central party (the bank), so as to protect the privacy of its customers.

In a manner similar to that described in US patent application Ser. No. 08/203,231, it can be ensured that the representation of $\mathcal{P}$ can be computed if and only if $\mathcal{P}$ performs more than u demonstrations, for an arbitrary pre-determined integer u. Hereto, $\mathcal{P}$ should commit on $(h, a_1, \ldots, a_u)$, where each of $a_1, \ldots, a_u$ is determined in the same way as a in FIG. 5. When performing the i-th demonstration with respect to $(h, a_1, \ldots, a_u)$, where $1 \leq i \leq u$, $\mathcal{P}$ uses the pair (h, a_i). Several approaches exist to implementing the commitment: one is to build a binary tree with the a_i's as the leaves, each other node being the one-way hash of the child nodes, and having the root node and h certified; another is to have $(h, a_1, \ldots, a_u)$ certified directly.

Embodying the inventive techniques based on the difficulty of computing RSA-roots is believed to be straightforward for those of ordinary skill in the art when studying the drawings in conjunction with the detailed descriptions; hence a detailed description is omitted here.

The protocol of FIG. 5 does not allow $\mathcal{P}$ to demonstrate formulas from propositional logic that have zero or more "AND" connectives and one "NOT" connective, in such a manner that additional information about its secrets is revealed if and only if demonstrating more than a predetermined number of times. Turning now to FIG. 6, a flowchart of a protocol for demonstrating that a satisfiable formula applies to a set of secrets, the formula connecting linear relations by zero or more "AND" connectives and at most one "NOT" connective, without revealing additional information about the secrets unless performed more than once, will now be described in detail.

As has been described in detail in relation to FIG. 3, any such formula can be expressed by the matrix equation

$$\begin{pmatrix} \alpha_{11} & \ldots & \alpha_{1,k-l} & 1 & 0 & \ldots & 0 \\ \vdots & & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & & \vdots & \vdots & \ddots & \ddots & 0 \\ \alpha_{l1} & \ldots & \alpha_{l,k-l} & 0 & \ldots & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{\pi(1)} \\ \vdots \\ \vdots \\ x_{\pi(k)} \end{pmatrix} = \begin{pmatrix} b_1 - f_1\epsilon \\ \vdots \\ \vdots \\ b_l - f_l\epsilon \end{pmatrix} \bmod q,$$

where $f_1, \ldots, f_l$ are numbers in $\mathbb{Z}_q$, and ϵ is a non-zero difference term. To enable $\mathcal{P}$ to demonstrate any satisfiable formula that connects linear relations by zero or more "AND" connectives and at most one "NOT" connective, it is assumed that $\mathcal{P}$ is allowed to reveal up to $l+1$ correction

factors, l of which are used to adjust the exponents pertaining to $(g_{\pi(k-l+1)}, \ldots, g_{\pi(k)})$, and one to adjust the exponent pertaining to h. (Again, alternatively restrictions may be placed upon the correction factors to limit the class of formulas that can be demonstrated, and by not allowing any restriction factors $\mathcal{P}$ can only anticipate one formula when computing a; the necessary modifications are believed straightforward to those of ordinary skill in the art.) Specifically, $\mathcal{P}$ at the start knows a randomly generated representation $(w_1, \ldots, w_k)$ of a with respect to $(g_1, \ldots, g_k)$, and the protocol is as follows.

The first line of Box **61** shows $\mathcal{P}$ computing a random number w_0 in $\mathbb{Z}_q$, which it will use to adjust the exponent corresponding to h. The next four lines show $\mathcal{P}$ computing computing $l+1$ correction factors, $e_0, e_1, \ldots, e_l$, all in $\mathbb{Z}_q$. As shown in the fifth line, $\mathcal{P}$ then sends (h, a), a digital certificate on (h, a), and the $l+1$ correction factors to $\mathcal{V}$: the remarks made for Box **51** apply here also.

The first line of Box **62** shows $\mathcal{V}$ generating a challenge number c in $\mathbb{Z}_{2^t}$, and the second line shows $\mathcal{V}$ sending c to $\mathcal{P}$.

The first line of Box **63** shows $\mathcal{P}$ computing the multiplicative inverse, δ, of the difference number ϵ. The difference number can be computed by $\mathcal{P}$ from one of the l linear equations that are specified by the matrix relation. The next four lines show $\mathcal{P}$ computing $k-l+1$ response numbers, and the sixth line shows $\mathcal{P}$ sending the response numbers to $\mathcal{V}$. The first five lines of Box **64** show $\mathcal{V}$ computing l numbers, $r_{k-l+1}, \ldots, r_k$ (the descriptions of the computations for r_{k-l+1} and r_k take up two lines each in the figure, which is only for display convenience). The sixth line shows $\mathcal{V}$ verifying the $k-l+1$ responses provided by $\mathcal{P}$. If the verification holds, then $\mathcal{V}$ accepts the proof. Not shown is $\mathcal{V}$ storing (h, a), c, and $(e_0 - r_0 \bmod q, r_1, \ldots, r_k)$ in a database.

The techniques detailed and referenced in the description of FIG. **5** all hold here as well. In particular, the inventive technique is particularly useful in case a central party maintains the database: in case $\mathcal{P}$ performs a second demonstration with respect to (h, a), responsive to different challenge, the representation $(x_1, \ldots, x_k)$ can be computed afterwards by the party controlling the database (and, instead of storing $(e_0 - r_0 \bmod q, r_1, \ldots, r_k)$, it suffices to store $(r_1(e_0 - r_0)^{-1} \bmod q, \ldots, r_k(e_0 - r_0)^{-1} \bmod q)$); similar techniques for greater database storage can be used; when the interests of the central party, which maintains the database, are not the same as the interests of $\mathcal{V}$,

the central party should in general require $\mathcal{V}$ to perform a signed proof with $\mathcal{P}$; whether the signed proof convinces of the particular formula that has been demonstrated, only that one out of a subclass of formulas has been demonstrated, or of merely the fact that $\mathcal{P}$ knows a representation of h, can be determined by which information about the correction factors and/or the demonstrated formula is hashed along; the number of times that $\mathcal{P}$ may use (h, a) in a demonstration can be any pre-determined integer; and, application to a situation in which $\mathcal{P}$ consists of $\mathcal{T}$ and $\mathcal{U}$, and to embodiments related to the difficulty of RSA-roots, are straightforward.

Still more generally, $\mathcal{P}$ can be enabled to demonstrate any satisfiable formula from propositional logic, atomic propositions being linear relations over $\mathbb{Z}_q$, in such a manner that more than one such demonstration reveals $(x_1, \ldots, x_k)$.

Any satisfiable formula from propositional logic that has "AND" connectives, more than one "NOT" connective, and no "OR" connectives, can be split into several separate proofs for sub-formulas, such that each proof for a sub-formula can be dealt with as in FIG. 5 or 6 (cf. the description of FIG. 3). For a signed demonstration of the formula with respect to (h, a), the proofs for the sub-formulas can all be performed in parallel, responsive to a single challenge c. When the demonstration for the formula has been separated into, say, m parallel proofs of sub-formulas, the proof of the i-th sub-formulas requires $\mathcal{P}$ to send a number a_i, in the same way as $\mathcal{P}$ has to send a number a in Boxes **51** and **61**. For a signed proof, the challenge c of $\mathcal{V}$ must now be computed as a one-way hashvalue of at least $(a_1, \ldots, a_m)$. Furthermore, $a_1, \ldots, a_m$ must be chosen by $\mathcal{P}$ such that their product, $\prod_{i=1}^{m} a_i$, is equal to an adjusted form of a. More precisely, $\mathcal{P}$ is allowed to reveal in the demonstration of the formula, up to $k+1$ correction factors, $(e_0, \ldots, e_k)$, that are used to adjust a to $a(h^{e_0} \prod_{i=1}^{k} g_i^{e_i})^{-1}$, and $\prod_{i=1}^{m} a_i$ must be equal to $a(h^{e_0} \prod_{i=1}^{k} g_i^{e_i})^{-1}$. As will be clear to those of ordinary skill in the art, this allows the representation $(x_1, \ldots, x_k)$ to be computed from any two transcripts responsive to different challenges. Specifically, in accordance with FIG. 6, for each i between 1 and m, $\mathcal{P}$ must reveal a response set for the i-th sub-formula proof, and $\mathcal{V}$ can verify the sub-formula proof in accordance with the verification relation described by the fourth line of Box **64**; by multiplying all verification relations, one for each sub-formula proof and hence m in total, and multiplying the outcome by $(\prod_{i=0}^{k} g_i^{e_i})^{-1}$, one relation of the form $a = h^{-z_0} g_1^{z_1} \cdots g_k^{z_k}$ results (this

computation need not actually be performed by $\mathcal{V}$) where each of the $k+1$ exponents $z_0, \ldots, z_k$ is a linear combination involving correction factors, response numbers of each of the i sub-formula proofs, the challenge value, and coefficients that specify the formula demonstrated by $\mathcal{P}$; and, by storing the challenge value and $(z_1/z_0 \bmod q, \ldots, z_k/z_0 \bmod q)$, the database maintaining party can compute the representation, $(x_1, \ldots, x_k)$, in case $\mathcal{P}$ performs two demonstrations with respect to (h, a).

Arbitrary satisfiable formulas from propositional logic can also involve "OR" connectives. To deal with such formulas, the inventive technique for demonstrating arbitrary formulas that has been described in relation to FIG. 4 can be applied, together with the inventive technique described in the preceding paragraph for ensuring that any two demonstrations reveal $(x_1, \ldots, x_k)$. Specifically, the formula to be proven by $\mathcal{P}$ is split up into sub-formula proofs, say m in total, each of which can be performed responsive to a challenge number. The m challenge numbers must be determined by $\mathcal{P}$ from a single challenge value of $\mathcal{V}$ (a one-way hashvalue in case of a signed proof), subject to restrictions that are in accordance with the formula that $\mathcal{P}$ has to demonstrate (cf. example 3). As detailed in the preceding paragraph, $\mathcal{P}$ may release up to $k+1$ correction factors in order to adjust a (and suitable restrictions may be imposed), and the adjusted a must be equal to $\prod_{i=1}^{k} a_i$, where a_i is a number sent by $\mathcal{P}$ in the i-th sub-formula proof (playing the same role as a in Box 61). Again, from all the m response sets, the description of the formula that is proven by $\mathcal{P}$, its challenge value, and the correction factors (if any), one obtains a single relation of the form $a = h^{-z_0} g_1^{z_1} \cdots g_k^{z_k}$. Storing the challenge value and $(z_1/z_0 \bmod q, \ldots, z_k/z_0 \bmod q)$ is sufficient to enable computation of $(x_1, \ldots, x_k)$ in case $\mathcal{P}$ performs any two demonstrations of satisfiable formula from propositional logic with respect to (h, a).

Again, a detailed description is omitted, as are a description of embodiments of which the security is related to the difficulty of computing RSA-roots, and a description of applying the inventive techniques to a situation in which $\mathcal{P}$ consists of two parties; those of ordinary skill in the art are believed to find little difficulty in providing these descriptions by studying the drawings in conjunction with the corresponding descriptions.

This concludes the detailed descriptions of the preferred embodiments. While these descriptions of the present invention have been given as examples, it will be appreciated by those of ordinary skill in the art that

various modifications, alternate configurations, and equivalents may be employed without departing from the spirit and scope of the present invention. For example, there are many essentially equivalent orders to evaluate expressions; ways to evaluate expressions; ways to order expressions, tests, and transmissions within flowchart boxes; ways to group operations into flowchart boxes; and ways to order flowchart boxes. The particular choices that have been made here are merely for clarity in exposition.

Certain variations and substitutions may be apparent to those of ordinary skill in the art. For example, instead of sending a, and hashing a when computing the challenge c for a signed proof, a collision-intractable hash of a can be used; of course, the verifications by $\mathcal{V}$ then have to be adjusted correspondingly. Furthermore, the inventive techniques can also be applied in a group that is not of prime order. In that case the secrets of $\mathcal{P}$ are no longer in a field, but in a ring that can have zero divisors, whence the choice for a group of prime order is believed preferable in many applications; however, this need not be a problem, because the zero-divisors may be hard to find (such as is the case when working in a multiplicative group having an order equal to equal to the product of two large primes), or it may even be advantageous for the application at hand that zero-divisors can be found. For example, by demonstrating that $Ax_1 = B \bmod 2q$, where A is even, $\mathcal{P}$ demonstrates that x_1 has one of two values, differing by q, without revealing which one. Likewise, in case the inventive techniques are based on RSA groups, the exponent v need not necessarily be a prime; for example, v can instead be taken to be the product of two primes. Again, the secrets of $\mathcal{P}$ then are in a ring that has zero divisors, and by taking v to be the product of two large primes it can be made infeasible for $\mathcal{P}$ to find zero-divisors, and sometimes it may be useful that zero-divisors can be found. Other variations and substitutions have been indicated and sometimes described in detail in the text.

It will also be obvious to those of ordinary skill in the art how parts of the inventive techniques and protocols disclosed here can be used to advantage. For example, the credential techniques described in US patent application Ser. No. 08/203,231 can be applied in conjunction with the inventive techniques of the present application: h, or (h, a) can be certified using a secret-key certificate, as described in US patent application Ser. No. 08/321,855; certificates can be issued by means of a restrictive blind issuing protocol, meaning that at least some of $x_1, \ldots, x_k$ are encoded into h by the

certificate issuing authority, while the certificate and (h, a) are hidden from the certificate issuing authority; and, when $\mathcal{V}$ is convinced that $\mathcal{P}$ knows representations of each of two numbers, h_1 and h_2, with respect to $(g_1, \ldots, g_k)$, linear relations can be demonstrated for the numbers in the two representations, by applying the inventive techniques to the quotient $h_1 h_2^{-1}$. One particular application of proving formulas for $h_1 h_2^{-1}$ is when h_2 is a pseudonym of $\mathcal{P}$: this allows $\mathcal{P}$ to demonstrate, possibly in the form of a sub-formula of a formula that it has to demonstrate, that the exponent indicating the identity of the pseudonym-owner is the same in both representations.

Another example pertains to so-called undeniable signatures, well-known in the art. With $h = g_1^{x_1} g_2^{x_2} g_3^{x_3}$ being the public key of the signer, and (x_1, x_2, x_3) its secret key, an undeniable signature on a message m can be taken to be any linear relation involving two of the numbers x_1, x_2, x_3, such that m is one of the coefficients of the linear relation (for example, $x_1 + m x_2 \bmod q$ can be taken). To confirm the signature, the method of FIG. 1 can be used; and to deny a signature, the method of FIG. 3 can be used. In both case, the zero-knowledge variation of the protocol should be used, which as described requires either a repetition of the four steps for small challenges, or a five-step protocol in which the first step is a commitment of $\mathcal{V}$ on its challenge. As will be appreciated, this method improves significantly upon an undeniable signature scheme well-known in the art (see, Chaum, D., Heyst, E., and Pfitzmann, B., "Cryptographically undeniable signatures, unconditionally secure for the signer," Advances in Cryptology-Crypto '91, Lecture Notes in Computer Science 576, Springer-Verlag, pp. 470–484). Furthermore, the same construction can be based on the difficulty of computing RSA-roots, as will be clear to those of ordinary skill in the art.

WHAT IS CLAIMED IS:

1. An apparatus for demonstrating that a plurality of secrets, $(x_1, \ldots, x_k)$, satisfy a formula from propositional logic, said formula connecting linear relations in said secrets by one or more of the logical connectives "AND," "OR" and "NOT," said apparatus comprising:

   committing means for forming

   $$h \leftarrow g_1^{x_1} \cdots g_k^{x_k},$$

   where $g_1, \ldots, g_k$ are substantially random numbers in a group in which computation of discrete logarithms is substantially infeasible; and

   demonstrating means for demonstrating knowledge of a representation of a first number with respect to a first vector of numbers, said first number and each number in said first vector of numbers being a product of powers of numbers in the set, $\{h, g_1, \ldots, g_k\}$, and their inverses.

2. An apparatus as in claim 1, wherein said formula is represented by a system of $l < k$ independent linear relations,

   $$\begin{pmatrix} \alpha_{11} & \ldots & \alpha_{1,k-l} & 1 & 0 & \ldots & 0 \\ \vdots & & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & & \vdots & \vdots & \ddots & \ddots & 0 \\ \alpha_{l1} & \ldots & \alpha_{l,k-l} & 0 & \ldots & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{\pi(1)} \\ \vdots \\ \vdots \\ x_{\pi(k)} \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_l \end{pmatrix} \bmod q,$$

   where $\pi(\cdot)$ is a permutation of $\{1, \ldots, k\}$ and q denotes the order of said group, and said first number is equal to

   $$h(\prod_{i=1}^{l} g_{\pi(k-l+i)}^{b_i})^{-1},$$

   and said first vector of numbers is equal to

   $$(g_{\pi(1)} \prod_{i=1}^{l} g_{\pi(k-l+i)}^{-\alpha_{i1}}, \ldots, g_{\pi(k-l)} \prod_{i=1}^{l} g_{\pi(k-l+i)}^{-\alpha_{i,k-l}}).$$

3. An apparatus as in claim 1, wherein said formula is represented by the linear inequality,

   $$x_{\pi(1)} \neq \alpha_1 + \alpha_2 x_{\pi(2)} \cdots + \alpha_k x_{\pi(k)} \bmod q,$$

where $\pi(\cdot)$ is a permutation of $\{1,\ldots,k\}$ and q denotes the order of said group, and said first number is equal to $g_{\pi(1)}$, and said first vector of numbers is equal to

$$(g_{\pi(1)}^{\alpha_1}h^{-1}, g_{\pi(1)}^{\alpha_2}g_{\pi(2)}, \ldots, g_{\pi(1)}^{\alpha_k}g_{\pi(k)}).$$

4. An apparatus as in claim 1, wherein said formula is represented by a system consisting of at least one independent linear equation and exactly one linear inequality,

$$\begin{pmatrix} \alpha_{11} & \ldots & \alpha_{1,k-l} & 1 & 0 & \ldots & 0 \\ \vdots & & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & & \vdots & \vdots & \ddots & \ddots & 0 \\ \alpha_{l1} & \ldots & \alpha_{l,k-l} & 0 & \ldots & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{\pi(1)} \\ \vdots \\ \vdots \\ x_{\pi(k)} \end{pmatrix} = \begin{pmatrix} b_1 - f_1\epsilon \\ \vdots \\ \vdots \\ b_l - f_l\epsilon \end{pmatrix} \bmod q,$$

where $\pi(\cdot)$ is a permutation of $\{1,\ldots,k\}$, q denotes the order of said group and ϵ is a non-zero difference number in $\mathbb{Z}_q$, and said first number is equal to

$$\prod_{i=1}^{l} g_{\pi(k-l+i)}^{f_i},$$

and said first vector of numbers is equal to

$$(h^{-1}\prod_{i=1}^{l} g_{\pi(k-l+i)}^{b_i}, g_{\pi(1)}\prod_{i=1}^{l} g_{\pi(k-l+i)}^{-\alpha_{i1}}, \ldots, g_{\pi(k-l)}\prod_{i=1}^{l} g_{\pi(k-l+i)}^{-\alpha_{i,k-l}}).$$

5. An apparatus as in one of claims 2 to 4, wherein said formula is a sub-formula of a larger formula, said larger formula containing at least one "OR" logical connective.

6. An apparatus as in claim 1, wherein said formula is represented by "R_1 OR $\cdots$ OR R_l" and each sub-formula R_i, for $1 \leq i \leq l$, connects linear relations in said secrets by zero or more "AND" connectives and at most one "NOT" connective.

7. An apparatus as in claim 1, wherein said formula has at least one "OR" logical connective, said apparatus further comprising simulating means for simulating, using self-chosen challenges and self-chosen responses, the demonstration of knowledge of at least one second number with respect to at least one second vector of numbers.

8. An apparatus as in claim 1, wherein said formula, or a sub-formula of said formula, is represented by

$$(x_{\pi(1)} = \alpha_1 + \sum_{i=2}^{k} \alpha_i x_{\pi(i)} \bmod q) \text{ OR } (x_{\rho(1)} = \beta_1 + \sum_{i=1}^{k} \beta_i x_{\rho(i)} \bmod q),$$

where $\pi(\cdot)$ and $\rho(\cdot)$ are permutations of $\{1, \ldots, k\}$ and q denotes the order of said group.

9. An apparatus as in one of claims 2 to 8, said demonstrating means further comprising:

randomness generating means, for generating at least one substantially random number in $\mathbb{Z}_q$;

initial witness computing means, for computing an initial witness in said group;

response generating means, for computing at least one response number in $\mathbb{Z}_q$, responsive to a challenge number; and

transmission means, for transmitting signals representative of at least said response numbers.

10. An apparatus as in claim 9, said challenge number being computed by application of a one-way hash function to at least h and said initial witness.

11. An apparatus as in claim 10, said one-way hash function further being applied to a partial description of said formula.

12. An apparatus as in claim 10, said one-way hash function further being applied to a description that uniquely represents said formula.

13. An apparatus as in claim 9, said demonstrating means further comprising correction factor computing means, for computing at least one correction factor in $\mathbb{Z}_q$, and said transmission means also transmitting signals representative of said at least one correction factor.

14. An apparatus as in claim 13, said challenge number being computed by application of a one-way hash function to at least h, said initial witness, at least partial information about said at least one correction factor and at least partial information about said formula.

15. An apparatus as in claim 1, said apparatus being operated by at least two parties, each of said at least two parties holding some but not all of said plurality of secrets.

16. An apparatus as in claim 2, said demonstrating means further comprising:

randomness generating means, for generating $k-l$ substantially random numbers, $(w_1, \ldots, w_{k-l})$, each in $\mathbb{Z}_q$:

initial witness computing means, for computing

$$a \leftarrow \prod_{i=1}^{k-l} g_{\pi(i)}^{w_i} \prod_{i=1}^{l} g_{\pi(k-l+i)}^{-\sum_{j=1}^{k-l} a_{i,j} w_j};$$

response generating means, for computing $k-l$ response numbers, $r_1, \ldots, r_{k-l}$, in response to a challenge number, c, according to $r_i \leftarrow c x_{\pi(i)} + w_i \bmod q$, for $1 \leq i \leq k-l$; and

transmission means, for transmitting signals representative of at least said $k-l$ response numbers.

17. An apparatus as in claim 2, said demonstrating means further comprising:

randomness generating means, for generating at least $k-l+z$ substantially random numbers, each in $\mathbb{Z}_q$, where $1 \leq z \leq l$;

initial witness computing means, for computing an initial witness, a, from said at least $k-l+z$ substantially random numbers;

correction factor generating means, for computing z correction factors, each in $\mathbb{Z}_q$, said correction factors satisfying a predetermined form;

response generating means, for computing at least $k-l$ response numbers in response to a challenge number; and

transmission means, for transmitting signals representative of at least said at least $k-l$ response numbers and said z correction factors.

18. An apparatus as in claim 2, said demonstrating means further comprising:

randomness generating means, for generating k substantially random numbers, $(w_1, \ldots, w_k)$, each in $\mathbb{Z}_q$;

initial witness computing means, for computing

$$a \leftarrow \prod_{i=1}^{k} g_i^{w_i};$$

correction factor generating means, for computing l correction factors, $e_1, \ldots, e_l$, according to

$$e_j \leftarrow w_{\pi(k-l+j)} + \sum_{i=1}^{k-l} \alpha_{ji} w_{\pi(i)} \bmod q,$$

for $1 \leq j \leq l$;

response generating means, for computing at least $k - l$ response numbers, each in $\mathbb{Z}_q$, in response to a challenge number, c; and

transmission means, for transmitting signals representative of at least said $k - l$ response numbers and said l correction factors.

19. An apparatus as in claim 3, said demonstrating means further comprising:

randomness generating means, for generating k substantially random numbers, $(w_1, \ldots, w_k)$, each in $\mathbb{Z}_q$;

initial witness computing means, for computing

$$a \leftarrow h^{-w_1} g_{\pi(1)}^{\sum_{i=1}^{k} \alpha_i w_i} \prod_{i=2}^{k} g_{\pi(i)}^{w_i};$$

response generating means, for computing k response numbers, $r_1, \ldots, r_k$, in response to a challenge number, c, according to $r_1 \leftarrow c\delta + w_1 \bmod q$ and $r_i \leftarrow c x_{\pi(i)} \delta + w_i \bmod q$ for $2 \leq i \leq k$, where δ is equal to

$$((\alpha_1 + \sum_{i=2}^{k} \alpha_i x_{\pi(i)}) - x_{\pi(1)})^{-1} \bmod q;$$

and

transmission means, for transmitting signals representative of at least said k response numbers.

20. An apparatus as in claim 4, said demonstrating means further comprising:

randomness generating means, for generating $k+1$ substantially random numbers, $(w_0, \ldots, w_k)$, each in $\mathbb{Z}_q$;

initial witness computing means, for computing

$$a \leftarrow \prod_{i=1}^{k} g_i^{w_i};$$

correction factor computing means, for computing $l+1$ correction factors, $e_0, \ldots, e_l$, according to $e_0 \leftarrow w_0$ and $e_j \leftarrow w_{\pi(k-l+j)} - b_j w_0 + \sum_{i=1}^{k-l} \alpha_{ji} w_{\pi(i)} \bmod q$, for $1 \leq j \leq l$;

response generating means, for computing $k-l+1$ response numbers, $r_0, \ldots, r_{k-l}$, in response to a challenge number, c, according to $r_0 \leftarrow c\delta + w_0 \bmod q$ and $r_i \leftarrow c x_{\pi(i)} \delta + w_{\pi(i)} \bmod q$ for $1 \leq i \leq k-l$, where δ is equal to $\epsilon^{-1} \bmod q$; and

transmission means, for transmitting signals representative of at least said $k-l+1$ response numbers and said $l+1$ correction factors.

21. An apparatus as in claim 8, said demonstrating means further comprising:

randomness generating means, for generating $k-1$ substantially random numbers, $(w_2, \ldots, w_k)$, each in $\mathbb{Z}_q$;

initial witness computing means, for computing

$$a_1 \leftarrow g_{\pi(1)}^{\sum_{i=2}^{k} \alpha_i w_i} \prod_{i=2}^{k} g_{\pi(i)}^{w_i};$$

challenge simulating means, for generating a substantially random self-chosen challenge number, c_2;

response simulating means, for generating $k-1$ substantially random self-chosen response numbers, $s_2, \ldots, s_k$;

initial witness simulating means, for computing

$$a_2 \leftarrow h^{-c_2} g_{\rho(1)}^{c_2 \beta_1 + \sum_{i=2}^{k} \beta_i s_i} \prod_{i=2}^{k} g_{\rho(i)}^{s_i};$$

response generating means, for computing $k-1$ response numbers, $r_2, \ldots, r_{k-l}$, responsive to a challenge number, c, according to $r_i \leftarrow c_1 x_{\pi(i)} + w_i \bmod q$ for $2 \leq i \leq k$, where c_1 is uniquely determined from c and c_2; and

transmission means, for transmitting signals representative of at least $(r_2, \ldots, r_k)$, $(s_2, \ldots, s_k)$ and one of c_1, c_2.

22. An apparatus for demonstrating that a plurality of secrets, $(x_1, \ldots, x_k)$, satisfy a formula from propositional logic, said formula connecting linear relations in said secrets by one or more of the logical connectives "AND," "OR" and "NOT," said apparatus comprising:

committing means for forming

$$h \leftarrow Y_1^{x_1} \cdots Y_k^{x_k} x_{k+1}^{v} \bmod n,$$

where n is the product of at least two distinct prime numbers, v is an integer, $Y_1, \ldots, Y_k$ are substantially random numbers of substantially high order in the group $\mathbb{Z}_n^*$, and x_{k+1} is a number in $\mathbb{Z}_n^*$; and

demonstrating means for demonstrating knowledge of a representation of a first number with respect to a first vector of numbers, said first number and all but one of the numbers in said first vector of numbers being a product of powers of numbers in the set, $\{h, Y_1, \ldots, Y_k\}$, and their inverses, and the last number in said first vector of numbers being v.

23. An apparatus as in claim 22, wherein said formula is represented by a system of $l < k$ independent linear relations,

$$\begin{pmatrix} \alpha_{11} & \ldots & \alpha_{1,k-l} & 1 & 0 & \ldots & 0 \\ \vdots & & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & & \vdots & \vdots & \ddots & \ddots & 0 \\ \alpha_{l1} & \ldots & \alpha_{l,k-l} & 0 & \ldots & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{\pi(1)} \\ \vdots \\ \vdots \\ x_{\pi(k)} \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_l \end{pmatrix} \bmod v,$$

where $\pi(\cdot)$ is a permutation of $\{1, \ldots, k\}$, and said first number is equal to

$$h \left(\prod_{i=1}^{l} Y_{\pi(k-l+i)}^{b_i}\right)^{-1} \bmod n,$$

and said first vector of numbers is equal to

$$(Y_{\pi(1)} \prod_{i=1}^{l} Y_{\pi(k-l+i)}^{-\alpha_{i1}} \bmod n, \ldots, Y_{\pi(k-l)} \prod_{i=1}^{l} Y_{\pi(k-l+i)}^{-\alpha_{i,k-l}} \bmod n, v).$$

24. An apparatus as in claim 22, wherein said formula is represented by the linear inequality,

$$x_{\pi(1)} \neq \alpha_1 + \alpha_2 x_{\pi(2)} \cdots + \alpha_k x_{\pi(k)} \bmod v,$$

where $\pi(\cdot)$ is a permutation of $\{1, \ldots, k\}$, and said first number is equal to $Y_{\pi(1)}$, and said first vector of numbers is equal to

$$(Y_{\pi(1)}^{\alpha_1} h^{-1} \bmod n, Y_{\pi(1)}^{\alpha_2} Y_{\pi(2)} \bmod n, \ldots, Y_{\pi(1)}^{\alpha_k} Y_{\pi(k)} \bmod n, v).$$

25. An apparatus as in claim 22, wherein said formula is represented by a system consisting of at least one independent linear equation and exactly one linear inequality,

$$\begin{pmatrix} \alpha_{11} & \ldots & \alpha_{1,k-l} & 1 & 0 & \ldots & 0 \\ \vdots & & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & & \vdots & \vdots & \ddots & \ddots & 0 \\ \alpha_{l1} & \ldots & \alpha_{l,k-l} & 0 & \ldots & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{\pi(1)} \\ \vdots \\ \vdots \\ x_{\pi(k)} \end{pmatrix} = \begin{pmatrix} b_1 - f_1\epsilon \\ \vdots \\ \vdots \\ b_l - f_l\epsilon \end{pmatrix} \bmod v,$$

where $\pi(\cdot)$ is a permutation of $\{1, \ldots, k\}$ and ϵ is a non-zero difference number in $\mathbb{Z}_v$, and said first number is equal to

$$\prod_{i=1}^{l} Y_{\pi(k-l+i)}^{f_i},$$

and said first vector of numbers is equal to

$$(h^{-1} \prod_{i=1}^{l} Y_{\pi(k-l+i)}^{b_i}, Y_{\pi(1)} \prod_{i=1}^{l} Y_{\pi(k-l+i)}^{-\alpha_{i1}}, \ldots, Y_{\pi(k-l)} \prod_{i=1}^{l} Y_{\pi(k-l+i)}^{-\alpha_{i,k-l}}, v).$$

26. An apparatus as in one of claims 23 to 25, wherein said formula is a sub-formula of a larger formula, said larger formula containing at least one "OR" logical connective.

27. An apparatus as in claim 22, wherein said formula is represented by "R_1 OR $\cdots$ OR R_l" and each sub-formula R_i, for $1 \leq i \leq l$, connects linear relations in said secrets by zero or more "AND" connectives and at most one "NOT" connective.

28. An apparatus as in claim 22, wherein said formula has at least one "OR" logical connective, said apparatus further comprising simulating means for simulating, using self-chosen challenges and self-chosen responses, the demonstration of knowledge of at least one second number with respect to at least one second vector of numbers.

29. An apparatus as in claim 22, wherein said formula, or a sub-formula of said formula, is represented by

$$(x_{\pi(1)} = \alpha_1 + \sum_{i=2}^{k} \alpha_i x_{\pi(i)} \bmod v) \text{ OR } (x_{\rho(1)} = \beta_1 + \sum_{i=1}^{k} \beta_i x_{\rho(i)} \bmod v),$$

where $\pi(\cdot)$ and $\rho(\cdot)$ are permutations of $\{1, \ldots, k\}$.

30. An apparatus as in one of claims 23 to 29, said demonstrating means further comprising:

randomness generating means, for generating at least one substantially random number in $\mathbb{Z}_v$ and one substantially random number in $\mathbb{Z}_n^*$;

initial witness computing means, for computing an initial witness in $\mathbb{Z}_n^*$;

response generating means, for computing at least one response number in $\mathbb{Z}_v$ and one response number in $\mathbb{Z}_n^*$, responsive to a challenge number; and

transmission means, for transmitting signals representative of at least said response numbers.

31. An apparatus as in claim 30, said challenge number being computed by application of a one-way hash function to at least h and said initial witness.

32. An apparatus as in claim 31, said one-way hash function further being applied to a partial description of said formula.

33. An apparatus as in claim 31, said one-way hash function further being applied to a description that uniquely represents said formula.

34. An apparatus as in claim 30, said demonstrating means further comprising correction factor computing means, for computing at least one correction factor in $\mathbb{Z}_v$, and said transmission means also

transmitting signals representative of said at least one correction factor.

35. An apparatus as in claim 34, said challenge number being computed by application of a one-way hash function to at least h, said initial witness, at least partial information about said at least one correction factor and at least partial information about said formula.

36. An apparatus as in claim 22, said apparatus being operated by at least two parties, each of said at least two parties holding some but not all of said plurality of secrets.