



(51) International Patent Classification:
H04L 9/32 (2006.01)

(21) International Application Number:
PCT/EP2014/072222

(22) International Filing Date:
16 October 2014 (16.10.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13306419.6 16 October 2013 (16.10.2013) EP

(71) Applicant: THOMSON LICENSING [FR/FR]; 1-5 rue
Jeanne d'Arc, F-92130 Issy-les-Moulineaux (FR).

(72) Inventors: JOYE, Marc; c/o Technicolor R&D France,
975 avenue des Champs Blancs, CS17616, F-35576
Cesson Seville (FR). LIBERT, Benoît; 114 rue Tête d'Or,
F-69006 Lyon (FR).

(74) Agents: AUMONIER, Sébastien et al.; Technicolor, 1-5
rue Jeanne d'Arc, F-92130 Issy-Les-Moulineaux (FR).

(81) Designated States (unless otherwise indicated, for every
kind of national protection available): AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,

BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM,
DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,
HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR,
KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG,
MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM,
PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC,
SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN,
TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every
kind of regional protection available): ARIPO (BW, GH,
GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ,
TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU,
TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE,
DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU,
LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK,
SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ,
GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— as to applicant's entitlement to apply for and be granted a
patent (Rule 4.17(ii))

Published:

— with international search report (Art. 21(3))

(54) Title: METHOD FOR GENERATING A QUASI-ADAPTIVE NON-INTERACTIVE ZERO-KNOWLEDGE PROOF AND
CORRESPONDING ELECTRONIC DEVICE

(57) Abstract: In one embodiment, it is proposed a method for generating a quasi-adaptive non-interactive zero-knowledge proof, said proof assessing a membership of a first vector to a hard language, which is a subspace of G^n , where G is a group and n is an integer. The method is executed by an electronic device, and is remarkable in that it comprises: generating a derived linearly homomorphic signature on a second vector, said second vector being defined as a function of said first vector, from a linearly homomorphic signatures comprised in a common reference string and a witness associated with said first vector; and generating a non-malleable element by using a public evaluation function of a smooth projective hash function, in combination with a hash function, said first vector, said witness and elements comprised in said common reference string, said proof being a combination of said derived linearly homomorphic signature and said non-malleable element.



Method for generating a quasi-adaptive non-interactive zero-knowledge proof and corresponding electronic device

Technical Field

The disclosure relates to cryptography, and more specifically, with the development of efficient
5 non-interactive zero-knowledge (e.g. "NIZK") proofs of membership in hard languages.

Background

This section is intended to introduce the reader to various aspects of art, which may be related to various aspects of the present invention that are described and/or claimed below. This discussion is believed to be helpful in providing the reader with background information to
10 facilitate a better understanding of the various aspects of the present invention. Accordingly, it should be understood that these statements are to be read in this light, and not as admissions of prior art. In the rest of the document, the following acronyms, which are well known for one skilled in the art of cryptography, are used: PPT means Probabilistic polynomial time; CRS means Common Reference String; NIZK means Non-interactive zero-knowledge (as already
15 mentioned); QA-NIZK means Quasi-adaptive non-interactive zero-knowledge; DLIN means Decision Linear; SDP means Simultaneous Double Pairing; k -SDP means Simultaneous k -wise Pairing; SXDH means Symmetric eXternal Diffie-Hellman; GS means Groth-Sahai and SPHF means Smooth Projective Hash Function. Some reminders on the concepts of Quasi-Adaptive NIZK Proofs, Simulation-Soundness and Relative Soundness, Bilinear Maps and Hardness
20 Assumptions, Linearly Homomorphic Structure-Preserving Signatures, Smooth Projective Hash functions are described in the section "Detailed description of embodiments" of the present document, before the description of the figures.

Non-interactive zero-knowledge (NIZK) proofs play a central role in cryptography. Until recently, the most efficient way to non-interactively prove membership in the linear
25 subspace spanned by vectors of group elements was to use the Groth-Sahai proof systems (the Groth-Sahai proof systems are detailed in the article "Efficient non-interactive proof systems for

“bilinear groups” by J. Groth, and A. Sahai, published in the conference proceedings of Eurocrypt’08, LNCS 4965, pp. 415–432, 2008).

Indeed, if $A \in \mathbb{Z}_p^{t \times n}$ is a matrix of rank $t < n$ (where t and n are integers, and $\mathbb{Z}_p$ is the additive group of integers modulo a prime number p), in order to prove membership in the language $\mathcal{L} = \{\vec{v} \in \mathbb{G}^n \mid \exists \vec{x} \in \mathbb{Z}_p^t \text{ s.t. } \vec{v} = g^{\vec{x}A}\}$, where $\mathbb{G}$ is a group, and g an element of $\mathbb{G}$,
 5 using NIZK proofs, the Groth-Sahai techniques require $O(n + t)$ group elements in their basic form.

They can be made simulation-sound by using techniques suggested in the following articles: “Simulation-sound NIZK proofs for a practical language and constant size group signatures”, by J. Groth, published in the conference proceedings of Asiacrypt 2006, LNCS 4284,
 10 pp. 444–459, 2006, “A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks” by J. Camenisch et al., published in the conference proceedings of Eurocrypt’09, LNCS 5479, pages 351–368, 2009, and “Tightly Secure Public-Key Encryption”, by D. Hofheinz, T. Jager., published in the conference proceedings of
 15 Crypto 2012, LNCS series, pp. 590-607, 2012.

However, these techniques involve Groth-Sahai proofs for quadratic pairing product equations, which results in rather long proofs. One-time simulation-soundness can be more efficiently achieved using techniques suggested in the following articles: “Round-Optimal Password-Based Authenticated Key Exchange”, by J. Katz and V. Vaikuntanathan, published in
 20 the conference proceedings of TCC’11, LNCS 6597, pp. 293–310, 2011, and “Non-Interactive CCA2-Secure Threshold Cryptosystems with Adaptive Security: New Framework and Constructions”, by B. Libert, M. Yung, published in the conference proceedings of TCC 2012, LNCS 7194, pp. 75–93, Springer, 2012.

However, in recent results, Jutla and Roy gave significantly more efficient solutions for
 25 quasi-adaptive NIZK proofs (QA-NIZK), where the common reference string (CRS) may depend on the language of which membership is being proved. While malleable in their original form (just like the Groth-Sahai proofs), the Jutla Roy QA-NIZK proofs can be modified to achieve a

weak form of non-malleability called relative simulation-soundness, which is sufficient for many applications like password-authenticated key exchange protocols and chosen-ciphertext-secure non-interactive threshold encryption.

Indeed, Jutla and Roy suggested an even more efficient way to achieve single-theorem relative soundness in the article : *“Relatively-Sound NIZKs and Password-Based Key-Exchange”* by C. Jutla and A. Roy published in the conference proceedings of PKC’12, LNCS series, pp. 485-503, 2012. More recently in the article *“Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces”*, by C. Jutla, A. Roy, published in the conference proceedings of Asiacrypt’13, LNCS series, 2013 and in the Cryptology ePrint Archive: Report 2013/109, 2013, C. Jutla and A. Roy showed how to make the proofs more compact at the expense of settling for computational soundness (rather than perfect soundness described in the previous mentioned article *“Efficient non-interactive proof systems for bilinear groups”*). Still, in their most efficient variant, the Jutla-Roy proofs still require $O(n - t)$ group elements, which remains linear in n and results in long proofs when $t \ll n$.

Hence, to prove membership in a linear subspace of dimension t spanned by vectors of dimension n , the relatively sound QA-NIZK proofs of Jutla and Roy require $O(n - t)$ group elements.

The proposed technique aims to reduce the number of group elements in relatively sound QA-NIZK proofs.

More precisely, the proposed technique aims to obtain a size of a proof that does not depend on the dimension of a considered subspace.

Indeed, the proposed technique aims to construct a relatively QA-NIZK proof system for a language corresponding to the following set: $\{\vec{v} \in \mathbb{G}^n \mid \vec{x} \in \mathbb{Z}_p^t \text{ s.t. } \vec{v} = g^{\vec{x} \cdot A}\}$ by using proofs made of a constant number of group elements. In other words, one goal of the proposed technique is to obtain a size of the proof that should not depend on the number of equations n or on the number of variables t . At first, this goal seems very difficult to achieve given that it would even outperforms Fiat-Shamir-like proofs derived from Σ protocols in the random oracle

model, as explained in the article "Random oracles are practical: A paradigm for designing efficient protocols" by M. Bellare and P. Rogaway, and published in the ACM CCS, pp. 62-73, 1993. Indeed, the Fiat-Shamir heuristic would give $O(t)$ size proofs here, as the length of proofs is inevitably linear in the number of variables.

5 Summary of invention

References in the specification to "one embodiment", "an embodiment", "an example embodiment", indicate that the embodiment described may include a particular feature, structure, or characteristic, but every embodiment may not necessarily include the particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the
10 same embodiment. Further, when a particular feature, structure, or characteristic is described in connection with an embodiment, it is submitted that it is within the knowledge of one skilled in the art to affect such feature, structure, or characteristic in connection with other embodiments whether or not explicitly described.

The present invention is directed to a method for generating a quasi-adaptive non-
15 interactive zero-knowledge proof, said proof assessing a membership of a first vector to a hard language, which is a subspace of $\mathbb{G}^n$, where $\mathbb{G}$ is a group and n is an integer. The method is executed by an electronic device, and it comprises:

generating a derived linearly homomorphic signature on a second vector, said second vector being defined as a function of said first vector, a linearly homomorphic signatures
20 comprised in a common reference string and a witness associated with said first vector; and

generating a non-malleable element by using a public evaluation function of a smooth projective hash function, in combination with a hash function, said first vector, said witness and elements comprised in said common reference string, said proof being a combination of said derived linearly homomorphic signature and said non-malleable element.

25 In a preferred embodiment, the method for generating is remarkable in that said hard language corresponds to a set which is equal to $\{\vec{v} \in \mathbb{G}^n \mid \vec{x} \in \mathbb{Z}_p^t \text{ s.t. } \vec{v} = g^{\vec{x}.A}\}$, where p is a

prime number, t is an integer, g is a generator of said group $\mathbb{G}$, and A is a $t \times n$ matrix of rank t whose elements are comprised in $\mathbb{Z}_p$.

In a preferred embodiment, the method for generating according is remarkable in that said witness $\vec{x}$ belongs to $\mathbb{Z}_p^t$, and is associated with said first vector $\vec{v}$ by a relationship

$$\vec{v} = g^{\vec{x} \cdot A}.$$

In a preferred embodiment, the method for generating is remarkable in that said linearly homomorphic signatures comprised in a common reference string comprise $2t$ elements, each element being defined by a vector $(z, r_1, \dots, r_k)$, where k is a integer greater or equal to two, corresponding to a linearly homomorphic signature on independent vectors, which are obtained from rows of a matrix g^A .

In a preferred embodiment, the method for generating according is remarkable in that generating a derived linearly homomorphic signature, and generating a non-malleable element further use a label.

In another embodiment, it is proposed a method for ciphering in an non-interactive and adaptively secure threshold cryptosystems with chosen-ciphertext security. Such method is executed by an electronic device, and is remarkable in that it comprises executing a method for generating a quasi-adaptive non-interactive zero knowledge proof as mentioned previously.

According to an exemplary implementation, the different steps of the method are implemented by a computer software program or programs, this software program comprising software instructions designed to be executed by a data processor of a relay module according to the disclosure and being designed to control the execution of the different steps of this method.

Consequently, an aspect of the disclosure also concerns a program liable to be executed by a computer or by a data processor, this program comprising instructions to command the execution of the steps of a method as mentioned here above.

This program can use any programming language whatsoever and be in the form of a source code, object code or code that is intermediate between source code and object code, such as in a partially compiled form or in any other desirable form.

The disclosure also concerns an information medium readable by a data processor and
5 comprising instructions of a program as mentioned here above.

The information medium can be any entity or device capable of storing the program. For example, the medium can comprise a storage means such as a ROM (which stands for "*Read Only Memory*"), for example a CD-ROM (which stands for "*Compact Disc - Read Only Memory*") or a microelectronic circuit ROM or again a magnetic recording means, for example a floppy
10 disk or a hard disk drive.

Furthermore, the information medium may be a transmissible carrier such as an electrical or optical signal that can be conveyed through an electrical or optical cable, by radio or by other means. The program can be especially downloaded into an Internet-type network.

Alternately, the information medium can be an integrated circuit into which the
15 program is incorporated, the circuit being adapted to executing or being used in the execution of the method in question.

According to one embodiment, an embodiment of the disclosure is implemented by means of software and/or hardware components. From this viewpoint, the term "module" can correspond in this document both to a software component and to a hardware component or
20 to a set of hardware and software components.

A software component corresponds to one or more computer programs, one or more sub-programs of a program, or more generally to any element of a program or a software program capable of implementing a function or a set of functions according to what is described here below for the module concerned. One such software component is executed by
25 a data processor of a physical entity (terminal, server, etc.) and is capable of accessing the hardware resources of this physical entity (memories, recording media, communications buses, input/output electronic boards, user interfaces, etc.).

Similarly, a hardware component corresponds to any element of a hardware unit capable of implementing a function or a set of functions according to what is described here below for the module concerned. It may be a programmable hardware component or a component with an integrated circuit for the execution of software, for example an integrated circuit, a smart card, a memory card, an electronic board for executing firmware etc.

It should also be noted that a step of obtaining an element/value in the present document can be viewed either as a step of reading such element/value in a memory unit of an electronic device or a step of receiving such element/value from another electronic device via communication means.

In another embodiment, the present invention is directed to an electronic device comprising a module configured to generate a quasi-adaptive non-interactive zero-knowledge proof, said proof assessing a membership of a first vector to a hard language, which is a subspace of $\mathbb{G}^n$, where $\mathbb{G}$ is a group and n is an integer. Such electronic device comprises:

a module configured to generate a derived linearly homomorphic signature on a second vector, said second vector being defined as a function of said first vector, a linearly homomorphic signatures comprised in a common reference string and a witness associated with said first vector;

a module configured to generate a non-malleable element by using a public evaluation function of a smooth projective hash function, in combination with a hash function, said first vector, said witness and elements comprised in said common reference string, said proof being a combination of said derived linearly homomorphic signature and said non-malleable element.

In a preferred embodiment, the electronic device is remarkable in that said hard language corresponds to a set which is equal to $\{\vec{v} \in \mathbb{G}^n \mid \vec{x} \in \mathbb{Z}_p^t \text{ s.t. } \vec{v} = g^{\vec{x}.A}\}$, where p is a prime number, t is an integer, g is a generator of said group $\mathbb{G}$, and A is a $t \times n$ matrix of rank t whose elements are comprised in $\mathbb{Z}_p$.

In a preferred embodiment, the electronic device is remarkable in that said witness $\vec{x}$ belongs to $\mathbb{Z}_p^t$, and is associated with said first vector $\vec{v}$ by a relationship $\vec{v} = g^{\vec{x}.A}$.

In a preferred embodiment, the electronic device is remarkable in that said linearly homomorphic signatures comprised in a common reference string comprise $2t$ elements, each element being defined by a vector $(z, r_1, \dots, r_k)$, where k is a integer greater or equal to two, corresponding to a linearly homomorphic signature on independent vectors, which are obtained from rows of a matrix g^A .

In a preferred embodiment, the electronic device is remarkable in that the module configured to generate a derived linearly homomorphic signature, and the module configured to generate a non-malleable element further use a label.

In another embodiment, the present invention is directed to an, an electronic device comprising a module configured to cipher in an non-interactive and adaptively secure threshold cryptosystems with chosen-ciphertext security. The electronic device comprises a module configured to generate a quasi-adaptive non-interactive zero-knowledge proof, said proof assessing a membership of a first vector to a hard language, which is a subspace of $\mathbb{G}^n$, where $\mathbb{G}$ is a group and n is an integer, that comprises:

a module configured to generate a derived linearly homomorphic signature on a second vector, said second vector being defined as a function of said first vector, a linearly homomorphic signatures comprised in a common reference string and a witness associated with said first vector; and

a module configured to generate a non-malleable element by using a public evaluation function of a smooth projective hash function, in combination with a hash function, said first vector, said witness and elements comprised in said common reference string, said proof being a combination of said derived linearly homomorphic signature and said non-malleable element.

Brief description of the figures

The above and other aspects of the invention will become more apparent by the following detailed description of exemplary embodiments thereof with reference to the attached drawings in which:

- 5 - Figures 1(a)-(d) present a relatively sound QA-NIZK proof system according to one embodiment of the invention;
- Figures 2(a)-(f) present a non-interactive threshold cryptosystem according to one embodiment of the invention;
- Figure 3 presents an example of an electronic device that can be used to perform one or
10 several steps of methods disclosed in the present document.

Detailed description of embodiments

First of all, before describing in details some embodiments of the invention, some reminders related to concepts related to the background of the present technique are detailed:

a) Reminders on Quasi-Adaptive NIZK Proofs

- 15 Let $\mathcal{R}$ be a relation that takes as input a statement x and a witness w such that $\mathcal{R}(x, w) = 1$ if and only if x belongs to the language $\mathcal{L}$. It should be noted that only languages $\mathcal{L}$ where it may be hard to distinguish random elements of $\mathcal{L}$ from elements outside $\mathcal{L}$ are considered for cryptographic purposes. For example, consider an abelian group $\mathbb{G} = \langle g \rangle$ of prime order p where the discrete logarithm problem is hard. If $\mathbf{A} \in \mathbb{Z}_p^{t \times n}$ is a matrix of rank
20 $t < n$, deciding the membership of a linear subspace $g^{\mathbf{A}} \in \mathbb{G}^{t \times n}$ is believed to be hard for carefully chosen groups: in other words, the language

$$\mathcal{L} = \{ \vec{v} \in \mathbb{G}^n \mid \exists \vec{x} \in \mathbb{Z}_p^t \text{ s.t. } \vec{v} = g^{\vec{x} \cdot \mathbf{A}} \}$$

is not efficiently recognizable. For such languages, proving the membership of a candidate $x \in \mathcal{L}$ is non-trivial. Whenever $x \in \mathcal{L}$, any element w such that $\mathcal{R}(x, w) = 1$ is called a witness for the membership of x in $\mathcal{L}$.

A non-interactive zero-knowledge (NIZK) proof for a relation $\mathcal{R}$ usually consists of three algorithms or methods noted (K, P, V) , where K is a randomized algorithm (or method) that takes as input a security parameter $\lambda \in \mathbb{N}$ (in unary) and outputs a common reference string (CRS); P is a randomized algorithm (or method) used by the prover on input of a statement x and a witness w such that $\mathcal{R}(x, w) = 1$ to generate a proof for the statement $x \in \mathcal{L}$; and the algorithm (or method) V is a deterministic algorithm (or method) which is run by the verifier to output a binary value (which is 1 if and only if the verifier is convinced that $x \in \mathcal{L}$; and 0 otherwise) on input of the CRS, a statement x and a proof π . The CRS should be seen as a set of common public parameters generated by some trusted party. The zero-knowledge property usually refers to the existence of a simulator S that takes as input a true statement $x \in \mathcal{L}$ but no witness. Instead of a witness, the simulator S uses a trapdoor τ associated with the CRS to generate simulated proofs π whose distribution is indistinguishable from real proofs π produced by the actual algorithm P using the witness. The intuition is that a proof π leaks nothing beyond the validity of the statement $x \in \mathcal{L}$.

Quasi-Adaptive NIZK (QA-NIZK) proofs are NIZK proofs where the CRS is allowed to depend on the specific language for which proofs have to be generated. The CRS is divided into a fixed part Γ , produced by an algorithm K_0 , and a language-dependent part ψ , produced by an algorithm K_1 . However, there should be a single simulator for the entire class of languages.

More formally, let $\lambda \in \mathbb{N}$ be a security parameter. For public parameters Γ produced by algorithm K_0 , let $\mathcal{D}_\Gamma$ be a probability distribution over a collection of relations $R = \{\mathcal{R}_\rho\}$ parameterized by a string ρ with an associated language $\mathcal{L}_\rho = \{x \mid \exists w \text{ s.t. } \mathcal{R}_\rho(x, w) = 1\}$.

In the following, it is considered proof systems where the prover and the verifier both take a label lbl as additional input. For example, this label can include the message-carrying part (g^r, M, h^r) of an Elgamal-like encryption. A tuple of algorithms (K_0, K_1, P, V) is a QA-NIZK proof system for R if there exists a PPT simulator (S_1, S_2) , such that, for any PPT adversaries $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{A}_3$, the properties hereunder stands.

In short, quasi-adaptive completeness means that honestly generated proofs are always accepted by the verifier. Quasi-adaptive soundness captures that it should be computationally infeasible for the prover to trick the verifier into accepting a proof for a false statement. As for the quasi-adaptive zero-knowledge property, it requires the existence of a simulator (S_1, S_2) that can emulate the behavior of the real prover P (which always generates proofs using the witnesses) without knowing the witnesses w : instead, (S_1, S_2) uses a simulation trapdoor τ_{sim} hidden in the CRS ψ to create simulated proofs. Yet, no probabilistic polynomial time (PPT) adversary should be unable to see the difference.

Quasi-Adaptive Completeness:

$$\Pr[\Gamma \leftarrow K_0(\lambda); \rho \leftarrow \mathcal{D}_\Gamma; \psi \leftarrow K_1(\Gamma, \rho); (x, w, lbl) \leftarrow \mathcal{A}_1(\Gamma, \psi, \rho); \pi \leftarrow P(\psi, x, w, lbl) \text{ s.t. } V(\psi, x, \pi, lbl) = 1 \text{ if } \mathcal{R}_\rho(x, w) = 1] = 1.$$

Quasi-Adaptive Soundness:

$$\Pr[\Gamma \leftarrow K_0(\lambda); \rho \leftarrow \mathcal{D}_\Gamma; \psi \leftarrow K_1(\Gamma, \rho); (x, w, lbl) \leftarrow \mathcal{A}_2(\Gamma, \psi, \rho) \text{ s.t. } V(\psi, x, \pi, lbl) = 1 \wedge \neg (\exists w \text{ s.t. } \mathcal{R}_\rho(x, w) = 1) \in \text{negl}(\lambda).$$

Quasi-Adaptive Zero-Knowledge:

$$\Pr[\Gamma \leftarrow K_0(\lambda); \rho \leftarrow \mathcal{D}_\Gamma; \psi \leftarrow K_1(\mu, \rho) \text{ s.t. } \mathcal{A}_3^{P(\psi, \dots)}(\Gamma, \psi, \rho) = 1] \approx \Pr[\Gamma \leftarrow K_0(\lambda); \rho \leftarrow \mathcal{D}_\Gamma; (\psi, \tau_{sim}) \leftarrow S_1(\Gamma, \rho) \text{ s.t. } \mathcal{A}_3^{S_2(\psi, \tau_{sim}, \dots)}(\Gamma, \psi, \rho) = 1,$$

where:

- $P(\psi, \dots)$ emulates the actual prover and outputs a proof π on input of $(x, w) \in \mathcal{R}_\rho$, and lbl .
- $S_2(\psi, \tau_{sim}, \dots)$ is an oracle that takes as input $x \in \mathcal{L}_\rho$ (i.e. for which there exists w such that $(x, w) \in \mathcal{R}_\rho$) as well as a label lbl , and outputs a simulated proof $\pi \leftarrow S_2(\psi, \tau_{sim}, x, lbl)$.

It is assumed that the CRS ψ contains an encoding of ρ , which is thus available to V . The definition of Quasi-Adaptive Zero-Knowledge requires a single simulator for the entire family of relations R .

b) Reminders on Simulation-Soundness and Relative Soundness

5 It is often useful to have a property called simulation-soundness, which requires that the adversary be unable to prove false statements even after having seen simulated proofs for possibly false statements.

Unbounded Simulation-Soundness: For any PPT adversary $\mathcal{A}_4$, it holds that

$$10 \quad \Pr[\Gamma \leftarrow K_0(\lambda); \rho \leftarrow \mathcal{D}_\Gamma; (\psi, \tau_{sim}) \leftarrow S_1(\Gamma, \rho); (x, w, lbl) \leftarrow \mathcal{A}_4^{S_2(\psi, \tau_{sim}, \dots)}(\Gamma, \psi, \rho) \text{ s.t. } V(\psi, x, \pi, lbl) = 1 \wedge \neg (\exists w \text{ s.t. } \mathcal{R}_\rho(x, w) = 1) \wedge (x, \pi, lbl) \notin Q] \in \text{negl}(\lambda).$$

where the adversary is allowed unbounded access to an oracle $S_2(\psi, \tau_{sim}, \dots)$ that takes as input statement-label pairs (x, lbl) (where x may be outside $\mathcal{L}_\rho$) and outputs simulated proofs $\pi \leftarrow S_2(\psi, \tau_{sim}, x, lbl)$, before updating the set $Q = Q \cup \{(x, \pi, lbl)\}$ which is initially empty.

15 In the weaker notion of one-time simulation-soundness, only one query to the S_2 oracle is allowed. In some applications (such as password-authenticated key exchange and chosen-ciphertext secure public-key encryption with publicly verifiable ciphertexts), one may settle for a weaker notion, called relative soundness by Jutla and Roy and described in the previous mentioned article entitled: “Relatively-Sound NIZKs and Password-Based Key-Exchange”, which
20 allows for more efficient proofs, especially in the single theorem case. Informally, relatively sound proof systems involve both a public verifier and a private verification algorithm, which has access to a trapdoor. For hard languages, the two verifiers should almost always agree on any adversarially-created proof.

Moreover, the private verifier should not accept a non-trivial proof for a false
25 statement, even if the adversary has already seen proofs for false statements.

A labeled single-theorem relatively sound QA-NIZK proof system is comprised of a quasi-adaptive labeled proof system (K_0, K_1, P, V) along with an efficient private verifier W and an efficient simulator (S_1, S_2) . Moreover, the following properties should hold for any PPT adversaries $(\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4)$.

5 **Quasi Adaptive Relative Single-Theorem Zero-Knowledge:**

$$\begin{aligned} & \Pr[\Gamma \leftarrow K_0(\lambda); \rho \leftarrow \mathcal{D}_\Gamma; \psi \leftarrow K_1(\mu, \rho); (x, w, lbl, s) \leftarrow \mathcal{A}_1^{V(\psi, \dots)}(\Gamma, \psi, \rho); \pi \leftarrow \\ & P(\psi, \rho, x, w, lbl) \text{ s.t. } \mathcal{A}_2^{V(\psi, \dots)}(\pi, s) = 1] \approx \Pr[\Gamma \leftarrow K_0(\lambda); \rho \leftarrow \mathcal{D}_\Gamma; \\ & (\psi, \tau) \leftarrow S_1(\Gamma, \rho); (x, w, lbl, s) \leftarrow \mathcal{A}_1^{W(\psi, \tau, \dots)}(\Gamma, \psi, \rho); = 1; \\ & \pi \leftarrow S_2(\psi, \rho, \tau, x, lbl) \text{ s.t. } \mathcal{A}_1^{W(\psi, \tau, \dots)}(\pi, s) = 1], \end{aligned}$$

10 **Quasi Adaptive Relative Single-Theorem Simulation-Soundness:**

$$\begin{aligned} & \Pr[\Gamma \leftarrow K_0(\lambda); \rho \leftarrow \mathcal{D}_\Gamma; (\psi, \tau) \leftarrow S_1(\Gamma, \rho); (x, lbl, s) \leftarrow \mathcal{A}_3^{W(\psi, \tau, \dots)}(\Gamma, \psi, \rho); \pi \leftarrow \\ & S_2(\psi, \rho, \tau, x, lbl) \text{ s.t. } (x', lbl', \pi') \leftarrow \mathcal{A}_4^{W(\psi, \tau, \dots)} \quad \text{with} \\ & (x, \pi, lbl) \neq (x', \pi', lbl') \wedge \nexists w' \text{ s.t. } \mathcal{R}_\rho(x', w') = 1 \wedge W(\psi, \tau, x', lbl', \pi') = 1] \in \text{negl}(\lambda). \end{aligned}$$

c) Reminders on Bilinear Maps and Hardness Assumptions

15 For simplicity, symmetric bilinear maps $e: \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ over groups of prime order p are used, but extensions to the asymmetric setting $e: \mathbb{G} \times \widehat{\mathbb{G}} \rightarrow \mathbb{G}_T$ are possible if an efficiently computable isomorphism $\Psi: \widehat{\mathbb{G}} \rightarrow \mathbb{G}$ is available. Modulo slight changes, the proposed technique can also work without such an isomorphism but, in this case, they rely on somewhat less standard cryptographic assumptions.

20 The following definition of the Decision Linear Problem (DLIN) in the group $\mathbb{G}$ of order p , also described in the article "Short group signatures", by D. Boneh et al., published in the conference proceedings of Crypto'04, LNCS 3152, pp. 41–55, 2004, can be stated as follows: the Decision Linear Problem (DLIN) is to distinguish between the distributions $(g, g^a, g^b, g^{ac}, g^{bd}, g^{c+d})$ and $(g, g^a, g^b, g^{ac}, g^{bd}, g^z)$, with $a, b, c, d \xleftarrow{R} \mathbb{Z}_p$, $z \xleftarrow{R} \mathbb{Z}_p$, and g a
25 generator of the group $\mathbb{G}$.

The Decision Linear Assumption is the intractability of DLIN for any PPT distinguisher $\mathcal{D}$.

The DLIN problem can be viewed as the problem of deciding whether three vectors $(g^a, 1_{\mathbb{G}}, g), (1_{\mathbb{G}}, g^b, g), (g^{ab}, g^{cd}, g^z)$ form a subspace of dimension two (which is the case when $z = c + d$) or three.

5 Sometimes it is used in the present document the notion of Simultaneous Double Pairing (SDP) assumption, which is weaker than DLIN. As noted in the article *“Group Encryption: Non-Interactive Realization in the Standard Model”* by J. Cathalo et al., and published in the conference proceedings of Asiacrypt’09, LNCS 5912, pp. 179–196, 2009, any algorithm solving SDP immediately yields a DLIN distinguisher.

10 The following definition of the Simultaneous Double Pairing problem (SDP) in $(\mathbb{G}, \mathbb{G}_T)$ can be formulated as follows: given group elements $(g_z, g_r, h_z, h_u) \in \mathbb{G}^4$, the SDP problem is to find a non-trivial triple $(z, r, u) \in \mathbb{G}^3 \setminus \{(1_{\mathbb{G}}, 1_{\mathbb{G}}, 1_{\mathbb{G}})\}$ such that $e(g_z, z) \cdot e(g_r, r) = 1_{\mathbb{G}_T}$ and $e(h_z, z) \cdot e(h_u, u) = 1_{\mathbb{G}_T}$.

15 The DLIN and the SDP problems can both be generalized to dimensions higher than three.

Indeed, as described in the article *“A Cramer-Shoup encryption scheme from the linear assumption and from progressively weaker linear variants”* by H. Shacham, published in the Cryptology ePrint Archive: Report 2007/074, 2007, the k -Linear Problem (k -LIN) in $\mathbb{G}$ is the problem of distinguishing the distributions

$$\mathcal{D}_1 = \left\{ (g_1, \dots, g_k, g, g_1^{a_1}, \dots, g_k^{a_k}, g^{\sum_{i=1}^k a_i}) \in \mathbb{G}^{2k+2} \mid g_1, \dots, g_k, g \xleftarrow{R} \mathbb{G}, a_1, \dots, a_k \xleftarrow{R} \mathbb{Z}_p \right\}$$

20 and

$$\mathcal{D}_2 = \left\{ (g_1, \dots, g_k, g, g_1^{a_1}, \dots, g_k^{a_k}, g^z) \in \mathbb{G}^{2k+2} \mid g_1, \dots, g_k, g \xleftarrow{R} \mathbb{G}, a_1, \dots, a_k, z \xleftarrow{R} \mathbb{Z}_p \right\}.$$

For each $k \geq 2$, the k -Linear Problem is known to remain generically hard even in the presence of an oracle that solves the $(k - 1)$ linear problem. The following assumption which is implied by the k -LIN assumption can be stated:

The Simultaneous k -wise Pairing (k -SDP) problem is, given a random tuple $((g_{1,z}, \dots, g_{k,z}, g_{1,r}, \dots, g_{k,r})) \in_R \mathbb{G}^{2k}$, to find a non-trivial vector $(z, r_1, \dots, r_k) \in \mathbb{G}^{k+1}$ such that

$$e(g_{j,z}, z) \cdot e(g_{j,r}, r_j) = 1_{\mathbb{G}_T} \text{ with } j \in \{1, \dots, k\}, \text{ and } z \neq 1_{\mathbb{G}}.$$

Considerer a k -linar instance $(g_{1,r}, \dots, g_{k,r}, g_{1,r}^{a_1}, \dots, g_{k,r}^{a_k}, \eta) \in \mathbb{G}^{2k+1}$, where the
 5 goal is to decide if $\eta = g^{\sum_{i=1}^k a_i}$ or not. For any non-trivial tuple $(z, r_1, \dots, r_k) \in \mathbb{G}^{k+1}$ satisfying the equalities $e(g_{j,z}^{a_j}, z) \cdot e(g_{j,r}, r_j) = 1_{\mathbb{G}_T}$ for each $j \in \{1, \dots, k\}$, the following relationship stands:

$$\eta = g^{\sum_{i=1}^k a_i} \Leftrightarrow e(g, \prod_{j=1}^k r_j) \cdot e(z, \eta) = 1_{\mathbb{G}_T}.$$

Hence, any algorithm solving k -SDP with non-negligible probability immediately implies
 10 a k -linear distinguisher.

d) Linearly Homomorphic Structure-Preserving Signatures

Linearly homomorphic SPS schemes are homomorphic signatures where messages and signatures live in the domain group $\mathbb{G}$ of a bilinear map. The authors of this report recently described in the article "Linearly Homomorphic Structure-Preserving Signatures and their
 15 Applications" by B. Libert et al., and published in the conference proceedings of Crypto 2013, the following one-time construction and proved its security under the SDP assumption.

Keygen(λ, n): given a security parameter and the dimension $n \in \mathbb{N}$ of vectors to be signed, $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$. Choose $g_z, g_r, h_z, h_u \xleftarrow{R} \mathbb{G}$. Then, for $i = 1$ to n , pick $\chi_i, \gamma_i, \delta_i \xleftarrow{R} \mathbb{Z}_p$, and compute $g_i = g_z^{\chi_i} g_r^{\gamma_i}$ and $h_i = h_z^{\chi_i} h_u^{\delta_i}$. The private key $sk =$
 20 $(\{\chi_i, \gamma_i, \delta_i\}_{i=1}^n)$ while the public key is $pk = (g_z, g_r, h_z, h_u, \{g_i, h_i\}_{i=1}^n)$.

Sign($sk, (M_1, \dots, M_n)$): to sign a vector $(M_1, \dots, M_n) \in \mathbb{G}^n$ using $sk = (\{\chi_i, \gamma_i, \delta_i\}_{i=1}^n)$, compute and return $(z, r, u) = (\prod_{i=1}^n M_i^{-\chi_i}, \prod_{i=1}^n M_i^{-\gamma_i}, \prod_{i=1}^n M_i^{-\delta_i}) \in \mathbb{G}^3$.

$\text{SignDerive}(pk, \{\omega_i, \sigma^{(i)}\}_{i=1}^{\ell})$: given a public key pk as well as ℓ pairs $(\omega_i, \sigma^{(i)})$, where $\omega_i \in \mathbb{Z}_p$ for each i , parse $\sigma^{(i)}$ as $\sigma^{(i)} = (z_i, r_i, u_i) \in \mathbb{G}^3$ for $i \in \{1, \dots, \ell\}$. Then, compute and return $\sigma = (z, r, u)$, where $z = \prod_{i=1}^{\ell} z_i^{\omega_i}$, $r = \prod_{i=1}^{\ell} r_i^{\omega_i}$ and $u = \prod_{i=1}^{\ell} u_i^{\omega_i}$.

$\text{Verify}(pk, \sigma, (M_1, \dots, M_n))$: given a purposed signature $\sigma = (z, r, u) \in \mathbb{G}^3$, and a vector
 5 $(M_1, \dots, M_n)$, return 1 if and only if $(M_1, \dots, M_n) \neq (1_{\mathbb{G}}, \dots, 1_{\mathbb{G}})$, and (z, r, u) satisfy

$$1_{\mathbb{G}_T} = e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^n e(g_i, M_i); \text{ and } 1_{\mathbb{G}_T} = e(h_z, z) \cdot e(h_u, u) \cdot \prod_{i=1}^n e(h_i, M_i).$$

One particularity of this scheme is that, even if the private key is available, it is difficult to find two distinct signatures on the same vector if the SDP assumption holds: by dividing out the two signatures, one obtains the solution of an SDP instance (g_z, g_r, h_z, h_u) contained in the
 10 public key.

Under the k -SDP assumption, the one-time linearly homomorphic structure-preserving signature previously described can be extended as follows.

$\text{Keygen}(\lambda, n)$: given a security parameter and the dimension $n \in \mathbb{N}$ of vectors to be signed, $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$. For $j = 1$ to k , choose generators $g_{j,z}, g_{j,r} \xleftarrow{R} \mathbb{G}$. Then,
 15 for $i = 1$ to n , $j = 1$ to k pick $\chi_i, \gamma_{j,i} \xleftarrow{R} \mathbb{Z}_p$, and compute $g_{j,i} = g_{j,z}^{\chi_i} g_{j,r}^{\gamma_{j,i}}$. The private key $sk = \left(\left\{ \chi_i, \{\gamma_{j,i}\}_{j=1}^k \right\}_{i=1}^n \right)$ while the public key is $pk = \left(\left\{ g_{j,z}, g_{j,r}, \{g_{j,i}\}_{i=1}^n \right\}_{j=1}^k \right)$.

$\text{Sign}(sk, (M_1, \dots, M_n))$: to sign a vector $(M_1, \dots, M_n) \in \mathbb{G}^n$ using $sk = \left(\left\{ \chi_i, \{\gamma_{j,i}\}_{j=1}^k \right\}_{i=1}^n \right)$, compute and return $\sigma = (z, r_1, \dots, r_k) \in \mathbb{G}^{k+1}$, where

$$z = \prod_{i=1}^n M_i^{-\chi_i} \text{ and } r_j = \prod_{i=1}^n M_i^{-\gamma_{j,i}}, \text{ with } j \in \{1, \dots, k\}.$$

20 $\text{SignDerive}(pk, \{\omega_i, \sigma^{(i)}\}_{i=1}^{\ell})$: given a public key pk as well as ℓ pairs $(\omega_i, \sigma^{(i)})$, where $\omega_i \in \mathbb{Z}_p$ for each i , parse $\sigma^{(i)}$ as $\sigma^{(i)} = (z_i, r_{i,1}, \dots, r_{i,k}) \in \mathbb{G}^{k+1}$ for $i \in \{1, \dots, \ell\}$. Then, compute and return $\sigma = (z, r_1, \dots, r_k)$, where $z = \prod_{i=1}^{\ell} z_i^{\omega_i}$, $r_j = \prod_{i=1}^{\ell} r_{i,j}^{\omega_i}$ for $j \in \{1, \dots, k\}$.

Verify($pk, \sigma, (M_1, \dots, M_n)$): given $\sigma = (z, r_1, \dots, r_k) \in \mathbb{G}^{k+1}$, and a vector $(M_1, \dots, M_n)$, return 1 if and only if $(M_1, \dots, M_n) \neq (1_{\mathbb{G}}, \dots, 1_{\mathbb{G}})$, and, for each $j \in \{1, \dots, k\}$, the following equation holds:

$$1_{\mathbb{G}_T} = e(g_{j,z}, z) \cdot e(g_{j,r}, r_j) \cdot \prod_{i=1}^n e(g_{j,i}, M_i).$$

5 e) Smooth Projective Hash functions

Introduced by Cramer and Shoup, in the article "Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure PublicKey Encryption", published in the conference proceedings of Eurocrypt'02, LNCS 2332, pp. 45–64, 2002, smooth projective hash functions (SPHF) are primitives where a secret key sk is mapped to a projective hashing key $pk = \mu(sk)$.

10 Such a function involves two distinct hashing algorithms $(\Lambda_{sk}(\cdot), \Lambda_{pk}(\cdot))$ such that, for any element $x \in \mathcal{L}$, the action of $\Lambda_{sk}(\cdot)$ on x is completely determined by $pk = \mu(sk)$. In particular, the witness w such that $\mathcal{R}(x, w) = 1$ allows publicly computing $\Lambda_{sk}(x)$ as

$$\Lambda_{sk}(x) = \Lambda_{pk}(x, w)$$

using the public evaluation algorithm $\Lambda_{pk}(\cdot)$. However, for any $x \notin \mathcal{L}$, the action of $\Lambda_{sk}(\cdot)$ on x is completely undetermined. For any $x \notin \mathcal{L}$, given $pk = \mu(sk)$, the value $\Lambda_{sk}(x)$ should be statistically indistinguishable from a random value. The intuition is that, for a given $pk = \mu(sk)$, there are many equally likely private keys sk , each of which gives a different private evaluation $\Lambda_{sk}(x)$ when $x \notin \mathcal{L}$.

A simple example of SPHF based on the DLIN assumption works as follows. The private key is of the form $sk = (x_0, x_1, x_2) \in \mathbb{Z}_p^3$ and the projective key is

20
$$\mu(sk) = (Y_1, Y_2) = (f^{x_1} g^{x_0}, h^{x_2} g^{x_0}),$$

where $f, g, h \in \mathbb{G}$ are public generators. The language $\mathcal{L}_{DLIN}$ is the set of

$$\mathcal{L}_{DLIN} = \left\{ (f^r, h^s, g^{r+s}) \in \mathbb{G}^3 \mid r, s \xleftarrow{R} \mathbb{Z}_p \right\}$$

of vectors that live in a two-dimensional subspace. Under the DLIN assumption, this language is clearly indistinguishable from random vectors of $\mathbb{G}^3$. For any $(V_1, V_2, V_3) \in \mathcal{L}_{DLIN}$, the private evaluation is computed as $\Lambda_{sk}(V_1, V_2, V_3) = V_1^{x_1} V_2^{x_2} V_3^{x_0}$. This private evaluation is publicly computable by anyone who knows the witness $w = (r, s) \in \mathbb{Z}_p^2$ such $(f^r, h^s, g^{r+s}) \in \mathbb{G}^3$.
 5 Indeed, it equals $\Lambda_{sk}(V_1, V_2, V_3) = \Lambda_{sk}(V_1, V_2, V_3, (r, s)) = Y_1^r Y_2^s$.

In contrast, if $(V_1, V_2, V_3) \notin \mathcal{L}_{DLIN}$, the private evaluation $\Lambda_{sk}(V_1, V_2, V_3) = V_1^{x_1} V_2^{x_2} V_3^{x_0}$ is completely undetermined from $pk = \mu(sk) = (Y_1, Y_2) = (f^{x_1} g^{x_0}, h^{x_2} g^{x_0})$, because pk leaks no information about x . The latter SPHF readily extends to the language of k -linear tuples.

10 As in the Cramer-Shoup cryptosystem, described in the article "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack", published in the conference proceedings of Crypto'98, LNCS 1462, pp. 13–25, 1998, 2-universal SPHFs are used in the following: given $pk = \mu(sk)$, for any distinct $x, x' \notin \mathcal{L}$ the joint distribution of $\{\Lambda_{sk}(x), \Lambda_{sk}(x')\}$ is statistically indistinguishable from a pair of random values in the range of
 15 $\Lambda_{sk}(x)$. This is typically achieved by combining an ordinary SPHF with a collision-resistant hash function.

f) After having given some reminders that are going to ease the understanding of embodiments of the invention, and before describing the Figures, the general concept of the invention can be described as follows:

20 In the previous mentioned article "Relatively-Sound NIZKs and Password-Based Key-Exchange", Jutla and Roy described a single-theorem relatively sound NIZK proof system that combines the Groth-Sahai proofs and the smooth projective hash functions of Cramer and Shoup. Smooth projective hash functions can be seen as designated verifier non-interactive zero-knowledge proofs: while they are only convincing to a specific verifier (as the verification
 25 algorithm requires the knowledge of the private verification trapdoor), they do provide a form of onetime simulation-soundness. In contrast, the Groth-Sahai (GS) proofs are publicly verifiable but they are malleable and do not provide relative soundness. However, Jutla and Roy

showed that GS proofs become relatively sound if they are suitably combined with a smooth-projective hash function. The relatively sound proofs can thus be seen as a clever combination of malleable publicly verifiable proofs and privately verifiable non-malleable proofs. The proposed technique relies on the idea of modifying the Jutla-Roy relatively sound proofs described in the previous mentioned articles "Relatively-Sound NIZKs and Password-Based Key-Exchange" and "Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces", by modifying their underlying malleable components and replacing them by one-time linearly homomorphic signatures. However, combining one-time linearly homomorphic signatures and smooth projective hash functions is non-trivial, as explained in the present document.

The present technique is built on the homomorphic signature as the ones depicted in the section "*d) Linearly Homomorphic Structure-Preserving Signatures*" in the present document. Specifically, the language-dependent CRS contains one-time linearly homomorphic signatures on the rows of the matrix $\rho \in \mathbb{G}^{t \times n}$. For each vector $\vec{v} \in \mathcal{L}_\rho$, the prover can use the witness $\vec{x} \in \mathbb{Z}_p^t$ to derive a one-time homomorphic signature (z, r, u) on $\vec{v}$. In order to obtain the sought-after relative simulation-soundness property, the difficulty is that it is necessary to combine two seemingly conflicting requirements: first, the prover should be able to publicly derive a homomorphic signature (z, r, u) on $\vec{v}$ from signatures that are available in the CRS; second, one should prevent unwanted manipulations in the derived signature. In particular, for a given honestly generated proof of some statement $\vec{v} = g^{\vec{x}, A}$, no one should be able to derive a proof for a related statement $(\rho', \vec{v}')$ or a new proof for the same statement.

To achieve this, properties of a 2-universal SPHF should be used: each proof of membership $(z, r, u, \pi_0) \in \mathbb{G}^4$ consists of a derived linearly homomorphic signature (z, r, u) on some vector that depends on $\vec{v}$ and a non-malleable part π_0 consisting of a SPHF public evaluation which is calculated as a function of the language ρ and the statement $\vec{v}$. More precisely, the SPHF evaluation π_0 calculated as a function of a hash value $\alpha = H(\rho, \vec{v}) \in \mathbb{Z}_p$ of the pair $(\rho, \vec{v})$ where $H: \{0,1\}^* \rightarrow \mathbb{Z}_p$ is an ordinary collision-resistant hash function. The relative soundness property is guaranteed by the properties of the underlying SPHF as it implies that, even if the adversary is allowed to see exactly one simulated proof for a

false statement, it will be unable to prove a false statement by itself. Indeed, the 2-universal property of the underlying SPHF guarantees that, for any two distinct vectors $\vec{v}, \vec{v}' \in \mathbb{G}^n$ outside the linear span of $g^A \in \mathbb{G}^{t \times n}$, the private evaluation algorithm $\Lambda_{sk}(\cdot)$ yields pairwise independent private evaluations: given $pk = \mu(sk)$, the joint distribution $\{\Lambda_{sk}(\vec{v}), \Lambda_{sk}(\vec{v}')\}$ is statistically indistinguishable from random. Hence, even if the adversary is given $\Lambda_{sk}(\vec{v})$ as part of a simulated proof of membership for $\vec{v}$, it is unable to predict the proof $\pi'_0 = \Lambda_{sk}(\vec{v}')$ that would make the private verifier accept π'_0 in a proof of membership for $\vec{v}'$. Since the private verifier and the public verifier almost always agree if the DLIN assumption holds, the proof system is guaranteed to be as sound to the public verifier as it is to the private one.

Finally, in order to prevent anyone from turning a proof for some statement $(\rho, \vec{v})$ into a different proof of the same statement, the following property also described, in the section “*d) Linearly Homomorphic Structure-Preserving Signatures*” in the present document, that two distinct signatures on the same vector immediately give a solution to the SDP problem, is used.

By having the prover carefully embed $\alpha = H(\rho, \vec{v}) \in \mathbb{Z}_p$ in the vector for which it computes a derived signature, it can be made sure that:

- (i) The correctness of the public SPHF evaluation π_0 becomes publicly verifiable;
- (ii) The adversary cannot falsely prove the membership of a vector outside the row space of $g^A \in \mathbb{G}^{t \times n}$ even after having seen one such simulated proof;
- (iii) The adversary will be unable to maul an observed proof in order to obtain a proof for some related statement.

Figures 1(a)-(d) present a relatively sound QA-NIZK proof system according to one embodiment of the invention.

In the following, vectors are always considered as row vectors unless stated otherwise. If $A \in \mathbb{Z}_p^{t \times n}$ is a matrix, $g^A \in \mathbb{G}^{t \times n}$ denotes the matrix obtained by exponentiating g using the entries of A .

We consider public parameters $\Gamma = (\mathbb{G}, \mathbb{G}_T, g)$ consisting of bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ with a generator $g \in \mathbb{G}$. Like in the article "Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces", we consider languages

$$\mathcal{L}_\rho = \{g^{\vec{x} \cdot \mathbf{A}} \in \mathbb{G} \mid \vec{x} \in \mathbb{Z}_p^t\}$$

that are parameterized by $\rho = g^{\mathbf{A}} \in \mathbb{G}^{t \times n}$, where $\mathbf{A} \in \mathbb{Z}_p^{t \times n}$ is a $t \times n$ matrix such that $t < n$.

5 As in the article "Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces", it is assumed that the distribution $\mathcal{D}_\Gamma$ is efficiently samplable: namely, there exists a PPT algorithm which outputs a pair $(\rho, \mathbf{A})$ describing a relation $\mathcal{R}_\rho$ and its associated language $\mathcal{L}_\rho$ according to the distribution $\mathcal{D}_\Gamma$. One such distribution is obtained by picking a uniform matrix $\mathbf{A} \xleftarrow{R} \mathbb{Z}_p^{t \times n}$ and computing $\rho = g^{\mathbf{A}} \in \mathbb{G}^{t \times n}$.

10 Under the DLIN assumption, it can be constructed a relatively sound QA-NIZK proof system where each proof only requires 4 group elements. Under the k -linear assumption, the proof length amounts to $O(k)$ elements for a CRS of size $O(kn)$.

As in the article "Relatively-Sound NIZKs and Password-Based Key-Exchange", relative soundness (using smooth projective hash functions) is achieved. To this end, it is needed to
15 encode the matrix $\rho = g^{\mathbf{A}} \in \mathbb{G}^{t \times n}$ as a $2t \times (2n + 1)$.

The initialization method (see Figure 1(a)) aims at obtaining from a security parameter λ , some parameters that are used in the other methods described in Figures 1(b) – (e).

20 More precisely, in a step, referenced 101, an electronic device obtain a symmetric bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with g a randomly chosen element in the group $\mathbb{G}$. Then, it is outputted the set of parameters $\Gamma = (\mathbb{G}, \mathbb{G}_T, g)$.

Again, the dimensions $\mathbf{A} \in \mathbb{Z}_p^{t \times n}$ can be either fixed or part of $\mathcal{L}_\rho$ so that t, n can be given as input to the CRS generation method described in Figure 1(b).

25 The CRS generation method (see Figure 1(b)), executed or implemented by an electronic device, takes as input the set of parameters Γ as well as a parameter $(G_{ij})_{1 \leq i \leq t, 1 \leq j \leq n} \in \mathbb{G}^{t \times n}$.

In a step referenced 102, the electronic device obtains/chooses randomly two n -vectors $\vec{d}$ and $\vec{e}$ defined as follows: $\vec{d} = (d_1, \dots, d_n) \in \mathbb{Z}_p^n$, and $\vec{e} = (e_1, \dots, e_n) \in \mathbb{Z}_p^n$, in order to define

the elements $\vec{W} = (W_1, \dots, W_t) = g^{A \cdot \vec{d}^T} \in \mathbb{G}^t$ and $\vec{Y} = (Y_1, \dots, Y_t) = g^{A \cdot \vec{e}^T} \in \mathbb{G}^t$. These elements are used later to define a projective hash function.

Then, in a step referenced 103, the electronic device generates a key pair (pk_{ots}, sk_{ots}) for the one-time linearly homomorphic signature, depicted in the section “d) Linearly Homomorphic Structure-Preserving Signatures” in the present document, in order to sign vectors in $\mathbb{G}^{2n+1}$. Let $pk_{ots} = ((\mathbb{G}, \mathbb{G}_T), g_z, g_r, h_z, h_w, \{g_i, h_i\}_{i=1}^{2n+1})$ be the public key and let $sk_{ots} = (\{\chi_i, \gamma_i, \delta_i\}_{i=1}^{2n+1})$ be the corresponding private key. For simplicity, the generation of (pk_{ots}, sk_{ots}) can re-use the generator g produced in output of step 101.

Then, in a step referenced 104, the electronic device uses the secret key sk_{ots} in order to generate one-time linearly homomorphic signatures $\{(z_i, r_i, u_i)\}_{i=1}^{2t}$ on the following independent vectors, which are obtained from the rows of the matrix $\rho = (G_{ij})_{1 \leq i \leq t, 1 \leq j \leq n}$:

$$\vec{H}_i = (G_{i,1}, \dots, G_{i,n}, Y_i, 1, \dots, 1) \in \mathbb{G}^{2n+1} \text{ with } i \in \{1, \dots, t\}$$

$$\vec{F}_i = (1, \dots, 1, W_i, G_{i,1}, \dots, G_{i,n}) \in \mathbb{G}^{2n+1} \text{ with } i \in \{1, \dots, t\}.$$

Then, in a step referenced 105, the electronic device obtains a collision-resistant hash function $H: \{0,1\}^* \rightarrow \mathbb{Z}_p$ to be used.

Then, in a step referenced 106, the electronic device determines the CRS ψ which comprises a first part CRS_1 that is only used by the prover and a second part CRS_2 which is only used by the verifier. These are defined as

$$CRS_1 = (\rho, pk_{ots}, \vec{W}, \vec{Y}, \{(z_i, r_i, u_i)\}_{i=1}^{2t}, H),$$

$$CRS_2 = (pk_{ots}, \vec{W}, \vec{Y}, H).$$

The simulation trapdoor τ_{sim} is $sk_{ots} = (\{\chi_i, \gamma_i, \delta_i\}_{i=1}^{2n+1})$ and the private verification trapdoor is $\tau_v = \{\vec{d}, \vec{e}\}$.

The proof generator method (see Figure 1(c)), executed or implemented by an electronic device, takes as input a candidate vector $\vec{v}$ being equal to $(v_1, \dots, v_n) \in \mathbb{G}^n$, a witness $\vec{x}$ being equal to $(x_1, \dots, x_t) \in \mathbb{Z}_p^t$ such that $\vec{v} = g^{\vec{x} \cdot A}$, a label lbl and elements Γ and ψ previously defined.

In a step referenced 107, the electronic device determines the value of $\alpha = H(\rho, \vec{v}, lbl) \in \mathbb{Z}_p$.

Then, in a step referenced 108, the electronic device, using the linearly homomorphic signatures $\{(z_i, r_i, u_i)\}_{i=1}^{2t}$ comprised in the CRS, derives a one-time linearly homomorphic signature (z, r, u) on the vector $\vec{v} = (v_1, \dots, v_n, \pi_0, v_1^\alpha, \dots, v_n^\alpha) \in \mathbb{G}^{2n+1}$, where $\pi_0 = \prod_{i=1}^t (W_i^\alpha Y_i)^{x_i}$. More precisely, it determines:

$$5 \quad z = \prod_{i=1}^t (z_{2i-1} \cdot z_{2i}^\alpha)^{x_i}, r = \prod_{i=1}^t (r_{2i-1} \cdot r_{2i}^\alpha)^{x_i}, u = \prod_{i=1}^t (u_{2i-1} \cdot u_{2i}^\alpha)^{x_i}.$$

Then, the electronic device outputs the proof π which is equal to $(z, r, u, \pi_0) \in \mathbb{G}^4$.

Hence, the generator method system delivers a relatively sound QA-NIZK proof that only requires $O(1)$ group elements. The size of the proof does not depend on the dimension of the considered subspace.

10 The public verification method (see Figure 1(d)) executed or implemented by an electronic device, takes as input a vector $\vec{v}$ being equal to $(v_1, \dots, v_n) \in \mathbb{G}^n$, a proof π , the CRS ψ previously defined, the witness $\vec{x}$ being equal to $(x_1, \dots, x_t) \in \mathbb{Z}_p^t$ such that $\vec{v} = g^{\vec{x} \cdot A}$, a label lbl and element $\Gamma = (\mathbb{G}, \mathbb{G}_T, g)$.

15 In a step, referenced 109, the electronic device verifies that the format of the received/obtained parameters is compliant with the expected one (i.e. each coordinate of the vector $\vec{v}$ must belong to the group $\mathbb{G}$, the proof should have the following format $(z, r, u, \pi_0) \in \mathbb{G}^4$, etc.).

Then, in a step, referenced 110, the electronic device determines the following value: $\alpha = H(\rho, \vec{v}, lbl) \in \mathbb{Z}_p$, and, it determines the vector $\vec{v} = (v_1, \dots, v_n, \pi_0, v_1^\alpha, \dots, v_n^\alpha) \in \mathbb{G}^{2n+1}$.

20 Then, in a step, referenced 111, the electronic device determines if the elements (z, r, u) comprised in said received/obtained proof π corresponds to a valid one time linearly homomorphic signature on the vector $\vec{v} = (v_1, \dots, v_n, \pi_0, v_1^\alpha, \dots, v_n^\alpha) \in \mathbb{G}^{2n+1}$.

The electronic device outputs an information (such as a value equal to 1) if the one time linearly homomorphic signature is valid. In the step 111, the electronic device verifies if the following equations stand (where elements g_z, g_r, h_z, h_u and $\{g_i, h_i\}_{i=1}^{2n+1}$ are obtained from the public key pk_{ots} generated in step 103 and comprised in the CRS ψ):

$$1_{\mathbb{G}_T} = e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^n e(g_i \cdot g_{i+n+1}^\alpha, v_i) \cdot e(g_{n+1}, \pi_0);$$

$$1_{\mathbb{G}_T} = e(h_z, z). e(h_u, u) . \prod_{i=1}^n e(h_i . h_{i+n+1}^\alpha, v_i) . e(h_{n+1}, \pi_0).$$

If the verification fails, the electronic device outputs a value equal to 0.

The private verification method (see Figure 1(e)) executed or implemented by an electronic device, takes as a vector $\vec{v}$ being equal to $(v_1, \dots, v_n) \in \mathbb{G}^n$, a proof π , a label lbl and elements Γ and ψ previously defined, and the element τ_v which corresponds to the verification trapdoor comprising the vectors $\vec{d}$ and $\vec{e}$: $\vec{d} = (d_1, \dots, d_n) \in \mathbb{Z}_p^n$, and $\vec{e} = (e_1, \dots, e_n) \in \mathbb{Z}_p^n$.

In a step, referenced 112, the electronic device determines that the format of the parameters provided as input fulfill the expected rule.

Then, in a step, referenced 113, the electronic device determines the value of $\alpha = H(\rho, \vec{v}, lbl) \in \mathbb{Z}_p$.

In a step referenced 114, the electronic device checks if the following equations stand (where elements g_z, g_r, h_z, h_u and $\{g_i, h_i\}_{i=1}^{2n+1}$ are obtained from the public key pk_{ots} generated in step 103):

$$1_{\mathbb{G}_T} = e(g_z, z). e(g_r, r) . \prod_{i=1}^n e(g_i . g_{i+n+1}^\alpha, v_i) . e(g_{n+1}, \pi_0);$$

$$1_{\mathbb{G}_T} = e(h_z, z). e(h_u, u) . \prod_{i=1}^n e(h_i . h_{i+n+1}^\alpha, v_i) . e(h_{n+1}, \pi_0).$$

If the check fails (i.e. at least one of the equation is not verified), the electronic device delivers an output value equals to zero.

Then, in a step referenced 115, the electronic device determines if $\pi_0 = \prod_{j=1}^n v_j^{e_j + \alpha d_j}$. The electronic device delivers an output value equals to zero if the equation is not verified. Otherwise, it delivers an output value equals to one.

The proposed relatively sound QA-NIZK proof system readily extends to rest on the k -SDP assumption with $k > 2$. The only required modification is to replace the linearly homomorphic signature based on the SDP assumption by the second linearly homomorphic signature described in the section “d) Linearly Homomorphic Structure-Preserving Signatures” in the present document. In each proof, the (z, r, u) part is thus replaced by a signature of the form $(z, r_1, \dots, r_k)$ while the smooth projective hash value remains unchanged. In this case, the

proof requires $k + 2$ group elements whereas combining the techniques of the articles *“Relatively-Sound NIZKs and Password-Based Key-Exchange”* and *“Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces”* requires $k(n + 1 - t)$ elements per proof. From a security standpoint, the following result can be proven: the proposed proof system is a relatively sound

5 QA-NIZK proof system if the SDP assumption holds in $(\mathbb{G}, \mathbb{G}_T)$ and if H is a collision-resistant hash function.

Figures 2(a)-(f) present a non-interactive threshold cryptosystem according to one embodiment of the invention.

Indeed, as detailed in the present section, it can be constructed an adaptively secure

10 CCA2-secure non-interactive threshold cryptosystem based on the DLIN assumption. Adaptively secure threshold cryptosystems with chosen-ciphertext security (in the sense of article *“Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack”* by C. Rackoff, D. Simon, and published in the conference proceedings of Crypto’91, LNCS 576, pp. 433–444, 1991) were previously proposed in the article *“Adaptively Secure Threshold Cryptography: Introducing Concurrency, Removing Erasures”* by S. Jarecki, A. Lysyanskaya, in the conference

15 proceedings of Eurocrypt’00, LNCS 1807, pp. 221–242, 2000, and in the article *“Adaptively Secure Feldman VSS and Applications to Universally-Composable Threshold Cryptography”* by M. Abe, S. Fehr, published in the conference proceedings of Crypto’04, LNCS 3152, pp. 317–334, 2004, but they require some interaction during the decryption process. Non-interactive

20 solutions were put forth in the three following articles: *“Threshold Public-Key Encryption with Adaptive Security and Short Ciphertexts”* by B. Qin et al., published in the conference proceedings of ICICS’10, LNCS 6476, pp. 62–76, 2010, *“Adaptively Secure Non-Interactive Threshold Cryptosystems”* by B. Libert and M. Yung, published in the conference proceedings of ICALP 2011, LNCS 6756, pp. 588–600, 2011, and *“Non-Interactive CCA2-Secure Threshold*

25 *Cryptosystems with Adaptive Security: New Framework and Constructions”* by B. Libert and M. Yung, published in the conference proceedings TCC 2012, LNCS 7194, pp. 75–93, Springer, 2012.

But, as detailed in the following, they are less efficient than the proposed technique. Under the k -linear assumption, if compared to previous constructions based on the same

assumption, the proposed technique (related to Figures 2(a)-(f)) shortens ciphertexts by $O(k)$ group elements. This result is important as non-interactive threshold schemes can potentially serve as building blocks for protocols in the multi-linear setting (see the article "Candidate Multilinear Maps from Ideal Lattices" by S. Garg et al., published in the conference proceedings of Eurocrypt 2013, LNCS series, pp. 1–17, 2013, and the article "Practical Multilinear Maps over the Integers" by J.-S. Coron et al., published in the conference proceedings of Crypto 2013). Indeed, the $(k - 1)$ -linear problem is easy in groups equipped with a k -linear map (as shown in the article "An Algebraic Framework for Diffie-Hellman Assumptions" by A. Escala et al., published in the conference proceedings of Crypto 2013, LNCS 8043, pp. 129–147, 2013) but instantiations where the k -linear assumption holds (as seems to be the case in the previous mentioned article "Practical Multilinear Maps over the Integers") can be used.

The proposed threshold scheme is fully non-interactive: namely, decryption servers never have to talk to each other during the decryption process and they only send one message to a dedicated entity, called combiner, that gather the decryption shares in order to recover the plaintext.

The key generation method (see Figure 2(a)) executed or implemented by an electronic device, takes a security parameter λ , a number t , and a number N enabling to define a (t, N) -threshold encryption scheme (where t parties (or electronic devices) are required to perform the deciphering).

In a step referenced 201, the electronic device obtains a bilinear group $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$.

In a step referenced 202, the electronic device obtains three elements (randomly generated) that belong to the group $\mathbb{G}$: $g, f, h \xleftarrow{R} \mathbb{G}$.

In a step referenced 203, the electronic device obtains nine random elements in $\mathbb{Z}_p$: $x_0, x_1, x_2 \xleftarrow{R} \mathbb{Z}_p$, $y_0, y_1, y_2 \xleftarrow{R} \mathbb{Z}_p$ and $w_0, w_1, w_2 \xleftarrow{R} \mathbb{Z}_p$. Then, it determines the six following elements: $X_1 = f^{x_1} g^{x_0}$, $X_2 = h^{x_2} g^{x_0}$, $Y_1 = f^{y_1} g^{y_0}$, $Y_2 = h^{y_2} g^{y_0}$, $W_1 = f^{w_1} g^{w_0}$ and $W_2 = h^{w_2} g^{w_0}$.

In a step referenced 204, the electronic device obtains Groth-Sahai CRS $(\vec{f}_1, \vec{f}_2, \vec{f}_3)$, to be used for providing the validity of decryption shares. More precisely, the electronic device

obtains two elements chosen randomly in the group $\mathbb{G}$: $f_1, f_2 \xleftarrow{R} \mathbb{G}$, as well as two elements chosen randomly in $\mathbb{Z}_p$: $\phi_1, \phi_2 \xleftarrow{R} \mathbb{Z}_p$. Then the Groth-Sahai CRS $(\vec{f}_1, \vec{f}_2, \vec{f}_3)$ are determined as follows:

$$\vec{f}_1 = (f_1, 1, g), \vec{f}_2 = (1, f_2, g), \text{ and } \vec{f}_3 = \vec{f}_1^{\phi_1} \vec{f}_2^{\phi_2} (1, 1, g).$$

5 In a step referenced 205, the electronic device obtains three random polynomials $P_0[Z], P_1[Z], P_2[Z] \in \mathbb{Z}_p[Z]$ of degree $t - 1$, such that $P_0(0) = x_0, P_1(0) = x_1$ and $P_2(0) = x_2$. Then, the following elements are determined: $X_{i,1} = f^{P_1(i)} g^{P_0(i)}$, as well as $X_{i,2} = h^{P_2(i)} g^{P_0(i)}$ for $i = 1$ to N .

10 In a step referenced 206, the electronic device obtains a collision-resistant hash function $H: \{0,1\}^* \rightarrow \mathbb{Z}_p$.

In a step referenced 207, the electronic device generates/obtains a key pair for the one time linearly homomorphic signature depicted in the section “d) Linearly Homomorphic Structure-Preserving Signatures” in the present document, with $n = 7$. The public key is equal to $(g_z, g_r, h_z, h_u, \{g_i, h_i\}_{i=1}^7)$ and the corresponding private key is $\{\chi_i, \gamma_i, \delta_i\}_{i=1}^7$.

15 In a step referenced 208, the electronic device generates/determines/obtains one-time linearly homomorphic signatures $\{Z_j, R_j, U_j\}_{j=1}^4$ on the following linearly independent vectors:

$$\begin{aligned} \vec{h}_1 &= (f, 1, g, Y_1, 1, 1, 1) \in \mathbb{G}^7, \vec{h}_2 = (1, h, g, Y_2, 1, 1, 1) \in \mathbb{G}^7, \\ \vec{h}_3 &= (1, 1, 1, W_1, f, 1, g) \in \mathbb{G}^7, \vec{h}_4 = (1, 1, 1, W_2, 1, h, g) \in \mathbb{G}^7. \end{aligned}$$

20 In a step referenced 209, the electronic device determines the vector of private key shares as being $\mathbf{SK} = (SK_1, \dots, SK_N)$ with $SK_i = (P_0(i), P_1(i), P_2(i)) \in \mathbb{Z}_p^3$ for each $i \in \{1, \dots, N\}$. The corresponding vector $\mathbf{VK} = (VK_1, \dots, VK_N)$ of verification keys is defined in such way that $VK_i = (X_{i,1}, X_{i,1}) \in \mathbb{G}^2$, for each $i \in \{1, \dots, N\}$. The public key is defined to be:

$$PK = (g, \vec{f}_1, \vec{f}_2, \vec{f}_3, X_1, X_2, Y_1, Y_2, W_1, W_2, g_z, g_r, h_z, h_u, \{g_i, h_i\}_{i=1}^7, \{Z_j, R_j, U_j\}_{j=1}^4, H).$$

25 The encryption method (see Figure 2(b)) executed or implemented by an electronic device, takes as input a message $M \in \mathbb{G}$ and a public key being defined as follows: $PK = (g, \vec{f}_1, \vec{f}_2, \vec{f}_3, X_1, X_2, Y_1, Y_2, W_1, W_2, g_z, g_r, h_z, h_u, \{g_i, h_i\}_{i=1}^7, \{Z_j, R_j, U_j\}_{j=1}^4, H)$.

Then, in a step referenced 210, the electronic device obtains two random elements in $\mathbb{Z}_p$: $\theta_1, \theta_2 \xleftarrow{R} \mathbb{Z}_p$. It determines the following elements:

$$C_0 = M \cdot X_1^{\theta_1} \cdot X_2^{\theta_2}, C_1 = f^{\theta_1}, C_2 = h^{\theta_2}, C_3 = g^{\theta_1 + \theta_2} \text{ and} \\ C_4 = (W_1^\alpha Y_1)^{\theta_1} \cdot (W_2^\alpha Y_2)^{\theta_2}, \text{ where } \alpha = H(C_0, C_1, C_2, C_3) \in \mathbb{Z}_p.$$

5 Then, in a step referenced 211, the electronic device determine a linearly homomorphic signature (Z, R, U) on the vector $(C_1, C_2, C_3, C_4, C_1^\alpha, C_2^\alpha, C_3^\alpha) \in \mathbb{G}^7$. More precisely, it determines such signature as follows:

$$Z = Z_1^{\theta_1} \cdot Z_2^{\theta_2} Z_3^{\theta_1 \cdot \alpha} Z_4^{\theta_2 \cdot \alpha}; \\ R = R_1^{\theta_1} \cdot R_2^{\theta_2} R_3^{\theta_1 \cdot \alpha} R_4^{\theta_2 \cdot \alpha}; \\ 10 \quad U = U_1^{\theta_1} \cdot U_2^{\theta_2} U_3^{\theta_1 \cdot \alpha} U_4^{\theta_2 \cdot \alpha}.$$

Then, the electronic device delivers the ciphertext $C = (C_0, C_1, C_2, C_3, C_4, Z, R, U) \in \mathbb{G}^8$.

The ciphertext verify method (see Figure 2(c)) executed or implemented by an electronic device, takes in input a public key PK and a ciphertext C .

15 The electronic device checks, in a step referenced 212, if the ciphertext comprises eight elements of $\mathbb{G}$ (i.e. the ciphertext C can be represented by $(C_0, C_1, C_2, C_3, C_4, Z, R, U) \in \mathbb{G}^8$).

The electronic device determines the value of $\alpha = H(C_0, C_1, C_2, C_3) \in \mathbb{Z}_p$.

The electronic device outputs a value equal to 1 if and only if the following equations are verified:

$$1_{\mathbb{G}_T} = e(g_z, Z) \cdot e(g_r, R) \cdot \prod_{i=1}^3 e(g_i \cdot g_{i+4}^\alpha, C_i) \cdot e(g_4, C_4); \\ 20 \quad 1_{\mathbb{G}_T} = e(h_z, Z) \cdot e(h_u, U) \cdot \prod_{i=1}^3 e(h_i \cdot h_{i+4}^\alpha, C_i) \cdot e(h_4, C_4).$$

Hence, (Z, R, U) must be a valid linearly homomorphic signature on the vector $(C_1, C_2, C_3, C_4, C_1^\alpha, C_2^\alpha, C_3^\alpha) \in \mathbb{G}^7$. Otherwise, the electronic device outputs a value equal to zero.

25 The share decrypt method (see Figure 2(d)) executed or implemented by an electronic device, takes as input a public key PK , a value i , a secret key SK_i and a ciphertext C . The secret key must be represented as follows: $SK_i = (P_0(i), P_1(i), P_2(i)) \in \mathbb{Z}_p^3$.

In a step, referenced 214, the electronic device executes the method described in Figure 2(c). In the case that the result is equal to zero, then, the electronic device outputs the value i and an information, noted $\perp$, indicating that the ciphertext is malformed. Otherwise, the electronic device executes the step referenced 215 which comprises the step of determining

5 $\hat{\mu}_i = (v_i, \vec{C}_{P_1}, \vec{C}_{P_2}, \vec{C}_{P_0}, \pi_{v_i})$, which consists of a partial decryption $v_i = C_1^{P_1(i)} \cdot C_2^{P_2(i)} \cdot C_3^{P_0(i)}$, commitments $\vec{C}_{P_0}, \vec{C}_{P_1}, \vec{C}_{P_2}$ respectively to $P_0(i), P_1(i), P_2(i)$, and a proof π_{v_i} that the following equations are satisfied:

$$v_i = C_1^{P_1(i)} \cdot C_2^{P_2(i)} \cdot C_3^{P_0(i)}, X_{i,1} = f^{P_1(i)} \cdot g^{P_0(i)} \text{ and } X_{i,2} = h^{P_2(i)} \cdot g^{P_0(i)}.$$

The commitments $\vec{C}_{P_1}, \vec{C}_{P_2}, \vec{C}_{P_0}$ and the proof π_{v_i} are generated using the CRS

10 $\mathbf{f} = (\vec{f}_1, \vec{f}_2, \vec{f}_3).$

The share verify method (see Figure 2(e)) executed or implemented by an electronic device, takes as input a public key PK , a verification key VK_i , a ciphertext C and the following value $(i, \hat{\mu}_i)$ (corresponding to the output of the share decrypt method).

In a step referenced 216, the electronic device parses the ciphertext C as

15 $(C_0, C_1, C_2, C_3, C_4, Z, R, U) \in \mathbb{G}^8$, and VK_i as $(X_{i,1}, X_{i,2}) \in \mathbb{G}^2$. If the $\hat{\mu}_i$ is equal to $\perp$, or if $\hat{\mu}_i$ can not be parsed properly as $(v_i, \vec{C}_{P_1}, \vec{C}_{P_2}, \vec{C}_{P_0}, \pi_{v_i})$, then it outputs a value equal to 0. Otherwise it outputs a value equal to 1 if π_{v_i} is a valid proof. In other situation, it outputs a value equal to 0.

The combine method (see Figure 2(f)) executed or implemented by an electronic device takes as input $PK, \mathbf{VK}, C, \{(i, \hat{\mu}_i)\}_{i \in S}$.

20 In a step referenced 217, for each $i \in S$, the electronic device parses the corresponding decryption share $\hat{\mu}_i$ as $(v_i, \vec{C}_{P_1}, \vec{C}_{P_2}, \vec{C}_{P_0}, \pi_{v_i})$, and return $\perp$, if the share verify method described in Figure 2(e) outputs a value equal to zero. Otherwise, the electronic device determines $v = \prod_{i \in S} v_i^{\Delta_{i,S}(0)}$, which equals $v = C_1^{x_1} \cdot C_2^{x_2} \cdot C_3^{x_0} = X_1^{\theta_1} \cdot X_2^{\theta_2}$ and in turn reveals the plaintext $M = C_0/v$.

25 The threshold encryption scheme of Figures 2(a)-(f) can be proved chosen-ciphertext secure under adaptive corruptions if the DLIN assumption holds. Moreover, the security reduction is tight as, up to negligible terms, the adversary's advantage is bounded by a constant multiplied by the maximal probability of breaking the underlying assumptions.

The following theorem can be set up: the above threshold cryptosystem is IND-CCA secure against adaptive corruptions assuming that: (i) H is collision-resistant; (ii) The DLIN assumption holds in $\mathbb{G}$. More precisely, the advantage of any PPT adversary $\mathcal{A}$ is at most

$$Adv(\mathcal{A}) \leq Adv^{CR-hash}(\lambda) + 3 \cdot Adv^{DLIN}(\lambda) + \frac{q+1}{2^\lambda - q},$$

where the first term of the right-hand-side member accounts for the maximal advantage of any PPT collision-finding algorithm for H .

If each element has a 256-bit representation on BN curves (as in the article “Pairing-Friendly Elliptic Curves of Prime Order” by P. Barreto, M. Naehrig, published in the conference proceedings of SAC’05, LNCS 3897, pp. 319–331, 2005) at the 128-bit security level, the ciphertext overhead amounts to 1792 bits. The DLIN-based scheme described in the previous mentioned article “Non-Interactive CCA2-Secure Threshold Cryptosystems with Adaptive Security: New Framework and Constructions” has a ciphertext overhead comprised of 14 group elements and a one-time signature with its verification key (or 4864 bits using Groth’s one-time signature; see the article “Simulation-sound NIZK proofs for a practical language and constant size group signatures” by J. Groth, published in the conference proceedings of Asiacrypt 2006, LNCS 4284, pp. 444–459, 2006). The recent results of Escala et al. (in the previous mentioned article “An Algebraic Framework for Diffie-Hellman Assumptions” allow reducing this overhead to 3328 bits. The very recent techniques of Jutla and Roy (in the previous mentioned article “Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces”) – which also work in the threshold setting although it was not explicitly stated in the previous mentioned article “Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces” – lead to ciphertexts comprised of 9 group elements under the DLIN assumption and $3k + 3$ under the k -linear assumption. Under DLIN, ciphertexts are further compressed by 11% while relying on the same assumption and retaining tight security.

Under the k -linear assumption, the improvement provided by the present technique becomes more important as the ciphertext reduces to $2k + 4$ group elements. Specifically, it is needed $k + 1$ elements for the second homomorphic signature described in the section “d) Linearly Homomorphic Structure-Preserving Signatures” in the present document, another

$(k + 1)$ elements to contain the k -linear instance, one element for the Cramer-Shoup-like proof π_0 and one element to carry the plaintext. This allows saving $O(k)$ group elements with respect to the techniques described in the previous mentioned article "Relatively-Sound NIZKs and Password-Based Key-Exchange", which require $3k + 3$ group elements per ciphertext.

5 In addition to this, the present technique may find applications in other protocols where decryption capabilities should be shared among several parties. Examples include auction protocols and privacy-preserving targeted advertising protocols (see the article "Targeted Advertising . . . And Privacy Too" by A. Juels, published in the conference proceedings of CT-RSA'01, LNCS 2020, pp. 408–424, 2001, for example).

10 Figure 3 presents an example of a device that can be used to perform one or several steps of methods disclosed in the present document.

Such device referenced 300 comprises a computing unit (for example a CPU, for "*Central Processing Unit*"), referenced 301, and one or more memory units (for example a RAM (for "*Random Access Memory*") block in which intermediate results can be stored temporarily
15 during the execution of instructions a computer program, or a ROM block in which, among other things, computer programs are stored, or an EEPROM ("*Electrically-Erasable Programmable Read-Only Memory*") block, or a flash block) referenced 302. Computer programs comprise instructions that can be executed by the computing unit. Such device 300 can also comprise a dedicated unit, referenced 303, constituting an input-output interface to
20 allow the device 300 to communicate with other devices. In particular, this dedicated unit 303 can be connected with an antenna (in order to perform communication without contacts), or with serial ports (to carry communications by the use of physical contacts). It should be noted that the arrows in Figure 3 signify that the linked unit can exchange data through buses for example together.

25 In an alternative embodiment, some or all of the steps of the method previously described, can be implemented in hardware in a programmable FPGA ("*Field Programmable Gate Array*") component or ASIC ("*Application-Specific Integrated Circuit*") component. In

another embodiment of the invention, the previously mentioned means correspond to a dedicated circuit (such as a FPGA component or an ASIC component).

In an alternative embodiment, some or all of the steps of the method previously described, can be executed on an electronic device comprising memory units and processing
5 units as the one disclosed in the Figure 3.

Claims

1. Method for generating a quasi-adaptive non-interactive zero-knowledge proof, said proof assessing a membership of a first vector to a hard language, which is a subspace of $\mathbb{G}^n$, where $\mathbb{G}$ is a group and n is an integer, said method being executed by an electronic device, and being characterized in that it comprises:

generating a derived linearly homomorphic signature on a second vector, said second vector being defined as a function of said first vector, a linearly homomorphic signatures comprised in a common reference string and a witness associated with said first vector; and

generating a non-malleable element by using a public evaluation function of a smooth projective hash function, in combination with a hash function, said first vector, said witness and elements comprised in said common reference string, said proof being a combination of said derived linearly homomorphic signature and said non-malleable element.

2. Method for generating according to claim 1, characterized in that said hard language corresponds to a set which is equal to $\{\vec{v} \in \mathbb{G}^n \mid \vec{x} \in \mathbb{Z}_p^t \text{ s.t. } \vec{v} = g^{\vec{x}.A}\}$, where p is a prime number, t is an integer, g is a generator of said group $\mathbb{G}$, and A is a $t \times n$ matrix of rank t whose elements are comprised in $\mathbb{Z}_p$.

3. Method for generating according to claim 2, characterized in that said witness $\vec{x}$ belongs to $\mathbb{Z}_p^t$, and is associated with said first vector $\vec{v}$ by a relationship $\vec{v} = g^{\vec{x}.A}$.

4. Method for generating according to any claims 2 to 3, characterized in that said linearly homomorphic signatures comprised in a common reference string comprise $2t$ elements, each element being defined by a vector $(z, r_1, \dots, r_k)$, where k is a

integer greater or equal to two, corresponding to a linearly homomorphic signature on independent vectors, which are obtained from rows of a matrix g^A .

- 5 5. Method for generating according to any claims 1 to 4, characterized in that generating a derived linearly homomorphic signature, and generating a non-malleable element further use a label.
- 10 6. Method for ciphering in an non-interactive and adaptively secure threshold cryptosystems with chosen-ciphertext security, said method being executed by an electronic device, and being characterized in that it comprises executing a method for generating a quasi-adaptive non-interactive zero knowledge proof according to claims 1 to 5.
- 15 7. A computer-readable and non-transient storage medium storing a computer program comprising a set of computer-executable instructions to implement a method for cryptographic computations when the instructions are executed by a computer, wherein the instructions comprise instructions, which when executed, configure the computer to perform a method for generating a quasi-adaptive non-interactive zero-knowledge proof of claims 1 to 5, and/or to perform a method
20 ciphering in an non-interactive and adaptively secure threshold cryptosystems with chosen-ciphertext security of claim 6.
- 25 8. Electronic device comprising a module configured to generate a quasi-adaptive non-interactive zero-knowledge proof, said proof assessing a membership of a first vector to a hard language, which is a subspace of $\mathbb{G}^n$, where $\mathbb{G}$ is a group and n is an integer, said electronic device being characterized in that it comprises:
30 a module configured to generate a derived linearly homomorphic signature on a second vector, said second vector being defined as a function of said first vector, a linearly homomorphic signatures comprised in a common reference string and a witness associated with said first vector; and

a module configured to generate a non-malleable element by using a public evaluation function of a smooth projective hash function, in combination with a hash function, said first vector, said witness and elements comprised in said common reference string, said proof being a combination of said derived linearly homomorphic signature and said non-malleable element.

9. Electronic device according to claim 8, characterized in that said hard language corresponds to a set which is equal to $\{\vec{v} \in \mathbb{G}^n \mid \vec{x} \in \mathbb{Z}_p^t \text{ s.t. } \vec{v} = g^{\vec{x}.A}\}$, where p is a prime number, t is an integer, g is a generator of said group $\mathbb{G}$, and A is a $t \times n$ matrix of rank t whose elements are comprised in $\mathbb{Z}_p$.

10. Electronic device according to claim 9, characterized in that said witness $\vec{x}$ belongs to $\mathbb{Z}_p^t$, and is associated with said first vector $\vec{v}$ by a relationship $\vec{v} = g^{\vec{x}.A}$.

11. Electronic device according to any claims 8 to 9, characterized in that said linearly homomorphic signatures comprised in a common reference string comprise $2t$ elements, each element being defined by a vector $(z, r_1, \dots, r_k)$, where k is an integer greater or equal to two, corresponding to a linearly homomorphic signature on independent vectors, which are obtained from rows of a matrix g^A .

12. Electronic device according to any claims 8 to 11, characterized in that the module configured to generate a derived linearly homomorphic signature, and the module configured to generate a non-malleable element further use a label.

13. Electronic device comprising a module configured to cipher in a non-interactive and adaptively secure threshold cryptosystems with chosen-ciphertext security, said electronic device being characterized in that it comprises a module configured to generate a quasi-adaptive non-interactive zero-knowledge proof, said proof

assessing a membership of a first vector to a hard language, which is a subspace of $\mathbb{G}^n$, where $\mathbb{G}$ is a group and n is an integer, that comprises:

a module configured to generate a derived linearly homomorphic signature on a second vector, said second vector being defined as a function of said first vector, a linearly homomorphic signatures comprised in a common reference string and a witness associated with said first vector; and

a module configured to generate a non-malleable element by using a public evaluation function of a smooth projective hash function, in combination with a hash function, said first vector, said witness and elements comprised in said common reference string, said proof being a combination of said derived linearly homomorphic signature and said non-malleable element.

1/5

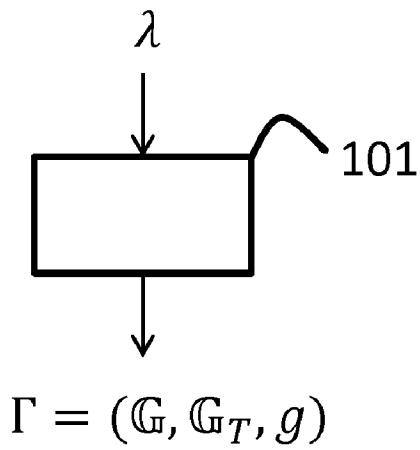


FIG. 1(a)

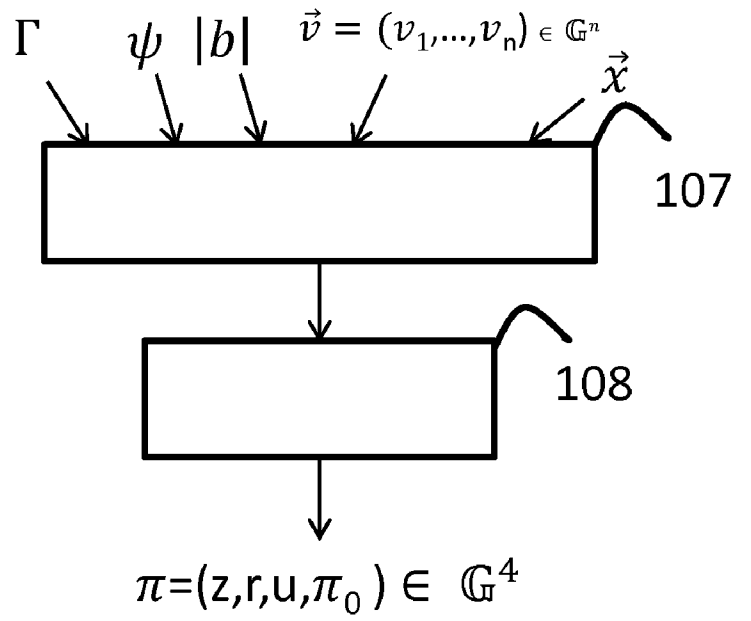


FIG. 1(c)

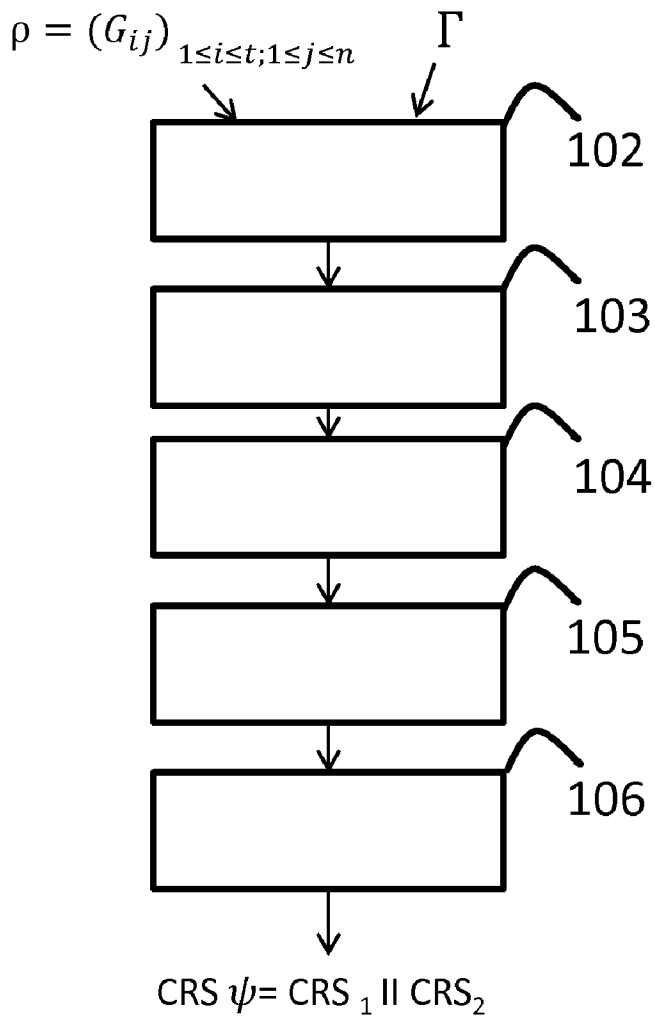


FIG. 1(b)

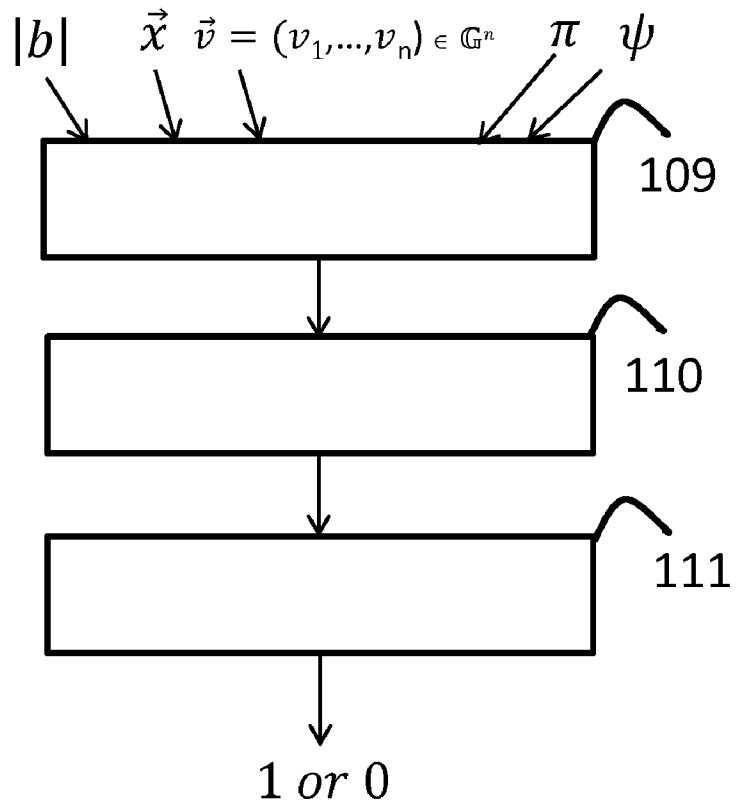


FIG. 1(d)

2/5

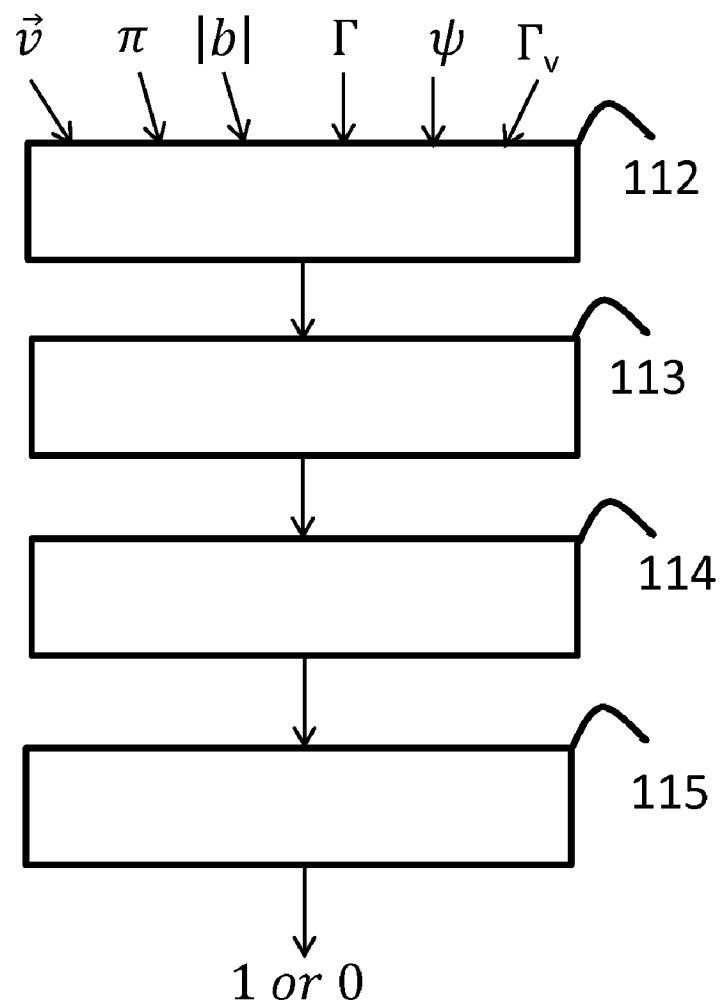


FIG. 1(e)

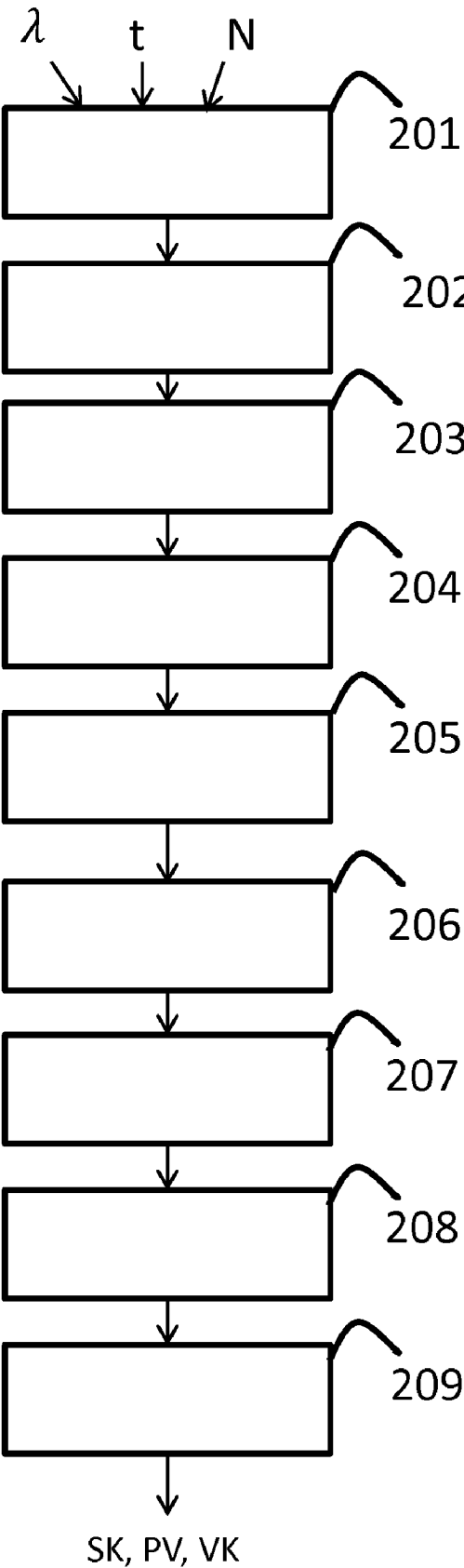


FIG. 2(a)

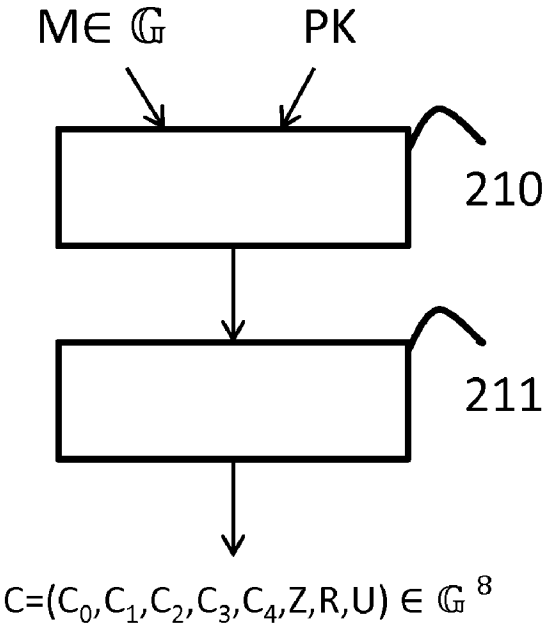


FIG. 2(b)

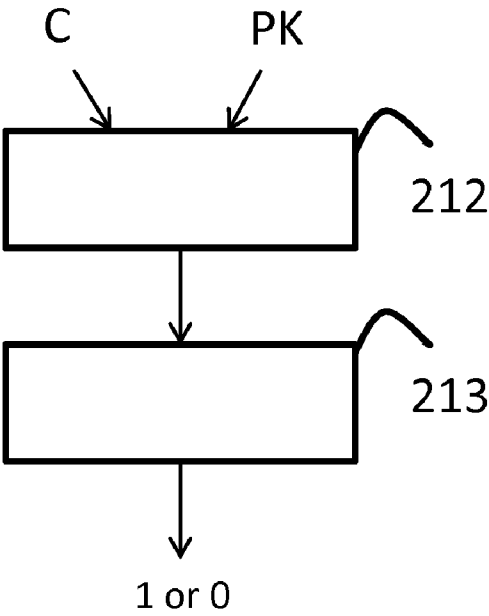


FIG. 2(c)

4/5

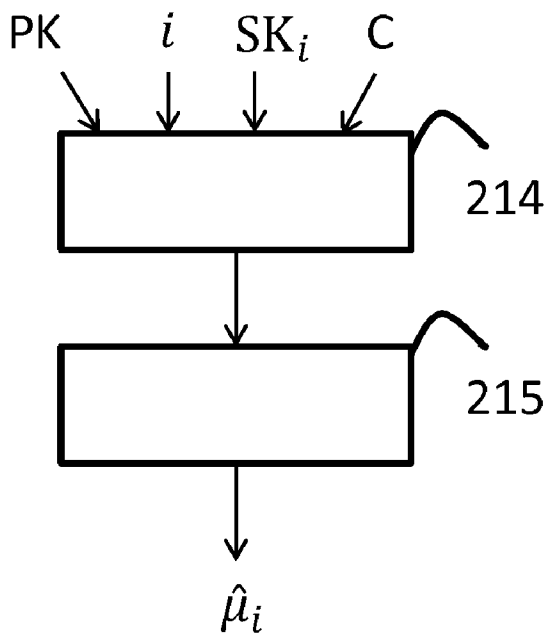


FIG. 2(d)

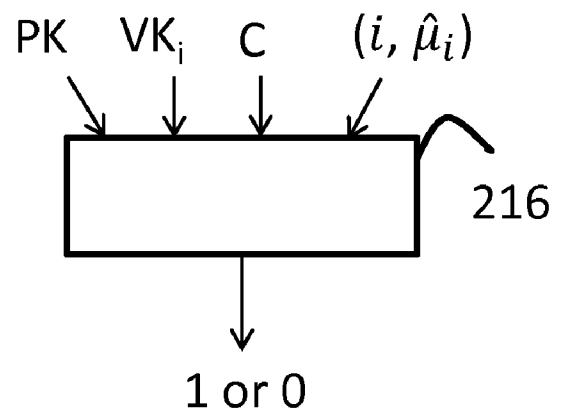


FIG. 2(e)

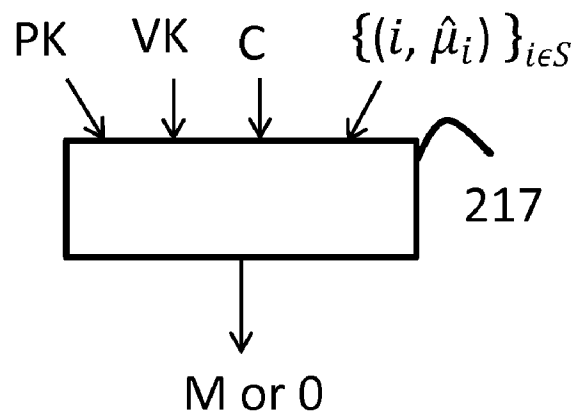


FIG. 2(f)

5/5

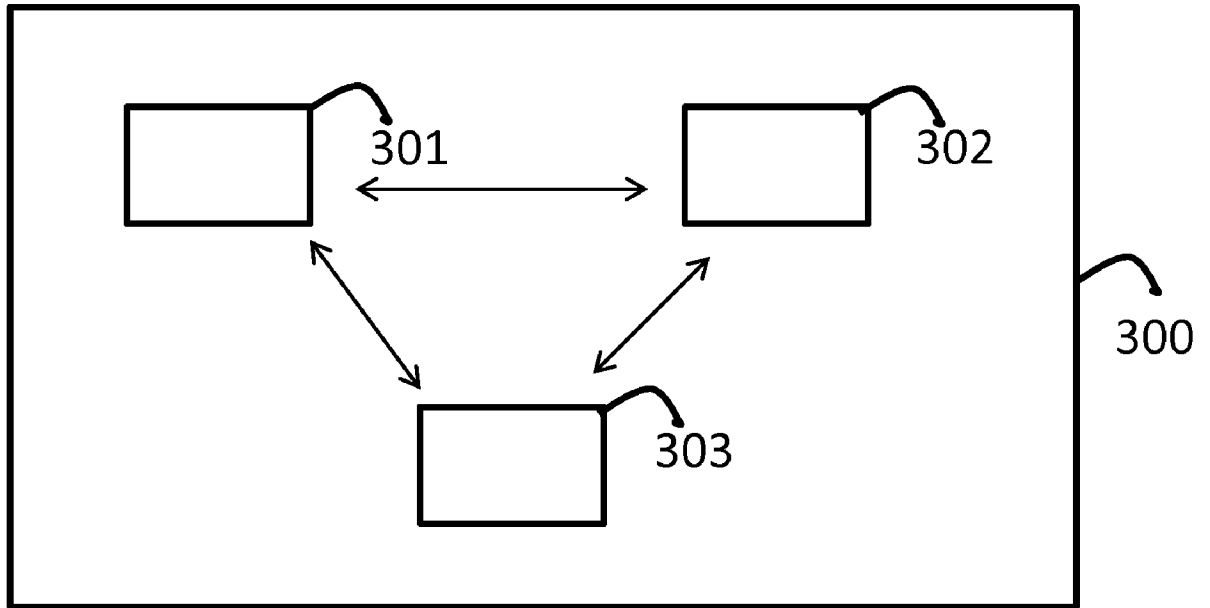


FIG. 3

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2014/072222

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L9/32
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JUTLA CHARANJIT S ET AL LI XUE XUELIOTATEE UQ EDU AU THE UNIVERSITY OF QUEENSLAND SCHOOL OF INFORMATION TECHNOLOGY AND ELECTRONIC: "Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces", 11 September 2013 (2013-09-11), LECTURE NOTES IN COMPUTER SCIENCE; [LECTURE NOTES IN COMPUTER SCIENCE], SPRINGER VERLAG, DE, PAGE(S) 1 - 20, XP047183040, ISSN: 0302-9743 cited in the application the whole document ----- -/--	1-13



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

12 January 2015

Date of mailing of the international search report

20/01/2015

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Di Felice, M

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2014/072222

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CHARANJIT JUTLA ET AL: "Relatively-Sound NIZKs and Password-Based Key-Exchange", INTERNATIONAL ASSOCIATION FOR CRYPTOLOGIC RESEARCH,, vol. 20120925:212556, 4 September 2012 (2012-09-04), pages 1-47, XP061006379, [retrieved on 2012-09-04] cited in the application the whole document -----	1-13
A	NUTTAPONG ATTRAPADUNG ET AL: "Efficient Completely Context-Hiding Quotable and Linearly Homomorphic Signatures", 26 February 2013 (2013-02-26), PUBLIC-KEY CRYPTOGRAPHY PKC 2013, SPRINGER BERLIN HEIDELBERG, BERLIN, HEIDELBERG, PAGE(S) 386 - 404, XP047022603, ISBN: 978-3-642-36361-0 the whole document -----	1-13