Identification and Privacy: Zero-Knowledge is not Enough[*]

Julien Bringer[1], Hervé Chabanne[1] and Thomas Icart[1,2]

[1]Sagem Sécurité
[2]Université du Luxembourg
name.surname@sagem.com

Abstract. At first glance, privacy and zero-knowledgeness seem to be similar properties. A scheme is private when no information is revealed on the prover and in a zero-knowledge scheme, communications should not leak provers' secrets.
Until recently, privacy threats were only partially formalized and some zero-knowledge (ZK) schemes have been proposed so far to ensure privacy. We here explain why the intended goal is not reached. Following the privacy model proposed by Vaudenay at Asiacrypt 2007, we then reconsider the analysis of these schemes and thereafter introduce a general framework to modify identification schemes leading to different levels of privacy. Our new protocols can be useful, for instance, for identity documents, where privacy is a great issue.
Furthermore, we propose efficient implementations of zero-knowledge and private identification schemes based on modifications of the GPS scheme. The security and the privacy are based on a new problem: the Short Exponent Strong Diffie-Hellman (SESDH) problem. The hardness of this problem is related to the hardness of the Strong Diffie-Hellman (SDH) problem and to the hardness of the Discrete Logarithm with Short Exponent (DLSE) problem. The security and privacy of these new schemes are proved in the random oracle paradigm.

1 Introduction

Contactless communications are used in many contexts: RFID tags, public transportation cards, electronic wallets or electronic passports. As these communications are based on radio frequency technology, it is often very simple to eavesdrop communications and even to impersonate a verifier. This doing, an adversary could either get personal information or track someone at range.

In order to protect the users' privacy, it is necessary to encrypt communications between devices and readers. To do so, an identification

[*] This work has been partially founded by the ANR T2TIT project.

scheme must be used to authenticate the device and to set up a session key. Many such schemes have been proposed so far, either based on symmetric cryptosystems [3,8,12,24,31,33,50] or asymmetric cryptosystems [15,17,20,35,37,41].

Among the few existing privacy models for identification devices – Juels and Weis [25], Burmenster, van Le and de Medeiros [28] and Vaudenay [48] – we choose to work with the latter as it is more general. Thereafter, we prove the soundness and the privacy of our new schemes by further adapting its model to take into account ZK specificities.

In this paper, we propose a general toolkit to achieve different levels of privacy for these schemes. We introduce three kinds of modifications to ZK schemes. Two of them need provers' identity to be hidden, whereas the last one can be used even when identities are public. Specific cryptographic techniques are associated to these modifications. The first modification is secure under the random oracle model [7], the second one requires a non-malleable [10] encryption scheme. The third modification is secure under the random oracle model and the Strong Diffie-Hellman (SDH) assumption (cf. [1]).

We also introduce the notion of public-identity privacy. The area of identity documents is an interesting example for this notion. In order to prove the authenticity of the information written on the document, an authentication protocol has to be used, for instance a proof of knowledge of an information written on the passport. Nevertheless this information cannot be considered as private. These documents are very frequently copied in order to keep traces of the owner, for instance in hotels or duty free area. Our goal is to obtain schemes where an eavesdropper cannot determine or trace any document even if he is aware of the valid identities. For instance, a classical Zero-Knowledge scheme cannot be public-identity private as the identity is enough to verify the validity of the protocol transcript.

We finally give examples, based on the toolkit, of efficient public-identity private protocols to bring a secure solution to this real application. This enables to compare speed, security and privacy of these schemes to the GPS scheme [17]. The security of these new schemes relies on the short exponent variant of the Decisionnal Diffie-Hellman (SEDDH) and on the new Short Exponent Strong Diffie-Hellman (SESDH) assumption introduced in this paper. These new problems enables to reduce the computational power needed for these applications.

1.1 Related Works

Zero-knowledge proofs have been introduced in [19] and zero-knowledge identification schemes in [11]. This kind of scheme is very interesting as it enables to perform an interactive proof of knowledge of secrets without revealing information. Many identification schemes have been developed and they are now envisaged in the context of travel documents. For instance, GPS and GQ [20] protocols have been proposed as identification schemes for passports in [32]. However, we prove that without modifications, such schemes do not respect privacy. In fact, this is not surprising since these protocols are not aimed at keeping the language secrets.

One idea in our propositions is to use verifiers' public keys in order to ensure the privacy of the provers' outputs. Therefore the Verifier is able to verify the authenticity of the prover thanks to his private key. Designated-Verifier Signature (DVS) schemes have been introduced in [23] and introduce the same idea. As the aim of this paper is to find private identification scheme, we do not investigate whether the proposed schemes can be transformed into a DVS scheme.

Depending of the scheme, a DVS scheme can be transformed into an identification scheme. There exist different types of DVS schemes: privately verifiable and publicly verifiable. The publicly verifiable schemes [23,29,43] have the property that given the public keys of the signer and the verifier, one can verify the signature. In this case, it is possible to identify the prover during the identification process. [40,27] are examples of privately verifiable DVS scheme. A weakness has been shown in [29] which implies that these schemes cannot be forward-private: given the secret of the signer, it is possible to determine whether he signs the message or not. We analyze the privacy of the scheme [40] in Appendix B([27] cannot be turned into a ZK identification scheme).

Jakobsson and Pointcheval proposed in [22] a mutual authentication scheme for devices with small power computation which shares some similarities with our work. This scheme has been broken in [51] and repaired. In [51], a scheme is also presented, which has also been broken and repaired later. The latter cannot be easily transform into a ZK scheme. Therefore, we only analyze the privacy of the repaired scheme in [22] in Appendix B.

1.2 Outlines

The paper is organized as follows: Section 2 describes the identification schemes studied in this paper in a general way and illustrates the lack of privacy for these schemes. Section 3 recalls Vaudenay's model and explains our modifications to it. Section 4 defines the computational assumption used in the paper. Section 5 defines our proposal of modified identification schemes to achieve privacy. Section 6 deals with the effective privacy and security of these schemes. Section 7 presents the outlines of the GPS scheme and the different modification applied on it. Section 8 deals with the possible implementations and compares the computation costs and the security for the different GPS based schemes. Section 9 concludes. Appendix A gives some details on the Section 2. Appendix B analyzes the privacy of two identification schemes in Vaudenay's model. Appendix C describes a possible modification of Vaudenay's model. Appendix D, E, F and G proves the security of the different schemes described in this paper. Appendix H describes another possible application of our work.

2 Zero-Knowledge and Identification Schemes

2.1 General Description

Among the different classes of identification protocol (ZK or closely related), Fiat-Shamir-like as [11,14,20,30,35,37], Schnorr-like as [15,17,34,41], Syndrome-Decoding-like [44,45,49] or others (e.g. [11,36,42,46]), we restrict ourselves to those based on arithmetic relations. We formalize such schemes as follows. Let $\mathcal{P}$ be a three-moves identification protocol between a prover and a verifier. Let $[A, c, B]$ a transcript of $\mathcal{P}$ with A, B sent by the prover and c by the verifier after reception of A. The protocol needs several algorithms:

- $\textsc{SetupAuthority}(1^k) \mapsto (KA_s, KA_p)$: a polynomial algorithm which outputs a private/public key pair for an authority. KA_p defines the group structure used in the scheme.
- $\textsc{SetupProver}_{KA_p}(1^k) \mapsto (s, I)$: a polynomial algorithm which outputs a private/public key pair of a prover. s is the secret linked to the identity I thanks to a one-way function Id: $Id(s) = I$.
- $\textsc{ComputeA}_{s,KA_p}() \mapsto (A, r_A)$: used to compute A thanks to a random value r_A.

- $\textsc{ComputeB}_{s,KA_p}(r_A, c) \mapsto B$: a polynomial algorithm used to compute B.
- $\textsc{Verify}(I, [A, c, B]) \mapsto x \in \{0, 1\}$: a function that checks whether the verifier identifies the prover with I.

This last algorithm checks if the following equation holds:

$$f(B) = A.g_I(c) \tag{1}$$

where f and g_I are two deterministic functions that depend on KA_p; the identity I of the prover is needed to compute g_I. Moreover, to avoid computation of B without the knowledge of s, f is one-way; more precisely the function $l_{I,A,c}$ which maps B to $\frac{f(B)}{A.g_I(c)}$ is a one-way trapdoor function: to compute efficiently B such that $l_{I,A,c}(B) = 1$, it is necessary to know the secret s and the value r_A.

Remark 1. A scheme can be multiple-round as if for instance c is a single bit. However, to resist to replay attacks, it is necessary to have exponentially many possible verifiers' outputs. In this paper, we focus on one-round protocols, but our analysis can be easily generalized to multiple-round protocols. A typical setting is c in $\left\{1, \ldots, 2^l\right\}$. If g_I is injective, two different c lead to two different couples (A, B), therefore the probability of success of a replay attack is negligible if l is large enough.

Definition 1. *A scheme is* ***sound*** *if there exists an extractor $\mathcal{E}$ which can retrieve a secret of one prover given several transcripts of the form $[A, c_i, B_i]$.*

This definition means that if an adversary is able to identify himself, he is in possession of a valid secret.

Definition 2. *A scheme is called* ***honest-verifier ZK*** *if there exists a simulator $\mathcal{S}$ able to simulate a protocol instance given the prover's identity I and a challenge c, i.e. such that $\mathcal{S}(c, I)$ outputs a pair A and B, where $[A, c, B]$ is a valid transcript identifying I.*

If such a simulator exists, a legitimate verifier cannot retrieve information on the secret. Indeed, because he is honest, he chooses c independently of A. As a consequence, once he has just chosen a c, if we are allowed to come back in the past and modify A to A', the c sent by the verifier is the same. In this case, we can use the simulator $\mathcal{S}(c, I)$ to compute A' and B'

and these values can be used in the protocol. This simulation is **perfect** if the simulator's outputs are indistinguishable from the legitimate prover's outputs. As the simulation has been made without the secret, verifiers have no advantage on the prover's secret.

2.2 Example of Privacy Leakage

We here explain through an example why this kind of ZK schemes do not respect privacy even if identities are hidden. Consider the Schnorr Identification scheme [41]: $\textsc{SetupAuthority}$ outputs a pair (KA_s, KA_p). KA_p defines a group $\mathbb{G}$, with a generator g of order q, where the discrete logarithm is a hard problem. $\textsc{SetupProver}$ randomly chooses s in $\{1, \dots, q-1\}$ and outputs $(s, I = g^s)$. $\textsc{Compute}$A outputs $(A = g^r, r)$ where r is random and $\textsc{Compute}\text{B}_{s,KA_p}(r, c) = r + sc \mod q$. $\textsc{Verify}(I, [A, c, B])$ checks whether $g^B = A.I^c$. Here, $f(B) = g^B$ and $g_I(c) = I^c$.

When a prover P proves his knowledge of the secret s related to his identity $I = g^s$(cf. Figure 1), an eavesdropper can record $[A, c, B]$. Therefore, he is able to recover I^c as $g^B A^{-1}$. If he captures transcripts from another prover, this eavesdropper gets different $I'^{c'}$ and c'. Computing $(I^c)^{c'}$ and $(I'^{c'})^c$ pairwise, the eavesdropper has a way to distinguish whether the provers have the same identity or not, even if the group order is unknown (In the case where the group order is known, I can be easily retrieve). Therefore, this scheme is traceable. Moreover, with several couples (c_i, I^{c_i}), one can compute I using the Bézout identity as $\gcd\left((c_i)_i\right)$ is certainly 1.

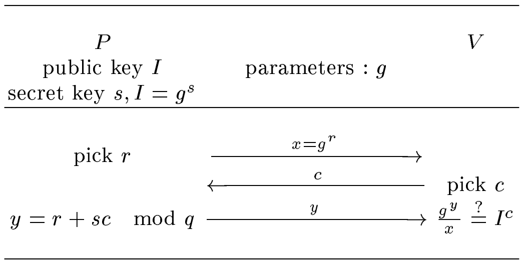

Fig. 1. Schnorr identification scheme

The same holds for other identification schemes where g_I is an exponentiation. Therefore, GPS [17], Fiat-Shamir [14] and GQ [20] protocols share the same weakness in regard of the respect of privacy. This leakage is also valid for every generalization of these schemes, such as GQ2 [37], Ong-Schnorr [35], Okamoto [34], a generalization of Fiat-Shamir [30], a generalization of Schnorr over RSA rings [15].

3 Vaudenay's Model

We recall in this section the model for privacy, correctness and soundness as described in [48]. Note that we slightly modify it in Section 3.2 in order to include zero-knowledgeness and to distinguish cases when provers' identities are public or hidden.

Following [48], we consider that provers are equipped with Contact-Less Device (CLD) to identify themselves. This device may either be RFID tags or contact-less smartcards. CLDs are transponders identified by a unique Serial Number (SN). Nevertheless, during the identification phase, a random virtual serial number (vSN) is used to address them, for instance as defined in the ISO/IEC 14443-3 standard [21]. CLDs can perform usual basic cryptographic functions. For instance, hash calculations [13], pseudo random generation [39] and public key encryption [2,4,9,16]. Furthermore, we assume CLDs can be compromised.

3.1 Description of the Model

In the model, an identification protocol is defined as functionalities, firstly to setup the system made of a verifier and several CLDs, secondly to run a protocol between CLDs and verifiers.
Setup Algorithms.

- $\textsc{SetupAuthority}(1^k) \mapsto (KA_s, KA_p)$ outputs a private/public key pair of an authority.
- $\textsc{SetupVerifier}_{KA_p}()$ generates a private/public key pair (KV_s, KV_p), possibly none. This pair can be used to protect communication between CLDs and verifiers.
- $\textsc{SetupCLDSecret}_{KA_p}(\text{SN})$ returns the parameters of the CLD identified by SN. This algorithm outputs a couple (s, I) where s is the private key of the CLD, I its public key and identity.
- $\textsc{SetupCLDState}_{KV_p}(\text{SN}, s, I)$ returns S, some data to initialize the internal memory of the CLD.
- $\textsc{SetupCLD}_{KV_p}(\text{SN})$ first uses $\textsc{SetupCLDSecret}$ then $\textsc{SetupCLDState}$, and stores the pair (I,SN) in a database.

Remark 2. As SETUPCLDSECRET$_{KA_p}$ does not use KV_p to compute s and I, the knowledge of KV_s does not leak any information on the secret s. For instance, if KA_p defines a group of generator g where the Computational Diffie-Hellman assumption holds, if the pair (KV_s, KV_p) is (v, g^v) for a random v and if $I = Id(s) = g^s$, the knowledge of v does not leak information on s.

Communication Protocol $\mathcal{P}$. The identification protocol between a CLD and a verifier is made of messages sent by the two parties. At the end of the protocol, the verifier checks the result. In this paper, we focus on the three-pass protocols described in Section 2.

Oracles. To formalize possible actions of adversaries, different oracles are defined to represent ways for adversaries to interact with verifiers or CLDs, or to eavesdrop communications. The use of different oracles leads to different privacy levels.

Given a public key KV_p, the adversaries have access to:

- CREATECLD(SN): creates a CLD with serial number SN initialized via SETUPCLD. At this point, it is a free CLD, i.e. not yet in the system.
- DRAWCLD(*distr*)$\mapsto$((vSN$_1$,b_1),...,(vSN$_n$,b_n)): this oracle moves a random subset of CLDs from the set of free CLDs into the set of drawn CLDs in the system. Virtual serial numbers (vSN$_1$,...,vSN$_n$), used to identify the drawn CLDs, are randomly chosen according to the distribution *distr*. If b_i is one, this indicates whether a CLD is used in the system. This oracle creates and keeps a table of correspondences $\mathcal{T}$ where $\mathcal{T}$(vSN)=SN. Adversaries have no knowledge of this table $\mathcal{T}$.
- FREE(vSN): moves the drawn CLD vSN to the set of free CLDs.
- LAUNCH() $\mapsto \pi$: makes the verifier launch a new protocol instance π.
- SENDVERIFIER$(m, \pi) \mapsto m'$ (*resp.* SENDCLD$(m', \pi) \mapsto m$): sends the message m to the verifier (*resp.* m' to the CLD) who responds m' (resp. m) in the protocol instance π.
- RESULT$(\pi) \mapsto x$: when π is a complete instance of $\mathcal{P}$, it returns 1 if the verifier succeeds in identifying a CLD from π and 0 otherwise.
- CORRUPT(vSN)$\mapsto S$: returns the internal state S of the CLD vSN.

Types of Adversaries.

- **Strong** adversary is allowed to use a priori all of the above oracles.
- **Destructive** adversary destroys the CLDs it corrupts. Therefore, after a query CORRUPT(vSN), no further query can be made on vSN.

- **Forward** adversary cannot use any oracle after one CORRUPT query, i.e. destructs the system when he corrupts one CLD.
- **Weak** adversary is not allowed to use the CORRUPT oracle.
- **Narrow** adversary is not allowed to use the RESULT oracle.

This defines 8 kinds of adversaries because Narrow adversaries may also have restriction on the use of the CORRUPT oracle. For instance, an adversary can be narrow and forward, he is then denoted narrow-forward.

Three security notions are defined in this model: correctness, soundness and privacy.

Definition 3. *A scheme is* **correct** *if the identification of a legitimate CLD fails only with negligible probability.*

Soundness. The definition of soundness below deals with active adversaries. Active adversaries are able to impersonate verifiers, CLDs and to eavesdrop and modify communications. This property is useful for privacy (cf. Lemma 1). Soundness of ZK schemes is already widely studied, even against active adversaries, see for instance [6]. Nevertheless, this definition is very specific to the model above.

Definition 4. *A scheme is* **sound** *if any polynomially bounded* **strong** *adversary cannot produce a protocol instance π of a communication leading to a positive result of the oracle* RESULT, *but with a negligible probability. π must neither be equal to some eavesdropped protocol transcript between a legitimate CLD and a legitimate verifier nor lead to the identification of a corrupted CLD.*

Remark 3. Obviously it means that a scheme is not sound if an adversary is able to modify on the fly outputs from a prover without affecting the identification result. Moreover, even if not addressed by the definition, replay attacks must be mitigated, that is why the probability that a legitimate verifier outputs twice the same values in a complete protocol instance should be negligible.

Privacy. Privacy is defined as an advantage of an adversary over the system. More precisely, there is a privacy leakage if the system cannot be simulated. To formalize this, [48] proposes to challenge the adversary once with the legitimate oracles and a second time with simulated oracles. In this setting, the adversary is free to define a game and an algorithm $\mathcal{A}$ to solve his game. If the two challenge's results are distinguishable, then there is a privacy leakage. A game with three phases is imposed. In the first phase, $\mathcal{A}$ has access to the whole system using oracles. In a second phase, the hidden table $\mathcal{T}$ constructed via DrawCLD oracle is transmitted to $\mathcal{A}$. In a third phase, $\mathcal{A}$, which is no longer allowed to use the oracles, outputs its result. A scheme is defined as **private** if for any game, all adversaries are trivial (see Def. 6).

Definition 5. *A* **blinded** *adversary uses simulated oracles instead of the oracles* Launch*,* SendVerifier*,* SendCLD *and* Result*. Simulations are made using an algorithm called a* **blinder** *denoted $\mathcal{B}$.*

To simulate oracles, a blinder has access neither to the private database nor to the secret key KV_s. We denote $\mathcal{A}^{\mathcal{B}}$ the algorithm $\mathcal{A}$ when executed using the blinder instead of legitimate oracles.

Definition 6. *An adversary is* **trivial** *if there exist a blinder $\mathcal{B}$ such that the difference $\left|\Pr\left[\mathcal{A}\ wins\right] - \Pr\left[\mathcal{A}^{\mathcal{B}}\ wins\right]\right|$ is negligible, i.e. he makes no clever use of the oracles.*

Hence, to prove privacy, it suffices to prove that an adversary cannot distinguish between the outputs of the blinder $\mathcal{B}$ and outputs made by legitimate oracles. As stated in [48], this definition of privacy is more general than anonymity and untraceability. As there are 8 kinds of adversaries, there are 8 different notions of privacy. In the sequel, we will always mention which notion we refer to.

Note that Corrupt queries are considered to always leak information on CLDs' identity. For instance, an adversary can systematically open CLDs in order to track them. In this model, such an adversary is considered as a trivial one because a blinded adversary will succeed the same way, as the Corrupt oracle is not simulated. Strong privacy is defined only to ensure that CLDs cannot be tracked using their outputs even if their secrets are known.

The following lemma established by Vaudenay in [48] emphasizes the link between soundness and privacy:

Lemma 1. *A scheme sound and narrow-weak (resp. narrow-forward) private is weak (resp. forward) private.*

The proof relies on the fact that an adversary is not able to simulate any CLD as the scheme is sound. This implies that the Result oracle is easily simulated by the blinder.

3.2 Adaptation of the Model for ZK Identification Schemes

Zero-Knowledgeness. Privacy implies that an adversary gains no advantage over the identity of CLD (nor on anything else in the system). Nevertheless, we still want to prove that even legitimate verifiers do not have any advantage on the CLD's secret. The difference between adversaries and legitimate verifiers is that legitimate verifiers know the secret KV_s while adversaries or provers are unaware of it. To prove the privacy, it is necessary to perfectly simulate CLDs' outputs from an adversary point of view. In the case where KV_s is not used, for instance in the schemes described in Section 2.1, it is equivalent to a verifier point of view; which is not the case when KV_s is needed. However our setup algorithms are defined to ensure that secret s and KV_s are independent (cf. Remark 2). A simulator against a honest verifier is thus authorized to use KV_s to simulate CLDs' outputs, as it gives it no advantage on s. *In fine*, the definition of **honest-verifier ZK** schemes remains unchanged from Def. 2 but the use of KV_s will be considered in the security analysis of our modified schemes (cf. Section 6.2 and Appendix D.3).

New Adversaries: Hidden-Identity and Public-Identity.
A hidden-identity adversary does not have the list of identities chosen during the setup phase initiated by the multiple calls to SetupCLD. A public-identity adversary has access to this list. In the following, we precise **HI** for hidden-identity if identities stay hidden, and **PI** for public-identity if they are public.

4 Computational Assumptions

In the following, we will note $\mathbb{G}$ a group structure. We assume that $\mathbb{G}$ is cyclic and that its –possibly unknown– order is q. We call g one generator of this group. The Discrete Logarithm(DL) problem can be defined as:

- Given g and g^a in $\mathbb{G}$ with a randomly chosen in $[0, ..., q-1]$,
- Compute a.

The discrete logarithm with short exponent (DLSE) problem is the usual DL problem but with short exponents instead of normal ones. It has been introduced in [47]. For instance, a is a random element in $[0, ..., S-1]$ where S is smaller than q. This defines the DLSE(S) problem.

The computational Diffie-Hellman (CDH) problem can be defined as:

- Given g, g^a and g^b with a and b randomly chosen in $[0, ..., q-1]$,
- Compute g^{ab}.

There exists a similar problem where the exponents are short. This problem is called the Short Exponent Computational Diffie-Hellman (SECDH) problem and has been introduced in [26]. It is made of two subproblems: the (short,full)SECDH problem and the (short,short)SECDH problem, following that only a is a short exponent and b is a random full exponent or a and b are short exponents. We call it (S_1, S_2)SECDH problems for $a \leq S_1$ and $b \leq S_2$. As a result, it is proved in [26] that the problems (S, S)SECDH and (q, S)SECDH are hard if and only if CDH and DLSE(S) problems are hard.

Let us define the Decisional Diffie-Hellman (DDH) problem:

- Given g, g^a, g^b with a and b randomly chosen in $[0, ..., q-1]$,
- Given $g^c = g^{ab}$ with probability $1/2$ and $g^c = g^d$ with probability $1/2$ with d randomly chosen in $[0, ..., q-1]$,
- Decide whether g^{ab} equals g^c.

The hardness of the DDH problem implies the hardness of the CDH problem. As for the CDH problem, the SEDDH problem is defined in [26]. It is also made of two subproblems according to the fact that a is short or a and b are short. The hardness of the (q, S)SEDDH problem and (S, S)SEDDH problem are equivalent to the hardness of the DLSE(S) problem combined with the hardness of the DDH problem. Let us now introduce a DDH oracle $\mathcal{O}_a^{DDH}$:

- Given g, g^d with d randomly chosen in $[0, ..., q-1]$
- Given g_1, an element of $\mathbb{G}$
- Returns whether g^{ad} equals g_1.

This oracle $\mathcal{O}_a^{DDH}$ solves the DDH problem associated to a.

The Strong Diffie-Hellman (SDH) problem is defined as:

- Given g, g^a and g^b with a and b randomly chosen in $[0, ..., q-1]$,
- Given access to the oracle $\mathcal{O}_a^{DDH}$
- Compute g^{ab}.

Definition 7. *We define the (S_1, S_2) Short Exponent Strong Diffie-Hellman problem ((S_1, S_2)SESDH).*

- *Given g, g^a and g^b with a randomly chosen in $[0, ..., S_1 - 1]$ and b randomly chosen in $[0, ..., S_2 - 1]$,*
- *Given access to the oracle $\mathcal{O}_a^{DDH}$*
- *Compute g^{ab}.*

As far as we know, this is the first definition of the SESDH problem.

Proposition 1. *Assume the hardness of the SDH problem and the hardness of the DLSE(S) problem, assume $q > 2S$, then the (q, S)SESDH problem is hard.*

Proof. First, we need to prove a lemma. We define an oracle $\mathcal{O}_S^{DLSE}$, which as inputs (g, g^a) determines whether a is in $[0, S-1]$. As a remark, this oracle can be transformed into the oracle $\mathcal{O}_{S'}^{DLSE}$ for $S' = \lfloor \frac{S}{\lambda} \rfloor$ smaller than S and for λ integer. This is possible because $\mathcal{O}_{S'}^{DLSE}(g, g^a) = \mathcal{O}_S^{DLSE}(g, g^{\lambda a})$.

Lemma 2. *Assume $q > 2S$, assume there exists an oracle $\mathcal{O}_S^{DLSE}$, then DLSE(S) can be solved in $\log_2(S)$ calls to $\mathcal{O}_S^{DLSE}$.*

Sketch of the proof. We describe an algorithm $\mathcal{A}$ which inputs g, g^a, S and outputs a if a in $[0, S-1]$. We assume S is a power of 2 to simplify the proof.

Algorithm $\mathcal{A}$. INPUT=(g, g^a, S), OUTPUT=a.
If g^a is not in $[0, S-1]$, we stop the algorithm. If g^a is the neutral element, then we output 0. Otherwise, if g^{2a} is in $[0, S-1]$, we output $\mathcal{A}(g, g^a, \frac{S}{2})$ else we output $\frac{S}{2} + \mathcal{A}(g, g^{a-S/2}, \frac{S}{2})$.

It is clear that this algorithm terminates in at most $\log_2(S)$ calls to the oracle $\mathcal{O}_S^{DLSE}$ and outputs a. □

Now assume there exists an oracle $\mathcal{O}_S^{SESDH}$ which solves in polynomial time the (q, S)SESDH problem. As input, it receives g, g^a, g^b and $\mathcal{O}_a^{DDH}$. We assume this oracle outputs an answer even if the inputs are not a valid (q, S)SESDH problem instance but a SDH problem instance. We distinguish two cases:

1. either this oracle solves the usual SDH problem with a non negligible probability,
2. or it solves the SDH problem with a negligible probability.

In the latter, we can transform the $\mathcal{O}_S^{SESDH}$ into an $\mathcal{O}_S^{DLSE}$ oracle: Given g, g^a, we want to determine whether a is in $[0, S-1]$.

- We randomly choose b in $[0, q-1]$ and we compute g^b. As we have chosen b, we can simulate the oracle $\mathcal{O}_b^{DDH}$.
- We execute $\mathcal{O}_S^{SESDH}(g, g^b, g^a, \mathcal{O}_b^{DDH})$ which outputs g^u.
- If $g^u = g^{ab}$, a is in $[0, S-1]$, else a is bigger than S.

This algorithm is a $\mathcal{O}_S^{DLSE}$ oracle. As demonstrated in the lemma, this implies that the DLSE(S) problem can be solved in polynomial time.

For these reasons, the hardness of the (q, S)SESDH problem is ensured by the hardness of the SDH problem and the DLSE problem. □

We do not manage to prove the hardness of the (S, S)SESDH and (S, q) SESDH problems. Nevertheless, these problems seem hard under the same assumption. We do not deal with the Small Exponent Gap Diffie-Hellman (GDH) problem as we do not have application for this problem in this paper. It can be proved equivalent to the GDH problem and the DLSE problem with a similar proof.

5 General Toolkits for Privacy

In the model presented previously, there exist non trivial narrow-weak adversaries against identification schemes as the ones defined in Section 2.1, even with hidden-identities (cf. Sec 2.2). We describe here general methods to enhance privacy of these schemes. We state their correctness, soundness, privacy and zero-knowledgeness properties in the next section.

5.1 Setup Algorithms

The SetupAuthority is the algorithm of the original scheme as defined in Section 2.1. KA_p defines a group $\mathbb{G}$ which is the group structure used in the identification process. If there exists a generator, we denote it g. Its order is possibly unknown. The SetupVerifier$_{KA_p}$ algorithm is described in Section 3. The SetupCLDSecret$_{KA_p}$ algorithm uses SetupProver as defined in Section 2.1 and SetupCLDState$_{KV_p}$ outputs the initial state $S = (KA_p, KV_p, s)$.

5.2 Modification of Identification Schemes

We give a general framework to bring various levels of privacy to existing protocols. Some examples of application are given in Section 7 and in

Appendix H. We start with a protocol $\mathcal{P}$ as defined in Section 2.1. In the sequel, we assume that g_I is an exponential function in base I (as it is the case for Fiat-Shamir-like schemes or Schnorr-like schemes).

Modification 1: Classical Introduction of the Random Oracle [7] This well-known modification is simply to change the first message into its hash. Let H be a random oracle. The verifier possesses a list L of the form $\{\ldots,(I,SN),\ldots\}$. This list is kept hidden. The protocol can be described as follows:

1. the CLD uses $\textsc{ComputeA}_{s,KA_p}() \mapsto (A, r_A)$ and outputs $H(A)$,
2. the verifier sends a random c to the CLD,
3. the CLD uses $\textsc{ComputeB}_{s,KA_p}(r_A, c) \mapsto B$ and outputs B.

A verifier should check amongst identities I in L whether $H(f(B)g_I(c)^{-1})$ and $H(A)$ are equal. We prove in the sequel that this modification gives a **HI** weak private protocol, **HI** sound and ZK (this last property is already known).

Modification 2: Use of an Encryption Scheme Let (E_{KV_p}, D_{KV_s}) be an encryption scheme. We suggest to use this scheme to cipher the first message sent to the verifier. The verifier possesses a list L of the form $\{\ldots,(I,SN),\ldots\}$ and this list is kept hidden. The protocol can be described as follows:

1. the CLD uses $\textsc{ComputeA}_{s,KA_p}() \mapsto (A, r_A)$ and outputs $E_{KV_p}(A)$,
2. the verifier sends a random c to the CLD,
3. the CLD uses $\textsc{ComputeB}_{s,KA_p}(r_A, c) \mapsto B$ and outputs B.

A verifier should check amongst identities I in L whether $D_{KV_s}\left(E_{KV_p}(A)\right) \cdot g_I(c)$ is equal to $f(B)$. If the encryption scheme is semantically secure [18] (IND-CPA), the modification leads to a narrow-strong private[1] protocol and ZK. Furthermore, if the encryption is non-malleable [10] (NM-CPA), the modification gives **HI** soundness and **HI** forward privacy.

Modification 3: a Public-Identity Private Protocol We present here a modification in order to obtain a **PI** forward private protocol, **PI**

[1] Strong adversaries against privacy are able to get all the secrets of any CLD, so **PI** strong adversaries and **HI** strong adversaries are equivalent.

sound, ZK and narrow-strong private. This modification is more specific: we assume that there exists a generator g of $\mathbb{G}$ and that KA_p defines an exponent e. We use the pair (v, g^v) as (KV_s, KV_p) and the random oracle H. The group structure must ensure the hardness of the SDH problem. Furthermore, we assume that $\textsc{ComputeA}_{s,KA_p}()$ outputs $(A = (g^e)^{r_A}, r_A)$. The verifier possesses a list L of the form $\{\ldots, (I, SN), \ldots\}$ and this list is public. The protocol can be described as follows:

1. the CLD uses $\textsc{ComputeA}_{s,KA_p}()$ twice to obtain (A_1, r_1) and (A_2, r_2), computes $A_1^v = (g^v)^{er_1}$, $A_2^v = (g^v)^{er_2}$ and outputs A_1, $H(A_1, A_2, A_1^v, A_2^v)$,
2. the verifier sends a random c to the CLD,
3. the CLD uses $\textsc{ComputeB}_{s,KA_p}(r_1 + r_2, c) \mapsto B$ and outputs B.

A verifier computes $A'_2 = \frac{f(B)}{A_1 g_I(c)}$ for identities I in L and checks whether $H(A_1, A'_2, A_1^v, A'^v_2)$ equals $H(A_1, A_2, A_1^v, A_2^v)$ using his secret v. This technique is illustrated in the Section 7 over the GPS scheme.

This modification has some similarities with the scheme in [22]. Firstly, a Diffie-Hellman key is used as input of the hash function. Secondly, almost all the computations can be made offline and do not reveal information on the secret of the prover. Nevertheless the scheme in [22] is not sound and not forward-private in the model. This difference exists because we used two Diffie-Hellman keys in the hash, linked to the final output of the CLD.

We discuss in Appendix C about the possibility to modify the simulation of the $\textsc{Result}$ oracle to get, with this modification, a strong private scheme.

6 Security Properties

In the sequel, we assume I is a generator of $\mathbb{G}$ and f is a bijection. All the proofs are detailed in the Appendix D. Note first that the modifications 1,2 and 3 in 5 keep the correctness of the original scheme.

6.1 Narrow Privacy

To prove narrow-weak privacy, it is sufficient to perfectly simulate any CLD without having any identity of tags.

Theorem 1.

- *If H is a random oracle then **Modification 1** defines a* **HI** *narrow-weak private scheme.*
- *If E_{KV_p} is IND-CPA secure then the **Modification 2** defines a* **PI** *narrow-strong private scheme.*
- *If H is a random oracle and if the SDH assumption holds then **Modification 3** defines a* **PI** *narrow-strong private scheme.*

This is the best narrow-privacy achievable for this construction. For **Modification 1**, if one CLD is corrupted, its identity is retrieved and it is possible to identify it thanks to protocol instances. For modifications **2** and **3**, an adversary who does not know the verifier's secret is not able to distinguish between simulated tags and legitimate tags even if he has all tags' secrets.

6.2 Zero-Knowledgeness

Narrow-privacy implies that no adversary is able to have an advantage on the identities. This property also implies that adversaries have no advantage over secrets as there exists a deterministic function Id such that $Id(s) = I$. Therefore, whatever the adversaries's strategy, they fail getting information on the secret just by eavesdropping or interrogating tags. Nevertheless, it differs from ZK so it is necessary to prove that legitimate verifiers cannot find information on these secrets:

Theorem 2. *Assume there exists a simulator $\mathcal{S}(c, I)$ to the original scheme, then it can be transformed into a simulator $\mathcal{S}_i(c, I)$ for the modification i. Hence, if the original scheme is honest-verifier ZK, then modifications **1**, **2** and **3** define honest-verifier ZK schemes.*

Note that $\mathcal{S}_3$ uses the key KV_s to simulate tags' outputs.

6.3 Soundness

Theorem 3. *Assume the soundness of the original scheme.*

- *If H is a random oracle and identities are kept hidden, then **Modification 1** defines a sound scheme.*
- *If E_{KV_p} is NM-CPA secure and the identities are kept hidden then **Modification 2** defines a sound scheme.*
- *If H is a random oracle, identities are public and the SDH assumption holds, then **Modification 3** defines a sound scheme.*

Note that for **Modification 2** with only IND-CPA encryption scheme, an adversary could modify a ciphertext. For instance, if the scheme is re-randomizable, an adversary can modify the encrypted part without changing the clear text. In this case, he is in possession of a ciphertext which has a non trivial relation with the original one. This could lead to a valid transcript, which means we cannot reach soundness.

Compared to **Modification 2**, identities can be public in **Modification 3**. The main reason is that an adversary will need to know the couples (I, I^v) to be able to impersonate the identity I, whereas we can prove that computing I^v is here as hard as the SDH assumption. In fact, **Modification 3** is a kind of generalization of **Modification 1** with computations made in the two bases g and g^v at the same time.

6.4 Privacy

Theorem 4. *Assume the soundness of the original scheme.*

- *If H is a random oracle then **Modification 1** defines a* **HI** *weak private (and not forward) scheme.*
- *If E_{KV_p} is NM-CPA secure then **Modification 2** defines a* **HI** *forward private scheme (and not destructive).*
- *If H is a random oracle and if the SDH assumption holds then **Modification 3** defines a* **PI** *forward private scheme (and not destructive).*

These results are due to Theorems 1, 3 and the Lemma 1. This is the best privacy achievable for this construction. **Modification 1** cannot lead to a forward private scheme because whenever the secret from a CLD is discovered thanks to the Corrupt oracle, all the outputs of this CLD can be traced.

Modifications **2** and **3** are not destructive private because we can prove there exists a way to compute transcripts such that the Result oracle cannot be simulated on them.

7 Variations Around the GPS Scheme

In this section we applied our modification to the well-known GPS scheme. The GPS scheme, described in [17] is a very efficient Zero-Knowledge identification scheme. Its security is based on the hardness of the DLSE problem. This scheme is correct, sound against passive adversary and statistically ZK but we have proved that it does not respect privacy.

More precisely, we have shown that information leak on the identity of the prover involved in such schemes.

The GPS scheme falls in the scheme we have described in Section 2.1. For this reason, the modifications presented in Section 5 lead to different schemes. We here describe the explicit schemes.

7.1 The original GPS Scheme

The computations are made over a group $\mathbb{G}$. Let g be a basis element of $\mathbb{G}$. Define three integers A, B and S. To be secure, GPS needs a group where the DLSE(S) problem is hard. The parameter B is related to the security against replay attack. Finally, A is related to the zero-knowledgeness of the scheme. The scheme is described in Figure 2.

$Prover$		$Verifier$
public key I	parameter : g	List L
secret key $s \in [0, S-1]$		
$I = g^s$		
pick $r_1 \in [0, A-1]$		
	$\xrightarrow{x=g^{r_1}}$	
	$\xleftarrow{c}$	pick $c \in [0, B-1]$
$y = r_1 + sc$	$\xrightarrow{y}$	
		check whether there exists $I \in L$
		such that $g^y x^{-1} = I^c\ textand$
		$0 \leq y \leq A - 1 + (B-1)(S-1)$

Fig. 2. GPS scheme

Each prover has an identity I and a secret s smaller than S with $I = g^s$. Because the DLSE(S) problem is hard, computing s given $I = g^s$ is infeasible. The aim of the GPS scheme is to enable a prover to prove his knowledge of the secret associated to his identity. Furthermore, this proof is made in order to leak no information on his secret to the verifier.

In a first step, the prover randomly chooses r_1 in $[0, A-1]$. Then he computes $x = g^{r_1}$ and he sends this value to the verifier. The verifier sends a challenge c randomly chosen in $[0, B-1]$. The prover responds with $y = r_1 + sc$. This computation is made without any modular reduction, this is one of the main differences with the Schnorr scheme. The verifier checks whether there exists $I \in L$ such that $g^y x^{-1} = I^c$ and whether

$0 \le y \le A - 1 + (B-1)(S-1)$. If these two conditions are verified, the prover is identified.

7.2 Modification 1: Hashed GPS

In [17], a trick is used in order to decrease the total amount of bits sent by the prover. It is possible to slightly modify the scheme using a pseudo-random function H with outputs length of l bits. The scheme is described in Figure 3.

$Prover$		$Verifier$
public key I	parameter : g	List L
secret key $s \in [0, S-1]$		
$I = g^s$		
pick $r_1 \in [0, A-1]$		
	$\xrightarrow{x=H(g^{r_1})}$	
	$\xleftarrow{c}$	pick $c \in [0, B-1]$
$y = r_1 + sc$	$\xrightarrow{y}$	
		check whether there exists $I \in L$
		such that $H(g^y I^{-c}) = x$ and
		$0 \le y \le A - 1 + (B-1)(S-1)$

Fig. 3. Hashed GPS

The difference is that the first message is no more g^{r_1} but $H(g^{r_1})$. For instance, if H is SHA-256, the first message is just 256-bit long instead of 1536-bit long. Nevertheless, this gain is counterbalanced as it implies that more computations are needed during the identification process (two hash computations are needed, one on the prover side and one on the verifier side).

To guarantee the security, the length of the hashed values must be bigger than B. In this case the probability to get pre-images of the function is bigger than the probability to succeed in impersonating provers through a replay attack. The authors of [17] suggest $B = 2^{32}$ and 50-bit long hash values.

As a result of the previous sections, we have the following lemma:

Lemma 3. *Assume the hardness of the DLSE*(S) *problem, assume H is a random oracle and assume $\frac{BS}{A}$ is negligible then the hashed GPS is hidden-identity weak private.*

7.3 Modification 2: Randomized GPS

In the sequel, we assume that the verifier possesses a pair of key (g^v, v) where v is a private random exponent while g^v is public. Depending on the application, this value could be stored on the prover side or sent at the beginning of the identification process. In order to guarantee its legitimacy, it should be sent signed by an authority. As a result of the previous section:

Lemma 4. *Assume the hardness of the DLSE(S) problem, the hardness of the DDH problem and assume $\frac{BS}{A}$ is negligible then Randomized GPS is sound against passive adversary, statistically ZK and narrow-strong private.*

P		V
public key I	parameters : g, g^v	secret key v
secret key $s \in [0, S-1]$		List L
$I = g^s$		
pick $r_1 \in [0, S-1]$		
pick $r_2 \in [0, A-1]$	$\xrightarrow{A_1=g^{r_1}, A_2=(g^v)^{r_1} g^{r_2}}$	
	$\xleftarrow{c}$	pick $c \in [0, B-1]$
$y = r_2 + sc$	$\xrightarrow{y}$	Compute
		$I = (\frac{g^y A_1^v}{A_2})^{1/c}$ and
		Check if $I \in L$ and
		$0 \leq y \leq A - 1 + B(S-1)$

Fig. 4. Randomized GPS

Proofs can be easily deduced from the proof of the modification 2 in Section 5.

The difference with the original GPS scheme is that we compute an encryption of g^{r_2} thanks to the El Gamal scheme with small exponent. This encryption scheme is IND-CPA if the SEDDH is a hard problem. If an NM-CPA encryption scheme is used, the privacy is enhanced to the HI forward privacy.

In this model, this scheme cannot be proved sound against active adversary because an adversary can compute offline a protocol transcript. This is possible because there exists a simulator of the GPS scheme. This simulator enables to compute y and $A = g^{r_2}$ from I and c. Then the adversary of the Randomized GPS scheme can randomly choose r_1 in $[0..S-1]$, can compute $A_1 = g^{r_1}$ and $A_2 = (g^v)^{r_1} g^{r_2}$ and output A_1, A_2, y.

7.4 Modification 3: Randomized Hashed GPS

As for the GPS scheme, it is possible to minimize the amount of bits sent thanks to a hash function H. The use of this function increases the soundness and the privacy of the scheme.

Lemma 5. *Assume the hardness of the DLSE*(S)*, the hardness of the SDH problem, assume* H *is a random oracle and assume* $\frac{BS}{A}$ *is negligible then Randomized hashed GPS is public-identity forward private, sound against active adversaries, narrow-strong private and statistically ZK.*

Even if the proof of this lemma is similar to the proof of the modification 3, we rewrite it in order to prove that the choice of the parameter, especially the size of the exponent, is appropriate. The proof is in Appendix E and Appendix F. The computation needed during the scheme can be summarized(cf. Fig 5):

P public key I secret key $s \in [0, S-1]$ $I = g^s$	parameters : g, g^v	V secret key v List L
pick $r_1 \in [0, A-1]$		
pick $r_2 \in [0, A-1]$	$\xrightarrow{z=H\left(A_1=g^{r_1}, g^{r_2}, A_1^v, (g^v)^{r_2}\right)}$	
	$\xleftarrow{c}$	pick $c \in [0, B-1]$
$y = r_1 + r_2 + sc$	$\xrightarrow{A_1, y}$	Check, if there exists $I \in L$ such that $z = H(A_1, (\frac{g^y}{A_1 I^c}), A_1^v, (\frac{g^y}{A_1 I^c})^v)$ and $0 \leq y \leq 2A - 2 + (B-1)(S-1)$

Fig. 5. Randomized Hashed GPS

1. the prover randomly chooses r_1 in $[0, A-1]$ and r_2 in $[0, A-1]$,
2. the prover computes $A_1 = g^{r_1}$, $A_1^v = (g^v)^{r_1}$, $A_2 = g^{r_2}$ and $A_2^v = (g^v)^{r_2}$,
3. the prover sends $H(A_1, A_2, A_1^v, A_2^v)$ and keeps A_1 and $r_1 + r_2$ in memory,
4. the verifier sends a random c in $[0, B-1]$,
5. the prover sends A_1 and $r_1 + r_2 + sc$
6. the verifier checks whether $z = H(A_1, \frac{g^y}{A_1 I^c}, A_1^v, (\frac{g^y}{A_1 I^c})^v)$ and $0 \leq y \leq 2A - 2 + (B-1)(S-1)$.

If the verification succeeds, the prover is identified. Furthermore, A_1^v can be a semi-static Diffie-Hellman key and can be used later to derive a session key.

As the scheme is now public-identity forward private, it could be used in common applications where identities are public. Moreover if they are sent by the provers on a covert channel (think of the Machine Readable Zone (MRZ) on identity documents) which cannot be eavesdrop, then the verification is simplified as the search in the list is not needed.

8 Implementations Issues

In this section, we firstly describe the parameters size and the possible trick to optimize the schemes. Secondly, we describe two possible adaptations of the randomized hashed GPS to get new properties.

8.1 Practical Implementation Setting

Parameters Description We describe two typical sets of parameter. Firstly, $\mathbb{G}$ is a subgroup of $\mathbb{Z}_p$ where p is a prime integer of 1536 bits. There must exist a prime integer q of at least 160-bit long that divides $p-1$. Let g be an element of order q and $\mathbb{G}$ is the group generated by g. In this implementation, S is 2^{160}. Computing a discrete logarithm in basis g of exponent smaller than S is assumed to require more than 2^{80} multiplications over $\mathbb{G}$. Depending on the desired level of security, B can lie in $\left[2^{32}, 2^{160}\right]$. The probability of success of a replay attack is $\frac{1}{\sqrt{B}}$. Finally, A must be greater than $B \times S \times 2^{80}$ to ensure the zero-knowledgeness. For instance, $B = 2^{80}$ gives $A = 2^{320}$.

Secondly, $\mathbb{G}$ is the group of point of an elliptic curve E_p over $\mathbb{F}_p$ where p is a 160-bit long prime. This group have a prime group order q. The values of $A = 2^{320}, B = 2^{80}$ and $S = 2^{160}$ can still be used. With this setting, it is not useful to use short exponents.

Precomputations for the GPS Scheme It is essential to find the best way to implement the schemes in order to provide a fast identification process. It is necessary to find solutions to accelerate the computation as computational power can be limited on some devices.

Instead of computing at each identification $H(g^{r_1})$ as described in Section 7.2, a possibility is to precompute these values and to store couples

$(r_1, H(g^{r_1}))$. These couples are referred as coupons. Therefore, at each identification, the prover only computes $r_1 + sc$. Usually r_1 is the output of a pseudo-random function which has a seed as input. It is thus possible to store r_1 efficiently. Instead of storing the first part of all the coupons, it is possible to store only the seed used in the pseudo-random function and to update it each time. Consequently, each coupon has the size of its hash value. If 4 KBytes of memories can be allocated to the storage of coupons with coupons of 50-bit long, this leads to 640 coupons stored.

Precomputations for the Randomized Hashed GPS Scheme The previous idea can be applied to the Randomized hashed GPS scheme. It is possible to precompute multiple $H(A_1, A_2, A_1^v, A_2^v)$ associated with one seed as in the above section. Depending on the underlying group structure, it is possible to store A_1. For instance, with elliptic curves over group of order of 160 bits, a group element can be represented with 161 bits. Therefore, a coupon with a hash output of 50-bit long requires 211 bits of memory. It is noticeable that in this case no elliptic curve computation is needed on the CLD, but only two additions and one multiplication over the integer and one update of the seed.

If the underlying group is $\mathbb{F}_p$ with p a 1536-bit long prime number, it does not seem realistic to store many A_1. In this case, the CLDs needs to be able to compute A_1. Nevertheless, this trick enables to decrease the computation needed. For instance with $B = 2^{32}$, $S = 2^{160}$ and $A = S \times B \times 2^{80} = 2^{272}$, the computations needed are just an exponentiation of 272 bits instead of an exponentiation of $4 \log_2(A) = 1048$ bits.

8.2 Adaptation of the Randomized Hashed GPS

A Version Enabling Mutual Authentication In this paper, we consider that CLDs are aware of the public key g^v. In an application, this public key can be previously sent accompanied by a certificate coming from an authority. In order to ensure a mutual authentication in these schemes, it is possible to slightly modify the Randomized Hashed GPS as in [22]. Nevertheless we need to add one move to our protocol. We assume that the CLD can use two different hash functions H_1 and H_2. In this case, the CLD can compute $H_1(A_1, A_2, A_1^v, A_2^v)$ and $H_2(A_1, A_2, A_1^v, A_2^v)$. In the first move it sends $H_1(A_1, A_2, A_1^v, A_2^v)$, in the second it receives a challenge c in the third it sends A_1 and y. In the new forth move, it receives a value z' and check whether $H_2(A_1, A_2, A_1^v, A_2^v) = z'$. This authentication of the reader does not imply an important online computation.

A private Scheme Enabling Fast Identification In a context where no covert channel exists, the prover's identity is unknown to the verifier. One would prefer that the identification does not need as many computation as the size of the list of identities. It is possible to achieve such a scheme. Randomized hashed with encryption GPS uses the asymmetric encryption scheme DHIES described in [1]. Its security is based on the SDH assumption and on the security of a symmetric encryption scheme $(\mathcal{E}, \mathcal{D})$ which should be IND-CCA2 [5]. DHIES is also IND-CCA2. We chose DHIES because it can be very efficiently inserted in Randomized Hashed GPS scheme.

DHIES relies on the El Gamal scheme. The public/private key pair is of the form (g^v, v). To encrypt a message M:

1. a random exponent r is chosen,
2. $A = g^r$ and $A^v = (g^v)^r$ are computed,
3. $K = H(A^v)$ is computed
4. $(A, \mathcal{E}_K(M))$ is the ciphertext of M.

To decrypt:

1. A^v is computed thanks to A and v,
2. $K = H(A^v)$ is computed
3. $\mathcal{D}_K(\mathcal{E}_K(M))$ is computed.

This scheme is secure under the SDH assumption or under the (q, S) SESDH assumption if r is always chosen smaller than S. The proof is totally equivalent to the one in [1], because the two problems have indistinguishable instances if the DLSE(S) problem is hard.

Lemma 6. *Assume the hardness of the (q, S)SESDH problem, assume H is a random oracle, assume $\frac{BS}{A}$ is negligible, assume $(\mathcal{E}, \mathcal{D})$ is IND-CCA2 then Randomized hashed with Encryption GPS is public-identity forward private, sound against active adversaries, narrow-strong private and statistically ZK.*

Proof are in Appendix G. In the Figure 6, we describe the resulting scheme.

As for the Randomized Hashed GPS, no exponentiation, hash computation or encryption have to be made online. Furthermore, the size of the cipher text is small if the underlying group is $\mathbb{Z}_p$.

P		V
public key I secret key $s \in [0, S-1]$ $I = g^s$	parameters : g, g^v	secret key v
pick $r_1 \in [0, A-1]$		
pick $r_2 \in [0, A-1]$	$\xrightarrow{z=H\left(A_1=g^{r_1}, A_2=g^{r_2}, A_1^v, A_2^v\right)}$	
	$\xleftarrow{c}$	pick $c \in [0, B-1]$
Compute $K = H(A_1^v)$		
$y = r_1 + r_2 + sc$	$\xrightarrow{A_1, B=\mathcal{E}_K(r_2), y}$	Compute $K = H(A_1^v)$ Compute $r_2 = \mathcal{D}_K(B)$ Compute $I = (g^y A_1^{-1} g^{-r_2})^{1/c}$ Check whether $z = H(A_1, A_2, A_1^v, A_2^v)$ $0 \leq y \leq 2A - 2 + (B-1)(S-1)$ check whether I is in the list

Fig. 6. Randomized hashed with encryption GPS

8.3 Comparisons

Table 1 summarizes the computation needed for CLDs during the identification. Furthermore, it sums up the soundness and the privacy of the schemes. PA and AA for passive and active adversary. A,B and S are the parameters defined for all the schemes in this paper. l is the length of the output of the hash function when it is needed. In this table, the computation column represents the number of bits of exponentiation needed by the prover.

Scheme	**Assumption**	**Soundness**	**Privacy**	**Computations (in bits)**	**Size of coupon**
					+Computations
GPS	DLSE(S)	PA	Not private	$\log_2(A)$	
Hashed GPS	DLSE(S)	HI AA	HI weak	$\log_2(A)$	$\log_2(l)$
Randomized GPS	(S, q)SEDDH	PA	Narrow-Strong	$2\log_2(S) + \log_2(A)$	
Randomized hashed GPS	(q, S)SESDH	PI AA	Narrow-Strong PI forward	$4\log_2(A)$	$\log_2(l)$ $\log_2(A)$
Randomized hashed GPS with encryption	(q, S)SESDH	PI AA	Narrow-Strong PI forward	$4\log_2(A)$ Encryption of $\log_2(A)$	$\log_2(A)$ Encryption of $\log_2(A)$

Table 1. Comparisons of schemes

9 Conclusion

We introduce the notion of Public-identity privacy in order to analyze the privacy of ZK schemes. We propose three general modifications to transform general ZK schemes into private schemes, following the desired privacy.

We also suggest several efficient identification schemes that are zero-knowledge and public-identity private. To be efficient, these schemes use short size exponents. Their securities are thus based on the (full,short) SESDH problem, a new problem equivalent to a combination of the SDH and DLSE problems.

These schemes ensure different levels of security. They increase the privacy of the original GPS scheme, thanks to a small amount of computations. They can be used in real life applications to preserve the privacy of the owners' device.

The Randomized hashed GPS scheme seems well suited for identity documents. For instance, is seems possible to write the cryptographic identity on the document in order to transmit this public value by an optical covert channel. The scheme could be used in order to prove the authenticity of the document. As randomized hashed GPS is public-identity forward private, owner's identity is not revealed by the contactless devices.

Acknowledgments We are grateful to Jean-Sébastien Coron to help us with the SESDH problem.

References

1. Michel Abdalla, Mihir Bellare, and Phillip Rogaway. The oracle Diffie-Hellman assumptions and an analysis of DHIES. In David Naccache, editor, *Topics in Cryptology - CT-RSA 2001, The Cryptographer's Track at RSA Conference 2001, San Francisco, CA, USA, April 8-12, 2001, Proceedings*, pages 143–158. Springer, 2001.
2. Jean-Philippe Aumasson, Matthieu Finiasz, Willi Meier, and Serge Vaudenay. TCHo: A hardware-oriented trapdoor cipher. In Josef Pieprzyk, Hossein Ghodosi, and Ed Dawson, editors, *Information Security and Privacy, 12th Australasian Conference, ACISP 2007, Townsville, Australia, July 2-4, 2007, Proceedings*, pages 184–199. Springer, 2007.
3. Gildas Avoine, Etienne Dysli, and Philippe Oechslin. Reducing time complexity in RFID systems. In Bart Preneel and Stafford E. Tavares, editors, *Selected Areas in Cryptography, 12th International Workshop, SAC 2005, Kingston, ON, Canada, August 11-12, 2005, Revised Selected Papers*, pages 291–306. Springer, 2005.
4. Lejla Batina, Nele Mentens, Kazuo Sakiyama, Bart Preneel, and Ingrid Verbauwhede. Low-cost elliptic curve cryptography for wireless sensor networks. In Levente Buttyán, Virgil D. Gligor, and Dirk Westhoff, editors, *Security and Privacy in Ad-Hoc and Sensor Networks, Third European Workshop, ESAS 2006, Hamburg, Germany, September 20-21, 2006, Revised Selected Papers*, pages 6–17. Springer, 2006.

5. Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In Hugo Krawczyk, editor, *Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings*, pages 26–45. Springer, 1998.
6. Mihir Bellare and Adriana Palacio. GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In Moti Yung, editor, *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, pages 162–177. Springer, 2002.
7. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.
8. Julien Bringer and Hervé Chabanne. Trusted-HB: a low-cost version of HB+ secure against man-in-the-middle attacks. *CoRR*, abs/0802.0603, 2008.
9. Benoît Calmels, Sébastien Canard, Marc Girault, and Hervé Sibert. Low-cost cryptography for privacy in RFID systems. In Josep Domingo-Ferrer, Joachim Posegga, and Daniel Schreckling, editors, *Smart Card Research and Advanced Applications, 7th IFIP WG 8.8/11.2 International Conference, CARDIS 2006, Tarragona, Spain, April 19-21, 2006, Proceedings*, pages 237–251. Springer, 2006.
10. Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000.
11. Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. *J. Cryptology*, 1(2):77–94, 1988.
12. Martin Feldhofer, Sandra Dominikus, and Johannes Wolkerstorfer. Strong authentication for RFID systems using the AES algorithm. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, pages 357–370. Springer, 2004.
13. Martin Feldhofer and Christian Rechberger. A case against currently used hash functions in RFID protocols. In Robert Meersman, Zahir Tari, and Pilar Herrero, editors, *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops, Montpellier, France, October 29 - November 3, 2006. Proceedings, Part I*, pages 372–381. Springer, 2006.
14. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, pages 186–194. Springer, 1986.
15. Marc Girault. An identity-based identification scheme based on discrete logarithms modulo a composite number. In *EUROCRYPT*, pages 481–486, 1990.
16. Marc Girault and David Lefranc. Public key authentication with one (online) single addition. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, pages 413–427. Springer, 2004.
17. Marc Girault, Guillaume Poupard, and Jacques Stern. On the fly authentication and signature schemes based on groups of unknown order. *J. Cryptology*, 19(4):463–487, 2006.
18. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

19. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989.
20. Louis C. Guillou and Jean-Jacques Quisquater. A “paradoxical“ indentity-based signature scheme resulting from zero-knowledge. In Shafi Goldwasser, editor, *Advances in Cryptology - CRYPTO '88, 8th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1988, Proceedings*, pages 216–231. Springer, 1988.
21. International Standards ISO/IEC. *ISO 14443-3: Identification cards – Contactless Integrated Circuit(s) Cards – Proximity Cards. Part 3: Initialization and Anticollision.* ISO, 2001.
22. Markus Jakobsson and David Pointcheval. Mutual authentication for low-power mobile devices. In Paul F. Syverson, editor, *Financial Cryptography, 5th International Conference, FC 2001, Grand Cayman, British West Indies, February 19-22, 2002, Proceedings*, pages 178–195. Springer, 2001.
23. Markus Jakobsson, Kazue Sako, and Russell Impagliazzo. Designated verifier proofs and their applications. In *EUROCRYPT*, pages 143–154, 1996.
24. Ari Juels and Stephen A. Weis. Authenticating pervasive devices with human protocols. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, pages 293–308. Springer, 2005.
25. Ari Juels and Stephen A. Weis. Defining strong privacy for RFID. In *PERCOMW '07: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 342–347, Washington, DC, USA, 2007. IEEE Computer Society. http://saweis.net/pdfs/JuelsWeis-RFID-Privacy.pdf.
26. Takeshi Koshiba and Kaoru Kurosawa. Short exponent Diffie-Hellman problems. In Feng Bao, Robert H. Deng, and Jianying Zhou, editors, *Public Key Cryptography - PKC 2004, 7th International Workshop on Theory and Practice in Public Key Cryptography, Singapore, March 1-4, 2004*, pages 173–186. Springer, 2004.
27. Fabien Laguillaumie and Damien Vergnaud. Designated verifier signatures: Anonymity and efficient construction from any bilinear map. In Carlo Blundo and Stelvio Cimato, editors, *Security in Communication Networks, 4th International Conference, SCN 2004, Amalfi, Italy, September 8-10, 2004, Revised Selected Papers*, pages 105–119. Springer, 2004.
28. Tri Van Le, Mike Burmester, and Breno de Medeiros. Universally composable and forward-secure RFID authentication and authenticated key exchange. In Feng Bao and Steven Miller, editors, *Proceedings of the 2007 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2007, Singapore, March 20-22, 2007*, pages 242–252. ACM, 2007.
29. Helger Lipmaa, Guilin Wang, and Feng Bao. Designated verifier signature schemes: Attacks, new security notions and a new construction. In Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings*, pages 459–471. Springer, 2005.
30. Silvio Micali and Adi Shamir. An improvement of the Fiat-Shamir identification and signature scheme. In Shafi Goldwasser, editor, *Advances in Cryptology - CRYPTO '88, 8th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1988, Proceedings*, pages 244–247. Springer, 1988.
31. David Molnar and David Wagner. Privacy and security in library RFID: issues, practices, and architectures. In Vijayalakshmi Atluri, Birgit Pfitzmann, and

Patrick Drew McDaniel, editors, *Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004, Washingtion, DC, USA, October 25-29, 2004*, pages 210–219. ACM, 2004.

32. Jean Monnerat, Serge Vaudenay, and Martin Vuagnoux. About machine-readable travel documents. RFID Security, 2007.
33. Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita. RFID privacy issues and technical challenges. 48(9):66–71, 2005.
34. Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In Ernest F. Brickell, editor, *Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings*, pages 31–53. Springer, 1992.
35. H. Ong and Claus-Peter Schnorr. Fast signature generation with a fiat shamir-like scheme. In *EUROCRYPT*, pages 432–440, 1990.
36. David Pointcheval. A new identification scheme based on the perceptrons problem. In *EUROCRYPT*, pages 319–328, 1995.
37. Jean-Jacques Quisquater and Louis Guillou. The new Guillou-Quisquater Scheme. In *Proceedings of the RSA 2000 conference*, 2000.
38. Ronald L. Rivest. On the notion of pseudo-free groups. In Moni Naor, editor, *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, pages 505–521. Springer, 2004.
39. Matthew J. B. Robshaw. Searching for compact algorithms: cgen. In Phong Q. Nguyen, editor, *Progressin Cryptology - VIETCRYPT 2006, First International Conferenceon Cryptology in Vietnam, Hanoi, Vietnam, September 25-28, 2006, Revised Selected Papers*, pages 37–49. Springer, 2006.
40. Shahrokh Saeednia, Steve Kremer, and Olivier Markowitch. An efficient strong designated verifier signature scheme. In Jong In Lim and Dong Hoon Lee, editors, *Information Security and Cryptology - ICISC 2003, 6th International Conference, Seoul, Korea, November 27-28, 2003, Revised Papers*, pages 40–54. Springer, 2003.
41. Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, pages 239–252. Springer, 1989.
42. Adi Shamir. An efficient identification scheme based on permuted kernels (extended abstract). In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, pages 606–609. Springer, 1989.
43. Ron Steinfeld, Laurence Bull, Huaxiong Wang, and Josef Pieprzyk. Universal designated-verifier signatures. In Chi-Sung Laih, editor, *Advances in Cryptology - ASIACRYPT 2003, 9th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, November 30 - December 4, 2003, Proceedings*, pages 523–542. Springer, 2003.
44. Jacques Stern. An alternative to the fiat-shamir protocol. In *EUROCRYPT*, pages 173–180, 1989.
45. Jacques Stern. A new identification scheme based on syndrome decoding. In Douglas R. Stinson, editor, *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, pages 13–21. Springer, 1993.
46. Jacques Stern. Designing identification schemes with keys of short size. In Yvo Desmedt, editor, *Advances in Cryptology - CRYPTO '94, 14th Annual Inter-*

national Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings, pages 164–173. Springer, 1994.

47. Paul C. van Oorschot and Michael J. Wiener. On Diffie-Hellman key agreement with short exponents. In *EUROCRYPT*, pages 332–343, 1996.
48. Serge Vaudenay. On privacy models for RFID. In *ASIACRYPT*, pages 68–87, 2007.
49. Pascal Véron. Improved identification schemes based on error-correcting codes. 8(1):57–69, 1996.
50. Stephen A. Weis, Sanjay E. Sarma, Ronald L. Rivest, and Daniel W. Engels. Security and privacy aspects of low-cost radio frequency identification systems. In Dieter Hutter, Günter Müller, Werner Stephan, and Markus Ullmann, editors, *Security in Pervasive Computing, First International Conference, Boppard, Germany, March 12-14, 2003, Revised Papers*, pages 201–212. Springer, 2003.
51. Duncan S. Wong and Agnes Hui Chan. Efficient and mutually authenticated key exchange for low power computing devices. In Colin Boyd, editor, *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, pages 272–289. Springer, 2001.

A Formal Description of Zero-Knowledge Schemes

Definition 8. *A scheme is* **sound** *if there exists an extractor such that given $[A, c_1, B_1]$ and $[A, c_2, B_2]$, two valid transcripts of $\mathcal{P}$, it is possible to compute the secret s linked to I.*

If the equality $f(B_1).g_I(c_2) = f(B_2).g_I(c_1)$ holds, it should be possible to recover s. A usual way to achieve such a property is to compute $g_I(c)$ as I^c. This doing, it is possible to retrieve the discrete logarithm of I in some bases. If g_I is the exponentiation function, and if identities are generator elements of the group structure, f is a bijection.

Remark 4. Even if the discrete logarithm is not the underlying problem, it seems difficult to achieve soundness without a homomorphic function g_I.

Definition 9. *A scheme is honest-verifier* **zero-knowledge** *if given c, it is possible to compute A and B using only I such that Equation (1) holds. A scheme is malicious verifier zero-knowledge if the probability to simulate a prover thanks to its identity is non negligible whatever the challenge c.*

A trivial strategy exists for the honest-verifier zero-knowledge simulator: computing a random B and computing $A = \frac{f(B)}{g_I(c)}$ for a given c.

B Privacy Analysis of Two Privately Verifiable Schemes

A publicly verifiable scheme is a scheme where the verification can be made thanks to the public information of the verifier and the prover. In this case, if the identity are public, it is possible to track the different provers thanks to their identities. As one of our aim is to create a public-identity private scheme, we have looked for privately verifiable scheme.

We did not find many privately verifiable ZK schemes in three moves which are the main topic of this paper. The best examples we found are the schemes by Saeednia *et al*'s in [40], adapted from a DVS scheme and the scheme of Jakobsson and Pointcheval in [22].

After a brief description of these schemes, we analyze their privacy in the model.

B.1 Privacy Analysis of the Saeednia *et al*'s Scheme [40]

We describe in Figure 7 the protocol in [40]. This scheme is based on the Schnorr signature scheme. The Discrete logarithm problem is hard in the underlying group. The order of the group is a large prime q.

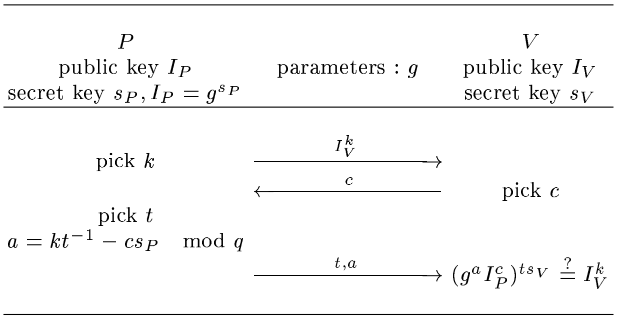

Fig. 7. Identification scheme from [40]

The protocol transcript is $\left[I_V^k, c, t, a\right]$. As a remark, the verification equality can be transformed into

$$(I_V^{s_P})^{c.t} = I_V^{k-at}.$$

Therefore, given the secret s_P, the transcript and I_V, it is possible to verify the previous equality. As a consequence, an adversary who has been able to find a secret of one prover, can retrieve the data emitted by this prover. Among the eavesdropped protocol transcript, the adversary retrieves which protocol has been made by this prover. Therefore, a narrow-forward adversary can determine with probability near of one if the transcript has been made by the CLD associated to s_P.

B.2 Privacy Analysis of the Jakobsson and Pointcheval Scheme [22]

This scheme is designed to ensure mutual authentication between a CLD and a verifier. Nevertheless, in the following, we focus on the authentication of the prover. The underlying group structure is chosen such that the discrete logarithm is a hard problem. We describe in Figure 8 the protocol. H is a cryptographic hash function.

P public key I_P secret key $s_P, I_P = g^{s_P}$	parameters : $\mathbb{G}, g, \mathrm{List}(ID)$	V public key I_V secret key s_V
pick t, p	$\xrightarrow{g^p, z=H(g^t, I_V, g^p, I_V^p)}$	
	$\xleftarrow{c}$	pick c
$y = t - cs_P \mod q$		
	$\xrightarrow{y}$	Check whether there exists $I_P \in L$ $z = H(g^y I_P^c, I_V, g^p, (g^p)^{s_V})$

Fig. 8. Jakobsson-Pointcheval identification scheme

We assume that the identities are public. In this context, the public parameters are :

- $\mathbb{G}$ and g where g is a generator of the group $\mathbb{G}$ of order q,
- I_V, the identity of the verifier
- a list of identities I_{P_i} of provers.

To prove the privacy leakage, we prove it is not possible to simulate the Result oracle using the public information. This proves that this scheme is at most narrow strong private.

Given the public parameters, we compute a valid protocol transcript. As this transcript is valid, if the oracles are not simulated, the Result oracle outputs true with probability 1. Otherwise, if the system is simulated, the simulation of the RESULT oracle, applied on this protocol transcript,7 is impossible because it is necessary to possess s_V or p to verify the validity.

To simulate the protocol transcript outputs by a prover P, we use the usual simulation of the Schnorr scheme. We assume that a challenge c is given.

1. We randomly choose y and compute $T = g^y I_P^c$,
2. We randomly choose p and compute g^p, I_V^p and$z = H(T, I_V, g^p, I_V^p)$.

As for the Schnorr scheme, this simulation is perfect. The resulting transcript is $[g^p, z, c, y]$. Nevertheless, it is not possible for any blinder, who has only access to the public information, to distinguish whether this transcript is valid or not. It is one of the properties proved in [22].

Here, we described an adversary who creates a transcript such that a simulated RESULT accept with probability about 1/2 and the genuine RESULT accepts with probability 1. This proves that this system is distinguishable from a random system.

Remark 5. Here we assume that the verifier public key is known from the prover. Otherwise, if the verifier has to send his public key at the beginning of the session, the scheme cannot be considered private anymore. If an adversary can send his own public key to the CLD, and then executing the whole protocol, he can retrieve, at the end of the protocol, the identity of the prover. It is thus legitimate to consider that the blinder does not have access to the secret key s_V.

Moreover in the presented model, this scheme is not sound.

These schemes do not lead to the highest level of privacy. Our aim is to propose a general toolkit enabling to adapt general ZK scheme to obtain a scheme with the desired privacy level. Furthermore, we propose an efficient scheme which achieves this privacy.

C Simulation of the Result Oracle.

We suggest to modify the simulation of the RESULT oracle defined in [48] to allow the blinder to use a legitimate RESULT oracle and to alter its outputs if wanted. Indeed, if this oracle is fully simulated then once all the internal memory of a tag is corrupted, it is possible to distinguish the legitimate oracle from a simulated one. We relax this constraint to be able to achieve strong privacy whereas this was proved to be impossible [48] in the original model.

This doing we get a strong Private scheme in the modified model. Strong privacy can be achieved by encrypting the two messages sent by tags with a non-malleable encryption scheme [10].

Theorem 5. *Let E_{KV_p} be an NM-CPA scheme. The modification defined with the transcript $\left[E_{KV_p}(A), c, E_{KV_p}(B)\right]$ leads to a* **PI** *strong private and* **HI** *sound scheme.*

Proof. Firstly, it is narrow-strong private and forward private as is the **Modification 2**. Furthermore, the simulation of the oracle Result is easy. To simulate this oracle, a blinder has to store all the outputs of the tag it had simulated. Once he receives a protocol transcript, either it comes from a simulated tag, and the simulated oracle answer is *true* or it does not come from a simulated tag, and the output of the simulated Result oracle is the output of the genuine Result oracle.

This perfectly simulates the oracle as no modification of a transcript could arise to a valid transcript because an NM-CPA scheme is used. Therefore, if a transcript is valid, it directly comes from a tag (simulated or legitimate) or it was made with a corrupted secret. □

Remark 6. Contrary to all the other schemes presented, it is necessary to compute a ciphertext on the tag side. In the other schemes, the first message can be pre-computed in order to minimize the computational power needed online. In this case, the minimal information needed to compute B are kept in memory.

D Proofs of Security

In the proofs, we suppose $g_I(c) = I^c$ although they can be generalized with any bijective exponentiation in base I.

D.1 Correctness

The modifications **1** and **3** remain correct in the random oracle model. The only possibility for the identification to fail is that a collision happens in the outputs of the random oracle. The probability of such an event is negligible. The **Modification 2** stays correct because the applied modifications are bijection. The probability is thus the same for the modified scheme as for the original one.

D.2 Narrow Privacy

Modification 1 To prove the privacy, it is necessary to prove that we can simulate the oracles defined in the model. In fact all of them can be clearly simulated except the SendVerifier oracle. This oracle represents the output of the devices. In the following, we construct a simulation and we prove that an adversary cannot distinguish between this simulation and the outputs of genuine devices.

Transcripts of protocol instances between a legitimate CLD and any verifier is of the form $[A' = H(A), c, B]$. If the CLD is a legitimate one, the equation $f(B) = A.I^c$ holds, otherwise if the CLD is simulated, A' and B are random values. To distinguish between a legitimate and a simulated CLD, an adversary $\mathcal{A}$ has to compute A thanks to A', B, c and I to verify the equality of $H(A)$ with A', because H is a one way function.

Given two values B and $c \neq 0$, it is easy to see that for all I there exists an A such that $f(B) = AI^c$ and for all A there exists an I such that $f(B) = AI^c$. This proves that for a couple (B, c) there are as many couples (I, A) as the group order. Given $H(A), B, c$, with $f(B) = AI^c$, an adversary has no information on the couple (I, A) if H is a one way function. Therefore if the group order is exponential, the probability of success of an adversary to compute (I, A) is negligible.

For the same reason, given different tuples $H(A_i), B_i, c_i$ with $f(B_i) = A_i I^{c_i}$, an adversary cannot compute one of the couple (I, A_i) in polynomial time. Therefore he cannot distinguish between a simulated CLD and a genuine one.

Modification 2 A legitimate CLD outputs $E_{KV_p}(A)$ and B. A simulated CLD outputs $E_{KV_p}(R_1)$ and R_2 where R_1 and R_2 are random. Even if s is known by any strong adversary, as E is semantically secure, distinguishing between $f(B)I^{-c}$ and $f(R_2)I^{-c}$ thanks to $E_{KV_p}(A)$ and $E_{KV_p}(R_1)$ is impossible. Therefore, the SendVerifier oracle can be easily and perfectly simulated. Moreover, the SendTag and Launch oracles are trivial to simulate. Therefore any adversary can be blinded.

Modification 3 As the adversary is strong, we assume he is aware of all the secrets. As a consequence, it is equivalent to consider that the c sent by the adversary is zero. Nevertheless the adversary cannot use his possibility to corrupt a CLD to have an advantage on the privacy.

In this context, an instance protocol between a legitimate CLD and a verifier is of the form $[A_1, H(A_1, A_2, A_1^v, A_2^v), B]$ with $f(B) = A_1 A_2$. If the CLD is simulated, the transcript is $[R_1, R_2, R_3]$ where R_1, R_2 and R_3 are random values.

Firstly, the adversary cannot distinguish the simulation if he does not make any call to the random oracle. From a given instance of the SDH problem, we simulate a CLD and we can run the adversary on it. Then we can simulate all the calls to the random oracle.

Assume $g^v, R_1, \mathcal{O}_v^{DDH}$ is a SDH instance. We randomly choose R_2 as long as the hashed length and R_3 as long as B and we output R_1, R_2, R_3. Then we run $\mathcal{A}$ on this instance. We assume that all the call to the random oracle are made of four group elements g_1, g_2, g_3, g_4 otherwise the oracle output a random value. If $g_1 \neq R_1$ the oracle output a random value. If $g_1 = R_1$, we check if $g_2 = \frac{f(R_3)}{R_1}$, and if $g_3 = g_1^v$ and $g_4 = g_2^v$ thanks to the oracle. If all these equalities hold, we output R_2.

If $\mathcal{A}$ succeeds, as the oracle does not reveal any information on its inputs, it means he has made the call $R_1, \frac{R_3}{R_1}, R_1^v, (\frac{R_3}{R_1})^v$. Therefore he has computed R_1^v and he had solved the SDH problem.

D.3 Zero-Knowledgeness

Assume we have a simulator for the original protocol. Therefore, there exists an algorithm $\mathcal{S}$ such that $\mathcal{S}(c, I)$ outputs A and B such that $f(B) = A.I^c$. $\mathcal{S}_1$ can be computed thanks to $\mathcal{S}$ just by modifying A to $H(A)$, $\mathcal{S}_2$ thanks to $\mathcal{S}$ by modifying A to $E_{KV_p}(A)$. The **Modification 3** requests the knowledge of $v = KV_s$ to be simulated. $\mathcal{S}_3$ uses $\mathcal{S}$ to compute A and B. It randomly chooses A_2 and computes $A_1 = \frac{A}{A_2}$. He computes $z = H(A_1, A_2, A_1^v, A_2^v)$ and outputs A_1 and z.

D.4 Soundness

Modification 1 As we have proved in proof of the hidden-identity narrow-weak privacy, an adversary has no advantage on the identity. Nevertheless, if he succeeds in the identification protocol, he has successfully computed A', B for a c such that $H(f(B)I^{-c}) = A'$. As H is a one way function, this means he has computed $f(B)I^{-c}$, and as a consequence he can compute I.

Modification 2 The proof is similar to the previous one, be cause the scheme is NM-CPA. As a consequence, if an adversary succeeds, he is able to compute the clear text which is $f(B)I^{-c}$.

Modification 3 In this proof, we assume it is always possible to compute f^v, the function which maps B to $(f(B))^v$ where v is the secret key of the legitimate verifier.

First we need to prove a lemma.

Lemma 7. *Computing I^v thanks to g^v, I, and any complete protocol instance from the CLD who has I as its identity is as hard as the SDH problem.*

Proof. A transcript is of the form $A_1, H(A_1, A_2, A_1^v, A_2^v), c, B$ as the device is genuine. Because the adversary possesses g^v and $I = g^s$, he can compute $A_2 = f(B)A_1^{-1}I^{-c}$.

Assume $\mathcal{A}$ computes g^{vs} successfully. We can separate two cases: either the adversary does not use the random oracle in his computation, or he does.

Case 1. As $\mathcal{A}$ does not use the random oracle, he succeeds with transcripts of the form A_1, c, B. Nevertheless, these transcripts can be computationally simulated. In this case, once $\mathcal{A}$ is used, he only has access to g^v and I. If he computes g^{vs}, it means he solves the CDH problem. As the hardness of the (q, S)SESDH problem is assumed, it is impossible to solve this later problem except with a negligible probability.

Case 2. $\mathcal{A}$ uses the random oracle. Given an (q, S)SESDH instance: $g^v, I = g^s$ and $\mathcal{O}_v^{DDH}$, we can make indistinguishable protocol transcripts. When we run $\mathcal{A}$ on (g^v, I), he interrogates the device associated to I. For each call to the device, we output first two random values A, R. Once the adversary sent the challenge c, we output B. This is our simulation of the device.

In the following, we assume that each call made by $\mathcal{A}$ to the random oracle are made of four elements of $\mathbb{G}$, g_1, g_2, g_3, g_4. Thanks to $\mathcal{O}_v^{DDH}$, these transcripts are indistinguishable from genuine transcripts. For each call to the random oracle, we can check whether the g_1 is one of the simulated A. If there is no match, we output a random value. Otherwise we compute $A_2 = f(B).A^{-1}.I^{-c}$ and we check if $g_2 = A_2$ and thanks to $\mathcal{O}_v^{DDH}$ if $g_3 = g_1^v$ and $g_4 = A_2^v$. In this case we output the R resulting from the device simulation.

To succeed, an adversary has to make a good call to the random oracle. In this latter case $\mathcal{A}$ has solved the (q, S)SESDH problem. Thanks to A_2^v and A^v, we can compute $f(B)^v = A^v A_2^v I^{vc}$ by definition of A_2. Therefore he has computed I^v.

□

As a random oracle, H is assumed to be collision resistant. Therefore, to succeed to the identification, the adversary has to compute a transcript A, R, c, B such that $R = H(A, A_2 = f(B)A^{-1}I^{-c}, A^v, A_2^v)$. If he succeeds, he has computed A_2^v and A^v, therefore he can compute I^v. This contradicts the previous lemma.

D.5 Privacy

In the following, we prove that modifications **2** and **3** are not destructive private.

Modification 2 Suppose a destructive adversary eavesdropped two tags T_1 and T_2. He flipped a coin and destroyed the CLD designated by the coin and gets its secret s_b. He selects the transcript of the communication of one CLD he had already in memory, for instance T_1. This transcript is of the form $\left[E_{KV_p}(\mathbf{A}), c, B\right]$. Using s_b, c and B, it is possible to compute $A' = \frac{f(B)}{(Id(s_b))^c}$ which could be the good A with probability $\frac{1}{2}$. He has no advantage on this as the scheme is semantically secure. He modifies c to c' and computes B' such that $[A', c', B']$ is a correct transcript of the original scheme. This is possible because he is aware of the secret. He sends $\left[E_{KV_p}(\mathbf{A}), c', B'\right]$ to the Result oracle. If the answer is positive, then the used CLD is the one who has s_b as secret. Otherwise, it is the other one. Therefore an adversary has an advantage on s_b on this game.

It is not possible to simulate this Result query if the tags had been simulated. If s_b is the secret linked to the simulated CLD, the genuine Result oracle will output *false* on the modified transcript. Because there is no way to distinguish whether the secret is linked to the CLD or not, the simulator has no advantage on tags' secrets and cannot determine whether the modified transcript should be accepted. Therefore the blinded adversary has no advantage on s_b. This imply that there is a privacy leakage.

Modification 3 The attack is exactly the same as the one explained for the modification **2**. Thanks to s, B and c, an adversary gets $A_1.A_2$. He modifies c to c' and compute B'. He sent $[A_1, z, c', B']$ to the RESULT oracle. Once again this is not possible to simulate the answer of this query.

E The Short Exponent Problem in the Security Proofs

In this section, we prove an equivalence between the hardness of the DLSE(S) problem and the hardness of a discrete logarithm problem where some bits are known. This last problem is strongly related to all the security proof in the next appendices.

Lemma 8. *Assume $\frac{S}{A}$ is negligible, given g^{r_1} with r_1 randomly chosen in $[0, A-1]$, given the $\log_2(A) - \log_2(S)$ most significant bits (m.s.b.) of r_1, computing the $\log_2(S)$ less significant bits (l.s.b.) of r_1 is equivalent to the DLSE(S) problem.*

Proof. We write $l = \log_2(S)$ and $r_1 = t2^l + r$. t represents the m.s.b while r is the unknown part.

Firstly if an algorithm can solve the DLSE(S), he can solve the discrete logarithm of $g^{r_1 - t2^l}$, and it is simple to find out r.

Secondly, if an algorithm solves the presented problem, given g^r with $r \leq S$, we can randomly choose $t \leq \frac{A}{S}$, then we compute $g^{t2^l + r}$ and we run the algorithm on these value, thus computing DLSE(S). □

As proved in [17], if $\frac{S}{A}$ is negligible, given $r_1 + r_2$, the $\log_2(S)$ l.s.b. of r_1 are computationally indistinguishable from a random $\log_2(S)$-bit long value. As a consequence, if the DLSE(S) is hard, this lemma ensures the hardness of computing the discrete logarithm of g^{r_1} while $r_1 + r_2$ is known with $r_1 < A$ and $r_2 < A$ and $\frac{S}{A}$ negligible. This problem arises in all the security proofs.

From this lemma, we can deduce the following corollaries:

Corollary 1. *Given $g^v, g^{r_1}, r_1 + r_2$ with $v < q$, $r_1 < A$ and $r_2 < A$ randomly chosen, assume $\frac{S}{A}$ is negligible then computing g^{vr_1} is equivalent to solve the (q, S)SECDH problem.*

Corollary 2. *Given $g^v, g^{r_1}, r_1 + r_2$ with $v < q$, $r_1 < A$ and $r_2 < A$ randomly chosen, given g^e equals to g^{vr_1} with probability $1/2$ and g^r with the same probability and a random r, assume $\frac{S}{A}$ is negligible then deciding whether g^e equals g^{vr_1} is equivalent to solve the (q, S)SEDDH problem.*

Corollary 3. *Given $g^v, g^{r_1}, r_1{+}r_2, \mathcal{O}_v^{DDH}$ with $v < q$, $r_1 < A$ and $r_2 < A$ randomly chosen, assume $\frac{S}{A}$ is negligible then computing g^{vr_1} is equivalent to solve the (q, S)SESDH problem.*

The proofs of the corollaries are straightforward. We prove the corollary 1. The others proofs are similar.

Proof. Assume there exists an adversary $\mathcal{A}$ able to compute r_1 given $g^v, g^{r_1}, r_1 + r_2$ with $v < q$, $r_1 < A$ and $r_2 < A$. We will use this adversary to solve a (q, S)SECDH problem instance. Given g^v, g^r with $r < S$, we randomly choose $t \leq \frac{A}{S}$ and $r_2 \leq A{-}1$ and we compute g^{r+tS} and $r_2{+}tS$. $r + tS$ has the same distribution as a random value in $[0, A - 1]$. $r_2 - r$ is computationally indistinguishable from a random value smaller than A because $r < S$, therefore $r_2 + tS = (r_2 - r) + (r + tS)$ is computationally indistinguishable from the sum of two values smaller than A.

$\mathcal{A}$ is thus able to compute $g^{v(r+tS)}$, and as we can compute g^{vtS}, we succeed to compute g^{vr}. □

F Security Proof of the Randomized Hashed GPS

Proof of Lemma 5.

F.1 Narrow Strong Privacy

Proof. To prove the privacy, it is necessary to prove that we can simulate the oracles defined in the model. In fact all of them can be clearly simulated except the SendVerifier oracle. This oracle represents the output of the devices. In the following, we construct a simulation and we prove that an adversary cannot distinguish between this simulation and the outputs of genuine devices.

As the adversary is strong, we assume he is aware of all the secrets. An instance protocol between a legitimate device and a verifier is of the form $g^{r_1}, H(g^{r_1}, g^{r_2}, g^{vr_1}, g^{vr_2}), r_1{+}r_2$. If the device is simulated, the transcript is $g^{r_1}, H(g^{r_1}, g^{r_2}, g^{vr_1}, g^{vr_2}), r_3$ where r_1, r_2 and $r_3 \leq A - 1 + S - 1$ are random values.

As H is a random oracle, the adversary has to compute $g^{r_1}, g^{r_2}, g^{vr_1}, g^{vr_2}$ from $g^v, g^{r_1}, r_1+r_2, H(g^{r_1}, g^{r_2}, g^{vr_1}, g^{vr_2})$ to distinguish simulated instances. Clearly, it is equivalent to compute g^{vr_1} from $g^v, g^{r_1}, r_1 + r_2, H(g^{vr_1})$. As $\frac{BS}{A}$ is negligible, we can use the lemma 8, therefore it is equivalent to consider $g^v, g^{r_1}, H(g^{vr_1})$ with $r_1 < S$.

Given an instance of the (q, S)SESDH problem $g^v, g^{r_1}, \mathcal{O}_v^{DDH}$, we randomly choose a value R and we run the adversary on the problem g^v, g^{r_1}, R. For each call g^b of the adversary to the random oracle, we check whether $g^b = g^{vr_1}$ thanks to $\mathcal{O}_v^{DDH}$. In this case, we output R as the value of the random oracle.

If the adversary succeeds, he has computed g^{vr_1} and therefore he has solved the (q, S)SESDH problem. □

F.2 Zero-Knowledge

Proof. Assume $\frac{BS}{A}$ is negligible. Therefore there exists a simulator for the original GPS protocol. There exists an algorithm $\mathcal{S}$ such that $\mathcal{S}(c, I)$ outputs g^r and y such that $g^y = g^r.I^c$. A simulator for the Randomized Hashed GPS scheme can be defined thanks to $\mathcal{S}$ by randomly choosing r_1 in $[0..A-1]$, and then by computing g^{r_1}, g^{vr_1} and $(g^r)^v$ and output $g^r, H(g^r, g^{r_1}, g^{vr}, g^{vr_1}), r_1 + y$. The value g^{vr} can be computed as we considered that the simulator has access to the private material of the verifier. □

F.3 Public-Identity Soundness

To succeed, an adversary has to make a protocol instance given a challenge. He can eavesdrop any communication but he cannot reuse communications he got from a device. He is neither allowed to create a transcript using the secret he gets from corrupting a device.

First we need to prove a lemma.

Lemma 9. *Assume the adversary runs a device associated to the identity I. Computing I^v thanks to g^v, I, and any protocol transcript is as hard as the (q, S)SESDH problem.*

Proof. A transcript is of the form $g^{r_1}, H(g^{r_1}, g^{r_2}, g^{vr_1}, g^{vr_2}), c, r_1 + r_2 + sc$ as the device is genuine. Because the adversary possesses g^v and $I = g^s$, he can compute $g^{r_2} = g^{r_1+r_2+sc}g^{-r_1}g^{-cs}$.

Assume $\mathcal{A}$ computes g^{vs} successfully. We can separate two cases: either the adversary does not use the random oracle in his computation, or he does.

Case 1. As $\mathcal{A}$ does not use the random oracle, he succeeds with transcripts of the form $g^{r_1}, c, r_1 + r_2 + sc$. Nevertheless, these transcripts can be computationally simulated because $\frac{BS}{A}$ is negligible. In this case, once $\mathcal{A}$ is used, he only has access to g^v and g^s. If he computes g^{vs}, it means he solves the (q, S)SECDH problem which is impossible except with a negligible probability.

Case 2. $\mathcal{A}$ uses the random oracle. Given an (q, S)SESDH problem instance: $g^v, I = g^s$ and $\mathcal{O}_v^{DDH}$, we can make indistinguishable protocol transcript. When we run $\mathcal{A}$ on (g^v, I), he interrogates the device associated to I. For each call to the device, we output first a random value R. Once the adversary sent the challenge c, we output g^{r_1}, y with $y \leq 2A - 2 + (B-1)(S-1)$ and $r_1 \leq A - 1$. This is our simulation of the device.

In the following, we assume that each call made by $\mathcal{A}$ to the random oracle are made of four elements of $\mathbb{G}$, g_1, g_2, g_3, g_4. Thanks to $\mathcal{O}_v^{DDH}$, these transcripts are indistinguishable from genuine transcripts. For each call to the random oracle, we can check whether the g_1 is one of the simulated g^{r_1}. If there is no match, we output a random value. Otherwise we compute $A_2 = g^y.g^{-r_1}.I^{-c}$ and we check thanks to $\mathcal{O}_v^{DDH}$ if $g_3 = g^{vr_1}$ and $g_4 = A_2^v$. In this case we output the R resulting from the device simulation. Nevertheless, in this later case $\mathcal{A}$ has solved the (q, S)SESDH problem. Thanks to A_2^v and g^{vr_1}, we can compute $g^{vy} = g^{vr_1} A_2^v I^{vc}$ by definition of A_2. Therefore he has computed I^v. □

In the following, we use the previous lemma to prove the public-identity soundness.

Proof. As a random oracle, H is assumed to be collision resistant. Therefore, to succeed in the identification, the adversary has to compute a transcript g_1, R, c, y such that $R = H(g_1, A_2 = g^y g_1^{-1} I^{-c}, g_1^v, A_2^v)$. If he succeeds, he has computed A_2^v and g_1^v, therefore he can compute g^{vs}. This contradicts the previous lemma. □

F.4 Public-Identity Forward Privacy

Proof. Following Vaudenay, a sound and narrow-forward private scheme is a forward private scheme. Therefore Randomized Hashed GPS is public-identity forward private. □

G Security Proof of the Randomized Hashed with Encryption GPS

Proof of the Lemma 6. First we need to prove a lemma for all the proof of this scheme.

Lemma 10. *Assume the hardness of the (q, S)SESDH problem, assume that $(\mathcal{E}, \mathcal{D})$ is an IND-CCA2 cipher then from an instance of the form $\left[g^{r_1}, \mathcal{E}_{H(g^{vr_1})}(r_2), H(g^{r_1}, g^{r_2}, g^{vr_1}, g^{vr_2}), r_1 + r_2\right]$, an adversary cannot compute g^{vr_1} except with a negligible probability.*

Proof. Assume such an adversary $\mathcal{A}$ exists. Given an instance of the (q, S)SESDH problem $g^v, g^r, \mathcal{O}_v^{DDH}$, we can perfectly simulate a tuple of the form $g^{r_1}, \mathcal{E}_{H(g^{vr_2})}(r_2), H(g^{r_1}, g^{r_2}, g^{vr_1}, g^{vr_2}), r_1 + r_2$.

Firstly, because $(\mathcal{E}, \mathcal{D})$ is an IND-CCA2 cipher, no information on r_2 are leaked if the random oracle is not called on g^{vr_1}. Therefore, it is equivalent to consider the tuple $g^{r_1}, H(g^{vr_1}), H(g^{r_1}, g^{r_2}, g^{vr_1}, g^{vr_2}), r_1+r_2$. Computing g^{vr_1} is equivalent to computing g^{vr_2}, consequently we can just focus on the tuple $g^{r_1}, H(g^{vr_1}), r_1 + r_2$.

Given an instance $g^v, g^r, \mathcal{O}_v^{DDH}$ of the (q, S)SESDH problem, we randomly choose R, $r_2 \leq A - 1$ and $t \leq \frac{A}{S}$ and we compute g^{tS+r_1} and $r_2 + tS$. We run $\mathcal{A}$ on $g^v, g^{r_1+tS}, R, tS + r_2$. As $\frac{S}{A}$ is negligible, $r_2 - r_1$ is computationally indistinguishable from a random element in $[0, A - 1]$.

In the case where the adversary makes no call to the random oracle, his success is equivalent to the (q, S)SECDH problem which is hard. In the case where the adversary makes call to the random oracle, we can check if he has computed g^{vr_1+vuS} thanks to $\mathcal{O}_v^{DDH}$. In this case, we can output R. The simulation is thus perfect.

If $\mathcal{A}$ succeeds with a good call to the random oracle, we can compute $g^{vr_1} = g^{vr_1+vuS}g^{-vuS}$. □

G.1 Narrow Strong Privacy

Proof. We prove that a simulation of the SendVerifier oracle can be built.

Assume the hardness of the (q, S)SESDH problem. This hardness implies that the DHIES scheme is IND-CCA2 if the key $K = H(g^{vr_1})$ is secure. This security, as stated in the previous lemma is ensured by the (q, S)SESDH problem. As the Randomized Hashed GPS is narrow strong private, it is possible to simulate output from devices using the simulation of the randomized GPS and the fact that any random value can simulate the text encrypted by the DHIES scheme. □

G.2 Zero-Knowledge

Proof. Assume $\frac{BS}{A}$ is negligible. Therefore there exists a simulator for the original GPS protocol. There exists an algorithm $\mathcal{S}$ such that $\mathcal{S}(c, I)$ outputs A and y such that $g^y = A.I^c$. A simulator for the Randomized Hashed with Encryption GPS scheme can be defined thanks to $\mathcal{S}$ by randomly choosing r_1 in $[0..A-1]$, and then by computing g^{r_1}, g^{vr_1}, A^v,$K = H(A^v)$ and $\mathcal{E}_K(r_1)$, the output is $A, H(A, g^{r_1}, A^v, g^{vr_1}), \mathcal{E}_K(r_1), r_1{+}y$. The value A^v can be computed as we considered that simulator has access to private material of verifier. □

G.3 Public-identity Soundness

Proof. In the following we assume the hardness of the (q, S)SESDH problem and we assume $\frac{BS}{A}$ is negligible. Therefore, the Randomized Hashed GPS is sound against active adversary. As demonstrated for this scheme, the value A_1^v cannot be computed under the (q, S)SESDH problem. This implies the security of the DHIES scheme. As a consequence, nothing can be learned on the value encrypted by DHIES. For this reason, an adversary against Randomized Hashed with Encryption GPS has no advantage compared to an adversary against Randomized Hashed GPS. This scheme is thus sound. □

G.4 Public-Identity Forward Privacy

Proof. Following the lemma of Vaudenay, a sound and narrow-forward private scheme is a forward private scheme. Therefore Randomized Hashed GPS is public-identity forward private. □

H Application of Modification 3 to GQ

RSA moduli The first following protocol is computed over RSA rings. Our modification is based on the difficulty of the SDH problem. It is necessary to describe RSA moduli N such that the SDH problem is hard on $\mathbb{Z}_N$. The integer N should be the product of two safe primes p and

q. If the SDH problem is hard on the multiplicative group of $\mathbb{Z}_p^*$ of order p' and on the multiplicative group of $\mathbb{Z}_q^*$ of order q' where $p' = \frac{p-1}{2}$ and $q' = \frac{q-1}{2}$, then it is hard over the multiplicative group of $\mathbb{Z}_{pq}$ of order $p'q'$. If we consider the multiplicative group in $(\mathbb{Z}_N)^*$ of order $p'q'$, the SDH problem is hard on it.

It has been proved in [38] that safe RSA rings are pseudo free groups. As a consequence, the computation of discrete logarithms is hard on these groups.

Modification of GQ Protocol We briefly describe in Figure 9 how it is possible to modify the GQ scheme. We need a generator g of the subgroup of $(\mathbb{Z}_N)^*$ of order $p'q'$ to implement it. The privacy is improved, from non private (or **HI** weak private with the **Modification 1**) to **PI** forward private.

P public key I, secret key s $s^e I = 1$	parameters : $KV_p = g^v, KA_p = (N = pq, g, e)$	V secret key $KV_s = v$ a public list $L = \{\ldots, (I, SN), \ldots\}$
pick r_1, r_2	$\xrightarrow{A_1 = g^{er_1}, z = H(A_1, A_2 = g^{er_2}, A_1^v, A_2^v)}$	
	$\xleftarrow{c}$	pick $c \in [0, e-1]$
	$\xrightarrow{B = g^{r_1 + r_2} s^c}$	Check, if it exists $I \in L$ such that $z = H(A_1, \frac{B^e}{A_1 I^{-c}}, A_1^v, \left(\frac{B^e}{A_1 I^{-c}}\right)^v)$

Fig. 9. Modification 3 of GQ