This is the Full Version of the Extended Abstract that appears
in the Proceedings of the Cryptographers Track at the RSA conference (CT-RSA '18)
(16 – 20 April 2018, San Francisco, CA, USA)

Reassessing Security of Randomizable Signatures

David Pointcheval[1,2] and Olivier Sanders[3]

[1] DIENS, École normale supérieure, CNRS, PSL Research University, Paris, France
[2] INRIA
[3] Orange Labs, Applied Crypto Group, Cesson-Sévigné, France

Abstract. The Camenisch-Lysyanskaya (CL) signature is a very popular tool in cryptography, especially among privacy-preserving constructions. Indeed, the latter benefit from their numerous features such as randomizability. Following the evolution of pairing-based cryptography, with the move from symmetric pairings to asymmetric pairings, Pointcheval and Sanders (PS) proposed at CT-RSA '16 an alternative scheme which improves performances while keeping the same properties. Unfortunately, CL and PS signatures raise concerns in the cryptographic community because they both rely on interactive assumptions that essentially state their EUF-CMA security. This lack of precise security assessment is obviously a barrier to a widespread use of these signatures and a reason for preferring other constructions, such as the ones relying on q-type assumptions.
In this paper, we study more thoroughly the security of these signatures and prove that it actually relies, for both constructions, on simple variants of the SDH assumption, assuming a slight modification of the original constructions. Our work thus shows that the CL and PS signature schemes offer similar security guarantees as those provided by several other constructions using bilinear groups, and so that one can benefit from their interesting features without jeopardizing security.

1 Introduction

Although introduced 40 years ago [DH76], digital signature is still a very active topic in cryptography (*e.g.* [Gro15,PS16,Gha16,LMPY16]). This is mostly due to the use of this primitive as a building block for more complex constructions that require advanced features.

For example, let us consider privacy-preserving mechanisms, such as group signatures [BMW03], direct anonymous attestations [BCC04] or e-cash systems [Cha82]. They usually require some entity to prove that some data (*e.g.* a coin, a key) is certified while remaining anonymous. Obviously, this entity cannot show the same certificate each time she needs to authenticate, otherwise one could easily trace her.

One solution could be to commit this certificate and then prove, in a zero-knowledge way, that the commitment opens to a valid signature on the data. In a bilinear setting, this can for example be done by using Groth-Sahai proofs [GS08]. Alternative solutions in the random oracle model (ROM) can be preferred if one favors efficiency. In either case, the complexity of the proof will increase with the number of elements to hide, hence the need to limit this number.

A very attractive feature for a signature scheme in such a context is called *randomizability*. It allows anyone to derive, from a valid signature σ, a new version σ' on the same message. To our knowledge, in a bilinear setting, the first construction achieving such a property was proposed by Camenisch and Lysyanskaya [CL04]. We call it the CL signatures in the following. Indeed, a CL signature σ can be randomized by selecting a random scalar t and raising each element of σ to this power t. The point is that initial σ and its new version are unlinkable under the DDH assumption [BCN+10], if one does not explicitly know the signed message, but just possibly as a committed value.

This explains the popularity of CL signatures among privacy-preserving constructions (*e.g.* [BCN+10,BFG+13,CPST15]): the users no longer have to commit the signature, but simply have to randomize σ before sending it.

Unfortunately, these signatures have an important drawback, their size is linear in the number of messages to be signed, or the length of the vector. This can be a problem for some applications, in particular, for anonymous credentials. However, this problem was recently solved by Pointcheval and Sanders [PS16] who proposed new signatures (called PS in the following), with the same features as CL ones, but with a short constant size (namely only 2 group elements, whatever the size of the vector to be signed).

Nevertheless, the nice features of CL and PS signatures come at a price: their security is proven under interactive assumptions, which raises concerns in the cryptographic community. In particular, this may be seen as a reason for preferring alternative signatures such as the ones of Libert *et al.* [LMPY16] (less efficient but proven under a standard assumption).

The problem of the validity of the computational assumptions underlying the security of a cryptographic scheme is not new. The use of an interactive assumption usually allows to design more efficient constructions but with an obviously questionable security analysis. Conversely, one can be more confident in the security of a scheme proven under a standard assumption but the latter usually entails lower performances. Between them, one can find different trade-offs with constructions proven under non-interactive, but still non-standard (*e.g.* q-type) assumptions.

Among the latter, a prominent example is the (non-randomizable) Boneh Boyen signature scheme [BB04,BB08,ASM06] (called BB in the following). One of its strengths is the fact that its security relies on an assumption (q-SDH) which seems independent of the scheme and which can be simply stated (and so evaluated). This partly explains the popularity of these signatures, at least compared to their randomizable counterparts.

1.1 Our Contribution

In this work, we aim at narrowing this gap by proving that both CL and PS signatures can be analyzed with non-interactive assumptions, and a slight variant can be proven EUF-CMA secure (*Existential Unforgeability against Adaptively Chosen Message Attacks*) under the latter. Such a result implicitly increases the confidence in the original schemes.

Starting from the PS signature, we first identify a simple variant of the q-SDH assumption [BB04,BB08] which underlies the security of this scheme. As for q-SDH, this new assumption is based on a sequence $(g, g^x, \ldots, g^{x^q})$ —actually one in each group to deal with the asymmetric setting of the PS signatures— along with very few additional elements, and requires to return an element $h^{\frac{1}{x+w}}$. However, because of the randomizability of PS signatures, h can be any element chosen by the adversary, which leads to a major problem: if one defines $h = g^w \cdot g^x$ (or equivalently $h = g^{(x+w)Q(x)}$), then one can trivially break this assumption. We therefore add another success condition which rules out such strategy by requiring that the discrete logarithm of h cannot be a (polynomial) multiple of $(x + w)$. We provide more details on this condition and on the way it is enforced in Section 3.

Unfortunately, as for BB signatures, this assumption can only be used to prove the security notion of EUF-wCMA (*Existential Unforgeability against weak Chosen Message Attacks*) where the adversary cannot adaptively choose the messages it submits to the signing oracle, but before the setup only.

Nevertheless, we can deal with this issue by requiring the signer to sign an additional message (recall that PS signatures can handle any number of messages) which can be either the hash of the original message or an additional random scalar of its choice. In practice, this simply means that when one asks for a signature on an r-vector message $\mathbf{m} = (m_1, \ldots, m_r)$, one actually receives a PS signature on $(m_1, \ldots, m_r, m')$ where m' may be equal to $H(m_1, \ldots, m_r)$ for some hash function H.

As we show in this paper, such a slight change is enough to avoid interactive assumptions. Moreover, due to the constant size of PS signatures, this does not impact efficiency, in particular in the case where m' is a hash value, since no additional value has to be sent.

Regarding this use of a hash function for computing m', one may think that it might cause further problems, especially when one needs to prove knowledge of signatures on committed messages. However, this does not bring any trouble, because the prover only has to prove knowledge of $r + 1$ scalars $(m_1, \ldots, m_r, m')$ and not that the last element m' is the hash of the previous ones, as we explain in this paper: the actual redundancy for this m' is not for the security property, but for an efficiency purpose only. This construction with a redundant m' thus leads to an EUF-CMA randomizable signature scheme, with provable security in the random oracle model.

Next, we apply the same methodology to the CL signature scheme, by identifying a different variant of the q-SDH assumption. We then show that this variant underlies the security of CL signatures assuming a modification similar to the one described for PS signatures. Due to the linear size of CL signatures, such a modification implies a slight increase of the complexity but this can be considered as a reasonable trade-off to avoid interactive assumptions.

Eventually, we prove that our variants of the q-SDH assumption hold in the generic bilinear group model. The simplicity of these new variants makes these proofs quite easy to follow, contrarily to the original generic proofs of the CL and PS signatures. An interesting outcome of these new security assessments of CL and PS signatures is the identification of strong links between these signatures and the ones of Boneh and Boyen [BB04,BB08]: all of them rely on the q-SDH assumption or some simple variants. Our results thus prove that the security of CL and PS signatures is not significantly weaker than the one of BB signatures. We argue that this result is particularly relevant in regard of the massive use (*e.g.* [BCN+10,BFG+13,LLY13,DLST14,CPST15]) of CL signatures, thanks to their randomizability property, and so, potentially, of their plug-in replacement proposed by Pointcheval and Sanders [PS16].

1.2 Related Work

The term "CL signature" can be confusing since it actually refers to various schemes. The first one was introduced in [CL03] and was proven under the strong RSA assumption [BP97,FO97] (*a.k.a.* flexible RSA problem [CS99]). It strongly differs from the one, introduced by the same authors two years later [CL04], that we consider in this paper. Indeed, the latter makes use of bilinear groups and achieves randomizability, contrarily to the former. In that paper [CL04], Camenisch and Lysyanskaya also describe an extension of the Boneh-Boyen signature scheme [BB04,BB08] handling several messages. This extension was later referred to as an SDH variant of CL signatures by some works (*e.g.* [Sch15]), which adds to the confusion.

We stress that, to our knowledge, all the variants (*e.g.* [Oka06,Sch15]) of CL signatures proven under different versions of the q-SDH assumption [BB04] are actually very different from the original bilinear scheme introduced in [CL04]. In particular, none of them achieves randomizability, which is one of the main features of the latter.

Gerbush *et al.* [GLOW12] managed to keep randomizability while relying on fixed-size assumptions but at the cost of using groups of composite order whose complexity is significantly higher [Fre10,Gui13] than the one of prime order groups.

Our results thus differ from previous ones since we prove the security of CL and PS signatures under variants of the q-SDH assumption without modifying their properties or impacting (significantly) their efficiency.

The technique to convert a EUF-wCMA signature scheme into an EUF-CMA-secure scheme is reminiscent of Krawczyk and Rabin's work [KR00] on chameleon hash functions. It can also be found in [BB08]. A similar approach was proposed in [AGHO11] to convert an EUF-RMA secure (*Existential Unforgeability against Random Message Attacks*) into a EUF-CMA secure signature scheme. Since EUF-RMA security can be generically stated into a non-interactive way (given q signatures on q random messages, it is hard to construct another one on a new message) our result might look natural. However, such a conversion comes at the cost of this additional generic assumption which is often very complex to evaluate (even in the generic group model). Moreover, this new assumption only underlies the EUF-RMA security and not the stronger EUF-wCMA, as in our case.

Conversely, in this work we start from the original q-SDH assumption and identify simple variants that underlie the security of CL and PS signatures. These variants are in particular different from the assumptions stating the EUF-RMA security of these schemes, and can easily be used to compare these schemes with each other but also with alternatives such as BB signatures. Furthermore, we get EUF-CMA security without modifying the basic construction, and so keeping the nice features.

1.3 Organization

We recall some definitions in Section 2 and present the non-interactive assumptions underlying CL and PS signatures in Section 3 (we postpone the proofs that they hold in the generic bilinear group model to Appendix A). In Section 4, we recall the PS signature scheme and then explain (and prove in Section 5) how to modify it to avoid interactive assumptions. We proceed similarly in Section 6 for the CL signatures.

2 Preliminaries

2.1 Bilinear Groups

Bilinear groups are a set of three cyclic groups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ of prime order p along with a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ that is

1. bilinear: for any $g \in \mathbb{G}_1, \widetilde{g} \in \mathbb{G}_2$, and $a, b \in \mathbb{Z}_p$, $e(g^a, \widetilde{g}^b) = e(g, \widetilde{g})^{ab}$;
2. non-degenerate: for any $g \in \mathbb{G}_1 \backslash \{1_{\mathbb{G}_1}\}$ and $\widetilde{g} \in \mathbb{G}_2 \backslash \{1_{\mathbb{G}_2}\}$, $e(g, \widetilde{g}) \neq 1_{\mathbb{G}_T}$;
3. efficient: for any $g \in \mathbb{G}_1$ and $\widetilde{g} \in \mathbb{G}_2$, $e(g, \widetilde{g})$ can be efficiently computed.

Galbraith, Paterson, and Smart [GPS08] defined three types of pairings: in type 1, $\mathbb{G}_1 = \mathbb{G}_2$; in type 2, $\mathbb{G}_1 \neq \mathbb{G}_2$ but there exists an efficient homomorphism $\phi : \mathbb{G}_2 \to \mathbb{G}_1$, while no efficient one exists in the other direction; in type 3, $\mathbb{G}_1 \neq \mathbb{G}_2$ and no efficiently computable homomorphism exists between $\mathbb{G}_1$ and $\mathbb{G}_2$, in either direction.

CL signatures, as most of the constructions in the early-age of pairing-based cryptography, use pairings of type 1. While an asymmetric variant of these signatures can easily be defined (*e.g.* [BCN⁺10]), Sanders and Pointcheval [PS16] recently pointed out that the latter does not take advantage of the entire capabilities of pairings of type 3. They therefore proposed a new signature scheme in this setting, with the same features as CL signatures, but with a constant complexity. Their construction can only be instantiated in type 3 bilinear groups, but this is not a significant drawback since the latter offer the best performances.

2.2 Digital Signature Schemes

Syntax. A digital signature scheme is defined by four algorithms:

- the parameter setup algorithm (`Setup`), on input a security parameter k, outputs the public parameters pp;
- the key generation algorithm (`Keygen`), on input the public parameters pp, outputs a pair of signing and verification keys $(\mathsf{sk}, \mathsf{pk})$ – we assume that sk contains pk, and that pk contains pp;
- the signing algorithm (`Sign`), on input the signing key sk and a message m, outputs a signature σ;
- the verification algorithm (`Verify`), on input the verification key pk, a message m, and its alleged signature σ, outputs 1 if σ is a valid signature on m under pk, and 0 otherwise.

Security Notion. The standard security notion for a signature scheme is *existential unforgeability under chosen message attacks* (EUF-CMA) [GMR88]: it means that it is hard, even given access to a signing oracle, to output a valid pair (m, σ) for a message m never asked to the signing oracle. It is defined using the following game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$:

- **Setup:** $\mathcal{C}$ runs the `Setup` and the `Keygen` algorithms to obtain $(pp, \mathsf{sk}, \mathsf{pk})$. The adversary is given the public key pk;
- **Queries:** $\mathcal{A}$ adaptively requests signatures on at most q messages $m_1, \ldots, m_q$. $\mathcal{C}$ answers each query by returning $\sigma_i \leftarrow \mathtt{Sign}(\mathsf{sk}, m_i)$;

- **Output:** $\mathcal{A}$ eventually outputs a message-signature pair (m^*, σ^*) and wins the game if $\texttt{Verify}(\mathsf{pk}, m^*, \sigma^*) = 1$ while $m^* \neq m_i\ \forall i \in [1, q]$.

A signature scheme is EUF-CMA secure if no probabilistic polynomial-time adversary $\mathcal{A}$ can win this game with non-negligible probability, for a polynomial number q of adaptive signing queries. A weaker security notion, named *existential unforgeability under weak chosen message attacks* (EUF-wCMA) [BB08], forces the adversary to provide the list of messages $m_1, \ldots, m_q$ to the challenger at the beginning of the game (before receiving the public key pk). Finally, we note that the stronger SUF-CMA (*strong unforgeability under chosen message attacks*, a.k.a. *non-malleability*) security notion is unachievable by a randomizable signature scheme. Indeed, it implies that no adversary, given a signature σ on m, can derive a new signature $\sigma^* \neq \sigma$ on the same message, which is exactly the opposite of randomizability.

3 Computational Assumptions

In this section, we first recall the LRSW assumption and the PS assumption, that underly the security of the Camenisch-Lysyanskaya signatures [CL04] and the Pointcheval-Sanders signatures [PS16], respectively. They are both interactive computational assumptions, and the latter was denoted "Assumption 1" in their paper, with a pairing of type 3, while the former does not explicitly require a pairing.

3.1 Interactive Assumptions

Definition 1 (LRSW Assumption). *Let $\mathbb{G}$ be a cyclic group of prime order p, with a generator g. For $(g, X = g^x, Y = g^y)$, with $x, y \xleftarrow{\$} \mathbb{Z}_p^*$, we define the oracle $\mathcal{O}(m)$ on input $m \in \mathbb{Z}_p$ that chooses a random $h \in \mathbb{G}^*$ and outputs the triple $T = (h, h^y, h^{x+mxy})$. Given (g, X, Y) and unlimited access to this oracle $\mathcal{O}$, no adversary can efficiently generate such a triple for a new scalar m^*, not asked to $\mathcal{O}$.*

The validity of a new tuple $T = (h, U, V)$ can be proven interactively in any group, or non-interactively checked with a pairing of type 1, $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, with $e(Y, h) = e(g, U)$ and $e(X, h \cdot U^m) = e(g, V)$.

Definition 2 (PS Assumption). *Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ a bilinear group of type 3, with g (resp. $\widetilde{g}$) a generator of $\mathbb{G}_1$ (resp. $\mathbb{G}_2$). For $(\widetilde{g}, \widetilde{X} = \widetilde{g}^x, \widetilde{Y} = \widetilde{g}^y)$, with $x, y \xleftarrow{\$} \mathbb{Z}_p^*$, we define the oracle $\mathcal{O}(m)$ on input $m \in \mathbb{Z}_p$ that chooses a random $h \in \mathbb{G}_1$ and outputs the pair $P = (h, h^{x+my})$. Given $(g, Y, \widetilde{g}, \widetilde{X}, \widetilde{Y})$ and unlimited access to this oracle $\mathcal{O}$, no adversary can efficiently generate such a pair, with $h \neq 1_{\mathbb{G}_1}$, for a new scalar m^*, not asked to $\mathcal{O}$.*

The validity of $P = (h, V)$ can be checked: $e(V, \widetilde{g}) = e(h, \widetilde{X} \cdot \widetilde{Y}^m)$.

3.2 Non-Interactive Assumptions

Since both above assumptions are interactive, this raises some concerns about the security of the randomizable signatures from [CL04] and [PS16], even if these assumptions are proven to hold in generic groups or generic bilinear groups. In particular, this may be considered as a good reason for preferring Boneh-Boyen signatures [BB08], whose security relies on the non-interactive q-SDH assumption [BB04]:

Definition 3 (q-SDH Assumption). *Let $(p, \mathbb{G}_1, \mathbb{G}_T, e)$ a bilinear group of type 1, with g a generator of $\mathbb{G}_1$. Given $(g, g^x, \ldots, g^{x^q})$, for $x \xleftarrow{\$} \mathbb{Z}_p^*$, no adversary can output a pair $(w, g^{\frac{1}{x+w}})$, with $w \in \mathbb{Z}_p^*$.*

In this work, we show that both the CL and PS signatures can be slightly modified to be proven EUF-CMA secure under the variants q-MSDH-1 and q-MSDH-2 (the "M" stands for "modified") of this assumption. They are both q-type assumptions, but non-interactive. The former will be required for the security of the PS signatures, while the latter will be required for the security of the CL signatures.

Definition 4 (q-MSDH-1 Assumption). *Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ a bilinear group of type 3, with g (resp. $\widetilde{g}$) a generator of $\mathbb{G}_1$ (resp. $\mathbb{G}_2$). Given $\{(g^{x^i}, \widetilde{g}^{x^i})\}_{i=0}^{q}$ along with $(g^a, \widetilde{g}^a, \widetilde{g}^{a\cdot x})$, for $a, x \overset{\$}{\leftarrow} \mathbb{Z}_p^*$, no adversary can output a tuple $(w, P, h^{\frac{1}{x+w}}, h^{\frac{a}{P(x)}})$ for some $h \in \mathbb{G}_1^*$, where P is a polynomial of degree at most q and w is a scalar such that $(X+w)$ and $P(X)$ are relatively prime.*

One can note that the validity of the output (w, P, U, V) can easily be verified since one can generate $\widetilde{g}^{P(x)}$ using the elements $(\widetilde{g}, \widetilde{g}^x, \ldots, \widetilde{g}^{x^q})$ and then checks whether $e(U, \widetilde{g}^{a\cdot x} \cdot \widetilde{g}^{a\cdot w}) = e(V, \widetilde{g}^{P(x)})$.

The first goal of this paper is to show that this assumption underlies the EUF-wCMA security of the PS signature scheme (see Section 4). It is therefore natural that, as the signature scheme itself, it only holds with pairings of type 3. The asymmetric nature of the latter implies that we need to provide both the sequences $(g, g^x, \ldots, g^{x^q})$ and $(\widetilde{g}, \widetilde{g}^x, \ldots, \widetilde{g}^{x^q})$ since we cannot use some isomorphism to compute the latter from the former. But we stress that this does not give more power to an adversary than the one it has with $(g, g^x, \ldots, g^{x^q})$ in a type 1 setting.

Therefore, this assumption mostly differs from the q-SDH one in two ways. First, the challenge also contains a tuple $(g^a, \widetilde{g}^a, \widetilde{g}^{a\cdot x})$. Second, we allow the adversary to return an element $h^{\frac{1}{x+w}}$ for any $h \in \mathbb{G}_1$. Nevertheless, to avoid trivial solutions, the adversary must additionally return $h^{\frac{a}{P(x)}}$ for some polynomial P that is not divisible by $X+w$. Intuitively, this implies that the adversary cannot build h as $g^{Q(x)}$ with Q a multiple of $X+w$ which prevents it from returning $h^{\frac{1}{x+w}}$. This is formally stated by the following theorem, proven in Appendix A.1 for completeness:

Theorem 5. *The q-MSDH-1 assumption holds in the generic bilinear group model for pairings of type 3: after Q group and pairing oracle queries, no adversary can solve the q-MSDH-1 problem with probability greater than $q(2q+5+Q)^2/p$.*

Definition 6 (q-MSDH-2 Assumption). *Let $(p, \mathbb{G}_1, \mathbb{G}_T, e)$ a bilinear group of type 1, with g a generator of $\mathbb{G}_1$. Given $\{(g^{x^i}, g^{b\cdot x^i})\}_{i=0}^{q+1}$ and $(g^a, g^{a\cdot b\cdot x})$, for $a, b, x \overset{\$}{\leftarrow} \mathbb{Z}_p^*$, no adversary can output a tuple $(w, P, h^{\frac{1}{x+w}}, h^{\frac{a}{x\cdot P(x)}})$ for some $h \in \mathbb{G}_1$, with P a polynomial of degree at most q and $w \neq 0$ a scalar such that $X+w$ and $P(X)$ are relatively prime.*

One can note the similarities between the q-MSDH-1 and the q-MSDH-2 assumptions which translate the similarities between the PS and the CL signatures. The main difference is the additional sequence $(g^b, g^{b\cdot x}, \ldots, g^{b\cdot x^{q+1}})$ which plays the same role as $(\widetilde{g}, \widetilde{g}^x, \ldots, \widetilde{g}^{x^q})$ in a q-MSDH-1 instance: it provides a way to check the validity of the output (see the remark below) through a pairing computation, while being useless to the adversary because of the "b" factor in the exponent.

We show, in Section 6, that this assumption underlies the EUF-wCMA security of the CL signature scheme. This is the second contribution of this paper. A proof that this assumption holds in the generic bilinear group model is provided in Appendix A.2 for completeness:

Theorem 7. *The q-MSDH-2 assumption holds in the generic bilinear group model for pairings of type 1: after Q group and pairing oracle queries, no adversary can solve the q-MSDH-2 problem with probability greater than $(q+1)(2q+6+Q)^2/p$.*

Remark 8. Proving the validity of a new q-MSDH-2 tuple requires an interaction since the elements provided in an instance are not enough to perform the verification non-interactively. Nevertheless, we can easily avoid this problem by forcing the adversary to also return $h^{\frac{1}{x(x+w)}}$. Indeed, in this case, the validity of new tuple $(w, P, U, V, W) = (w, P, h^{\frac{1}{x+w}}, h^{\frac{1}{x(x+w)}}, h^{\frac{a}{x\cdot P(x)}})$ could be verified by checking whether:

1. $e(U, g) = e(V, g^x)$
2. $e(U \cdot V^w, g^{a \cdot b \cdot x}) = e(W, g^{b \cdot x \cdot P(x)})$

While this assumption is weaker than q-MSDH-2, it can still be used to prove security of CL signatures as explained in Section 6.3. However, we choose to keep the q-MSDH-2 assumption as it is to highlight the similarities with the q-MSDH-1 assumption.

Eventually, another assumption will appear in the security analysis, the SDL assumption [BCN+10] which extends the standard discrete logarithm (DL) assumption to the bilinear setting:

Definition 9 (SDL Assumption). *Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ a bilinear group, with g (resp. $\widetilde{g}$) a generator of $\mathbb{G}_1$ (resp. $\mathbb{G}_2$). Given (g, g^x) and $(\widetilde{g}, \widetilde{g}^x)$, for some $x \xleftarrow{\$} \mathbb{Z}_p$, no adversary can output the scalar x.*

One can note that for pairings of type 1, the SDL assumption is actually the classical Discrete Logarithm (DL) assumption. This is also clear that if one can break the SDL assumption, then given (g, g^x) and $(\widetilde{g}, \widetilde{g}^x)$ from a q-MSDH-1 instance, one can extract x and so solve the q-MSDH-1 problem with non-negligible probability. Hence, the q-MSDH-1 assumption implies the SDL assumption (with pairings of type 3). And this is clear that the q-MSDH-2 assumption implies the DL assumption, and thus the SDL assumption (with pairings of type 1). Hence, all our results will just require the q-MSDH-1 or q-MSDH-2 assumptions, since they imply the SDL assumption.

4 The Pointcheval-Sanders Signatures

We first recall the basic PS signature scheme on r-vector messages $(m_1, \ldots, m_r) \in \mathbb{Z}_p^r$, whose security proof relies on the interactive PS assumption (see Definition 2). The main feature is the constant size of the signature, independently of the value r. However, the keys depend on this value.

4.1 The Basic Pointcheval-Sanders Signature Scheme

- $\mathtt{Setup}(1^k)$: Given a security parameter k, this algorithm outputs $pp \leftarrow (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$. These bilinear groups must be of type 3. In the following, we denote $\mathbb{G}_i^*$ the subset of generators $\mathbb{G}_i \backslash \{1_{\mathbb{G}_i}\}$, for $i = 1, 2$;
- $\mathtt{Keygen}(pp)$: This algorithm selects $\widetilde{g} \xleftarrow{\$} \mathbb{G}_2^*$ and $(x, y_1, \ldots, y_r) \xleftarrow{\$} (\mathbb{Z}_p^*)^{r+1}$, computes $(\widetilde{X}, \widetilde{Y}_1, \ldots, \widetilde{Y}_r) \leftarrow (\widetilde{g}^x, \widetilde{g}^{y_1}, \ldots, \widetilde{g}^{y_r})$, and sets $\mathsf{sk} \leftarrow (x, y_1, \ldots, y_r)$ and $\mathsf{pk} \leftarrow (\widetilde{g}, \widetilde{X}, \widetilde{Y}_1, \ldots, \widetilde{Y}_r)$.
- $\mathtt{Sign}(\mathsf{sk}, m_1, \ldots, m_r)$: This algorithm selects a random $h \xleftarrow{\$} \mathbb{G}_1^*$ and outputs $\sigma \leftarrow (h, h^{(x + \sum y_j \cdot m_j)})$.
- $\mathtt{Verify}(\mathsf{pk}, (m_1, \ldots, m_r), \sigma)$: This algorithm parses σ as (σ_1, σ_2) and checks whether $\sigma_1 \neq 1_{\mathbb{G}_1}$ and $e(\sigma_1, \widetilde{X} \cdot \prod \widetilde{Y}_j^{m_j}) = e(\sigma_2, \widetilde{g})$ are both satisfied, or not. In the positive case, it outputs 1, and 0 otherwise.

One can note that a signature $\sigma = (\sigma_1, \sigma_2)$ is randomizable, by raising both σ_1 and σ_2 to a same non-zero power. The invariant is the discrete logarithm of σ_2 in basis σ_1, hence, the unlinkability relies on the DDH assumption. The unforgeability (EUF-CMA) has been proven to hold [PS16], under the PS assumption that is interactive. However, the weaker security notion EUF-wCMA can be proven under the q-MSDH-1 assumption (see Section 5.1 for the proof):

Theorem 10. *The basic PS signature scheme achieves EUF-wCMA security under the q-MSDH-1 assumption, where q is a bound on the number of messages asked by the adversary to get signed.*

While weak chosen-message attacks might be enough in several contexts, where the honest messages to be signed are known in advance, this is always better to achieve the highest security level, that means EUF-CMA.

4.2 The Modified Pointcheval-Sanders Signature Scheme

In order to achieve the EUF-CMA security level, we extend the vector with one more component, that gives a degree a freedom, and allows to use the same approach as with chameleon hash functions [KR00] (see also [BB08]). The signature scheme is defined as follows:

- $\texttt{Setup}(1^k)$: Given a security parameter k, this algorithm outputs $pp \leftarrow (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$. These bilinear groups must be of type 3. In the following, we denote $\mathbb{G}_i^*$ the subset of generators $\mathbb{G}_i \backslash \{1_{\mathbb{G}_i}\}$, for $i = 1, 2$;
- $\texttt{Keygen}(pp)$: This algorithm selects $\widetilde{g} \overset{\$}{\leftarrow} \mathbb{G}_2^*$ and $(x, y_1, \ldots, y_{r+1}) \overset{\$}{\leftarrow} (\mathbb{Z}_p^*)^{r+2}$, computes $(\widetilde{X}, \widetilde{Y}_1, \ldots, \widetilde{Y}_{r+1}) \leftarrow (\widetilde{g}^x, \widetilde{g}^{y_1}, \ldots, \widetilde{g}^{y_{r+1}})$, and sets the keys $\mathsf{sk} \leftarrow (x, y_1, \ldots, y_{r+1})$ and $\mathsf{pk} \leftarrow (\widetilde{g}, \widetilde{X}, \widetilde{Y}_1, \ldots, \widetilde{Y}_{r+1})$;
- $\texttt{Sign}(\mathsf{sk}, \mathbf{m} = (m_1, \ldots, m_r))$: This algorithm selects random $h \overset{\$}{\leftarrow} \mathbb{G}_1^*$ and $m' \overset{\$}{\leftarrow} \mathbb{Z}_p$ and outputs $\sigma \leftarrow (m', h, h^{(x + \sum_{j=1}^r y_j \cdot m_j + y_{r+1} \cdot m')})$.
- $\texttt{Verify}(\mathsf{pk}, \mathbf{m} = (m_1, \ldots, m_r), \sigma)$: This algorithm parses σ as (m', σ_1, σ_2) and checks whether $\sigma_1 \neq 1_{\mathbb{G}_1}$ and $e(\sigma_1, \widetilde{X} \cdot \prod_{j=1}^r \widetilde{Y}_j^{m_j} \cdot \widetilde{Y}_{r+1}^{m'}) = e(\sigma_2, \widetilde{g})$ are both satisfied or not. In the positive case, it outputs 1, and 0 otherwise.

Actually, this is exactly the previous signature scheme in dimension $r + 1$ instead of r, and the last component of the vector is randomly chosen, and appended to the signature. The security of this construction is formally stated by the following theorem, proven in Section 5.2:

Theorem 11. *The modified PS signature scheme achieves EUF-CMA security under the q-MSDH-1 assumption, where q is a bound on the number of adaptive signing queries.*

Whereas the previous construction was just wEUF-CMA, it was fully-randomizable, since the only random element in the signature was the generator h. This new construction is EUF-CMA, but just weakly randomizable, for the generator h only and not m'. By generating m' in a deterministic way, one gets both a shorter and a randomizable signature.

4.3 Avoiding the Additional Element

Whereas the computational assumption becomes a non-interactive one, the new construction slightly increases the size of the signature, since it must contain the additional element m'. Moreover, it also cancels the full randomizability of the signature since m' cannot itself be randomized. But the purpose of m', in the security proof, is to provide a degree of freedom. This is possible to define it in a deterministic way from the r-vector message $\mathbf{m}$, as $m' \leftarrow H(\mathbf{m})$, where H is a hash function onto $\mathbb{Z}_p$. Then, the security proof still holds, in the random oracle model [BR93] for H. However, the simulator guesses one of the hashing queries $\mathbf{m}^*$ to be the vector message in the output forgery. For this one, $H(\mathbf{m}^*)$ is programmed at random. If the guess is correct this leads to an attack to either the q-MSDH-1 problem or the SDL problem. Since this guess succeeds with probability $1/q_H$ (where q_H is the number of hashing queries), this reduction has a success probability divided by q_H, compared to the above reduction. In addition, in the q-MSDH-1 assumption, q is now the number q_H of hashing queries, and not just the number of signing queries:

Corollary 12. *The modified PS signature scheme, with $m' \leftarrow H(\mathbf{m})$, achieves EUF-CMA security under the q-MSDH-1 assumption in the random oracle model, where q is a bound on the number of hashing queries.*

Of course, the proof then makes use of the random oracle model (ROM), but we should recall that PS signatures (as CL ones) are mostly used in combination with NIZK proofs (*e.g.* [BCN$^+$10], [CPST15], etc) that already make use of the ROM. As a consequence, proving the security of the signature itself in the ROM does not impact much the security of the global construction.

In addition, it is worthy to note that efficient proofs of knowledge of a signature remain possible, despite the use of the hash function for $m' \leftarrow H(\mathbf{m})$. Indeed, to prove knowledge of a signature on a r-vector message $\mathbf{m}$, one simply has to run the protocol from [PS16, Section 6.2] for a signature on $(\mathbf{m}, H(\mathbf{m})) \in \mathbb{Z}_p^{r+1}$. In particular, it is not necessary to prove that the tuple is well-formed, and namely that last component m' is indeed $H(\mathbf{m})$, since it could have been any random scalar for the security of the scheme.

A subtlety arises in the verification process, when one uses the hash function H. Indeed, the `Verify` algorithm can take either $(\mathbf{m}, \sigma_1, \sigma_2)$ or $(\mathbf{m}, m' = H(\mathbf{m}), \sigma_1, \sigma_2)$ as input. In any case, the EUF-CMA security proof does not expect any property from last component m', which is very interesting if one needs to prove knowledge of a signature.

Finally, we note that the case of blind signature is trickier when a hash function is involved. Indeed, blind extraction remains possible (one must simply run the protocol from [PS16, Section 6.1] on $(\mathbf{m}, m') = (\mathbf{m}, H(\mathbf{m}))$ without proving well-formedness of this pair) but then the `Verify` algorithm must check that m' is indeed $H(\mathbf{m})$. Therefore, if one needs to combine blind signatures and zero-knowledge proofs of knowledge of the latter, then one should avoid this technique and use instead the scheme of section 4.2.

5 New Security Proofs for PS Signatures

5.1 Proof of Theorem 10

We just provide the proof of Theorem 10 in the single-message case. The r-vector message case is let to the reader, since it is similar to the next proof of Theorem 11. We thus show that the single-message PS signature scheme is EUF-wCMA secure under the q-MSDH-1 assumption. Let $\mathcal{A}$ be an adversary, succeeding with probability ε within time t. $\mathcal{A}$ first sends a list of messages $(w_1, \ldots, w_q)$ to the challenger which generates a public key $\mathsf{pk} = (\widetilde{g}, \widetilde{X}, \widetilde{Y}_1)$. At the end of the game, $\mathcal{A}$ is expected to return a forgery (σ_1^*, σ_2^*) on $w \neq w_i\ \forall i \in [1, q]$.

Let $(\mathsf{g}, \mathsf{g}^x, \ldots, \mathsf{g}^{x^q}) \in \mathbb{G}_1^{q+1}$, $(\widetilde{\mathsf{g}}, \widetilde{\mathsf{g}}^x, \ldots, \widetilde{\mathsf{g}}^{x^q}) \in \mathbb{G}_2^{q+1}$ and $(\mathsf{g}^a, \widetilde{\mathsf{g}}^a, \widetilde{\mathsf{g}}^{a \cdot x}) \in \mathbb{G}_1 \times \mathbb{G}_2^2$ be a random q-MSDH-1 instance, for which an adversary should return a tuple $(w, P, h^{\frac{1}{x+w}}, h^{\frac{a}{P(x)}})$ for some $h \in \mathbb{G}_1$.

The challenger $\mathcal{C}$ generates $g \leftarrow \mathsf{g}^{\prod_{i=1}^q (x+w_i)}$ and $\widetilde{g} \leftarrow \widetilde{\mathsf{g}}^{\prod_{i=1}^q (x+w_i)}$, using the elements from the sequences $(\mathsf{g}, \mathsf{g}^x, \ldots, \mathsf{g}^{x^q}) \in \mathbb{G}_1^{q+1}$ and $(\widetilde{\mathsf{g}}, \widetilde{\mathsf{g}}^x, \ldots, \widetilde{\mathsf{g}}^{x^q}) \in \mathbb{G}_2^{q+1}$, since this is a polynomial of degree q. It then also sets $\widetilde{X} \leftarrow \widetilde{\mathsf{g}}^{a \cdot x}$ and $\widetilde{Y}_1 \leftarrow \widetilde{\mathsf{g}}^a$ to define the public key $(g, \widetilde{g}, \widetilde{X}, \widetilde{Y}_1)$. This implicitly sets $\mathsf{sk} \leftarrow (x' = \frac{a \cdot x}{\prod_{i=1}^q (x+w_i)}, y_1' = \frac{a}{\prod_{i=1}^q (x+w_i)})$.

To generate signatures for the q queried messages w_j, $\mathcal{C}$ chooses $t_j \xleftarrow{\$} \mathbb{Z}_p^*$ and outputs $(w_j, (\mathsf{g}^{\prod_{i \neq j}^q (x+w_i)})^{t_j}, (\mathsf{g}^a)^{t_j})$. The second element can be computed from the sequence $(\mathsf{g}, \mathsf{g}^x, \ldots, \mathsf{g}^{x^q}) \in \mathbb{G}_1^{q+1}$, since this is a polynomial of degree $q-1$.

One can note that for each pair $(\sigma_1 = (\mathsf{g}^{\prod_{i \neq j}^q (x+w_i)})^{t_j}, \sigma_2 = (\mathsf{g}^a)^{t_j})$, if one sets $h \leftarrow \sigma_1$, then $\sigma_2 = h^{x' + w_j y_1'}$, with h a random group element, since t_j is random and non-zero. Hence, (σ_1, σ_2) is a valid signature of the message w_j.

Eventually, $\mathcal{A}$ outputs a forgery $\sigma^* = (\sigma_1^*, \sigma_2^*)$ on a message $w \neq w_j$, for $j = 1, \ldots, q$. Since σ^* must be valid, we have $e(\sigma_1^*, \widetilde{X} \cdot \widetilde{Y}_1^w) = e(\sigma_2^*, \widetilde{g})$, and so $e(\sigma_1^*, \mathsf{g}^{a(x+w)}) = e(\sigma_2^*, \widetilde{\mathsf{g}}^{\prod_{i=1}^q (x+w_i)})$: (σ_1^*, σ_2^*) is of the form $(h^{\frac{1}{x+w}}, h^{\frac{a}{\prod_{i=1}^q (x+w_i)}})$, for some $h \in \mathbb{G}_1^*$. Let $P(X) = \prod_{i=1}^q (X + w_i)$. Since $w \neq w_i$ the polynomial $X + w$ and $P(X)$ are relatively prime. Therefore, $(w, P, \sigma_1^*, \sigma_2^*)$ is a valid answer to the q-MSDH-1 challenge. Since $\mathcal{C}$ never aborts, its probability of success is essentially the same as the one of $\mathcal{A}$. □

5.2 Proof of Theorem 11

Actually, the proof is very similar to the previous one: we can show that the EUF-wCMA security of the single-message PS signature scheme implies the EUF-CMA security of the r-vector message modified PS signature scheme, under the SDL assumption, which is already implied by the q-MSDH-1 assumption.

More specifically, two cases can appear between the signed messages $\mathbf{m}^{(j)} = (m_1^{(j)}, \ldots, m_r^{(j)})$ and $m'^{(j)}$ for $j = 1, \ldots, q$ and the output message $\mathbf{m}^* = (m_1^*, \ldots, m_r^*)$ and m'^*: either $\sum_{i=1}^r y_i m_i^* + y_{r+1} m'^* \neq \sum_{i=1}^r y_i m_i^{(j)} + y_{r+1} m'^{(j)} \bmod p$ for all $j = 1, \ldots, q$, or not.

In the former case (*i.e.*, with non-negligible probability, $\sum_{i=1}^r y_i m_i^* + y_{r+1} m'^* \neq \sum_{i=1}^r y_i m_i^{(j)} + y_{r+1} m'^{(j)} \bmod p$ for all j), the challenger $\mathcal{C}$ generates, from a q-MSDH-1 instance, as in the previous proof, a public key $\mathsf{pk} = (g, \widetilde{g}, \widetilde{X}, \widetilde{Y}_1)$ along with valid signatures $(\sigma^{(j)} = (\sigma_1^{(j)}, \sigma_2^{(j)}))$ for the single-message PS signature scheme on randomly chosen messages $(w_1, \ldots, w_q)$. Next, it extends the public key to $\mathsf{pk}' = (\widetilde{g}, \widetilde{X}, \widetilde{Y}_1, \widetilde{Y}_2, \ldots, \widetilde{Y}_{r+1})$ with $\widetilde{Y}_i \leftarrow \widetilde{Y}_1^{u_i}$, for $u_i \xleftarrow{\$} \mathbb{Z}_p^*$, for $i = 2, \ldots, r+1$. This implicitly defines $\mathsf{sk}' \leftarrow (x, y_1, y_2 = u_2 y_1, \ldots, y_{r+1} = u_{r+1} y_1)$.

For the j-th signing query $\mathbf{m}^{(j)} = (m_1^{(j)}, \ldots, m_r^{(j)})$, the challenger $\mathcal{C}$ sets $m'^{(j)} \leftarrow u_{r+1}^{-1}(w_j - \sum_{i=1}^r u_i m_i^{(j)}) \bmod p$ (with $u_1 = 1$). Since $y_1 w_j = \sum_{i=1}^r y_1 u_i m_i^{(j)} + y_1 u_{r+1} m'^{(j)} = \sum_{i=1}^r y_i m_i^{(j)} + y_{r+1} m'^{(j)} \bmod p$, the tuple $(m'^{(j)}, \sigma_1^{(j)}, \sigma_2^{(j)})$ is a valid signature of $\mathbf{m}^{(j)} = (m_1^{(j)}, \ldots, m_r^{(j)})$.

From a forgery $\sigma^* = (m'^*, \sigma_1^*, \sigma_2^*)$ on an r-vector message $\mathbf{m}^* = (m_1^*, \ldots, m_r^*)$ that is different from any $\mathbf{m}^{(j)}$, we additionally know that, if one sets $w^* \leftarrow \sum_{i=1}^r u_i m_i^* + u_{r+1} m'^* \bmod p$, $y_1 w^* \neq y_1 w_j \bmod p$, for $j = 1, \ldots, q$. Hence, (σ_1^*, σ_2^*) is a valid forgery for the new message m^* under the single-message PS signature scheme, which leads to an attack against the q-MSDH-1 assumption: for $P(X) = \prod_{i=1}^q (X + w_i)$, $(w^*, P, \sigma_1^*, \sigma_2^*)$ is a valid answer to the q-MSDH-1 challenge.

In the latter case (*i.e.*, with non-negligible probability $\sum_{i=1}^r y_i m_i^* + y_{r+1} m'^* = \sum_{i=1}^r y_i m_i^{(j)} + y_{r+1} m'^{(j)} \bmod p$ for some j): the challenger $\mathcal{C}$ generates, from an SDL instance $(g, Y = g^y, \widetilde{g}, \widetilde{Y} = \widetilde{g}^y)$, a public key: it chooses a random scalar x, and random scalars a_i, b_i, for $i = 1, \ldots, r+1$, to set $\widetilde{X} = \widetilde{g}^x$ and $\widetilde{Y}_i = \widetilde{g}^{a_i} \widetilde{Y}^{b_i}$, which implicitly sets $y_i = a_i + y b_i$. For the j-th signing query $\mathbf{m}^{(j)} = (m_1^{(j)}, \ldots, m_r^{(j)})$, the challenger $\mathcal{C}$ chooses $m'^{(j)} \xleftarrow{\$} \mathbb{Z}_p$, and sets $\sigma_1^{(j)} \leftarrow g^{t_j}$ and $\sigma_2^{(j)} \leftarrow (g^{x + \sum_{i=1}^r a_i m_i^{(j)} + a_{r+1} m'^{(j)}} \times Y^{\sum_{i=1}^r b_i m_i^{(j)} + b_{r+1} m'^{(j)}})^{t_j}$, which is equal to $(\sigma_1^{(j)})^{x + \sum_{i=1}^r y_i m_i^{(j)} + y_{r+1} m'^{(j)}}$. This is thus a valid signature of $\mathbf{m}^{(j)}$.

In case of forgery, $\sum_{i=1}^r y_i m_i^* + y_{r+1} m'^* = \sum_{i=1}^r y_i m_i^{(j)} + y_{r+1} m'^{(j)} \bmod p$ means $\widetilde{g}^{\sum_{i=1}^r a_i m_i^{(j)} + a_{r+1} m'^{(j)}} \times \widetilde{Y}^{\sum_{i=1}^r b_i m_i^{(j)} + b_{r+1} m'^{(j)}} = \widetilde{g}^{\sum_{i=1}^r a_i m_i^* + a_{r+1} m'^*} \times \widetilde{Y}^{\sum_{i=1}^r b_i m_i^* + b_{r+1} m'^*}$ and thus

$$\widetilde{g}^{\sum_{i=1}^r a_i (m_i^* - m_i^{(j)}) + a_{r+1}(m'^* - m'^{(j)})} = \widetilde{Y}^{\sum_{i=1}^r b_i (m_i^{(j)} - m_i^*) + b_{r+1}(m'^{(j)} - m'^*)}.$$

Since the b_i's are random (hidden by the a_i's in the y_i's), $\sum_{i=1}^r b_i (m_i^{(j)} - m_i^*) + b_{r+1}(m'^{(j)} - m'^*) = 0 \bmod p$ with probability $1/p$. Excepted in this unlikely case, one breaks the SDL problem.

As a consequence, none of the two cases can happen with non-negligible probability. □

Remark 13. Pointcheval and Sanders also introduced in [PS16] a variant of their signature scheme which allows to sign committed messages. Such a variant requires to add a tuple $(g, \{Y_i = g^{y_i}\}_i)$, for a generator $g \xleftarrow{\$} \mathbb{G}_1^*$ in the public key. In the previous security proofs, the element g was generated as $\mathsf{g}^{\prod_{i=1}^q (x + w_i)}$, while the secret value y_1' was implicitly set as $\frac{a}{\prod_{i=1}^q (x + w_i)}$. Therefore the element g^a provided in a q-MSDH-1 instance is exactly the element Y_1. The pair (g, Y_1) can then be extended to Y_i's the same way the $\widetilde{Y}_i$'s are generated from $(\widetilde{g}, \widetilde{Y}_1)$. As a consequence, our modified scheme still supports this essential feature, and the security still relies on the q-MSDH-1 assumption only (the second case in the above proof leads to an attack against the SDL problem).

6 The Camenisch-Lysyanskaya Signatures

6.1 The Basic Camenisch-Lysyanskaya Signature Scheme

The Camenisch-Lysyanskaya (CL) signature scheme was introduced in [CL04]. We here recall the version (denoted C in their paper) allowing to sign r-vector messages:

- $\mathtt{Setup}(1^k)$: Given a security parameter k, this algorithm outputs $pp \leftarrow (p, \mathbb{G}_1, \mathbb{G}_T, e)$, for a bilinear group of type 1. In the following, we denote $\mathbb{G}_1^* = \mathbb{G}_1 \backslash \{1_{\mathbb{G}_1}\}$;

- $\mathtt{Keygen}(pp)$: This algorithm selects $g \xleftarrow{\$} \mathbb{G}_1^*$ and $(x, y_1, \ldots, y_r) \xleftarrow{\$} (\mathbb{Z}_p^*)^{r+1}$, computes $(X, Y_1, \ldots, Y_r) \leftarrow (g^x, g^{y_1}, \ldots, g^{y_r})$, and sets $\mathsf{sk} \leftarrow (x, y_1, \ldots, y_r)$ and $\mathsf{pk} \leftarrow (g, X, Y_1, \ldots, Y_r)$;
- $\mathtt{Sign}(\mathsf{sk}, \mathbf{m} = (m_1, \ldots, m_r))$: This algorithm selects a random $\sigma_1 = h \xleftarrow{\$} \mathbb{G}_1^*$ and computes the following elements:

  $$\sigma_i \leftarrow \sigma_1^{y_i}, \text{ for } i = 2, \ldots, r; \qquad \tau_i \leftarrow \sigma_i^{y_1}, \text{ for } i = 1, \ldots, r; \qquad \mu \leftarrow \sigma_1^x \prod_{i=1}^{r} \sigma_i^{x \cdot y_1 \cdot m_i}.$$

  It then returns the signature $\sigma = (\{(\sigma_i, \tau_i)\}_{i=1}^r, \mu)$;
- $\mathtt{Verify}(\mathsf{pk}, \mathbf{m} = (m_1, \ldots, m_r), \sigma)$: This algorithm parses σ as $(\{(\sigma_i, \tau_i)\}_{i=1}^r, \mu)$ and checks if all the following equations hold:

  $$e(\sigma_1, Y_i) = e(\sigma_i, g), \text{ for } i = 2, \ldots, r \text{ and } e(\sigma_i, Y_1) = e(\tau_i, g), \text{ for } i = 1, \ldots, r$$
  $$e(\sigma_1 \prod_{i=1}^{r} \tau_i^{m_i}, X) = e(\mu, g).$$

  If this is the case, then it outputs 1. Else, it returns 0.

The main feature is the randomizability of the signature. The unforgeability (EUF-CMA) has been proven to hold [CL04], under the LRSW assumption that is interactive. However, the weaker security notion EUF-wCMA can be proven under the q-MSDH-2 assumption (the proof can be found in Section 6.3):

Theorem 14. *The basic CL signature scheme achieves EUF-wCMA security under the q-MSDH-2 assumption, where q is a bound on the number of messages asked by the adversary to get signed.*

For the EUF-CMA security level, one has to do a similar modification as for the PS signature scheme.

6.2 The Modified Camenisch-Lysyanskaya Signature Scheme

Here again, the idea is to sign the vector $\mathbf{m} = (m_1, \ldots, m_r)$ with an additional component m'. Contrarily to the case of PS signatures, this adds two elements of $\mathbb{G}_1$ to σ. This is due to the fact that the latter is linear in the number of messages to be signed. Signing m' thus increases further the size of the signature but this may be considered as a reasonable trade-off to avoid the interactive LRSW assumption, for achieving EUF-CMA.

- $\mathtt{Setup}(1^k)$: Given a security parameter k, this algorithm outputs $pp \leftarrow (p, \mathbb{G}_1, \mathbb{G}_T, e)$, for a bilinear group of type 1. In the following, we denote $\mathbb{G}_1^* = \mathbb{G}_1 \backslash \{1_{\mathbb{G}_1}\}$;
- $\mathtt{Keygen}(pp)$: This algorithm selects $g \xleftarrow{\$} \mathbb{G}_1$ and $(x, y_1, \ldots, y_{r+1}) \xleftarrow{\$} (\mathbb{Z}_p^*)^{r+2}$, computes $(X, Y_1, \ldots, Y_{r+1}) \leftarrow (g^x, g^{y_1}, \ldots, g^{y_{r+1}})$, and sets the keys $\mathsf{sk} \leftarrow (x, y_1, \ldots, y_{r+1})$ and $\mathsf{pk} \leftarrow (g, X, Y_1, \ldots, Y_{r+1})$;
- $\mathtt{Sign}(\mathsf{sk}, \mathbf{m} = (m_1, \ldots, m_r))$: This algorithm selects random $\sigma_1 = h \xleftarrow{\$} \mathbb{G}_1^*$ and $m' \xleftarrow{\$} \mathbb{Z}_p$, and computes the following elements:

  $$\sigma_i \leftarrow \sigma_1^{y_i}, \text{ for } i = 2, \ldots, r+1 \text{ and } \tau_i \leftarrow \sigma_i^{y_1}, \text{ for } i = 1, \ldots, r+1$$
  $$\mu \leftarrow \sigma_1^x \cdot \sigma_{r+1}^{x \cdot y_1 \cdot m'} \cdot \prod_{i=1}^{r} \sigma_i^{x \cdot y_1 \cdot m_i}.$$

  It then returns the signature $\sigma = (m', \{(\sigma_i, \tau_i)\}_{i=1}^{r+1}, \mu)$;
- $\mathtt{Verify}(\mathsf{pk}, \mathbf{m} = (m_1, \ldots, m_r), \sigma)$: This algorithm first parses the signature σ as $(m', \{(\sigma_i, \tau_i)\}_{i=1}^{r+1}, \mu)$ and checks if all the following equations hold:

  $$e(\sigma_1, Y_i) = e(\sigma_i, g), \text{ for } i = 2, \ldots, r+1$$
  $$e(\sigma_i, Y_1) = e(\tau_i, g), \text{ for } i = 1, \ldots, r+1$$
  $$e(\sigma_1 \cdot \tau_{r+1}^{m'} \cdot \prod_{i=1}^{r} \tau_i^{m_i}, X) = e(\mu, g).$$

  If this is the case, then it outputs 1. Else, it returns 0.

But as above for the PS signatures (see Section 4.3), one looses the full-randomizability, since m' cannot be re-randomized. One can again get both randomizability and shorter signatures at once by using a hash function H onto $\mathbb{Z}_p$: $m' \leftarrow H(\mathbf{m})$. In any case, the randomness provided by the additional message (either m' or the hash value) allows to prove the unforgeability of this modified CL signature scheme under a non-interactive assumption. This is formally stated by the following theorem (the proof can be found in Section 6.3):

Theorem 15. *The modified CL signature scheme achieves EUF-CMA security under the q-MSDH-2 assumption, where q is a bound on the number of adaptive signing queries.*

6.3 New Security Proofs for CL Signatures

For both security proofs of either EUF-wCMA of the basic CL signature scheme[1] or EUF-CMA of the modified CL signature scheme, the output forgery $\sigma^* = (m'^*, \{(\sigma_i^*, \tau_i^*)\}_{i=1}^{r+1}, \mu^*)$ on $\mathbf{m}^* = (m_1^*, \dots, m_r^*)$ can be of two types:

- Type 1 forgeries: for some signature $(m', \{(\sigma_i, \tau_i)\}_{i=1}^{r+1}, \mu)$ returned by the signing oracle, for an adversarially chosen message $\mathbf{m} = (m_1, \dots, m_r)$, we have
$$g^{m_1} \cdot Y_{r+1}^{m'} \cdot \prod_{i=2}^{r} Y_i^{m_i} = g^{m_1^*} \cdot Y_{r+1}^{m'^*} \cdot \prod_{i=2}^{r} Y_i^{m_i^*};$$
- Type 2 forgeries: for any signature $(m', \{(\sigma_i, \tau_i)\}_{i=1}^{r+1}, \mu)$ returned by the signing oracle, for an adversarially chosen message $\mathbf{m} = (m_1, \dots, m_r)$, we have
$$g^{m_1} \cdot Y_{r+1}^{m'} \cdot \prod_{i=2}^{r} Y_i^{m_i} \neq g^{m_1^*} \cdot Y_{r+1}^{m'^*} \cdot \prod_{i=2}^{r} Y_i^{m_i^*}.$$

We will show that, in both proofs, the two types of forgery lead to an attack against either the DL problem or the q-MSDH-2 problem.

Type 1 Forgeries. From a DL instance (g, g^x), the challenger $\mathcal{C}$ selects random pairs of non-zero scalars in $\mathbb{Z}_p^*$, (u, y_1) and (α_i, β_i) for $i = 2, \dots, r$ and defines the public key $\mathsf{pk} = (g, X \leftarrow g^u, Y_1 \leftarrow g^{y_1}, Y_2 \leftarrow (g^x)^{\alpha_2} \cdot g^{\beta_2}, \dots, Y_r \leftarrow (g^x)^{\alpha_r} \cdot g^{\beta_r})$, which implicitly defines the secret key $\mathsf{sk} = (x' = u, y_1, y_2 = \alpha_2 \cdot x + \beta_2, \dots, y_r = \alpha_r \cdot x + \beta_r)$.

Using u, y_1, and pk, the challenger $\mathcal{C}$ is able to generate signatures on any message submitted by $\mathcal{A}$. Indeed, for a message $\mathbf{m} = (m_1, \dots, m_r) \xleftarrow{\$} \mathbb{Z}_p^r$, it selects a random scalar $t \xleftarrow{\$} \mathbb{Z}_p^*$ and computes:

$$\sigma_1 \leftarrow g^t, \qquad \sigma_i \leftarrow Y_i^t, \text{ for } i = 2, \dots, r,$$
$$\tau_i \leftarrow \sigma_i^{y_1}, \text{ for } i = 1, \dots, r, \qquad \mu \leftarrow \sigma_1^u \prod_{i=1}^{r} \sigma_i^{u \cdot y_1 \cdot m_i}.$$

The resulting signature $(\{(\sigma_i, \tau_i)\}_{i=1}^r, \mu)$ is definitely valid and correctly distributed since t is random and non-zero.

Let $\sigma^* = (\{(\sigma_i^*, \tau_i^*)\}_{i=1}^r, \mu^*)$ be the type 1 forgery on $\mathbf{m}^* = (m_1^*, \dots, m_r^*)$ returned by $\mathcal{A}$. Since this is a type 1 forgery, there is one of the above simulated message-signature pairs $(\{(\sigma_i, \tau_i)\}_{i=1}^r, \mu)$ for $\mathbf{m} = (m_1, \dots, m_r)$ such that
$$g^{m_1} \prod_{i=2}^{r} Y_i^{m_i} = g^{m_1^*} \prod_{i=2}^{r} Y_i^{m_i^*},$$

[1] for a uniform notation with the modified CL signature scheme, we can just assume $m' = y_{r+1} = 0$ and $\sigma_{r+1} = \tau_{r+1} = Y_{r+1} = 1_{\mathbb{G}_1}$.

which means

$$m_1 + \sum_{i=2}^{r} (\alpha_i \cdot x + \beta_i)\, m_i = m_1^* + \sum_{i=2}^{r} (\alpha_i \cdot x + \beta_i)\, m_i^* \bmod p,$$
$$x \times \sum_{i=2}^{r} \alpha_i (m_i - m_i^*) = (m_1^* - m_1) + \sum_{i=2}^{r} \beta_i (m_i^* - m_i),$$

Since $\mathbf{m}^* \neq \mathbf{m}$ and the α_i's are perfectly hidden to the adversary in the Y_i's, with probability $1/p$, $A = \sum_{i=2}^{r} \alpha_i (m_i - m_i^*) = 0 \bmod p$. Unless this bad case happens, $\mathcal{C}$ can recover x and solve the DL problem. Such a problem is included in a q-MSDH-$\mathsf{2}$ instance, and from x, one can break this instance.

Type 2 Forgeries. Let $((\mathsf{g}, \mathsf{g}^b), (\mathsf{g}^x, \mathsf{g}^{b\cdot x}), \ldots, (\mathsf{g}^{x^{q+1}}, \mathsf{g}^{b\cdot x^{q+1}}))$ and $(\mathsf{g}^a, \mathsf{g}^{a\cdot b\cdot x})$ be a q-MSDH-$\mathsf{2}$ instance.

Let $(\mathbf{m}_j)_{j=1}^q = (m_1^{(j)}, \ldots, m_r^{(j)})_{j=1}^q$ be the set of message queries from $\mathcal{A}$. For $j = 1, \ldots, q$, we define $w_j = m_1^{(j)} + \sum_{i=2}^{r} y_j \cdot m_i^{(j)}$, where $y_2, \ldots, y_r$ are random scalars generated by the challenger $\mathcal{C}$. The latter then sets the public key $\mathsf{pk} = (g \leftarrow \mathsf{g}^{b\cdot x \prod_{j=1}^q (x+w_j)}, X \leftarrow \mathsf{g}^{a\cdot b\cdot x}, Y_1 \leftarrow \mathsf{g}^{b \prod_{j=1}^q (x+w_j)}, Y_2 \leftarrow g^{y_2}, \ldots, Y_r \leftarrow g^{y_r})$ which implicitly defines $\mathsf{sk} = (x' = \frac{a}{\prod_{j=1}^q (x+w_j)}, y_1 = \frac{1}{x}, y_2, \ldots, y_r)$.

To generate the j-th signature on $\mathbf{m}_j$, $\mathcal{C}$ generates a random scalar $t \xleftarrow{\$} \mathbb{Z}_p^*$ and returns $(\sigma_1, \tau_1) \leftarrow (\mathsf{g}^{t\cdot x \prod_{i\neq j}^q (x+w_i)}, \mathsf{g}^{t \prod_{i\neq j}^q (x+w_i)})$ and $(\sigma_i, \tau_i) \leftarrow (\sigma_1^{y_i}, \tau_1^{y_i})$, for $i = 2, \ldots, r$ along with $\mu = (\mathsf{g}^a)^t$. This is a valid signature on $\mathbf{m} = (m_1, \ldots, m_r)$[2] under pk since:

$$\begin{aligned}
e(\sigma_1, Y_i) &= e(\sigma_1, g^{y_i}) = e(\sigma_1^{y_i}, g) = e(\sigma_i, g), \text{ for } i = 2, \ldots, r;\\
e(\sigma_1, Y_1) &= e(\mathsf{g}^{t\cdot x \prod_{i\neq j}^q (x+w_j)}, Y_1) = e(\mathsf{g}^{t \prod_{i\neq j}^q (x+w_j)}, Y_1^x) = e(\tau_1, g);\\
e(\sigma_i, Y_1) &= e(\sigma_1^{y_i}, Y_1) = e(\sigma_1, Y_1)^{y_i} = e(\tau_1, g)^{y_i} = e(\tau_1^{y_i}, g) = e(\tau_i, g),\\
&\quad \text{for } i = 2, \ldots, r;\\
e(\sigma_1 \prod_{i=1}^{r} \tau_i^{m_i}, X) &= e(\sigma_1 \cdot \tau_1^{m_1} \cdot \tau_1^{\sum_{i=2}^r y_i m_i}, X) = e(\tau_1^x \cdot \tau_1^{w_j}, \mathsf{g}^{a\cdot b\cdot x})\\
&= e(\mathsf{g}^{t \prod_i^q (x+w_i)}, \mathsf{g}^{a\cdot b\cdot x}) = e(\mathsf{g}^{t\cdot a}, \mathsf{g}^{b\cdot x \cdot \prod_i^q (x+w_i)})\\
&= e(\mathsf{g}^{t\cdot a}, g) = e(\mu, g).
\end{aligned}$$

Moreover, the signature is correctly distributed since t is random and non-zero.

Let $\sigma^* = (\{(\sigma_i^*, \tau_i^*)\}_{i=1}^r, \mu^*)$ be the type 2 forgery on $\mathbf{m}^* = (m_1^*, \ldots, m_r^*)$ returned by $\mathcal{A}$. The validity implies that

$$\begin{aligned}
e(\sigma_1^*, Y_i) &= e(\sigma_i^*, g), \text{ for } i = 2, \ldots, r\\
e(\sigma_i^*, Y_1) &= e(\tau_i^*, g), \text{ for } i = 1, \ldots, r\\
e(\sigma_1^* \prod_{i=1}^{r} (\tau_i^*)^{m_i}, X) &= e(\mu^*, g).
\end{aligned}$$

Therefore, we have: $\sigma_i^* = (\sigma_1^*)^{y_i}$, for $i = 2, \ldots, r$ and $\sigma_i^* = (\tau_i^*)^x$, for $i = 1, \ldots, r$. So the last verification equation can be rewritten as:

$$e((\sigma_1^*)^{1+\frac{m_1^* + \sum_{i=2}^r y_i m_i^*}{x}}, \mathsf{g}^{a\cdot b\cdot x}) = e(\mu^*, \mathsf{g}^{b\cdot x \prod_{j=1}^q (x+w_j)}).$$

This means that $(\mu^*)^{\frac{x \prod_{j=1}^q (x+w_j)}{a}} = (\sigma_1^*)^{(x+m_1^*+\sum_{i=2}^r y_i \cdot m_i^*)}$. Let us set $w = m_1^* + \sum_{i=2}^r y_i \cdot m_i^*$, for $h = (\sigma_1^*)^{x+w}$, we then have $\sigma_1^* = h^{\frac{1}{x+w}}$ and $\mu^* = h^{\frac{a}{x \prod_{j=1}^q (x+w_j)}}$. Since this is a type 2 forgeries,

[2] We remove the superscript (j) in the following to simplify the notations.

$w \neq m_1 + \sum_{i=2}^{r} y_i \cdot m_i = w_j \bmod p$, for $j = 1, \ldots, q$. Therefore, the polynomials $X + w$ and $X \prod_{j=1}(X + w_j)$ are relatively prime, which means that $(w, P, \sigma_1^*, \mu^*)$ is a valid solution to the q-MSDH-2 instance.

Remark 16. One can note that the forgery returned by $\mathcal{A}$ contains τ_1 such that $\tau_1 = \sigma_1^{\frac{1}{x}}$. This means that the challenger is also able to return $h^{\frac{1}{x(x+w)}}$, which can be useful if one wants to rely on the assumption described in remark 8.

Proof of Theorem 15. This proof is quite similar to the previous one, except that the last component m' allows to replace fixed signing queries by adaptively-chosen messages in the type 2 forgery case. Indeed, $\mathcal{C}$ now generates random w_j to construct the public key and answers the j-th signing query $\mathbf{m} = (m_1, \ldots, m_r)$ by setting $m' \leftarrow w_j - m_1 - \sum_{i=2}^{r} y_i \cdot m_i$, which is also random-looking, if w_j is randomly chosen.

7 Conclusion

In this paper, we have provided a new security assessment of CL and PS signatures. Our results prove that the interactive assumptions of the original evaluations [CL04,PS16] can easily be avoided, assuming a slight modification of the constructions. They can indeed be replaced by some simple variants of the q-SDH assumption [BB04] that we analyze in the generic bilinear group model.

Our work thus shows that CL and PS schemes offer the same level of confidence than those already relying on q-type assumptions. In particular, this proves that the use of these signatures should no longer be considered as a trade-off between efficiency and security and so that it is possible even in a sensitive context (*e.g.* electronic payment).

Acknowledgments

This work was supported in part by the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013 Grant Agreement no. 339563 – CryptoCloud) and by the French ANR Project ANR-16-CE39-0014 PERSOCLOUD.

References

AGHO11. Masayuki Abe, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Optimal structure-preserving signatures in asymmetric bilinear groups. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 649–666. Springer, Heidelberg, August 2011.

ASM06. Man Ho Au, Willy Susilo, and Yi Mu. Constant-size dynamic k-TAA. In Roberto De Prisco and Moti Yung, editors, *SCN 06*, volume 4116 of *LNCS*, pages 111–125. Springer, Heidelberg, September 2006.

BB04. Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73. Springer, Heidelberg, May 2004.

BB08. Dan Boneh and Xavier Boyen. Short signatures without random oracles and the SDH assumption in bilinear groups. *Journal of Cryptology*, 21(2):149–177, April 2008.

BCC04. Ernest F. Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick McDaniel, editors, *ACM CCS 04*, pages 132–145. ACM Press, October 2004.

BCN$^+$10. Patrik Bichsel, Jan Camenisch, Gregory Neven, Nigel P. Smart, and Bogdan Warinschi. Get shorty via group signatures without encryption. In Juan A. Garay and Roberto De Prisco, editors, *SCN 10*, volume 6280 of *LNCS*, pages 381–398. Springer, Heidelberg, September 2010.

BFG$^+$13. David Bernhard, Georg Fuchsbauer, Essam Ghadafi, Nigel P. Smart, and Bogdan Warinschi. Anonymous attestation with user-controlled linkability. *Int. J. Inf. Sec.*, 12(3):219–249, 2013.

BMW03. Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 614–629. Springer, Heidelberg, May 2003.

BP97. Niko Bari and Birgit Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 480–494. Springer, Heidelberg, May 1997.

BR93. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73. ACM Press, November 1993.

Cha82. David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *CRYPTO'82*, pages 199–203. Plenum Press, New York, USA, 1982.

CL03. Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, editors, *SCN 02*, volume 2576 of *LNCS*, pages 268–289. Springer, Heidelberg, September 2003.

CL04. Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 56–72. Springer, Heidelberg, August 2004.

CPST15. Sébastien Canard, David Pointcheval, Olivier Sanders, and Jacques Traoré. Divisible E-cash made practical. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 77–100. Springer, Heidelberg, March / April 2015.

CS99. Ronald Cramer and Victor Shoup. Signature schemes based on the strong RSA assumption. In *ACM CCS 99*, pages 46–51. ACM Press, November 1999.

DH76. Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

DLST14. Nicolas Desmoulins, Roch Lescuyer, Olivier Sanders, and Jacques Traoré. Direct anonymous attestations with dependent basename opening. In Dimitris Gritzalis, Aggelos Kiayias, and Ioannis G. Askoxylakis, editors, *CANS 14*, volume 8813 of *LNCS*, pages 206–221. Springer, Heidelberg, October 2014.

FO97. Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 16–30. Springer, Heidelberg, August 1997.

Fre10. David Mandell Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 44–61. Springer, Heidelberg, May 2010.

Gha16. Essam Ghadafi. Short structure-preserving signatures. In Kazue Sako, editor, *CT-RSA 2016*, volume 9610 of *LNCS*, pages 305–321. Springer, Heidelberg, February / March 2016.

GLOW12. Michael Gerbush, Allison B. Lewko, Adam O'Neill, and Brent Waters. Dual form signatures: An approach for proving security from static assumptions. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 25–42. Springer, Heidelberg, December 2012.

GMR88. Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, April 1988.

GPS08. Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

Gro15. Jens Groth. Efficient fully structure-preserving signatures for large messages. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 239–259. Springer, Heidelberg, November / December 2015.

GS08. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Heidelberg, April 2008.

Gui13. Aurore Guillevic. Comparing the pairing efficiency over composite-order and prime-order elliptic curves. In Michael J. Jacobson Jr., Michael E. Locasto, Payman Mohassel, and Reihaneh Safavi-Naini, editors, *ACNS 13*, volume 7954 of *LNCS*, pages 357–372. Springer, Heidelberg, June 2013.

KR00. Hugo Krawczyk and Tal Rabin. Chameleon signatures. In *NDSS 2000*. The Internet Society, February 2000.

LLY13. Kwangsu Lee, Dong Hoon Lee, and Moti Yung. Aggregating CL-signatures revisited: Extended functionality and better efficiency. In Ahmad-Reza Sadeghi, editor, *FC 2013*, volume 7859 of *LNCS*, pages 171–188. Springer, Heidelberg, April 2013.

LMPY16. Benoît Libert, Fabrice Mouhartem, Thomas Peters, and Moti Yung. Practical"signatures with efficient protocols" from simple assumptions. In Xiaofeng Chen, XiaoFeng Wang, and Xinyi Huang, editors, *ASIACCS 16*, pages 511–522. ACM Press, May / June 2016.

Oka06. Tatsuaki Okamoto. Efficient blind and partially blind signatures without random oracles. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 80–99. Springer, Heidelberg, March 2006.

PS16. David Pointcheval and Olivier Sanders. Short randomizable signatures. In Kazue Sako, editor, *CT-RSA 2016*, volume 9610 of *LNCS*, pages 111–126. Springer, Heidelberg, February / March 2016.

Sch15. Sven Schäge. Tight security for signature schemes without random oracles. *Journal of Cryptology*, 28(3):641–670, July 2015.

A Hardness in the Generic Bilinear Group Model

A.1 Proof of Theorem 5

Let $(g, g^x, \ldots, g^{x^q}) \in \mathbb{G}_1^{q+1}$, $(\widetilde{g}, \widetilde{g}^x, \ldots, \widetilde{g}^{x^q}) \in \mathbb{G}_2^{q+1}$ and $(g^a, \widetilde{g}^a, \widetilde{g}^{a \cdot x}) \in \mathbb{G}_1 \times \mathbb{G}_2^2$ be a q-MSDH-1 instance. We recall that the goal of the adversary is to return a tuple $(w, P, h^{\frac{1}{x+w}}, h^{\frac{a}{P(x)}})$ for some $h \in \mathbb{G}_1^*$ where P and w are such that $P(X)$ and $X + w$ are relatively prime.

In the generic bilinear group model, the adversary $\mathcal{A}$ can obtain new elements only through queries to group operations (addition of the scalar exponents) or pairing (multiplication of the scalar exponents) oracles. Since we consider pairings of type 3, the elements $h^{\frac{1}{x+w}}$ and $h^{\frac{a}{P(x)}}$ returned by $\mathcal{A}$ can only be combinations of g^a and g^{x^i} for $i = 0, \ldots, q$. So, the q-MSDH-1 assumption holds if it is not possible to find scalars (u, u') and $\{(v_i, v_i'\}_{i=0}^q$, such that:

$$h^{\frac{1}{x+w}} = (g^a)^u \prod_{i=0}^{q} (g^{x^i})^{v_i} \qquad\qquad h^{\frac{a}{P(x)}} = (g^a)^{u'} \prod_{i=0}^{q} (g^{x^i})^{v_i'}.$$

Let r be defined by $h = g^r$. Then the previous equations are equivalent to:

$$g^r = [(g^a)^u \prod_{i=0}^{q} (g^{x^i})^{v_i}]^{x+w} \qquad\qquad g^{a \cdot r} = [(g^a)^{u'} \prod_{i=0}^{q} (g^{x^i})^{v_i'}]^{P(x)}.$$

In the following, we associate each group element of $\mathbb{G}_1$ with polynomials whose formal variables are the unknown scalars involved in the q-MSDH1 challenge, namely x and a. We first prove that no adversary is able to symbolically produce a valid tuple and then show that an accidental validity is very unlikely.

From the previous relations we get:

$$a(x+w)[a \cdot u + \sum_{i=0}^{q} v_i \cdot x^i] = P(x)[a \cdot u' + \sum_{i=0}^{q} v_i' \cdot x^i].$$

By considering each member of this equation as a polynomial in the variable a:

$$(i) \quad u = 0 \qquad\qquad (ii) \quad (x+w)(\sum_{i=0}^{q} v_i \cdot x^i) = u' \cdot P(x) \qquad\qquad (iii) \quad \sum_{i=0}^{q} v_i' \cdot x^i = 0.$$

The second equation implies that $(x + w)$ divides $P(x)$, which is impossible since these two polynomials are assumed to be relatively prime. Therefore, $u' = (\sum_{i=0}^{q} v_i \cdot x^i) = 0$, which means that $r = (x + w)[a \cdot u + \sum_{i=0}^{q} v_i \cdot x^i] = 0$ and so that $h = 1_{\mathbb{G}_1}$. It is therefore impossible to symbolically construct a tuple $(w, P, h^{\frac{1}{x+w}}, h^{\frac{a}{P(x)}})$

Now, let us evaluate the probability that two different polynomials involved in the answers of the different oracles evaluate to the same value. The $(2q + 5)$ elements provided in the q-MSDH-1 challenge are associated with polynomials of degree at most q. Therefore, the polynomials resulting from oracle queries are of degree at most $2q$ due to the pairing computation. After Q oracle queries there are then at most $(2q + 5 + Q)^2/2$ pairs of such polynomials that could evaluate to the same value. By the Schwartz-Zippel, such an event occurs with probability $q(2q + 5 + Q)^2/p$, which is negligible.

A.2 Proof of Theorem 7

Let $((g, g^b), (g^x, g^{b \cdot x}), \ldots, (g^{x^{q+1}}, g^{b \cdot x^{q+1}}))$ and $(g^a, g^{a \cdot b \cdot x})$ be a q-MSDH-2 instance. We recall that the goal of the adversary is to return a tuple $(w, P, h^{\frac{1}{x+w}}, h^{\frac{a}{x \cdot P(x)}})$ for some $h \in \mathbb{G}_1^*$ where P and w are such that $P(X)$ and $X + w$ are relatively prime.

This proof is very similar to the previous one since the q-MSDH-1 and q-MSDH-2 assumptions provides almost the same elements in $\mathbb{G}_1$. Again, we define r as the element such that $h = g^r$, which gives us the following polynomial relations in the variables x, a and b:

1. $r = (x + w)(u_{1,1} \cdot a + u_{1,2} \cdot a \cdot b \cdot x + \sum_{i=0}^{q} v_{1,i} \cdot x^i + \sum_{i=0}^{q} v'_{1,i} \cdot b \cdot x^i)$
2. $r \cdot a = x \cdot P(x)(u_{2,1} \cdot a + u_{2,2} \cdot a \cdot b \cdot x + \sum_{i=0}^{q} v_{2,i} \cdot x^i + \sum_{i=0}^{q} v'_{2,i} \cdot b \cdot x^i)$

for some scalars $\{u_{j,1}, u_{j,2}, v_{j,i}, v'_{j,i}\}_{j=1,i=0}^{j=2,i=q+1}$. These equations give us:

$$\begin{gathered} a(x + w)(u_{1,1} \cdot a + u_{1,2} \cdot a \cdot b \cdot x + \textstyle\sum_{i=0}^{q} v_{1,i} \cdot x^i + \sum_{i=0}^{q} v'_{1,i} \cdot b \cdot x^i) = \\ x \cdot P(x)(u_{2,1} \cdot a + u_{2,2} \cdot a \cdot b \cdot x + \textstyle\sum_{i=0}^{q} v_{2,i} \cdot x^i + \sum_{i=0}^{q} v'_{2,i} \cdot b \cdot x^i) \end{gathered}$$

If we consider each member as a polynomial in a, we can conclude that:

– $(x + w)(u_{1,1} + u_{1,2} \cdot b \cdot x) = 0$
– $(x + w)(\sum_{i=0}^{q} v_{1,i} \cdot x^i + \sum_{i=0}^{q} v'_{1,i} \cdot b \cdot x^i) = x \cdot P(x)(u_{2,1} + u_{2,2} \cdot b \cdot x)$
– $\sum_{i=0}^{q} v_{2,i} \cdot x^i + \sum_{i=0}^{q} v'_{2,i} \cdot b \cdot x^i = 0$ $(*)$

Now, if we focus on the second equation we can note that each member is a polynomial of degree 1 in b, which implies that:

– $(x + w) \sum_{i=0}^{q} v_{1,i} \cdot x^i = u_{2,1} \cdot x \cdot P(x)$
– $(x + w) \sum_{i=0}^{q} v_{1,i} \cdot b \cdot x^i = u_{2,2} \cdot x^2 \cdot P(x)$

Since $w \neq 0$, each of these equations implies that $(x + w)$ divides $P(x)$, which is impossible since these polynomials are relatively prime. Therefore, we have $u_{2,1} = u_{2,2} = 0$ which, together with relation $(*)$, implies that $r = 0$. It is therefore impossible to symbolically produce a valid answer to the q-MSDH-2 problem.

It now remains to evaluate the probability of an accidental equality when evaluating the different pairs of polynomials. The q-MSDH-2 assumption provides $2q + 6$ elements which can be associated with polynomials of degree at most $q + 1$. Each of the Q oracle queries leads then to a polynomial of degree at most $2(q + 1)$. By the Schwartz-Zippel lemma, one can conclude that the probability that two pairs, among the $(2q + 6 + Q)^2/2$ possible ones, evaluate to the same value is smaller than $(q + 1)(2q + 6 + Q)^2/p$, which is negligible.