Schnorr Signatures are Tightly Secure in the ROM under a Non-interactive Assumption

Gavin Cho[1], Georg Fuchsbauer[2], and Adam O'Neill[1]

[1] Manning CICS, UMass Amherst, {gkcho,adamoneill}@umass.edu
[2] TU Wien, firstname.lastname@tuwien.ac.at

Abstract. We show that the widely-used Schnorr signature scheme meets existential unforgeability under chosen-message attack (EUF-CMA) in the random oracle model (ROM) if the circular discrete-logarithm (CDL) assumption, a new, non-interactive and falsifiable variant of the discrete-log (DL) problem we introduce, holds in the underlying group. Notably, our reduction is *tight*, meaning the constructed adversary against CDL has essentially the same running time and success probability as the assumed forger. This is crucial for justifying the size of the underlying group used in practice. To our knowledge, we are the first to exhibit such a reduction. Indeed, prior work required interactive and non-falsifiable assumptions (Bellare and Dai, INDOCRYPT 2020) or additional idealized models beyond the ROM like the algebraic group model (Fuchsbauer *et al.*, EUROCRYPT 2020). We justify CDL by showing it holds in two carefully-chosen idealized models that idealize different aspects of it. Namely, we show that CDL is as hard as DL in these models.

Keywords: Schnorr signatures · tight security · ECDSA conversion function

1 Introduction

1.1 Background and Main Results

Schnorr signatures and our focus. The Schnorr signature scheme [Sch90] (recalled below), specifically in the form of EdDSA [BDL+12] implemented over twisted Edwards curves, is one of the most widely used pieces of cryptography today. For example, it is used in SSH/SSL, and in Bitcoin since the Taproot soft-fork upgrade in November 2021 [WNR20]. There is a rich theory behind the scheme's security, with tantalizing open questions. The initial result of Pointcheval and Stern (PS) [PS96] showed the scheme meets existential unforgeability (EUF-CMA) in the random oracle model (ROM) [BR93] assuming the discrete-logarithm (DL) assumption holds in the underlying group. However, the PS result has the two major downsides, which have persisted despite much follow-on work: (1) the proof relies on the artificial ROM, and (2) the reduction given in the proof is *lossy*. In this work, we focus on overcoming (2).

The problem of tight security. Given a forger against the Schnorr signature scheme in the ROM with success probability p_{succ}, the PS result says there is an adversary solving DL in the underlying group with similar running time and having success probability $\Theta(p_{\mathsf{succ}}^2/q_H)$, where q_H is the number of RO queries made by the forger. Unfortunately, the discrepancy between the success probabilities of the assumed forger and the constructed DL adversary makes the result meaningless in practice. For example, suppose we implement the scheme over twisted Edwards curves of 256-bit order, which are conjectured to have 128-bit security for DL. Conservatively assuming at most 2^{64} RO queries, the PS result then tells us the scheme has $128/2 - 64 = 0$ bits of security for EUF-CMA! This is despite the lack of any known attack on the scheme short of solving DL after several decades of cryptanalysis.

Can we do better? Prior work [FPS20] showed tight security of Schnorr in additional idealized models such as the ROM combined with the algebraic group model (AGM) [FKL18], which makes assumptions on the adversary's strategy. Other work [NSW09, Sho23, CLMQ21] proved Schnorr secure directly in the generic group model (GGM) [Nec94, Sho97], making specific assumptions on the hash function. However, such idealized models are arguably better suited for analyzing simpler *assumptions* rather than the scheme itself.[3]

On the other hand, another sequence of works [PV05, GBL08, Seu12, FJS19] culminated in showing that there *is no* tight "generic" reduction in the ROM (even under a minimal formulation of security) for Schnorr signatures under *any* "representation-independent" non-interactive assumption (see below). Later, Bellare and Dai (BD) [BD20] showed that there *is* a generic reduction that loses *only* a q_H factor, thereby surpassing the "square-root barrier" but still falling short of a completely tight reduction.

This gap matters: going back to our previous example, we would get $128 - 64 = 64$ bits of security for the scheme, which is insufficient for practical applications. BD also rely on a new *interactive* assumption[4] they call multi-base DL, which is a variant of the one-more discrete logarithm assumption [BNPS03]. Since the challenger must answer discrete-log queries by the adversary, multi-base DL is not a falsifiable assumption [Nao03].

We ask whether we can eliminate *both* downsides of their result, namely:

> *Is there a* completely tight *reduction in the ROM proving EUF-CMA of Schnorr signatures from a non-interactive and falsifiable assumption?*

Due to the above-mentioned impossibility results, such an assumption has to be *representation-dependent*, meaning depend on the specific group representation (see further discussion below). It is *a priori* unclear (to us, at least) what such an assumption would look like.

[3] For example, these idealized models are subject to *uninstantiability* results [Den02, Zha22], so must be used with caution. Complex and interactive problems shown to hold in these models are more likely to fall prey to the reach of these results.

[4] Note that the assumption is *single-query*, whereas directly assuming the security of the signature scheme would be *multi-query*.

Our results. We resolve the above question in the affirmative. To do so, our central conceptual contribution is the *circular discrete-logarithm* (CDL) problem, a new, non-interactive, falsifiable (at least under a specific formulation of the assumption), and representation-dependent variant of DL. We show that if CDL holds then the Schnorr signature scheme is secure in the underlying group, with a *completely tight* reduction in the ROM. Then, using carefully-chosen idealized models that idealize differrent aspect of our assumption in appropriate elliptic-curve groups (*e.g.*, twisted Edwards curves), we show that breaking CDL has essentially the same complexity as breaking DL. Consequently, if one believes in these models for analyzing DL/CDL in appropriate elliptic-curve groups, then the security level of the scheme in such groups shown by our results matches that indicated by decades of cryptanalysis.

1.2 Technical Overview

The Schnorr signature scheme. The Schnorr signature scheme is defined over a group G of prime order p generated by $g \in \mathsf{G}$ and uses a hash function $H\colon \mathsf{G} \times \{0,1\}^* \to \mathbb{Z}_p$. A secret key x is uniformly sampled from $\mathbb{Z}_p$, *i.e.*, $x \leftarrow\!\!\$\, \mathbb{Z}_p$, and defines a public key $h \leftarrow g^x$. A signature on a message $m \in \{0,1\}^*$ is computed by sampling $r \leftarrow\!\!\$\, \mathbb{Z}_p$, setting $R \leftarrow g^r$, computing $c \leftarrow H(R\|m)$ and returning (R, s) with $s \leftarrow (r + cx) \bmod p$. A signature (R, s) on m under public key h is verified by checking $(g^s = R \cdot h^c)$, where $c \leftarrow H(R\|m)$. We denote the scheme by $\mathsf{Sch}[\mathsf{G}, H]$ and usually drop H when working in the ROM.

The circular discrete-logarithm problem. Let G be as above. Let $f\colon \mathsf{G} \to \mathbb{Z}_p$ be an efficient function that we call a *conversion function* following the terminology regarding ECDSA [oST13]. We say that the *circular discrete-logarithm* (CDL) problem holds in G for f if, given uniformly sampled $h \in \mathsf{G}$, it's hard to find (R, z) such that

$$g^z = R \cdot h^{f(R)} \tag{1}$$

and $f(R) \neq 0$. One can think of f as a very simple function, not (necessarily) a cryptographic hash function used to instantiate Schnorr. We say that CDL holds for G if there exists f such that CDL holds in G for f. (We exclude the all-zeros function, since CDL trivially holds for it.) We denote the CDL for f in G as $\mathrm{CDL}[\mathsf{G}, f]$ and just CDL for G as $\mathrm{CDL}[\mathsf{G}]$. The "circularity" in CDL is that in the solution equation R occurs both in the base and the exponent, mirroring this peculiar property of Schnorr's verification equation. An equivalent formulation is, given $h \leftarrow\!\!\$\, \mathsf{G}$, find $a, b \in \mathbb{Z}_p$ such that $f(g^a/h^b) = b$. Indeed, this immediately yields a CDL solution with $z = a$ and $R = g^a/h^b$.

It is easy to see that if $\mathrm{CDL}[\mathsf{G}, f]$ holds for some function f, then DL holds in G. On the other hand, note that even for DL-hard G, $\mathrm{CDL}[\mathsf{G}, f]$ does cannot hold for *every* function f. In particular, it is necessary that no value has a large preimage under f: Suppose $t \in \mathbb{Z}_p^*$ does; then a uniform $z \leftarrow\!\!\$\, \mathbb{Z}_p^*$, together with $R := g^z/h^t$ breaks CDL if $f(R) = t$, which happens with probability proportional to the size of $f^{-1}(t)$ in $\mathbb{Z}_p^*$. In Section 4, we show that, when modelling G as an

elliptic-curve generic group [GS22], this assumption on f is not only necessary but also sufficient for CDL to hold. For an elliptic-curve group, a simple example of a function f satisfying this assumption (as previously argued in [GS22]) is the *ECDSA conversion function* [oST13], which maps a point $P = (x, y)$ on the curve to its reduced x-coordinate, *i.e.*, $f\colon (x, y) \mapsto x \bmod p$.

Tight reduction from Schnorr to CDL. Our main result is a completely tight reduction in the ROM from EUF-CMA of $\mathsf{Sch}[\mathsf{G}]$ to CDL$[\mathsf{G}]$. Namely, given a forger against $\mathsf{Sch}[\mathsf{G}]$, we construct an adversary solving CDL$[\mathsf{G}, f]$ with essentially the same running-time and success probability as the assumed forger. The only assumption we make on f is that $|f^{-1}(0)|$, the number of elements f maps to 0, is small. This is implied by the condition on f discussed above, which is necessary for CDL$[\mathsf{G}, f]$ to hold in the first place.

Given an instance $h \in \mathsf{G}$ of CDL$[\mathsf{G}, f]$, the reduction sets h as the public key for the forger, so it does not know the corresponding secret key. When run, the forger makes hash queries and signing queries. Its signing queries are simulated by the CDL adversary as in the standard proof of Schnorr in [PS96]. Namely, on query m_i, the reduction picks $s_i, c_i \leftarrow\!\!\$\ \mathbb{Z}_p$, sets $R_i \leftarrow g^{s_i}/h^{c_i}$, and programs the random oracle at $R_i \| m_i$ to be c_i, returning (R_i, s_i) as the signature. (Since R_i is uniform, the probability that the RO is already defined at $R_i \| m_i$ is negligible.)

To simulate the hash queries, the intuition is that we embed outputs of f into the answers; at the same time, we need to ensure that the returned values are uniformly distributed values in $\mathbb{Z}_p$, independent of the adversary's view. For this, on query $R \| m$, the reduction picks $a, b \leftarrow\!\!\$\ \mathbb{Z}_p$ and programs the RO at $R \| m$ to $(f(R \cdot h^a \cdot g^b) + a) \bmod p$. We argue that *no matter what f is*, the hash values satisfy are independent and uniform.

Now, consider a successful forgery (R, s) on some m, thus

$$g^s = R \cdot h^c \ , \tag{2}$$

where c is the RO response for $R \| m$. Since m is different from the queried messages,[5] the query $R \| m$ was made explicitly (either by the adversary or the game when verifying the forgery), that is, the RO was not programmed during a signing query. Let thus a, b be the values chosen by the reduction when answering this RO query, that is, $c = f(R \cdot h^a \cdot g^b) + a$ (modulo p). Together with Eq. (2), this yields $g^s = R \cdot h^{f(R \cdot h^a \cdot g^b) + a}$, which implies

$$g^s \cdot g^b = (R \cdot h^a \cdot g^b) \cdot h^{f(R \cdot h^a \cdot g^b)} \ .$$

Thus, $(R^* := R \cdot h^a \cdot g^b, z^* := (s + b) \bmod p)$ is solution to CDL as long as $f(R^*) \neq 0$, which by our initial assumption on f holds with high probability.

Analyzing CDL in the EC-GGM. To gain confidence in a new computational hardness assumption in prime-order groups, it has become standard to

[5] Note that we can actually show *strong* unforgeability, namely that it's even hard for the adversary to forge a new signature on an already-signed message, since $(R, s, m) \neq (R_i, s_i, m_i)$ for all i implies $R \| m \neq R_i \| m_i$ for all i, since s_i is uniquely determined by R_i and m_i.

analyze it in the generic group model (GGM) [Nec94, Sho97], where the adversary only gets (random) labels of group elements and has access to an oracle to compute the group operation. Concretely, given the labels of two group elements, the oracle returns the label of the product of the group elements.

This model is however syntactically ill-defined when there is a function taking as input group elements, for example the conversion function in ECDSA [oST13], which we conjecture to be a suitable choice for CDL in such groups. To analyze the security of ECDSA (and variants thereof), Groth and Shoup (GS) [GS22] introduce the *elliptic-curve GGM*, where the labels are random group elements from an elliptic curve (conditioned on preserving simple properties, namely the identity element and inverses).

In Section 4 we analyze CDL$[\mathsf{G}, f]$ for arbitrary G and $f\colon \mathsf{G} \to \mathbb{Z}_p$, where $p = |\mathsf{G}|$, in the EC-GGM model, which uses labels from G. We show that CDL holds conditioned on the aforementioned (necessary) property of f: no element in its range can have a large preimage. In particular, we show that the advantage of any adversary is bounded by $((q+1)\cdot\mathsf{MaxSize}_f+27q^2+39q+15)/(p-1)$, where $\mathsf{MaxSize}_f := \max_{t\in\mathbb{Z}_q}\{|f^{-1}(t)|\}$ is the largest preimage and q is the number of group-oracle queries made by the adversary. Note that we use exactly the same property as GS do on the conversion function to prove security of ECDSA in the EC-GGM. As the ECDSA conversion function is 2-to-1, in this case the bound for CDL is essentially the same as the bound for DL [Sho97, GS22]. In other words, CDL in an appropriate elliptic-curve group and for the ECDSA conversion function is about as hard as DL in the EC-GGM.

Analyzing CDL in the algebraic bijective ROM. We also consider idealizing f. Particularly, we look at how the ECDSA conversion function is modeled in security analyses of ECDSA that idealize the conversion function; however, our analysis is again not tied to the ECDSA conversion function and works for other functions that have similar structure but are more "random" than the ECDSA conversion function. In fact, for such functions, this idealization is even for meaningful than for the ECDSA conversion function. Thus, it seems reasonable to use such an idealized model in our case.

Initial results on ECDSA's security by Brown [Bro02] modeled the conversion function as a RO (in addition to using the GGM), which ignores its obvious structure. To better capture the structure, the *bijective* ROM was proposed [FKP16, FKP17, HK23]. In this model $f = \psi \circ \Pi \circ \varphi$, where φ maps from G to $\mathbb{A} := \{0,1\}^L$, Π maps from $\mathbb{A}$ to $\mathbb{B} := [2^L - 1]$, and ψ maps from $\mathbb{B}$ to $\mathbb{Z}_p$. Here φ and ψ are standard-model functions, while Π is modeled as a bijective RO.

In fact, CDL for the ECDSA conversion function is a special case of the *semi-logarithm problem* (SLP) introduced by Brown [Bro05] and generalized by Fersch, Kiltz, and Poettering (FKP) [FKP17, Definition 6] with $\rho_0(u,v) = u$ and $\rho_1(u,v) = -v$. FKP show a loose reduction from SDL to DL in the BROM (and hence get a loose reduction for ECDSA). We would like a *tight* reduction in the case of CDL. We manage to do this by additionally assuming that the adversary is *algebraic* wrt. its queries to Π, *i.e.*, we use the algebraic BRO model

(ABROM) recently proposed to analyze blind ECDSA [QCY21]. Specifically, we show that the advantage of any CDL adversary in the ABROM is bounded by the hardness of DL in G plus $(2q^2 + 2q + q \cdot \mathsf{MaxSize}_\psi)/p$, where q is the number of queries made by the adversary. Our algebraic assumption on the adversary is important here. If we showed a tight reduction from CDL to DL in the BROM, that would imply (by our main result) a tight reduction from security of Schnorr signatures to DL in the ROM, which would be surprising.

1.3 Discussion

Comparison to ratio-based tightness. While the focus of their work is on the multi-user setting, Kiltz, Masny and Pan (KMP) [KMP16] give a two-link chain of reductions going from single-user EUF-CMA of Schnorr signatures to DL. The first reduction [KMP16, Lemma 3.5] goes from passive impersonation of Schnorr's identification protocol to DL, and the second goes from EUF-CMA of Schnorr signatures to the former. While they claim the first reduction is tight, their criterion for tightness is "ratio-based," namely requiring roughly equal *time-to-success ratios* of the assumed adversary and the constructed one. Unfortunately, as discussed by Bellare and Dai (BD) in [BD20, Appendix B], this criterion is problematic and the aforementioned reduction (which, as in [PS96] uses rewinding), has a substantial running-time blowup. As in [BD20], we employ a notion of tightness that requires roughly equal success probabilities and running times *individually*, avoiding such problems.

The second reduction [KMP16, Theorem 1.1] loses a factor q_H even under ratio-based tightness. Overall, as already argued by [BD20], in the single-user setting KMP do not improve on the required size of underlying group for Schnorr signatures versus classical results.

Relation to instantiability. CDL is related to the problem of instantiating Schnorr signatures (*i.e.*, replacing its RO with a concrete hash function) under a weak security notion called *universal unforgeability under no-message attack* (UUF-NMA) used to show prior *im*possibility results [PV05, GBL08, Seu12, FJS19]. In UUF-NMA, the forgery message is random and given to the adversary, and the adversary gets no signing queries. As compared to instantiating Schnorr signatures under UUF-CMA, CDL differs in that there is no message and f takes solely a group element as input. We also stress that in the above-mentioned impossibility results, UUF-NMA is only considered in the ROM.

Representation-dependence of CDL. Our reduction from EUF-CMA of Schnorr signatures to CDL does not contradict prior impossibility results because the most general of these results [FJS19] only applies to underlying problems that are representation-invariant, *i.e.*, an instance-solution pair remains valid when the representation of the underlying group is changed. However, CDL is representation-dependent. Indeed, Eq. (1) may hold for one group representation but not another, as the value of $f(R)$ depends on the representation.

FALSIFIABILITY OF CDL. For given G, f, CDL$[\mathsf{G}, f]$ is a clearly a falsifiable assumption in the sense of [Nao03]. For our main result, we can also rely on a seeminly weaker, *non-falsiable* CDL$[\mathsf{G}]$ that there merely *exists* an f such that CDL$[\mathsf{G}, f]$ holds, even if we don't know what it is. This is because we only use the conversion function f *in proofs*, not in real life. (We do not require modification to Schnorr signatures at all.) There have been previous instances of primitives occurring only in proofs, *e.g.* [BM14, GJO16, MOZ22], but to our knowledge it is novel in the context of Schnorr signatures. Note that we will also need that $|f^{-1}(0)|$ for such f is "small." The advantage of the best adversary breaking CDL$[\mathsf{G}, f]$ for a given resource usage and the meaning of "small" then determines our bound on the security of Schnorr signatures.

CDL AS A STEPPING-STONE. An important problem left open by our work is whether for every group G of order p (or those of interest) there is a function $f\colon \mathsf{G} \to \mathbb{Z}_p$ such that there is a reduction from CDL$[\mathsf{G}, f]$ to some more-standard assumption in G. For example, constructing f for which there is a tight reduction from CDL$[\mathsf{G}, f]$ to DL in G would, by composition, yield a tight reduction from EUF-CMA of $\mathsf{Sch}[\mathsf{G}]$ to DL in G. Interestingly, this would *not* contradict the above-mentioned impossibility results because our reduction is *non-generic*, since it computes f. Even a loose reduction would be of interest to corroborate CDL. In general, a construction of f could introduce other assumptions. We stress that Schnorr signatures themselves and their instantiation in practice using cryptographic hashing would remain unaffected.

EXTENSIONS TO SCHNORR. Many recent works build upon Schnorr signatures, particularly to achieve signature schemes with advanced functionalities such as aggregate signatures [CGKN21], blind signatures [FW24], multisignatures [NRS21], ring signatures [YEL+21], and threshold signatures [KG24]. (We give some representative citations to recent work, not an exhaustive list.) Schnorr signatures were also used to give adaptor signatures [AEE+21]. We leave it as an open problem to extend our results to these settings.

2 Preliminaries

NOTATION. If $\overrightarrow{v}$ is a vector then $|\overrightarrow{v}|$ is its length (the number of its coordinates) and v_i is its i-th coordinate. Strings are identified with vectors over $\{0,1\}$, so that $|Z|$ denotes the length of a string Z and Z_i denotes its i-th bit. By ε we denote the empty string or vector. By $x\|y$ we denote the concatenation of strings x, y. If S is a finite set, then $|S|$ denotes it size and we let $x \leftarrow\!\!\$\ S$ denote picking an element of S uniformly at random and assigning it to x.

Algorithms may be randomized unless otherwise indicated. If A is an algorithm, we let $y \leftarrow A^{\mathrm{O}_1,\cdots}(x_1,\ldots;\omega)$ denote running A on inputs $x_1,\ldots$ and coins ω, with oracle access to $\mathrm{O}_1,\ldots$, and assigning the output to y. Moreover, by $y \leftarrow\!\!\$\ A^{\mathrm{O}_1,\cdots}(x_1,\ldots)$ we denote picking ω at random and letting $y \leftarrow A^{\mathrm{O}_1,\cdots}(x_1,\ldots;\omega)$. We let $\mathbf{Out}(A^{\mathrm{O}_1,\cdots}(x_1,\ldots))$ denote the set of all possible outputs of A when run on inputs $x_1,\ldots$ and with oracle access to $\mathrm{O}_1,\ldots$. Running

time is worst-case, which for an algorithm with access to oracles means across all possible replies from the oracles. We use $\perp$ (bot) as a special symbol to denote rejection, and it is assumed not to be in $\{0,1\}^*$.

Let $f\colon A \to B$ be a function. We let $\mathsf{Size}_f(b) := |f^{-1}(b)|$ for all $b \in B$, $\mathsf{MaxSize}_f := \max_{b\in B} \mathsf{Size}_f(b)$. For sets A, B such that $|A| = |B|$, we let $\mathrm{Inj}(A, B)$ denote the set of injection from A to B.

GAMES. We use the code-based game-playing framework of BR [BR06]. By $\Pr[\mathrm{G} \Rightarrow y]$ we denote the probability that the execution of game G results in this output being y. In games, integer variables, set variables, boolean variables and string variables are assumed initialized, respectively, to 0, the empty set $\emptyset$, the boolean false and $\perp$.

2.1 Schnorr Signatures and Their Security

SIGNATURE SCHEMES AND THEIR SECURITY. A *signature scheme* with message space MS is a tuple of algorithms $\mathsf{DS} = (\mathsf{DS.K}, \mathsf{DS.S}, \mathsf{DS.V})$ that work as follows:

- $\mathsf{DS.K}$: The key-generation algorithm outputs a key pair (vk, sk). (We suppress the security parameter for simplicity.)
- $\mathsf{DS.S}(sk, m)$: On inputs a signing key sk and a message $m \in \mathsf{MS}$, the signing algorithm outputs a signature σ.
- $\mathsf{DS.V}(vk, \sigma, m)$: On inputs a verification key vk, signature σ, and message, $m \in \mathsf{MS}$, the verification algorithm outputs a bit.

For correctness, we require that

$$\Pr\left[\mathsf{DS.V}\big(vk, \mathsf{DS.S}(sk, m), m\big) \Rightarrow 1\right] = 1$$

for all $(sk, vk) \in \mathbf{Out}(\mathsf{DS.K})$ and all $m \in \mathsf{MS}$, where the probability is over the coins for $\mathsf{DS.S}$.

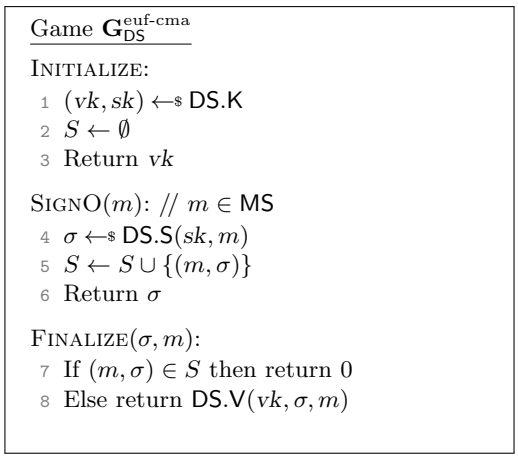

Fig. 1: Game defining (strong) EUF-CMA of DS.

Game $\mathbf{G}^{\mathrm{dl}}_{\mathsf{G},g}$	Game $\mathbf{G}^{\mathrm{cdl}}_{\mathsf{G},g,f}$
INITIALIZE: 1 $x \leftarrow\!\!\$\ \mathbb{Z}_p^*$ 2 $h \leftarrow g^x$ 3 Return h	INITIALIZE: 1 $x \leftarrow\!\!\$\ \mathbb{Z}_p^*$ 2 $h \leftarrow g^x$ 3 Return h
FINALIZE(x'): 4 Return $(x = x')$	FINALIZE(R, z): 4 Return $\big(f(R) \neq 0 \ \wedge\ g^z = R \cdot h^{f(R)}\big)$

Fig. 2: Games defining DL and circular DL problems.

SCHNORR SIGNATURES. Let G be a cyclic group of prime order $p = |\mathsf{G}|$, generated by g. Let $H\colon \mathsf{G} \times \{0,1\}^* \to \mathbb{Z}_p$ be a hash function. The *Schnorr signature scheme* [FS87, Sch90] $\mathsf{Sch}[\mathsf{G}, H] = (\mathsf{Sch.K}, \mathsf{Sch.S}, \mathsf{Sch.V})$ with message space $\{0,1\}^*$ works as follows. Algorithm $\mathsf{Sch.K}$ chooses $x \leftarrow\!\!\$\ \mathbb{Z}_p$, sets $X \leftarrow g^x$, and returns $(vk = X, sk = x)$. Algorithm $\mathsf{Sch.S}$ on input x, m chooses $r \leftarrow\!\!\$\ \mathbb{Z}_p$, sets $R \leftarrow g^r$ and $c \leftarrow H(R\|m)$, then returns $\big(R, (r + cx) \bmod p\big)$. Algorithm $\mathsf{Sch.V}$ on inputs $X, (R, s), m$ returns $\big(g^s = R \cdot X^c\big)$ where $c \leftarrow H(R\|m)$. Correctness is straightforward to check. In the ROM, we denote the scheme by $\mathsf{Sch}[\mathsf{G}]$.

EUF-CMA. We define the (strong) existential unforgeability under chosen-message attack (EUF-CMA). Let $\mathsf{DS} = (\mathsf{DS.K}, \mathsf{DS.S}, \mathsf{DS.V})$ with message space MS. For an adversary A, we let its (strong) EUF-CMA advantage against DS be $\mathbf{Adv}^{\text{euf-cma}}_{\mathsf{DS}}(\mathrm{A}) = \Pr\left[\mathrm{G}^{\text{euf-cma,A}}_{\mathsf{DS}} \Rightarrow 1\right]$, where the game is in Figure 1.

2.2 Discrete-Logarithm Problem

We recall the *discrete-logarithm* (DL) problem. Let G be a group of prime order $p = |\mathsf{G}|$, generated by g. For an adversary A, we let its DL-advantage against G, g be $\mathbf{Adv}^{\mathrm{dl}}_{\mathsf{G},g}(\mathrm{A}) = \Pr\left[\mathrm{G}^{\mathrm{dl,A}}_{\mathsf{G},g} \Rightarrow 1\right]$, where the game is in Figure 2.

3 Tight Security of Schnorr Signatures under CDL

We provide our new assumption then proceed to give a tight reduction of EUF-CMA security of Schnorr signatures in the ROM to our assumption.

3.1 Circular Discrete-Logarithm Problem

We introduce the *circular discrete-logarithm* (CDL) problem. Let G be a group of prime order $p = |\mathsf{G}|$, generated by g. Let $f\colon \mathsf{G} \to \mathbb{Z}_p$. For an adversary A we let its CDL-advantage against G, g, f be $\mathbf{Adv}^{\mathrm{cdl}}_{\mathsf{G},g,f}(\mathrm{A}) = \Pr\left[\mathbf{G}^{\mathrm{cdl}}_{\mathsf{G},g,f}(\mathrm{A}) \Rightarrow 1\right]$ where the game is in Figure 2.

If we want to assume that there exists no efficient adversary that solves CDL, the condition $f(R) \neq 0$ in the FINALIZE procedure is essential. Otherwise,

consider the adversary who has some $R^* \in \mathsf{G}$ such that $f(R^*) = 0$ hard-coded along with $z^* = \mathsf{DLog}_{\mathsf{G},g}(R^*)$, and simply outputs (R^*, z^*) as a valid CDL solution. This adversary would have advantage 1. The assumption would thus be wrong, even though no one might *know* such an adversary. This is analogous to collision-resistance of hash functions, for which an adversary always exists (*cf.* [Rog06]). By adding the condition $f(R) \neq 0$, we simply avoid such issues. The assumption remains strong enough for all our applications.

3.2 Main Result

Theorem 1 *Let* G *be a group of prime order* p*. Let* A *be an adversary against the Schnorr signature scheme* $\mathsf{Sch}[\mathsf{G}]$ *in the ROM and assume* A *makes at most* q_s *queries to the signing oracle and* q_h *queries to the hash oracle. Let* $f\colon \mathsf{G} \to \mathbb{Z}_p$ *be arbitrary and efficient. Then there exists an adversary* B *with running time roughly the same as* A *plus simulation overhead proportional to* q_s *and* $q_h \cdot T_f$*, where* T_f *is the time to compute* f*, such that*

$$\mathbf{Adv}^{\text{euf-cma}}_{\mathsf{Sch}[\mathsf{G}]}(\mathrm{A}) \leq \mathbf{Adv}^{\text{cdl}}_{\mathsf{G},g,f}(\mathrm{B}) + \frac{q_s(q_s+q_h) + q_h \cdot \mathsf{Size}_f(0)}{p} \,. \tag{3}$$

Remark 1. When $\mathsf{Size}_f(0)$ is small, the additive term in the RHS of Equation 3 is consistent with the quadratic lower-bound in the adversary's probability of (generically) solving DL in G. Indeed, for any f, one can break $\mathrm{CDL}[\mathsf{G}, f]$ by computing DL of h.

Remark 2. The running-time of the adversary constructed in our reduction depends on the running-time for f. Thus, if f is inefficient to compute, it will affect tightness of our reduction. However, all candidate f's we consider in our work are extremely efficient, *e.g.*, they output some bits of the input.

Remark 3. It is standard practice in implementations of Schnorr signatures to prepend the public key h to inputs to the hash function H. This ensures domain separation between users and implies better bounds for multi-user security. For simplicity we omit to do this, but our result readily extends to this setting. Namely, in the reduction, the CDL adversary handles the hash queries $h' \| R \| m$ by checking if $h = h'$. If so, it handles the query as in our simulation below. If not, it returns a random value (answering consistently across repeated queries).

Next, we prove the above theorem by formalizing the ideas laid out in the Introduction (Section 1.2).

Proof. We consider a sequence of games defined in Figure 3. Games in boxes (G_1 and G_3) contain the boxed lines, whereas they are ignored for the other games. Note that G_0 is equivalent to $\mathbf{G}^{\text{euf-cma}}_{\mathsf{Sch}[\mathsf{G}]}$.

$\underline{\mathrm{G}_0 \to \mathrm{G}_1}$. We start with analyzing how the probability of returning 1 changes from G_0 to G_1. The probability of aborting in line 6 in G_1 during any signing query is at most $(q_s + q_h)/p$ since R is uniformly sampled and there are at most

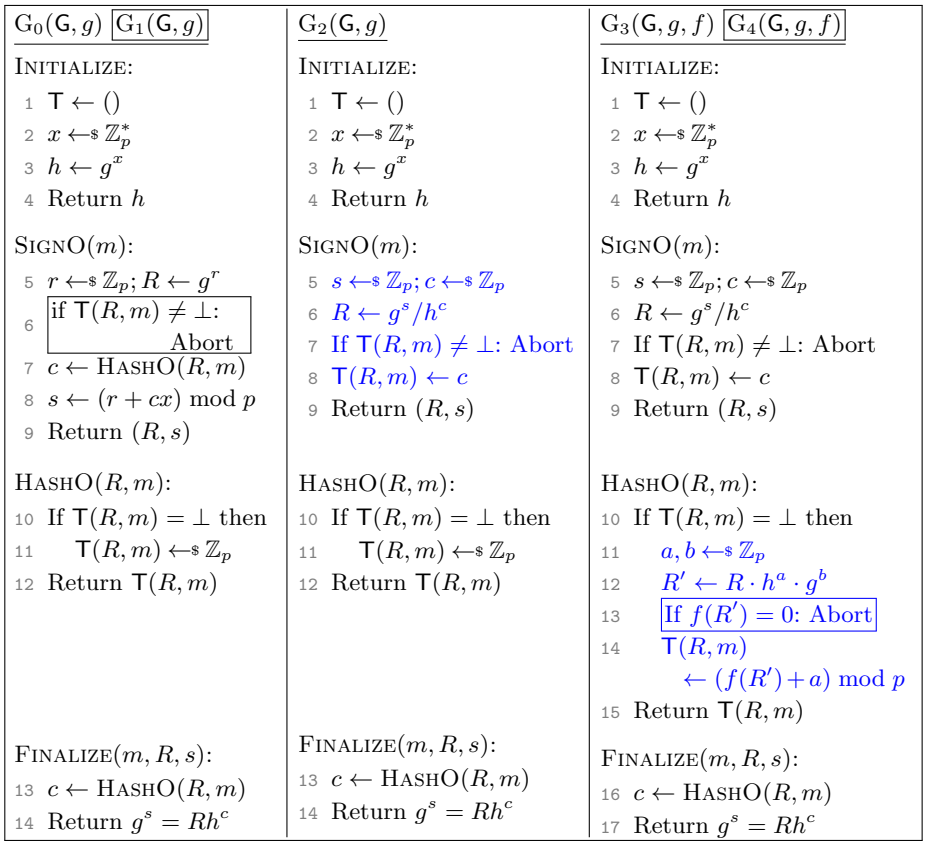

Fig. 3: Games for the proof of Theorem 1. Changes are highlighted in blue.

$q_s + q_h$ possible $(\cdot, m)$ pairs defined in T it could hit. By a union bound over all signing queries, the probability of G_1 aborting is at most $q_s(q_s + q_h)/p$. Thus, by the Fundamental Lemma of Game Playing [BR06] we have $\Pr\left[\mathrm{G}_0^{\mathrm{A}} \Rightarrow 1\right] \leq \Pr\left[\mathrm{G}_1^{\mathrm{A}} \Rightarrow 1\right] + q_s(q_s + q_h)/p$.

$\underline{\mathrm{G}_1 \to \mathrm{G}_2}$. In G_2, we change the signing oracle so that it doesn't require knowledge of the secret key x. We claim these games are equivalent. This is because, if $\mathsf{T}(R, m)$ has not been defined yet, the distribution of (R, s, c) in both games is uniform in $\mathsf{G} \times \mathbb{Z}_p \times \mathbb{Z}_p$ conditioned on $g^s = R \cdot h^c$. In other words, this hop corresponds to a reordering of how these variables are defined. We thus have $\Pr\left[\mathrm{G}_1^{\mathrm{A}} \Rightarrow 1\right] = \Pr\left[\mathrm{G}_2^{\mathrm{A}} \Rightarrow 1\right]$.

$\underline{\mathrm{G}_2 \to \mathrm{G}_3}$. In G_3, we change how we answer hash queries. We argue that the distribution of responses of the hash oracle in G_3 is equivalent to the distribution of responses in G_2. In G_2, the distribution of the value sampled in line 11 is uniform in $\mathbb{Z}_p$. This is also the case in G_3: Since a is uniform in $\mathbb{Z}_p$, so is the

Adversary $\mathrm{B}(h)$

$\mathsf{T} \leftarrow ()$; $i \leftarrow 1$
$(m, R, s) \leftarrow\!\!\$\ \mathrm{A}^{\textsc{SignO},\textsc{HashO}}(\mathsf{G}, g, h)$
$\textsc{HashO}(R, m)$ //ensure that value is defined
Let j be such that $R = R_j$ and $m = m_j$. If such j does not exist, abort
$R^* \leftarrow R \cdot h^{a_j} g^{b_j}$; $s^* \leftarrow (s + b_j) \bmod p$
Return (R^*, s^*)

$\textsc{SignO}(m)$:

$s \leftarrow\!\!\$\ \mathbb{Z}_p$; $c \leftarrow\!\!\$\ \mathbb{Z}_p$; $R \leftarrow g^s / h^c$
If $\mathsf{T}(R, m) \neq \bot$: Abort
$\mathsf{T}(R, m) \leftarrow c$
Return (R, s)

$\textsc{HashO}(R_i, m_i)$:

If $\mathsf{T}(R_i, m_i) = \bot$ then
  $a_i, b_i \leftarrow\!\!\$\ \mathbb{Z}_p$
  $R' \leftarrow R_i \cdot h^{a_i} \cdot g^{b_i}$
  If $f(R') = 0$: Abort
  $\mathsf{T}(R_i, m_i) \leftarrow (f(R') + a_i) \bmod p$
  $i \leftarrow i + 1$
Return $\mathsf{T}(R_i, m_i)$

Fig. 4: CDL-adversary B for the proof of Theorem 1.

value $(f(R') + a) \bmod p$. Moreover, conditioned on any *fixed* a, $R' = R \cdot h^a \cdot g^b$ is uniform in G because b is uniform in $\mathbb{Z}_p$. Thus $\Pr\left[\mathrm{G}_2^{\mathrm{A}} \Rightarrow 1\right] = \Pr\left[G_3^{\mathrm{A}} \Rightarrow 1\right]$.

$\underline{\mathrm{G}_3 \to \mathrm{G}_4}$. This hop introduces an abort whenever a hash query is made and $f(R') = 0$. Since R' is independent and uniformly distributed (as was just shown), the probability of any query aborting is $|f^{-1}(0)|/p$. By a union bound over all queries, the probability of aborting is at most $q_h \cdot |f^{-1}(0)|/p$. Thus, we get $\Pr\left[\mathrm{G}_3^{\mathrm{A}} \Rightarrow 1\right] \leq \Pr\left[\mathrm{G}_4^{\mathrm{A}} \Rightarrow 1\right] + q_h \cdot \mathsf{MaxSize}_f(0)/p$.

$\underline{\mathrm{G}_4}$. Combining the hops above yields

$$\mathbf{Adv}^{\text{euf-cma}}_{\mathsf{Sch}[\mathsf{G}]}(\mathrm{A}) \leq \mathbf{Adv}^{\mathrm{G}_4}_{\mathsf{G},g,f}(\mathrm{A}) + \frac{q_s(q_s + q_h) + q_h \cdot \mathsf{MaxSize}_f(0)}{p} \ , \tag{4}$$

Finally, consider adversary B defined in Figure 4. To prove the theorem, it remains to show that

$$\mathbf{Adv}^{\mathrm{G}_4}_{\mathsf{G},g,f}(\mathrm{A}) \leq \mathbf{Adv}^{\text{cdl}}_{\mathsf{G},g,f}(\mathrm{B}) \ ,$$

which combined with Eq. (4) completes the proof. To do so, we show that whenever A returns a strong forgery (m, R, s) in G_4, then B returns a solution to the given CDL instance. A wins if

$$g^s = R \cdot h^c \quad \text{with } c \leftarrow \mathsf{T}(R, m) \ . \tag{5}$$

We claim that $\mathsf{T}(R, m)$ must have been defined during some call to HashO, which in turn means that if A wins, B does not abort in line 4. The reason is that (m, R, s) is different from all (m_i, R_i, s_i), which consist of the i-th query m_i to SignO together with its response (R_i, s_i). Since for a signature (R, s) on m, the value s is determined by R and m, this condition is equivalent to $R\|m$ being different from all $R_i\|m_i$. Thus $\mathsf{T}(R, m)$ was not defined during a call to SignO and, if nowhere else, was defined via the call to HashO in line 3.

So let j be the HashO query that defined $\mathsf{T}(R, m)$, and let a_j, b_j be the values that were sampled during the call; thus

$$c = \mathsf{T}(R, m) = \left(f(R \cdot h^{a_j} \cdot g^{b_j}) + a_j\right) \bmod p\,.$$

Together with Eq. (5), this yields $g^s = R \cdot h^{f(R \cdot h^{a_j} \cdot g^{b_j}) + a_j}$. Multiplying by g^{b_j} yields $g^{s+b_j} = (R \cdot h^{a_j} \cdot g^{b_j}) \cdot h^{f(R \cdot h^{a_j} \cdot g^{b_j})}$. We note that $f(R \cdot h^{a_j} \cdot g^{b_j}) \neq 0$ as assured by the abort condition added in G_4. Together this shows that $R^* := R \cdot h^{a_j} \cdot g^{b_j}, s^* := (s + b_j) \bmod p$, the values returned by B, constitute a valid solution to the given CDL instance. □

4 CDL in the Elliptic-Curve GGM

The EC-GGM. The *Elliptic-curve generic group model* (EC-GGM) [GS22] was introduced to cover constructions (such as ECDSA) that define a function which takes as input group elements. In contrast to Shoup's [Sho97] original GGM, in the EC-GGM, the "encodings" of the group elements are not random strings but random points on a concrete elliptic curve E, which has a prime number p of points. (Using multiplicative notation, we let 1_E denote the identity element.)

In more detail, in security games defined in the EC-GGM, in the beginning, the challenger chooses a random injective *encoding function* $\tau\colon \mathbb{Z}_p \to E$ which preserves trivial relations; in particular, $\tau(0)$ is the identity element and if $\tau(i) = P$ then $\tau(-i) = P^{-1}$. To "compute" in the group, parties have two oracles: Map(i), for $i \in \mathbb{Z}_p$, returns $\tau(i)$; computing linear combinations of group elements is done by calling Add(c_1, P_1, c_2, P_2) for $c_i \in \mathbb{Z}_p$ and $P_i \in E$, for $i = 1, 2$, which returns $\tau(c_1 \cdot \tau^{-1}(P_1) + c_2 \cdot \tau^{-1}(P_2))$.

Note that for schemes (and assumptions) defined over elliptic curve groups (and in particular, if there is a function whose domain is the group), this model is more realistic than the pure GGM (and makes sense syntactically). *E.g.*, in the GGM, one can show ECDSA strongly unforgeable – although it is malleable – which is not possible in the EC-GGM.

Proof overview of CDL in the EC-GGM. To argue that CDL holds in the EC-GGM, we follow the common approach for GGM proofs, also taken by Groth and Shoup [GS22]: instead of randomly sampling τ in the beginning of the game, we sample entries of the form (i, P) on the fly as required; this does not change the adversary's view. The game is defined in Figure 5.

We then simplify the game and abort whenever the "lazy" sampling does not succeed at the first try (including the boxes with a single frame in Figure 5). Let-

Game $\mathbf{G}^{\text{ec-ggm-cdl}}_{E,f}$; $\boxed{\mathbf{G}^{\text{abrt-cdl}}_{E,f}}$; $\boxed{\boxed{\mathbf{G}^{\text{symb-cdl}}_{E,f}}}$

INITIALIZE:
1 $\tau \leftarrow \{(0, 1_E)\}$
2 $g' \leftarrow \text{MAP}(1)$
3 $\boxed{h \leftarrow \text{MAP}(X); \text{ Return } (g', h)}$
4 $x \leftarrow\!\!\$\ \mathbb{Z}_p^*; h \leftarrow \text{MAP}(x)$
5 Return (g', h)

MAP(i):
6 If $i \notin \text{Dom}(\tau)$
7 $\quad P \leftarrow\!\!\$\ E^*$
8 $\quad \boxed{\text{if } P \in \text{Ran}(\tau) \text{ then abort}}$
9 $\quad$ while $P \in \text{Ran}(\tau) : P \leftarrow\!\!\$\ E^*$
10 $\quad \tau \leftarrow \tau \cup \{(-i, P^{-1}), (i, P)\}$
11 Return $\tau(i)$

ADD(c_1, P_1, c_2, P_2):
12 For $j = 1, 2$: if $P_j \notin \text{Ran}(\tau)$
13 $\quad i \leftarrow\!\!\$\ \mathbb{Z}_p^*$
14 $\quad \boxed{\text{if } i \in \text{Dom}(\tau) \text{ then abort}}$
15 $\quad$ while $i \in \text{Dom}(\tau) : i \leftarrow\!\!\$\ \mathbb{Z}_p^*$:
16 $\quad \tau \leftarrow \tau \cup \left\{(-i, P_j^{-1}), (i, P_j)\right\}$
17 Return $\text{MAP}\big(c_1 \cdot \tau^{-1}(P_1) + c_2 \cdot \tau^{-1}(P_2)\big)$

FINALIZE(R, z):
18 $\boxed{\text{if } R \notin \text{Ran}(\tau) \text{ or } z \notin \text{Dom}(\tau) \text{ then return } 0}$
19 Return $\big(f(R) \neq 0 \ \wedge\ \text{MAP}(z) = \text{ADD}(1, R, f(R), h)\big)$

Fig. 5: Original and variants of the EC-GGM game for the circular discrete-log problem.

ting q denote the number of the adversary's group oracle queries, the difference to the original game is $\mathcal{O}(q^2/p)$. We next define a *symbolic* game where the secret value x is represented by an indeterminate X and the domain of τ now consists of (linear) polynomials in X (the game including all boxes in Figure 5). Again, using a standard GGM argument that relies on the Schwartz-Zippel Lemma, the difference to the previous game is $\mathcal{O}(q^2/p)$. Finally, we argue that the adversary's probability in winning the symbolic game is bounded by $\mathsf{MaxSize}_f/p$.

We note that, compared to Groth and Shoup we avoid the use of asymptotics and give a precise concrete analysis of our assumption in the EC-GGM. Additionally, the proof applies to much more general f than the ECDSA conversion function [oST13], as we simply require a bound on the largest preimage set. (For the ECDSA conversion function, this is two.)

Theorem 2 *Let E be an elliptic curve of prime order p and $f\colon E \to \mathbb{Z}_p$. Let* A *be an adversary against CDL for f in the EC-GGM with E that makes at most*

q queries to any of its oracles. Then

$$\mathbf{Adv}^{\text{ec-ggm-cdl}}_{E,f}(\mathrm{A}) \leq \frac{(q+1)\cdot \mathsf{MaxSize}_f + 27q^2 + 39q + 15}{p} .$$

Proof. We proceed via a sequence of games as defined in Figure 5 and analyze their differences.

ec-ggm-cdl→abrt-cdl. First consider the boxed code in lines 8 and 14. Consider the probability that in $\mathbf{G}^{\text{ec-ggm-cdl}}_{E,f}$, during any call to MAP or ADD, a value P or i is sampled (uniformly at random) for which there is already an assignment in τ (which we call a "collision"). MAP is called once in lines 2 and 4 and at most q times by the adversary, and ADD is called at most q times by the adversary. Each call to MAP samples at most one value of P (for which either P, P^{-1} either both already have an assignment in τ or both do not), and similarly each call to ADD samples at most two values i and possibly a value P when calling MAP in line 17. Now, the probability of a collision in line 2 is zero and in line 4 is $2/(p-1)$. Then, in the worst case the adversary makes q ADD queries. On the first such call, the probability of a collision in line 13 (in either of its possible executions) is at most $4/(p-1) + 6/(p-1)$ and in line 17 is at most $8/(p-1)$. If no collisions happened, then the probability of one happening in the second call is bounded by $10/(p-1) + 12/(p-1) + 14/(p-1)$, and so on.

By a union bound, the overall probability of a collision is thus bounded by

$$\sum_{j=1}^{3q+1} \frac{2j}{p-1} = \frac{(3q+1)(3q+2)}{p-1} = \frac{9q^2+9q+2}{p-1} .$$

Finally, consider the boxed code in line 18. The probability that the adversary wins in $\mathbf{G}^{\text{ec-ggm-cdl}}_{E,f}$ although there is no assignment already in τ for one (or both) of its output elements R and z is $1/(p-1)$.

Thus, overall, the difference between the probabilities of the games $\mathbf{G}^{\text{ec-ggm-cdl}}_{E,f}$ and $\mathbf{G}^{\text{abrt-cdl}}_{E,f}$ outputting 1 is bounded by $(9q^2+9q+3)/(p-1)$.

abrt-cdl→symb-cdl. Consider game $\mathbf{G}^{\text{symb-cdl}}_{E,f}$, but where we sample x (in line 4), as in $\mathbf{G}^{\text{abrt-cdl}}_{E,f}$. If we furthermore replace the checks "if $i \in \mathrm{Dom}(\tau)$" (lines 6 and 14), where i is now a linear polynomial in X, by "if $i(x) \in \mathrm{Dom}(\tau)$" (as well as all other occurrences of i by $i(x)$) then we obtain a game that is distributed like the previous game $\mathbf{G}^{\text{abrt-cdl}}_{E,f}$.

The difference between games $\mathbf{G}^{\text{abrt-cdl}}_{E,f}$ and $\mathbf{G}^{\text{symb-cdl}}_{E,f}$ outputting 1 is thus bounded by the probability C that for any two $i \neq i' \in \mathrm{Dom}(\tau)$, we have $i(x) = i'(x)$. INITIALIZE creates 4 polynomials (± 1 and $\pm X$), a call to ADD creates up to six polynomials (four in line 16 and two in line 10 when line 17 is executed), while a call to MAP creates fewer; and FINALIZE creates at most 2 different polynomials (since it aborts if one of its arguments is not yet in τ). Note that, as long as no collision occurs, all polynomials are independent of x. Thus, C is upper-bounded by the probability that when sampling a random evaluation point (out of p possible ones) any two out of $6q+6$ different polynomials (which are

lines) intersect. Since $6q+6$ lines intersect in at most $\sum_{i=1}^{6q+6}(i-1) = \sum_{i=1}^{6q+5} i = \frac{1}{2}(6q+5)(6q+6) = (6q+5)(3q+3)$ points, we have

$$C = \frac{(3q+2)(6q+6)}{p-1} = \frac{18q^2+30q+12}{p-1} \ .$$

symb-cdl. Consider an output (R, z) by the adversary in game $\mathbf{G}^{\text{symb-cdl}}_{E,f}$ that makes FINALIZE return 1. Since $R \in \text{Ran}(\tau)$, there exist $a, b \in \mathbb{Z}_p$ s.t.

$$\tau^{-1}(R) = a + bX \ .$$

Moreover, $\text{MAP}(z) = \text{ADD}\big(1, R, f(R), h)\big)$ implies

$$\tau(z) = \tau\big(\tau^{-1}(R) + t \cdot \tau^{-1}(h)\big) \text{ with } \ t := f(R) \neq 0 \ .$$

Since τ is injective and $\tau(X) = h$, this implies

$$\tau^{-1}(R) = z - tX \ ,$$

thus $a = z$ and $b = -t \neq 0$. Consider the point when R is added to the range of τ. Any "fresh" input R to ADD will be associated to a constant polynomial in line 16. Since $\tau^{-1}(R)$ is non-constant, R must have been added to τ during a call to MAP in line 10. Moreover this call to MAP must have been made by the experiment in lines 3 or 17, since the adversary can only call MAP on constant polynomials. When MAP is called on some fresh $i = a+bX$, the value R is picked uniformly and independently of a and b. The probability that any R satisfies $f(R) = -b$ is thus bounded by $\mathsf{MaxSize}_f/p$. As the adversary can create at most q values this way, and moreover we could have $f(h) = -1$, the probability that the adversary wins game symb-cdl is bounded by $(q+1) \cdot \mathsf{MaxSize}_f/p$.

Adding to this the differences between the previous games yields the bound of the theorem. □

5 CDL in the Algebraic Bijective RO Model

THE ALGEBRAIC BIJECTIVE RANDOM ORACLE MODEL. The *Algebraic Bijective Random Oracle Model* (ABROM) [QCY21] is a combination of the bijective random oracle (BRO) model [FKP16] and the algebraic group model (AGM) [FKL18], originally introduced to analyze blind ECDSA. It idealizes the ECDSA conversion function $f : \mathsf{G}^* \to \mathbb{Z}_p$ in a similar manner as the BRO, by decomposing f into three independent functions $f := \psi \circ \Pi \circ \varphi$, where φ maps from G^* to $\mathbb{A} := \{0,1\}^L$, Π maps from $\mathbb{A}$ to $\mathbb{B} := [2^L - 1]$, and ψ maps from $\mathbb{B}$ to $\mathbb{Z}_p$. Functions φ and ψ are standard-model (non-idealized), while Π is modeled as a bijective random oracle. The forward direction Π and its inverse Π^{-1} are accessible via the BRO and BRO^{-1} oracles, respectively. The conditions imposed on φ, Π, ψ in our result are meant to ensure f preserves the essential structure of the ECDSA conversion function such as invertibility and being 2-to-1.

Like the AGM, the ABROM keeps track of all "seen" group elements in a vector $\overrightarrow{U}$. We make a slight tweak to the model such that the domain of f and φ

is G instead of G^*, to better represent the CDL conversion function. The BRO oracle now takes as a vector representation $\overrightarrow{p}$ where we let $R = \prod_i U_i^{p_i}$ and outputs some $\beta \in \mathbb{B}$ corresponding to $\Pi(\varphi(R))$. In other words, to call Π on R, the adversary needs to provide a representation for some preimage R of α under φ. This preimage of α under φ is then added to $\overrightarrow{U}$ The BRO^{-1} oracle takes as input some $\beta \in \mathbb{B}$ and outputs some $\alpha \in \mathbb{A}$ corresponding to $\Pi^{-1}(\beta)$. Unlike the AGM, we do *not* require the adversary to give representations for group elements it outputs. Formally, for an adversary A, the CDL game in the ABROM is defined in Figure 6 as G_0. We let A's advantage be $\mathbf{Adv}^{\text{abro-cdl}}_{\mathsf{G},g,\varphi,\psi}(\mathrm{A}) = \Pr[\mathrm{G}_0 \Rightarrow 1]$.

We recall the following definition from [FKP16] before stating our result.

Definition 4. *(Semi-Injective Function) Let* G *be a prime order group* p *and* $\mathbb{A}$ *be a set. A function* $\varphi : \mathsf{G} \to \mathbb{A}$ *is called* semi-injective *if (a) its range* $\varphi(\mathsf{G}) \subseteq \mathbb{A}$ *is efficiently decidable and (b) it is either injective or 2-to-1 with* $\varphi(X) = \varphi(Y)$ *always implying* $Y \in \{X, X^{-1}\}$.

Theorem 3 *Let* G *be a group of prime order* p*. Let* A *be a CDL-adversary in the ABROM making at most* q *queries to its oracles. Let* $\mathbb{A} = \{0,1\}^L$ *and* $\mathbb{B} = [2^L - 1]$ *such that* $2^L \geq p$*. Let* $\varphi\colon \mathsf{G} \to \mathbb{A}$ *be semi-injective, and let* $\psi\colon \mathbb{B} \to \mathbb{Z}_p$ *be arbitrary. Assume* ϕ, ψ *are efficiently computable. Then there exists a DL-adversary* B *(shown in Figure 8) with running time roughly the same as* A *plus simulation overhead proportional to* q *such that*

$$\mathbf{Adv}^{\text{abro-cdl}}_{\mathsf{G},g,\varphi,\psi}(\mathrm{A}) \leq \mathbf{Adv}^{\text{dl}}_{\mathsf{G},g}(\mathrm{B}) + \frac{(q+1) \cdot \mathsf{MaxSize}_\psi + q^2 + 2q}{p} \,. \tag{6}$$

Proof. We show the result through a sequence of game hops shown in Figures 6 and 7. and proceed to analyze their differences.

$\underline{\mathrm{G}_0 \to \mathrm{G}_1.}$ In G_1, we lazily sample Π and introduce abort conditions in lines 11, 16, and 24. Note that the oracle responses in G_0 are identically distributed to the oracle responses in G_1 if G_1 does not abort. Thus, $\Pr[G_0^{\mathrm{A}} \Rightarrow 1] \leq \Pr[G_1^{\mathrm{A}} \Rightarrow 1] + \Pr[G_1^{\mathrm{A}} \text{ aborts}]$. We now analyze the probability that G_1 aborts. On the i-th query (to either BRO or BRO^{-1}), the probability of aborting is at most $(i-1)/2^L$, and the probability of aborting in line 24 is $q/2^L$. Thus,

$$\Pr[\mathrm{G}_1^{\mathrm{A}} \text{ aborts}] = \sum_{i=1}^{q} \frac{i}{2^L} = \frac{q(q+1)}{2^{L+1}} \,.$$

$\underline{\mathrm{G}_1 \to \mathrm{G}_2.}$ In G_2, in answering a BRO^{-1} query we resample α if the originally sampled α is in the range of φ, so that this time we learn the representation. We set $R := g^{s'} h^{t'}$ where s' and t' are uniformly sampled from $\mathbb{Z}$. Since g and h are generators, this results in a uniformly random R. We then set $\alpha \leftarrow \varphi(R)$. Because φ is either injective or 2-to-1, any $\alpha \in \varphi(\mathsf{G})$ is equally likely to be chosen, so this method of resampling maintains the uniform distribution on α. Thus, $\Pr[G_1^{\mathrm{A}} \Rightarrow 1] = \Pr[G_2^{\mathrm{A}} \Rightarrow 1]$.

Game $\mathrm{G}_0(\mathsf{G}, g, \varphi, \psi)$

INITIALIZE:
1 $\Pi \leftarrow\!\!\$\ \mathrm{Inj}(\mathbb{A}, \mathbb{B})$
2 $x \leftarrow\!\!\$\ \mathbb{Z}_p^*$
3 $h \leftarrow g^x$
4 $\overrightarrow{U} \leftarrow (g, h)$
5 Return h

BRO($\overrightarrow{p}$):
6 $R \leftarrow \prod_i U_i^{p_i}$
7 $\overrightarrow{U} \leftarrow \overrightarrow{U} \| R$
8 $\alpha \leftarrow \varphi(R)$
9 $\beta \leftarrow \Pi(\alpha)$
10 Return β

BRO$^{-1}(\beta)$:
11 $\alpha \leftarrow \Pi^{-1}(\beta)$
12 If $\alpha \in \varphi(\mathsf{G})$:
13 $\quad (R, R^{-1}) \leftarrow \varphi^{-1}(\alpha)$
14 $\quad \overrightarrow{U} \leftarrow \overrightarrow{U} \| R \| R^{-1}$
15 Return α

FINALIZE(R, z):
16 $\alpha \leftarrow \varphi(R)$
17 $\beta \leftarrow \Pi(\alpha)$
18 $c \leftarrow \psi(\beta)$
19 Return $(g^z = R h^{f(R)})$

Game $\mathrm{G}_1(\mathsf{G}, g, \varphi, \psi)$

INITIALIZE:
1 $\Pi \leftarrow \emptyset$
2 $x \leftarrow\!\!\$\ \mathbb{Z}_p^*$
3 $h \leftarrow g^x$
4 $\overrightarrow{U} \leftarrow (g, h)$
5 Return h

BRO($\overrightarrow{p}$):
6 $R \leftarrow \prod_i U_i^{p_i}$
7 $\overrightarrow{U} \leftarrow \overrightarrow{U} \| R$
8 $\alpha \leftarrow \varphi(R)$
9 If $(\alpha, \cdot) \in \Pi$: Return $\Pi(\alpha)$
10 $\beta \leftarrow\!\!\$\ \mathbb{B}$
11 If $(\cdot, \beta) \in \Pi$: Abort
12 $\Pi \leftarrow \Pi \cup \{(\alpha, \beta)\}$
13 Return β

BRO$^{-1}(\beta)$:
14 If $(\cdot, \beta) \in \Pi$: Return $\Pi^{-1}(\beta)$
15 $\alpha \leftarrow\!\!\$\ \mathbb{A}$
16 If $(\alpha, \cdot) \in \Pi$: Abort
17 If $\alpha \in \varphi(\mathsf{G})$:
18 $\quad (R, R^{-1}) \leftarrow \varphi^{-1}(\alpha)$
19 $\quad \overrightarrow{U} \leftarrow \overrightarrow{U} \| R \| R^{-1}$
20 $\Pi \leftarrow \Pi \cup \{(\alpha, \beta)\}$
21 Return α

FINALIZE(R, z):
22 $\alpha \leftarrow \varphi(R)$
23 If $(\alpha, \cdot) \in \Pi$ then $\beta \leftarrow \Pi(\alpha)$
24 Else $\beta \leftarrow\!\!\$\ \mathbb{B}$; If $(\cdot, \beta) \in \Pi$: Abort
25 $c \leftarrow \psi(\beta)$
26 Return $(g^z = R h^c)$

Fig. 6: Games G_0, G_1 for the proof of Theorem 3. Changes are highlighted in blue.

$\mathrm{G}_2 \to \mathrm{G}_3$. In G_3, the first change we make is to use vectors $\overrightarrow{s}$ and $\overrightarrow{t}$ to keep track of each "seen" element U_i, so that $U_i = g^{s_i} h^{t_i}$. This change is purely for bookkeeping and does not affect the behavior of the oracles. The second change is that we add abort conditions in lines 15 and 23, which we will use later.

We now analyze the probability of G_3 aborting. On each BRO query, the probability of aborting in a given execution of line 15 is at most $\mathsf{MaxSize}_\psi / p$ since β is uniformly sampled. Thus, by union bound over all the queries, the probability of aborting in line 15 during the execution of the entire game is at most $q \cdot \mathsf{MaxSize}_\psi / p$. The probability of aborting in line 23 is at most q/p since

Game $G_2(\mathsf{G}, g, \varphi, \psi)$

INITIALIZE:
$\Pi \leftarrow \emptyset$
$x \leftarrow\!\!\$\ \mathbb{Z}_p^*$
$h \leftarrow g^x$
$\overrightarrow{U} \leftarrow (g, h)$
Return h

BRO($\overrightarrow{p}$):
$R \leftarrow \prod_i U_i^{p_i}$
$\overrightarrow{U} \leftarrow \overrightarrow{U} \| R \| R^{-1}$
$\alpha \leftarrow \varphi(R)$
If $(\alpha, \cdot) \in \Pi$: Return $\Pi(\alpha)$
$\beta \leftarrow\!\!\$\ \mathbb{B}$
If $(\cdot, \beta) \in \Pi$: Abort
$\Pi \leftarrow \Pi \cup \{(\alpha, \beta)\}$
Return β

BRO$^{-1}(\beta)$:
If $(\cdot, \beta) \in \Pi$: Return $\Pi^{-1}(\beta)$
$\alpha \leftarrow\!\!\$\ \mathbb{A}$
If $\alpha \in \varphi(\mathsf{G})$:
  $s', t' \leftarrow\!\!\$\ \mathbb{Z}_p$
  $R \leftarrow g^{s'} h^{t'}$
  $\alpha \leftarrow \varphi(R)$
  $\overrightarrow{U} \leftarrow \overrightarrow{U} \| R \| R^{-1}$
If $(\alpha, \cdot) \in \Pi$: Abort
$\Pi \leftarrow \Pi \cup \{(\alpha, \beta)\}$
Return α

FINALIZE(R, z):
$\alpha \leftarrow \varphi(R)$
If $(\alpha, \cdot) \in \Pi$ then $\beta \leftarrow \Pi(\alpha)$
Else $\beta \leftarrow\!\!\$\ \mathbb{B}$; If $(\cdot, \beta) \in \Pi$: Abort
$c \leftarrow \psi(\beta)$
Return $(g^z = Rh^c)$

Game G_3/$\boxed{G_4}(\mathsf{G}, g, \varphi, \psi)$

INITIALIZE:
$\Pi \leftarrow \emptyset$
$x \leftarrow\!\!\$\ \mathbb{Z}_p^*$
$h \leftarrow g^x$
$\overrightarrow{U} \leftarrow (g, h)$
$\overrightarrow{s} \leftarrow (1, 0)$; $\overrightarrow{t} \leftarrow (0, 1)$ // $U_i = g^{s_i} h^{t_i}$
Return h

BRO($\overrightarrow{p}$):
$R \leftarrow \prod_i U_i^{p_i}$
$\overrightarrow{U} \leftarrow \overrightarrow{U} \| R \| R^{-1}$
$\alpha \leftarrow \varphi(R)$
If $(\alpha, \cdot) \in \Pi$: Return $\Pi(\alpha)$
$s' \leftarrow \langle \overrightarrow{s}, \overrightarrow{p} \rangle$; $t' \leftarrow \langle \overrightarrow{t}, \overrightarrow{p} \rangle$ //$R = g^{s'} h^{t'}$
$\overrightarrow{s} \leftarrow \overrightarrow{s} \| s' \| -s'$
$\overrightarrow{t} \leftarrow \overrightarrow{t} \| t' \| -t'$
$\beta \leftarrow\!\!\$\ \mathbb{B}$
If $-t' = \psi(\beta)$: Abort
If $(\cdot, \beta) \in \Pi$: Abort
$\Pi \leftarrow \Pi \cup \{(\alpha, \beta)\}$
Return β

BRO$^{-1}(\beta)$:
If $(\cdot, \beta) \in \Pi$: Return $\Pi^{-1}(\beta)$
$\alpha \leftarrow\!\!\$\ \mathbb{A}$
If $\alpha \in \varphi(\mathsf{G})$:
  $s', t' \leftarrow\!\!\$\ \mathbb{Z}_p$
  If $-t' = \psi(\beta)$: Abort
  $\overrightarrow{s} \leftarrow \overrightarrow{s} \| s' \| -s'$
  $\overrightarrow{t} \leftarrow \overrightarrow{t} \| t' \| -t'$
  $R \leftarrow g^{s'} h^{t'}$
  $\alpha \leftarrow \varphi(R)$
  $\overrightarrow{U} \leftarrow \overrightarrow{U} \| R \| R^{-1}$
If $(\alpha, \cdot) \in \Pi$: Abort
$\Pi \leftarrow \Pi \cup \{(\alpha, \beta)\}$
Return α

FINALIZE(R, z):
$\alpha \leftarrow \varphi(R)$
If $(\alpha, \cdot) \in \Pi$ then $\beta \leftarrow \Pi(\alpha)$
Else $\beta \leftarrow\!\!\$\ \mathbb{B}$; If $(\cdot, \beta) \in \Pi$: Abort
$c \leftarrow \psi(\beta)$
If $g^z = Rh^c$
  $\boxed{\text{If } R \notin \overrightarrow{U}\text{: Abort}}$
  Return 1
Else return 0

Fig. 7: Games $G_2 - G_4$ for the proof of Theorem 3. Changes are highlighted in blue.

t' is uniformly sampled. Thus,

$$\Pr\left[\,\mathrm{G}_2^{\mathrm{A}} \Rightarrow 1\,\right] \leq \Pr\left[\,\mathrm{G}_3^{\mathrm{A}} \Rightarrow 1\,\right] + q/p + q \cdot \mathsf{MaxSize}_\psi/p\ .$$

$\underline{\mathrm{G}_3 \to \mathrm{G}_4}$. In G_4, we introduce an abort in line 37 if the value R in A's output is not in $\overrightarrow{U}$. We have

$$\Pr\left[\,\mathrm{G}_3^{\mathrm{A}} \Rightarrow 1\,\right] \leq \Pr\left[\,\mathrm{G}_4^{\mathrm{A}} \Rightarrow 1\,\right] + \Pr\left[\,\mathrm{G}_4^{\mathrm{A}} \text{ aborts}\,\right]\ .$$

We claim that $\Pr\left[\,\mathrm{G}_4^{\mathrm{A}} \text{ aborts}\,\right] \leq \mathsf{MaxSize}_\psi/p$. First, note that if $(\alpha, \cdot) \in \Pi$ in line 33 then the abort in line 37 won't be executed. This is because the BRO and BRO^{-1} procedures always add preimages of α under φ to $\overrightarrow{U}$. Thus, if the game aborts in line 37, the value of α will be "fresh," meaning $(\alpha, \cdot) \notin \Pi$. In this case, $\beta \in \mathbb{B}$ is uniform and independent of A's output (R, z). Thus, over the choice of β, the probability that the if-check in line 37 evaluates to true is at most $\mathsf{MaxSize}_\psi/p$ because there is a unique c satisfying the equation.

$\underline{\text{The DL-Adversary.}}$ Finally, we show that there is a DL-adversary B (given in Figure 8) such that

$$\mathbf{Adv}^{\mathrm{G}_4}_{\mathsf{G},g,\varphi,\psi}(\mathrm{A}) \leq \mathbf{Adv}^{\mathrm{dl}}_{\mathsf{G},g}(\mathrm{B})\ , \tag{7}$$

which combined with the above completes the proof. Namely, we show that on any run of G_4 which outputs 1, B returns the discrete log of h. Note that G_4 outputs 1 iff A returns a valid (R, z) pair and $R \in \overrightarrow{U}$. So, there exists some i such that $R = g^{s_i} h^{t_i}$. Thus, $g^z h^{-f(R)} = g^{s_i} h^{t_i}$. Moreover, $t_i + f(R)$ is non-zero as assured by the abort conditions in lines 15 and 23, so $x := (z - s_i)/(t_i + f(R))$ is indeed the discrete log of h.

□

Acknowledgments. We thank Yilei Chen for collaboration in the early stages of this work and Dan Brown for helpful feedback. The second author was funded by the Vienna Science and Technology Fund (WWTF) [10.47379/VRG18002].

References

AEE+21. Lukas Aumayr, Oguzhan Ersoy, Andreas Erwig, Sebastian Faust, Kristina Hostáková, Matteo Maffei, Pedro Moreno-Sanchez, and Siavash Riahi. Generalized channels from limited blockchain scripts and adaptor signatures. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part II*, volume 13091 of *LNCS*, pages 635–664. Springer, Heidelberg, December 2021. 7

BD20. Mihir Bellare and Wei Dai. The multi-base discrete logarithm problem: Tight reductions and non-rewinding proofs for Schnorr identification and signatures. In Karthikeyan Bhargavan, Elisabeth Oswald, and Manoj Prabhakaran, editors, *INDOCRYPT 2020*, volume 12578 of *LNCS*, pages 529–552. Springer, Heidelberg, December 2020. 2, 6

Adversary $\mathrm{B}(h)$

$\Pi \leftarrow \emptyset$
$\overrightarrow{U} \leftarrow (g, h)$
$\overrightarrow{s} \leftarrow (1, 0)$; $\overrightarrow{t} \leftarrow (0, 1)$ // $U_i = g^{s_i} h^{t_i}$
$(R, z) \leftarrow\!\!\$\ \mathrm{A}^{\mathrm{BRO},\mathrm{BRO}^{-1}}(\mathsf{G}, g, h)$
If $R \notin \overrightarrow{U}$ return 0
Let i such that $R = U_i$
$x \leftarrow (z - s_i)/(t_i + f(R))$
Return x

$\mathrm{BRO}(R, \overrightarrow{p})$:

$R \leftarrow \prod_i U_i^{p_i}$
$\overrightarrow{U} \leftarrow \overrightarrow{U} \| R$
$\alpha \leftarrow \varphi(R)$
If $(\alpha, \cdot) \in \Pi$: Return $\Pi(\alpha)$
$s' \leftarrow \langle \overrightarrow{s}, \overrightarrow{p} \rangle$; $t' \leftarrow \langle \overrightarrow{t}, \overrightarrow{p} \rangle$ //$R = g^{s'} h^{t'}$
$\overrightarrow{s} \leftarrow \overrightarrow{s} \,\|\, s' \| - s'$
$\overrightarrow{t} \leftarrow \overrightarrow{t} \,\|\, t' \| - t'$
$\beta \leftarrow\!\!\$\ \mathbb{B}$
If $(\cdot, \beta) \in \Pi$: Abort
If $-t' = \psi(\beta)$: Abort
$\Pi \leftarrow \Pi \cup \{(\alpha, \beta)\}$
Return β

$\mathrm{BRO}^{-1}(\beta)$:

If $(\cdot, \beta) \in \Pi$: Return $\Pi^{-1}(\beta)$
$\alpha \leftarrow\!\!\$\ \mathbb{A}$
If $\alpha \in \varphi(\mathsf{G})$:
  $s', t' \leftarrow\!\!\$\ \mathbb{Z}_p$
  If $-t' = \psi(\beta)$: Abort
  $\overrightarrow{s} \leftarrow \overrightarrow{s} \,\|\, s' \,\|\, -s'$
  $\overrightarrow{t} \leftarrow \overrightarrow{t} \,\|\, t' \,\|\, -t'$
  $R \leftarrow g^{s'} h^{t'}$
  $\alpha \leftarrow \varphi(R)$
  $\overrightarrow{U} \leftarrow \overrightarrow{U} \| R \| R^{-1}$
If $(\alpha, \cdot) \in \Pi$: Abort
$\Pi \leftarrow \Pi \cup \{(\alpha, \beta)\}$
Return α

Fig. 8: DL-adversary B for the proof of Theorem 3.

BDL+12. Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-speed high-security signatures. *Journal of Cryptographic Engineering*, 2(2):77–89, September 2012. 1

BM14. Christina Brzuska and Arno Mittelbach. Using indistinguishability obfuscation via UCEs. In Palash Sarkar and Tetsu Iwata, editors, *ASI-*

ACRYPT 2014, Part II, volume 8874 of *LNCS*, pages 122–141. Springer, Heidelberg, December 2014. 7

BNPS03. Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. *Journal of Cryptology*, 16(3):185–215, June 2003. 2

BR93. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, November 1993. 1

BR06. Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, Heidelberg, May / June 2006. 8, 11

Bro02. Daniel R. L. Brown. Generic groups, collision resistance, and ECDSA. Cryptology ePrint Archive, Report 2002/026, 2002. https://eprint.iacr.org/2002/026. 5

Bro05. Daniel R.L. Brown. On the provable security of ECDSA. In Ian F. Blake, Gadiel Seroussi, and Nigel P Smart, editors, *Advances in Elliptic Curve Cryptography*, London Mathematical Society Lecture Note Series, pages 21–40. Cambridge University Press, 2005. 5

CGKN21. Konstantinos Chalkias, François Garillot, Yashvanth Kondi, and Valeria Nikolaenko. Non-interactive half-aggregation of eddsa and variants of schnorr signatures. In Kenneth G. Paterson, editor, *Topics in Cryptology - CT-RSA 2021 - Cryptographers' Track at the RSA Conference 2021, Virtual Event, May 17-20, 2021, Proceedings*, volume 12704 of *Lecture Notes in Computer Science*, pages 577–608. Springer, 2021. 7

CLMQ21. Yilei Chen, Alex Lombardi, Fermi Ma, and Willy Quach. Does fiat-shamir require a cryptographic hash function? In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 334–363, Virtual Event, August 2021. Springer, Heidelberg. 2

Den02. Alexander W. Dent. Adapting the weaknesses of the random oracle model to the generic group model. In Yuliang Zheng, editor, *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 100–109. Springer, Heidelberg, December 2002. 2

FJS19. Nils Fleischhacker, Tibor Jager, and Dominique Schröder. On tight security proofs for Schnorr signatures. *Journal of Cryptology*, 32(2):566–599, April 2019. 2, 6

FKL18. Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 33–62. Springer, Heidelberg, August 2018. 2, 16

FKP16. Manuel Fersch, Eike Kiltz, and Bertram Poettering. On the provable security of (EC)DSA signatures. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 1651–1662. ACM Press, October 2016. 5, 16, 17

FKP17. Manuel Fersch, Eike Kiltz, and Bertram Poettering. On the one-per-message unforgeability of (EC)DSA and its variants. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part II*, volume 10678 of *LNCS*, pages 519–534. Springer, Heidelberg, November 2017. 5

FPS20. Georg Fuchsbauer, Antoine Plouviez, and Yannick Seurin. Blind schnorr signatures and signed ElGamal encryption in the algebraic group model. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 63–95. Springer, Heidelberg, May 2020. 2

FS87. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, August 1987. 9

FW24. Georg Fuchsbauer and Mathias Wolf. Concurrently secure blind schnorr signatures. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 124–160. Springer, 2024. 7

GBL08. Sanjam Garg, Raghav Bhaskar, and Satyanarayana V. Lokam. Improved bounds on security reductions for discrete log based signatures. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 93–107. Springer, Heidelberg, August 2008. 2, 6

GJO16. Vipul Goyal, Aayush Jain, and Adam O'Neill. Multi-input functional encryption with unbounded-message security. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 531–556. Springer, Heidelberg, December 2016. 7

GS22. Jens Groth and Victor Shoup. On the security of ECDSA with additive key derivation and presignatures. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part I*, volume 13275 of *LNCS*, pages 365–396. Springer, Heidelberg, May / June 2022. 4, 5, 13

HK23. Dominik Hartmann and Eike Kiltz. Limits in the provable security of ECDSA signatures. In Guy N. Rothblum and Hoeteck Wee, editors, *TCC 2023, Part IV*, volume 14372 of *LNCS*, pages 279–309. Springer, Heidelberg, November / December 2023. 5

KG24. Chelsea Komlo and Ian Goldberg. Arctic: Lightweight and stateless threshold schnorr signatures. *IACR Cryptol. ePrint Arch.*, page 466, 2024. 7

KMP16. Eike Kiltz, Daniel Masny, and Jiaxin Pan. Optimal security proofs for signatures from identification schemes. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 33–61. Springer, Heidelberg, August 2016. 6

MOZ22. Alice Murphy, Adam O'Neill, and Mohammad Zaheri. Instantiability of classical random-oracle-model encryption transforms. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part IV*, volume 13794 of *LNCS*, pages 323–352. Springer, Heidelberg, December 2022. 7

Nao03. Moni Naor. On cryptographic assumptions and challenges (invited talk). In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 96–109. Springer, Heidelberg, August 2003. 2, 7

Nec94. V. I. Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55(2):165–172, 1994. 2, 5

NRS21. Jonas Nick, Tim Ruffing, and Yannick Seurin. MuSig2: Simple two-round Schnorr multi-signatures. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 189–221, Virtual Event, August 2021. Springer, Heidelberg. 7

NSW09. Gregory Neven, Nigel P. Smart, and Bogdan Warinschi. Hash function requirements for schnorr signatures. *J. Math. Cryptol.*, 3(1):69–87, 2009. 2

oST13. National Institute of Standards and Technology. Digital signature standard (dss). fips 186-4. Tech. rep., U.S. Department of Commerce, 2013. 3, 4, 5, 14

PS96. David Pointcheval and Jacques Stern. Security proofs for signature schemes. In Ueli M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 387–398. Springer, Heidelberg, May 1996. 1, 4, 6

PV05. Pascal Paillier and Damien Vergnaud. Discrete-log-based signatures may not be equivalent to discrete log. In Bimal K. Roy, editor, *ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 1–20. Springer, Heidelberg, December 2005. 2, 6

QCY21. Xianrui Qin, Cailing Cai, and Tsz Hon Yuen. One-more unforgeability of blind ECDSA. In Elisa Bertino, Haya Shulman, and Michael Waidner, editors, *ESORICS 2021, Part II*, volume 12973 of *LNCS*, pages 313–331. Springer, Heidelberg, October 2021. 6, 16

Rog06. Phillip Rogaway. Formalizing human ignorance. In Phong Q. Nguyen, editor, *Progress in Cryptology - VIETCRYPT 06*, volume 4341 of *LNCS*, pages 211–228. Springer, Heidelberg, September 2006. 10

Sch90. Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 239–252. Springer, Heidelberg, August 1990. 1, 9

Seu12. Yannick Seurin. On the exact security of Schnorr-type signatures in the random oracle model. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 554–571. Springer, Heidelberg, April 2012. 2, 6

Sho97. Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, Heidelberg, May 1997. 2, 5, 13

Sho23. Victor Shoup. The many faces of schnorr. Cryptology ePrint Archive, Paper 2023/1019, 2023. https://eprint.iacr.org/2023/1019. 2

WNR20. Pieter Wuille, Jonas Nick, and Tim Ruffing. Schnorr signatures for secp256k1. Bitcoin Improvement Proposal, 2020. See https://github.com/bitcoin/bips/blob/master/bip-0340.mediawiki. 1

YEL+21. Tsz Hon Yuen, Muhammed F. Esgin, Joseph K. Liu, Man Ho Au, and Zhimin Ding. DualRing: Generic construction of ring signatures with efficient instantiations. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 251–281, Virtual Event, August 2021. Springer, Heidelberg. 7

Zha22. Mark Zhandry. To label, or not to label (in generic groups). In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part III*, volume 13509 of *LNCS*, pages 66–96. Springer, Heidelberg, August 2022. 2