A preliminary version of this paper appears in the proceedings of INDOCRYPT 2020. This is the full version.

The Multi-Base Discrete Logarithm Problem: Tight Reductions and Non-Rewinding Proofs for Schnorr Identification and Signatures

Mihir Bellare[1] Wei Dai[2]

October 2020

Abstract

We introduce the Multi-Base Discrete Logarithm (MBDL) problem. We use this to give reductions, for Schnorr and Okamoto identification and signatures, that are non-rewinding and, by avoiding the notorious square-root loss, tighter than the classical ones from the Discrete Logarithm (DL) problem. This fills a well-known theoretical and practical gap regarding the security of these schemes. We show that not only is the MBDL problem hard in the generic group model, but with a bound that matches that for DL, so that our new reductions justify the security of these primitives for group sizes in actual use.

[1] Department of Computer Science & Engineering, University of California, San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: mihir@eng.ucsd.edu. URL: http://cseweb.ucsd.edu/~mihir/. Supported in part by NSF grant CNS-1717640 and a gift from Microsoft.

[2] Department of Computer Science & Engineering, University of California, San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: weidai@eng.ucsd.edu. URL: http://cseweb.ucsd.edu/~weidai/. Supported in part by a Powell Fellowship and grants of first author.

Contents

1 Introduction

It would not be an exaggeration to say that Schnorr identification and signatures [45] are amongst the best-known and most influential schemes in cryptography. With regard to practical impact, consider that Ed25519, a Schnorr-derived signature scheme over twisted Edwards curves [12], is used, according to IANIX [34], in over 200 different things. (OpenSSL, OpenSSH and GnuPG to name a tiny fraction.) Meanwhile the algebraic structure of the Schnorr schemes has resulted in their being the basis for many advanced primitives including multi- [39, 6, 3, 38], ring- [2, 32] and threshold- [50, 37] signatures.

Proving security of these schemes has accordingly attracted much work. Yet, all known standard-model proofs [44, 1, 36] exhibit a gap: the proven bound on adversary advantage (success probability) is much inferior to (larger than) the one that cryptanalysis says is "true." (The former is roughly the square-root of the latter. Accordingly we will refer to this as the square-root gap.)

The square-root gap is well known and acknowledged in the literature. Filling this long-standing and notorious gap between theory and practice is the subject of this paper. We start with some background.

SCHNORR SCHEMES. Let $\mathbb{G}$ be a group of prime order p, and $g \in \mathbb{G}$ a generator of $\mathbb{G}$. We let $\mathsf{ID} = \mathsf{SchID}[\mathbb{G}, g]$ denote the Schnorr identification scheme [45] (shown in Figure 4). The security goal for it is IMP-PA (impersonation under passive attack [24]). The Schnorr signature scheme $\mathsf{DS} = \mathsf{SchSig}[\mathbb{G}, g]$ [45] is derived from ID via the Fiat-Shamir transform [25] (also shown in Figure 4). The security goal for it is UF (unforgeability under chosen-message attack [30]) in the ROM (random oracle model [10]).

Recall that, $\mathbb{G}, g$ being public, the DL problem is for an adversary, given $X = g^x$, to recover x. Since we will introduce variants, we may, for emphasis, refer to DL itself as the "basic" version of the discrete-logarithm problem. Existing standard-model proofs for both ID and DS [44, 1, 36] are based on the assumed hardness of DL. The heart of the proof for DS, and the cause of the square-root gap, is the rewinding reduction in the proof for ID. This makes ID the first and most basic concern.

THE SITUATION WITH ID. The simplest proof of IMP-PA for $\mathsf{ID} = \mathsf{SchID}[\mathbb{G}, g]$ uses the Reset Lemma of [7]. It shows that, roughly:

$$\epsilon^{\text{imp-pa}}(t) \leq \sqrt{\epsilon^{\text{dl}}(t)} + \frac{1}{p} \,, \tag{1}$$

where $\epsilon^{\text{imp-pa}}(t)$ is the probability of breaking IMP-PA security of ID in time t and $\epsilon^{\text{dl}}(t)$ is the probability of breaking DL in time t. To draw quantitative conclusions about $\epsilon^{\text{imp-pa}}(t)$ as required in practice, however, we now also need to estimate $\epsilon^{\text{dl}}(t)$. The accepted way to do this is via the Generic Group Model (GGM) bound [47], believed to be accurate for elliptic curve groups. It says that

$$\epsilon^{\text{dl}}(t) \approx \frac{t^2}{p} \,. \tag{2}$$

Putting together the two equations above, we get, roughly:

$$\epsilon^{\text{imp-pa}}(t) \leq \frac{t}{\sqrt{p}} \,. \tag{3}$$

There is, however, no known attack matching the bound of Eq. (3). Indeed, the best known time t attack on ID is via discrete-log computation and thus has the considerably lower success probability of t^2/p. For example if $p \approx 2^{256}$ the best known attack against ID gives a time $t = 2^{80}$ attacker a success probability of $t^2/p = 2^{-96}$, but Eq. (3) only rules out a success probability of $t/\sqrt{p} = 2^{-48}$.

The proof is thus under-estimating security by a fairly large margin.

Accordingly in practice the proof is largely viewed as a qualitative rather than quantitative guarantee, group sizes being chosen in ad hoc ways. Improving the reduction of Eq. (1) to bring the theory more in line with the indications of cryptanalysis has been a long-standing open question.

TIERS AND KNOBS. Before continuing with how we address this question, we draw attention to the two-tiered framework of a security proof for a scheme S (above, $\mathsf{S} = \mathsf{ID}$) based on the assumed hardness of some problem P (above, P=DL). The first tier is the reduction from P. It is represented above by Eq. (1). The second tier is the estimate of the security of P itself, made (usually) in an idealized model such as the GGM [47] or AGM (Algebraic Group Model) [27]. It is represented above by Eq. (2). Both tiers are necessary to draw quantitative conclusions. This two-tier structure is an accepted one for security proofs, and widely, even if not always explicitly, used.

In this structure, we have the flexibility of choice of P, making this a "knob" that we can tune. Creative and new choices of P have historically been made, and been very valuable in cryptography, yielding proofs for existing schemes and then going on to be useful beyond. Historically, a classical example of such a (at the time, new) P is the Diffie-Hellman problem; the schemes S whose proof this allows include the Diffie-Hellman secret-key exchange [21] and the El Gamal public-key encryption scheme [23]. An example P closer to our work is the One-More Discrete Logarithm (OMDL) problem [5], which has by now been used to prove many schemes S [7, 20, 43, 26, 22]. But this knob-tuning approach is perhaps most visible in the area of bilinear maps, where many choices of problem P have been made, justified in the GGM, and then used to prove security of many schemes S. In the same tradition, we ask, how can we tune the knob to fill the square-root gap? Our answer is a choice of P we call MBDL.

MBDL. Our Multi-Base Discrete Logarithm (MBDL) problem is a variant of the One-More Discrete Logarithm (OMDL) problem of [5]. Continue to fix a cyclic group $\mathbb{G}$ and generator g of $\mathbb{G}$. In MBDL, the adversary is given a challenge $Y \in \mathbb{G}$, a list $X_1, X_2, \ldots, X_n \in \mathbb{G}^*$ of generators of $\mathbb{G}$, and access to an oracle DLO that, on input i, W, returns $\mathsf{DL}_{\mathbb{G},X_i}(W)$, the discrete logarithm of W, *not in base* g, *but in base* X_i. To win it must find $\mathsf{DL}_{\mathbb{G},g}(Y)$, the discrete logarithm of the challenge Y to base g, *while making at most one call to* DLO *overall*, meaning it is allowed to take the discrete log of at most one group element. (But this element, and the base X_i, can be chosen as it wishes.) The number of bases n is a parameter of the problem, so that one can refer to the n-MBDL problem or assumption. (Our results will rely only on 1-MBDL, but we keep the definition general for possible future applications.) The restriction to at most one DLO call is necessary, for if even two are allowed, $\mathsf{DL}_{\mathbb{G},g}(Y)$ can be obtained as $\mathrm{DLO}(1, Y) \cdot \mathrm{DLO}(1, g)^{-1} \bmod p$ where $p = |\mathbb{G}|$.

CORE RESULTS. We suggest that the square-root gap of Eq. (1) is a manifestation of an unformalized strength of the discrete logarithm problem. We show that this strength is captured by the MBDL problem. We do this by giving a proof of IMP-PA security of the Schnorr identification scheme $\mathsf{ID} = \mathsf{SchID}[\mathbb{G}, g]$ with a *tight* reduction from 1-MBDL: letting $\epsilon^{\text{1-mbdl}}(t)$ be the probability of breaking the 1-MBDL problem in time t, Theorem 4.1 says that, roughly:

$$\epsilon^{\text{imp-pa}}(t) \leq \epsilon^{\text{1-mbdl}}(t) + \frac{1}{p} \,. \tag{4}$$

Eq. (4) does not suffer from the square-root gap of Eq. (1). Progress. But this is in the first of the two tiers discussed above. Turning to the second, we ask, how hard is MBDL? Theorem 5.1 shows that, in the GGM, roughly:

$$\epsilon^{\text{1-mbdl}}(t) \approx \frac{t^2}{p} \,. \tag{5}$$

That is, 1-MBDL problem has essentially the same GGM quantitative hardness as DL. Putting

Schnorr Identification				
t	ϵ	$\log(p_1)$	$\log(p_2)$	Speedup $s = (\log(p_1)/\log(p_2))^3$
2^{80}	2^{-48}	256	208	1.9
2^{64}	2^{-64}	256	192	2.4
2^{100}	2^{-156}	512	356	3

Schnorr Signatures					
t	q_h	ϵ	$\log(p_1)$	$\log(p_2)$	Speedup $s = (\log(p_1)/\log(p_2))^3$
2^{80}	2^{60}	2^{-48}	316	268	1.6
2^{64}	2^{50}	2^{-64}	306	242	2.0
2^{100}	2^{80}	2^{-156}	592	436	2.5

Figure 1: **Speedups yielded by our results for the Schnorr identification scheme** $\mathsf{ID} = \mathsf{SchID}[\mathbb{G}, g]$ **(top) and signature scheme** $\mathsf{DS} = \mathsf{SchSig}[\mathbb{G}, g]$ **(bottom).** The target for the first is that IMP-PA adversaries with running time t should have advantage at most ϵ. We show the log of the group size p_i required for this under prior results ($i = 1$), and our results ($i = 2$). Assuming exponentiation in $\mathbb{G}$ is cubic-time, we then show the speedup ratio of scheme algorithms. The target for the second is that UF adversaries with running time t, making q_h queries to H, should have advantage at most ϵ, and the table entries are analogous.

Eqs. (4) and (5) together, we get (roughly) the following improvement over Eq. (3):

$$\epsilon^{\text{imp-pa}}(t) \leq \frac{t^2}{p} \ . \tag{6}$$

This bound is tight in the sense that it matches the indications of cryptanalysis.

A direct indication of the practical value of this improvement is that, for a given target level of provable security, we increase efficiency. Thus suppose that, for some chosen values of ϵ, t, we want to pick the group $\mathbb{G}$ to ensure $\epsilon^{\text{imp-pa}}(t) \leq \epsilon$. Eq. (6) allows us to use smaller groups than Eq. (3). Since scheme algorithms have running time cubic in the log of $p = |\mathbb{G}|$, this results in a performance improvement. Figure 1 says that this improvement can range from 1.9x to 3x.

What has been gained? A natural question is that our results rely on a new assumption (MBDL), so what has been gained? Indeed, MBDL, as with any new assumption, should be treated with caution. However, it seems that improving Eq. (1) to something like Eq. (4) under the basic DL assumption is out of reach and likely not possible, and thus that, as indicated above, the apparent strength of the Schnorr schemes indicated by cryptanalysis is arising from stronger hardness properties of the discrete log problem not captured in the basic version. We are trying to understand and formalize this hardness via new problems that tightly imply security of the Schnorr primitives.

Of course it would not be hard to introduce *some* problem which allows this. But we believe MBDL, and our results, are "interesting" in this regard, for the following reasons. First, MBDL is not a trivial reformulation of the IMP-PA security of ID, meaning we are not just assuming the square-root problem out of existence. Second, and an indication of the first, is that the proof of the IMP-PA security of ID from MBDL (see "Reduction approach" below) is correspondingly not trivial. Third, the use of MBDL is not confined to Schnorr identification; as we also discuss below under "MBDL as a hub," it already has many further applications and uses, and we imagine even more will arise in the future.

REDUCTION APPROACH. The proof of Eq. (1) uses a rewinding argument that exploits the special soundness property of the Schnorr identification scheme, namely that from two compatible transcripts —this means they are accepting and have the same commitment but different challenges— one can extract the secret key. To find the discrete log, in base g, of a given challenge Y, the discrete log adversary $\mathcal{B}$ plants the challenge as the public key X and performs two, related runs of the given IMP-PA adversary, hoping to get two compatible transcripts, in which case it can extract the secret key and solve its DL instance. The Reset Lemma [7] says it is successful with probability roughly the square of the IMP-PA advantage of $\mathcal{A}$, leading to the square-root in Eq. (1).

Recall that our 1-mbdl adversary $\mathcal{B}$ gets input a challenge Y whose discrete logarithm *in the usual base g it must find*, just like a DL adversary. To get Eq. (4) we must avoid rewinding. The question is how and why the ability to take one discrete logarithm in some random base X_1 helps to do this and get a tight reduction. Our reduction deviates from prior ones by *not* setting Y to the public key. Instead, it sets X_1 to the public key. Then, it performs a *single* execution of the given IMP-PA adversary $\mathcal{A}$, "planting" Y in the communication in such a way that success of $\mathcal{A}$ in impersonating the prover yields $\mathsf{DL}_{\mathbb{G},g}(Y)$. This planting step makes one call to $\mathrm{DLO}(1,\cdot)$, meaning asks for a discrete logarithm in base X_1 of some W that depends on the execution. The full proof is in Section 4.

MBDL AS A HUB. Having identified MBDL, we find that its applicability extends well beyond what is discussed above, making it a hub. Here we briefly discuss further results from MBDL.

The Schnorr signature scheme $\mathsf{DS} = \mathsf{SchSig}[\mathbb{G}, g]$ has a proof of UF-security in the ROM under the basic DL assumption [44, 41, 1, 36]. The bound —recalled in Eq. (15)— continues to exhibit the square-root gap. Theorem 4.3 gives a square-root avoiding reduction from 1-MBDL to fill this gap. Figure 1 shows resulting speedup factors of 1.6x to 2.5x for Schnorr signatures.

Security above refers to the single-user setting. Our results extend to tightly reduce the multi-user IMP-PA security of $\mathsf{SchID}[\mathbb{G}, g]$ to 1-MBDL, and analogously for signatures. This can be shown directly, but is also a consequence of general results of [36].

The situation for the Okamoto identification and signature schemes [42] is analogous to that for Schnorr, meaning the reductions in the current security proofs, from DL, use rewinding and has the square-root loss. In Appendix A we give results for Okamoto that are analogous to our results for Schnorr, meaning reductions from 1-MBDL that avoid the square root.

There's more. In a follow-up work, we also give reductions from MBDL that improve security of the following: (1) Bellare-Neven multi-signatures [6] (2) Abe, Ohkubo, Suzuki 1-out-of-n (ring/group) signatures [2] and (3) Schnorr-based threshold signatures [50].

RELATED WORK. One prior approach to resolving the square-root gap has been to use *only* an idealized model like the GGM [47] or AGM [27]. Thus, Shoup [47] directly showed that $\epsilon^{\text{imp-pa}}(t) \leq t^2/p$ in the GGM. Fuchsbauer, Plouviez and Seurin [28] give, in the AGM, a tight reduction from DL to the UF security of $\mathsf{DS} = \mathsf{SchSig}[\mathbb{G}, g]$. These results correspond to a setting of the knob, in the above-discussed two-tier framework, that is maximal: P is the target scheme itself (here Schnorr identification or signatures), so that the first tier is trivial and the second tier directly proves the scheme secure in the idealized model.

But it is well understood that idealized models have limitations. Proofs in the GGM assume the adversary does not exploit the representation of group elements. In the AGM, it is assumed that, whenever an adversary provides a group element Z, it is possible to extract its representation as a product of known powers of prior group elements. This is analogous to a "knowledge of exponent assumption" [19, 31, 8]. However, even in a typical elliptic curve group, an adversary can quite easily create group elements without "knowing" such a representation. The maximal setting of knob (working purely in an idealized model) means the security guarantee on the scheme is fully

subject to the limitations of the idealized model.

With MBDL, we, instead make a non-trivial, moderate setting of the knob. Our tight reductions from MBDL, such as Eq. (4), are in the standard model, and make no GGM or AGM-like assumptions on adversaries. It is of course true that we justify MBDL in the GGM (Theorem 5.1), but we are limiting the use of the idealized model to show security for a purely number-theoretic problem, namely MBDL. The first direct benefit is better security guarantees for the schemes. The second is that MBDL is a hub. As discussed above, we can prove security of many schemes from it, which reduces work compared to proving them all from scratch in idealized models, and also increases understanding by identifying a problem that is at the core of many things.

Another prior approach to improving reduction tightness has been to change metrics, measuring tightness, not via success probability and running time taken individually, but via their ratio [36]. This however does not translate to actual, numeric improvements. To discuss this further, let IMP-KOA denote impersonation under key-only attack. (That is, IMP-PA for adversaries making zero queries to their transcript oracle.) Kiltz, Masny and Pan (KMP) [36] define a problem they call 1-IDLOG that is a restatement of ("precisely models," in their language) the IMP-KOA security of $\mathsf{ID} = \mathsf{SchID}[\mathbb{G}, g]$. Due to the zero knowledge of ID, its IMP-PA security reduces tightly to its IMP-KOA security and thus to 1-IDLOG. Now, KMP [36] give a reduction of 1-IDLOG to DL that is ratio-tight, meaning preserves ratios of advantage to running time. This, however, uses rewinding, and is not tight in our sense, incurring the usual square-root loss when one considers running time and advantage separately. In particular the results of KMP do not seem to allow group sizes any smaller than allowed by the classical Eq. (1). Our reductions, in contrast, are tight for advantage and time taken individually, and across the full range for these values, and numerical estimates (Figure 1) show clear improvements over what one gets from Eq. (1). Also our results establish 1-IDLOG tightly (not merely ratio-tightly) under 1-MBDL. We discuss ratio-tightness further in Appendix B.

Discussion. Measuring quality of a reduction in terms of bit security effectively only reflects the resources required to attain an advantage close to 1. Under this metric, whether one starts from Eq. (1) or Eq. (4), one concludes that $\mathsf{ID} = \mathsf{SchID}[\mathbb{G}, g]$ has $\log_2(|\mathbb{G}|)/2$-bits of security. This reflects bit security being a coarse metric. The improvement offered by Eq. (4) over Eq. (1) becomes visible when one considers the full curve of advantage as a function of runtime, and is visible in Figure 1.

While new assumptions (like MBDL) should of course be treated with caution, cryptographic research has a history of progress through introducing them. For example, significant advances were obtained by moving from the CDH assumption to the stronger DDH one [40, 18]. Pairing-based cryptography has seen a host of assumptions that have had many further applications, including the bilinear Diffie-Hellman (BDH) assumption of [16] and the DLIN assumption of [15]. The RSA Φ-Hiding assumption of [17] has since found many applications. This suggests that the introduction and exploration of new assumptions, which we continue, is an interesting and productive line of research.

There is some feeling that "interactive" or "non-falsifiable" assumptions are undesirable. However, it depends on the particular assumption. There are interactive assumptions that are unbroken and successful, like OMDL [5], while many non-interactive ones have been broken. It is important that it be possible to show an assumption is false, but this is possible even for assumptions that are classified as "non-falsifiable;" for example, knowledge-of-exponent assumptions have successfully been shown to be false through cryptanalysis [8]. (The latter result assumes DL is hard.) MBDL is similarly amenable to cryptanalytic evaluation.

2 Preliminaries

NOTATION. If n is a positive integer, then $\mathbb{Z}_n$ denotes the set $\{0,\ldots,n-1\}$ and $[n]$ or $[1..n]$ denote the set $\{1,\ldots,n\}$. We denote the number of coordinates of a vector $\boldsymbol{x}$ by $|\boldsymbol{x}|$. If $\boldsymbol{x}$ is a vector then $|\boldsymbol{x}|$ is its length (the number of its coordinates), $\boldsymbol{x}[i]$ is its i-th coordinate and $[\boldsymbol{x}] = \{\,\boldsymbol{x}[i] : 1 \le i \le |\boldsymbol{x}|\,\}$ is the set of all its coordinates. A string is identified with a vector over $\{0,1\}$, so that if x is a string then $x[i]$ is its i-th bit and $|x|$ is its length. By ε we denote the empty vector or string. The size of a set S is denoted $|S|$. For sets D, R let $\mathrm{FNS}(D,R)$ denote the set of all functions $f : D \to R$.

Let S be a finite set. We let $x \xleftarrow{\$} S$ denote sampling an element uniformly at random from S and assigning it to x. We let $y \leftarrow A^{\mathrm{O}_1,\cdots}(x_1,\ldots;r)$ denote executing algorithm A on inputs $x_1,\ldots$ and coins r with access to oracles $\mathrm{O}_1,\ldots$ and letting y be the result. We let $y \xleftarrow{\$} A^{\mathrm{O}_1,\cdots}(x_1,\ldots)$ be the resulting of picking r at random and letting $y \leftarrow A^{\mathrm{O}_1,\cdots}(x_1,\ldots;r)$. We let $[A^{\mathrm{O}_1,\cdots}(x_1,\ldots)]$ denote the set of all possible outputs of A when invoked with inputs $x_1,\ldots$ and oracles $\mathrm{O}_1,\ldots$. Algorithms are randomized unless otherwise indicated. Running time is worst case.

GAMES. We use the code-based game playing framework of [11]. (See Fig. 3 for an example.) Games have procedures, also called oracles. Amongst these are INIT and a FIN. In executing an adversary $\mathcal{A}$ with a game Gm, procedure INIT is executed first, and what it returns is the input to $\mathcal{A}$. The latter may now call all game procedures except INIT, FIN. When the adversary terminates, its output is viewed as the input to FIN, and what the latter returns is the game output. By $\Pr[\mathrm{Gm}(\mathcal{A})]$ we denote the event that the execution of game Gm with adversary $\mathcal{A}$ results in output true. In writing game or adversary pseudocode, it is assumed that boolean variables are initialized to false, integer variables are initialized to 0 and set-valued variables are initialized to the empty set $\emptyset$. When adversary $\mathcal{A}$ is executed with game Gm, the running time of the adversary, denoted $\mathrm{T}_{\mathcal{A}}$, assumes game procedures take unit time to respond. By $\mathrm{Q}^{\mathrm{O}}_{\mathcal{A}}$ we denote the number of queries made by $\mathcal{A}$ to oracle O in the execution. These counts are all worst case.

GROUPS. Let $\mathbb{G}$ be a group of order p. We will use multiplicative notation for the group operation, and we let $1_{\mathbb{G}}$ denote the identity element of $\mathbb{G}$. We let $\mathbb{G}^* = \mathbb{G} \setminus \{1_{\mathbb{G}}\}$ denote the set of non-identity elements, which is the set of generators of $\mathbb{G}$ if the latter has prime order. If $g \in \mathbb{G}^*$ is a generator and $X \in \mathbb{G}$, the discrete logarithm base g of X is denoted $\mathsf{DL}_{\mathbb{G},g}(X)$, and it is in the set $\mathbb{Z}_{|\mathbb{G}|}$.

3 The Multi-Base Discrete-Logarithm Problem

We introduce the multi-base discrete-logarithm (MBDL) problem. It is similar in flavor to the one-more discrete-logarithm (OMDL) problem [5], which has found many applications, in that it gives the adversary the ability to take discrete logarithms. For the rest of this Section, we fix a group $\mathbb{G}$ of prime order $p = |\mathbb{G}|$, and we fix a generator $g \in \mathbb{G}^*$ of $\mathbb{G}$. Recall that $\mathsf{DL}_{\mathbb{G},g} : \mathbb{G} \to \mathbb{Z}_p$ is the discrete logarithm function in $\mathbb{G}$ with base g.

DL AND OMDL. We first recall the standard discrete logarithm (DL) problem via game $\mathbf{G}^{\mathrm{dl}}_{\mathbb{G},g}$ on the left of Figure 2. INIT provides the adversary, as input, a random challenge group element Y, and to win it must output $y' = \mathsf{DL}_{\mathbb{G},g}(Y)$ to FIN. We let $\mathbf{Adv}^{\mathrm{dl}}_{\mathbb{G},g}(\mathcal{A}) = \Pr[\mathbf{G}^{\mathrm{dl}}_{\mathbb{G},g}(\mathcal{A})]$ be the discrete-log advantage of $\mathcal{A}$.

In the OMDL problem [5], the adversary can obtain many random challenges $Y_1, Y_2, \ldots, Y_n \in \mathbb{G}$. It has access to a discrete log oracle that given $W \in \mathbb{G}$ returns $\mathsf{DL}_{\mathbb{G},g}(W)$. For better comparison with MBDL, let's allow just one query to this oracle. To win it must compute the discrete logarithms of two group elements from the given list $Y_1, Y_2, \ldots, Y_n \in \mathbb{G}$. The integer $n \ge 2$ is a parameter of the problem.

Game $\mathbf{G}^{\mathrm{dl}}_{\mathbb{G},g}$	Game $\mathbf{G}^{\mathrm{mbdl}}_{\mathbb{G},g,n}$

INIT:
1 $p \leftarrow \lvert\mathbb{G}\rvert$; $y \leftarrow\!\!\$\ Z_p$; $Y \leftarrow g^y$
2 Return Y
 	
FIN(y'):
3 Return $(y = y')$
	INIT:
	1 $p \leftarrow \lvert\mathbb{G}\rvert$; $y \leftarrow\!\!\$\ Z_p$; $Y \leftarrow g^y$
	2 For $i = 1, \ldots, n$ do
	3 $\quad x_i \leftarrow\!\!\$\ \mathbb{Z}_p^*$; $X_i \leftarrow g^{x_i}$
	4 Return $Y, X_1, \ldots, X_n$
	
	DLO(i, W): // One query
	5 Return $\mathsf{DL}_{\mathbb{G},X_i}(W)$
	
	FIN(y'):
	6 Return $(y = y')$

Figure 2: Let $\mathbb{G}$ be a group of prime order $p = |\mathbb{G}|$, and let $g \in \mathbb{G}^*$ be a generator of $\mathbb{G}$. Left: Game defining standard discrete logarithm problem. Right: Game defining (n, m)-multi-base discrete logarithm problem. Recall $\mathsf{DL}_{\mathbb{G},X}(W)$ is the discrete logarithm of $W \in \mathbb{G}$ to base $X \in \mathbb{G}^*$.

MBDL. In the MBDL problem we introduce, we return, as in DL, to there being a single random challenge point Y whose discrete logarithm in base g the adversary must compute. It has access to an oracle DLO to compute discrete logs, but rather than in base g as in OMDL, to bases that are public, random group elements $X_1, X_2, \ldots, X_n$. It is allowed *just one* query to DLO. (As we will see, this is to avoid trivial attacks.) The integer $n \geq 1$ is a parameter of the problem.

Proceeding formally, consider game $\mathbf{G}^{\mathrm{mbdl}}_{\mathbb{G},g,n}$ on the right in Fig. 2, where $n \geq 1$ is an integer parameter called the number of bases. The adversary's input, as provided by INIT, is a random challenge group element Y together with random generators $X_1, X_2, \ldots, X_n$. It can call oracle DLO with an index $i \in [n]$ and any group element $W \in \mathbb{G}$ of its choice to get back $\mathsf{DL}_{\mathbb{G},X_i}(W)$. Just one such call is allowed. At the end, the adversary wins the game if it outputs $y' = \mathsf{DL}_{\mathbb{G},g}(Y)$ to FIN. We define the mbdl-advantage of $\mathcal{A}$ by

$$\mathbf{Adv}^{\mathrm{mbdl}}_{\mathbb{G},g,n}(\mathcal{A}) = \Pr[\mathbf{G}^{\mathrm{mbdl}}_{\mathbb{G},g,n}(\mathcal{A})] \ .$$

DISCUSSION. By n-MBDL we will refer to the problem with parameter n. It is easy to see that if n-MBDL is hard then so is n'-MBDL for any $n' \leq n$. Thus, the smaller the value of n, the weaker the assumption. For our results, 1-MBDL, the weakest assumption in the series, suffices.

We explain why at most one DLO query is allowed. Suppose the adversary is allowed two queries. It could compute $a = \mathrm{DLO}(1, Y) = \mathsf{DL}_{\mathbb{G},X_1}(Y)$ and $b = \mathrm{DLO}(1, g) = \mathsf{DL}_{\mathbb{G},X_1}(g)$, so that $X_1^a = Y$ and $X_1^b = g$. Now the adversary returns $y' \leftarrow ab^{-1} \bmod p$ and we have $g^{y'} = (g^{b^{-1}})^a = X_1^a = Y$, so the adversary wins.

As evidence for the hardness of MBDL, Theorem 5.1 proves good bounds on the adversary advantage in the generic group model (GGM). It is also important to consider non-generic approaches to the discrete logarithm problem over elliptic curves, including index-calculus methods and Semaev polynomials [48, 46, 49, 35, 29], but, to the best of our assessment, these do not yield attacks on MBDL that beat the GGM bound of Theorem 5.1.

The MBDL problem as we have defined it can be generalized to allow multiple DLO queries with the restriction that at most one query is allowed per base, meaning for each i there can be at most one $\mathrm{DLO}(i, \cdot)$ query. In this paper, we do not need or use this extension. We have found applications based on it, but not pursued them because we have been unable to prove security of this extended version of MBDL in the GGM. We consider providing such a GGM proof an intriguing

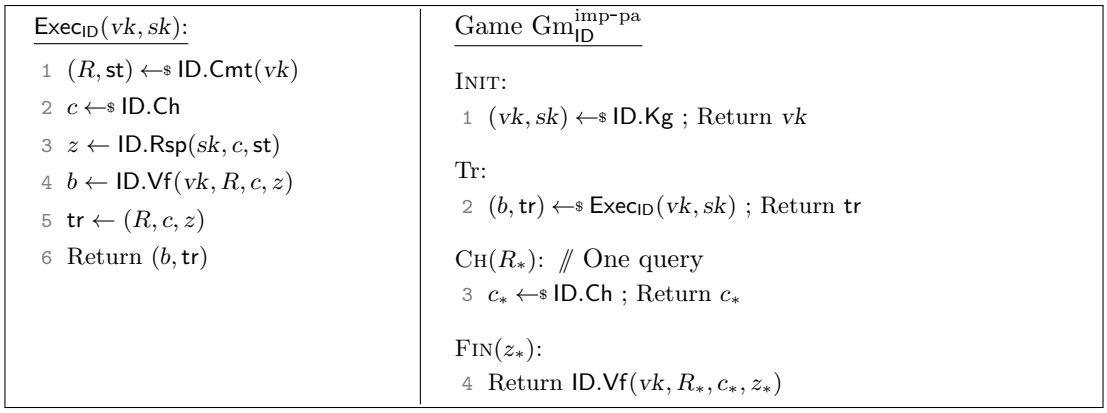


Figure 3: Left: Algorithm defining an honest execution of the canonical identification scheme ID given key pair (sk, vk). Right: Game defining IMP-PA security of ID.

open question, resolving which would open the door to several new applications.

Our formalizations of DL and MBDL fix the generator g. See [4] for a discussion of fixed versus random generators.

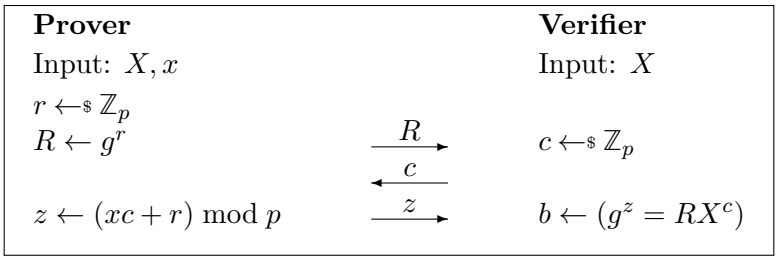
4 Schnorr Identification and Signatures from MBDL

In this section, we give a *tight* reduction of the IMP-PA security of the Schnorr identification scheme to the 1-MBDL problem and derive a corresponding improvement for Schnorr signatures.

Identification Schemes. We recall that a (canonical) identification scheme [1] ID (see Figure 4 for an example) is a 3-move protocol in which the prover sends a first message called a commitment, the verifier sends a random challenge, the prover sends a response that depends on its secret key, and the verifier makes a decision to accept or reject based on the conversation transcript and the prover's public key. Formally, ID is specified by algorithms ID.Kg, ID.Cmt, ID.Rsp, and ID.Vf, as well as a set ID.Ch of challenges Via $(vk, sk) \leftarrow\!\!\$\ \mathsf{ID.Kg}$, the key generation algorithm generates public verification key vk and associated secret key sk. Algorithms ID.Cmt and ID.Rsp are the prover algorithms. The commitment algorithm ID.Cmt takes input the public key vk and returns a commitment message R to send to the verifier, as well as a state st for the prover to retain. The deterministic response algorithm ID.Rsp takes input the secret key sk, a challenge $c \in \mathsf{ID.Ch}$ sent by the verifier, and a state st, to return a response z to send to the verifier. The deterministic verification algorithm ID.Vf takes input the public key and a conversation transcript R, c, z to return a decision $b \in \{\mathsf{true}, \mathsf{false}\}$ that is the outcome of the protocol.

An honest execution of the protocol is defined via procedure $\mathsf{Exec}_{\mathsf{ID}}$ shown in the upper left of Fig. 3. It takes input a key pair $(vk, sk) \in [\mathsf{ID.Kg}]$ to return a pair (b, tr) where $b \in \{\mathsf{true}, \mathsf{false}\}$ denotes the verifier's decision whether to accept or reject and $\mathsf{tr} = (R, c, z)$ is the transcript of the interaction. We require that ID schemes satisfy *(perfect) completeness*, namely that for any $(vk, sk) \in [\mathsf{ID.Kg}]$ and any $(b, \mathsf{tr}) \in [\mathsf{Exec}_{\mathsf{ID}}(sk, vk)]$ we have $b = \mathsf{true}$.

Impersonation under passive attack (IMP-PA) [24] is a security metric asking that an adversary not in possession of the prover's secret key be unable to impersonate the prover, even given access to honestly generated transcripts. Formally, consider the game $\mathrm{Gm}_{\mathsf{ID}}^{\text{imp-pa}}$ given in the right column of Fig. 3. An adversary has input the public key vk returned by Init. It then has access to honest transcripts via the oracle Tr. When it is ready to convince the verifier, it submits its commitment R_*



ID.Kg:

1 $x \leftarrow\!\!\$\ \mathbb{Z}_{|\mathbb{G}|}$; $X \leftarrow g^x$; Return (X, x)

ID.Cmt(X):

2 $r \leftarrow\!\!\$\ \mathbb{Z}_{|\mathbb{G}|}$; $R \leftarrow g^r$; Return (R, r)

ID.Rsp(x, c, r):

3 $z \leftarrow (xc + r) \mod |\mathbb{G}|$

4 Return z

ID.Vf(X, R, c, z):

5 $b \leftarrow (g^z = X^c R)$; Return b

DS.Kg:

1 $x \leftarrow\!\!\$\ \mathbb{Z}_{|\mathbb{G}|}$; $X \leftarrow g^x$

2 Return (X, x)

DS.Sign$^{\mathrm{H}}(x, m)$:

3 $r \leftarrow\!\!\$\ \mathbb{Z}_{|\mathbb{G}|}$; $R \leftarrow g^r$

4 $c \leftarrow \mathrm{H}(R, m)$

5 $z \leftarrow (xc + r) \mod |\mathbb{G}|$

6 Return (R, z)

DS.Vf$^{\mathrm{H}}(X, m, \sigma)$:

7 $(R, z) \leftarrow \sigma$

8 $c \leftarrow \mathrm{H}(R, m)$

9 Return $(g^z = X^c R)$

Figure 4: Let $\mathbb{G}$ be a group of prime order $p = |\mathbb{G}|$ and let $g \in \mathbb{G}^*$ be a generator of $\mathbb{G}$. The Schnorr ID scheme $\mathsf{ID} = \mathsf{SchID}[\mathbb{G}, g]$ is shown pictorially at the top and algorithmically at the bottom left. At the bottom right is the Schnorr signature scheme $\mathsf{DS} = \mathsf{SchSig}[\mathbb{G}, g]$, using $\mathrm{H} : \mathbb{G} \times \{0,1\}^* \to \mathbb{Z}_p$.

to oracle Ch. We allow only one query to Ch. In response the adversary obtains a random challenge c_*. It must now output a response z_* to Fin, and the game returns true iff the transcript is accepted by ID.Vf. The R_*, c_* at line 4 are, respectively, the prior query to Ch, and the response chosen at line 3. We define the IMP-PA advantage of $\mathcal{A}$ against ID as $\mathbf{Adv}^{\text{imp-pa}}_{\mathsf{ID}}(\mathcal{A}) = \Pr[\mathrm{Gm}^{\text{imp-pa}}_{\mathsf{ID}}(\mathcal{A})]$, the probability that the game returns true.

Schnorr identification scheme and prior results. Let $\mathbb{G}$ be a group of prime order $p = |\mathbb{G}|$, and $g \in \mathbb{G}^*$ a generator of $\mathbb{G}$. We recall the Schnorr identification scheme [45] $\mathsf{ID} = \mathsf{SchID}[\mathbb{G}, g]$ in Fig. 4. The public key $vk = X = g^x \in \mathbb{G}$ where $sk = x \in \mathbb{Z}_p$ is the secret key. The commitment is $R = g^r \in \mathbb{G}$, and r is returned as the prover state by the commitment algorithm. Challenges are drawn from $\mathsf{ID.Ch} = \mathbb{Z}_p$, and the response z and decision b are computed as shown.

The IMP-PA security of $\mathsf{ID} = \mathsf{SchID}[\mathbb{G}, g]$ based on DL is proven by a rewinding argument. The simplest analysis is via the Reset Lemma of [7]. It leads to the following (cf. [7, Theorem 2], [9, Theorem 3]). Let $\mathcal{A}$ be an adversary attacking the IMP-PA security of ID. Then there is a discrete log adversary $\mathcal{B}$ such that

$$\mathbf{Adv}^{\text{imp-pa}}_{\mathsf{ID}}(\mathcal{A}) \le \sqrt{\mathbf{Adv}^{\mathrm{dl}}_{\mathbb{G},g}(\mathcal{B})} + \frac{1}{p} \,. \tag{7}$$

Additionally, the running time $\mathrm{T}_{\mathcal{B}}$ of $\mathcal{B}$ is roughly $2\,\mathrm{T}_{\mathcal{A}}$ plus simulation overhead $\mathcal{O}(\mathrm{Q}^{\mathrm{Tr}}_{\mathcal{A}} \cdot T^{\mathrm{exp}}_{\mathbb{G}})$, where $T^{\mathrm{exp}}_{\mathbb{G}}$ is the time for an exponentiation in $\mathbb{G}$.

Our result. We show that the IMP-PA-security of the Schnorr identification scheme reduces *tightly* to the 1-MBDL problem. The reduction does *not* use rewinding. Our mbdl-adversary $\mathcal{B}$

solves the 1-MBDL problem by running the given imp-pa adversary $\mathcal{A}$ just once, so the mbdl-advantage, and running time, of the former, are about the same as the imp-pa advantage, and running time, of the latter. Refer to Section 2 for notation like $\mathrm{T}_{\mathcal{A}}, \mathrm{Q}^{\mathrm{Tr}}_{\mathcal{A}}$.

Theorem 4.1 *Let $\mathbb{G}$ be a group of prime order $p = |\mathbb{G}|$, and let $g \in \mathbb{G}^*$ be a generator of $\mathbb{G}$. Let $\mathsf{ID} = \mathsf{SchID}[\mathbb{G}, g]$ be the Schnorr identification scheme. Let $\mathcal{A}$ be an adversary attacking the imp-pa security of ID. Then we can construct an adversary $\mathcal{B}$ (shown explicitly in Figure 5) such that*

$$\mathbf{Adv}^{\text{imp-pa}}_{\mathsf{ID}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{mbdl}}_{\mathbb{G},g,1}(\mathcal{B}) + \frac{1}{p} \ . \tag{8}$$

Additionally, $\mathrm{T}_{\mathcal{B}}$ is roughly $\mathrm{T}_{\mathcal{A}}$ plus simulation overhead $\mathcal{O}(\mathrm{Q}^{\mathrm{Tr}}_{\mathcal{A}} \cdot T^{\mathrm{exp}}_{\mathbb{G}})$.

Proof of of Theorem 4.1: Recall that, when reducing IMP-PA security of Schnorr to DL, the constructed dl adversary $\mathcal{B}$ sets the target point Y to be the public key X. It is natural to take the same approach in our case. The question is how to use the discrete logarithm oracle DLO to avoid rewinding and get a tight reduction. But this is not clear and indeed the DLO oracle does not appear to help towards this.

Our reduction deviates from prior ones by *not* setting the target point Y to be the public key. Instead we look at a successful impersonation by $\mathcal{A}$. (Simulation of $\mathcal{A}$'s transcript oracle Tr is again via the honest-verifier zero-knowledge property of the scheme.) Adversary $\mathcal{A}$ provides R_*, receives c_* and then returns z_* satisfying $g^{z_*} = R_* X^{c_*}$, where X is the public key. Thus, $\mathcal{A}$ effectively computes the discrete logarithm of $R_* X^{c_*}$. We make this equal our mbdl challenge Y, meaning $\mathcal{B}$, on input Y, arranges that $Y = R_* X^{c_*}$. If it can do this successfully, the z_* returned by $\mathcal{A}$ will indeed be $\mathsf{DL}_{\mathbb{G},g}(Y)$, which it can output and win.

But how can we arrange that $Y = R_* X^{c_*}$? This is where the DLO oracle enters. Adversary $\mathcal{B}$ gives X as input to $\mathcal{A}$, meaning the public key is set to the group generator relative to which $\mathcal{B}$ may compute discrete logarithms. Now, when $\mathcal{A}$ provides R_*, our adversary $\mathcal{B}$ returns a challenge c_* that ensures $Y = R_* X^{c_*}$. This means $c_* = \mathsf{DL}_{\mathbb{G},X}(Y R_*^{-1})$, and this is something $\mathcal{B}$ can compute via its DLO oracle.

Some details include that the X returned by Init is a generator, while the public key is a random group element, so they are not identically distributed, and that the challenge computed via DLO must be properly distributed. The analysis will address these.

For the formal proof, consider the games of Figure 5. Procedures indicate (via comments) in which games they are present. Game Gm_1 includes the boxed code at line 2 while Gm_0 does not. The games implement the transcript oracle via the zero-knowledge simulation rather than using the secret key, but otherwise Gm_0 is the same as game $\mathrm{Gm}^{\text{imp-pa}}_{\mathsf{ID}}$ so we have

$$\begin{aligned}\mathbf{Adv}^{\text{imp-pa}}_{\mathsf{ID}}(\mathcal{A}) &= \Pr[\mathrm{Gm}_0(\mathcal{A})] \\ &= \Pr[\mathrm{Gm}_1(\mathcal{A})] + (\Pr[\mathrm{Gm}_0(\mathcal{A})] - \Pr[\mathrm{Gm}_1(\mathcal{A})]) \ .\end{aligned}$$

Games $\mathrm{Gm}_0, \mathrm{Gm}_1$ are identical-until-bad, so by the Fundamental Lemma of Game Playing [11] we have

$$\Pr[\mathrm{Gm}_0(\mathcal{A})] - \Pr[\mathrm{Gm}_1(\mathcal{A})] \leq \Pr[\mathrm{Gm}_1(\mathcal{A}) \text{ sets } \mathsf{bad}] \ .$$

Clearly $\Pr[\mathrm{Gm}_1(\mathcal{A}) \text{ sets } \mathsf{bad}] \leq 1/p$. Now we can work with Gm_1, where the public key X is a random element of $\mathbb{G}^*$ rather than of $\mathbb{G}$. We claim that

$$\Pr[\mathrm{Gm}_1(\mathcal{A}) = \Pr[\mathrm{Gm}_2(\mathcal{A})] \ . \tag{9}$$

Adversary $\mathcal{B}^{\text{DLO}}$:

1 $(Y, X) \leftarrow\!\!\$\ \text{INIT}()$; $z_* \leftarrow\!\!\$\ \mathcal{A}^{\text{CH},\text{TR}}(X)$; Return z_*

CH(R_*):

2 $W \leftarrow R_*^{-1} \cdot Y$; $c_* \leftarrow \text{DLO}(1, W)$; Return c_*

TR:

3 $z \leftarrow\!\!\$\ \mathbb{Z}_p$; $c \leftarrow\!\!\$\ \mathbb{Z}_p$; $R \leftarrow g^z \cdot X^{-c}$; Return (R, c, z)

Game Gm_0 / Gm_1 / Gm_2

INIT: // Games Gm_0, $\boxed{\text{Gm}_1}$

1 $p \leftarrow |\mathbb{G}|$; $y \leftarrow\!\!\$\ \mathbb{Z}_p$; $Y \leftarrow g^y$; $x \leftarrow\!\!\$\ \mathbb{Z}_p$

2 If $(x = 0)$ then $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{x \leftarrow\!\!\$\ \mathbb{Z}_p^*}$

3 $X \leftarrow g^x$; Return (Y, X)

INIT: // Game Gm_2

4 $p \leftarrow |\mathbb{G}|$; $y \leftarrow\!\!\$\ \mathbb{Z}_p$; $Y \leftarrow g^y$; $x \leftarrow\!\!\$\ \mathbb{Z}_p^*$; $X \leftarrow g^x$; Return (Y, X)

CH(R_*): // Games Gm_0, Gm_1

5 $c_* \leftarrow\!\!\$\ \mathbb{Z}_p$; Return c_*

CH(R_*): // Game Gm_2

6 $W \leftarrow R_*^{-1} \cdot Y$; $c_* \leftarrow \mathsf{DL}_{\mathbb{G},X}(W)$; Return c_*

Tr(W): // Games $\text{Gm}_0, \text{Gm}_1, \text{Gm}_2$

7 $z \leftarrow\!\!\$\ \mathbb{Z}_p$; $c \leftarrow\!\!\$\ \mathbb{Z}_p$; $R \leftarrow g^z \cdot X^{-c}$; Return (R, c, z)

FIN(z_*): // Games Gm_0, Gm_1

8 Return $(g^{z_*} = X^{c_*} R_*)$

FIN(z_*): // Games Gm_2

9 Return $(\ z_* = \mathsf{DL}_{\mathbb{G},g}(X^{c_*} R_*)\ )$

Figure 5: Top: MBDL adverary $\mathcal{B}$ for Theorem 4.1, based on IMP-PA adversary $\mathcal{A}$. Bottom: Games for proof of Theorem 4.1.

We now justify this. At line 4, game Gm_2 picks x directly from $\mathbb{Z}_p^*$, just like Gm_1, and also rewrites FIN in a different but equivalent way. The main thing to check is that CH in Gm_2 is equivalent to that in Gm_1, meaning line 6 results in c_* being uniformly distributed in $\mathbb{Z}_p$. For this regard R_*, X as fixed and define the function $f_{R_*,X} : \mathbb{G} \to \mathbb{Z}_p$ by $f_{R_*,X}(Y) = \mathsf{DL}_{\mathbb{G},X}(R_*^{-1} Y)$. The adversary has no information about Y prior to receiving c_* at line 6, so the claim is established if we show that $f_{R_*,X}$ is a bijection. This is true because $X \in \mathbb{G}^*$ is a generator, which means that the function $h_{R_*,X} : \mathbb{Z}_p \to \mathbb{G}$ defined by $h_{R_*,X}(c_*) = R_* X^{c_*}$ is the inverse of $f_{R_*,X}$. This establishes Eq. (9).

We now claim that adversary $\mathcal{B}$, shown in Fig. 5, satisfies

$$\Pr[\text{Gm}_2(\mathcal{A})] \leq \mathbf{Adv}^{\text{mbdl}}_{\mathbb{G},g,1}(\mathcal{B}) \ . \tag{10}$$

Putting this together with the above completes the proof, so it remains to justify Eq. (10). Adversary $\mathcal{B}$ has access to oracle DLO as per game $\mathbf{G}^{\text{mbdl}}_{\mathbb{G},g,1}$. In the code, CH and Tr are subroutines defined by $\mathcal{B}$ and used to simulate the oracles of the same names for $\mathcal{A}$. Adversary $\mathcal{B}$ has input the challenge Y whose discrete logarithm in base g it needs to compute, as well as the base X

relative to which it may perform one discrete log operation. It runs $\mathcal{A}$ on input X, so that the latter functions as the public key, which is consistent with Gm_2. The subroutine Ch uses DLO to produce c_* the same way as line 6 of Gm_2. It simulates Tr as per line 7 of Gm_2. If Gm_2 returns true at line 9 then we have $g^{z_*} = X^{c_*} R_* = W R_* = R_*^{-1} Y R_* = Y$, so $\mathcal{B}$ wins. ▮

Quantitative comparison. Concrete security improvements are in the end efficiency improvements, because, for a given security level, we can use smaller parameters, and thus the scheme algorithms are faster. Here we quantify this, seeing what Eq. (8) buys us over Eq. (7) in terms of improved efficiency for the identification scheme.

We take as goal to ensure that any adversary $\mathcal{A}$ with running time t has advantage $\mathbf{Adv}^{\text{imp-pa}}_{\mathsf{ID}}(\mathcal{A}) \leq \epsilon$ in violating IMP-PA security of $\mathsf{ID} = \mathsf{SchID}[\mathbb{G}, g]$. Here t, ϵ are parameters for which many choices are possible. For example, $t = 2^{90}$ and $\epsilon = 2^{-32}$ is one choice, reflecting a 128-bit security level, where we define the bit-security level as $\log_2(t/\epsilon)$. The cost of scheme algorithms is the cost of exponentiation in the group, which is cubic in the representation size $k = \log p$ of group elements. So we ask what k must be to provably ensure the desired security. Equations (7) and (8) will yield different choices of k, denoted k_1 and k_2, with $k_2 < k_1$. We will conclude that Eq. (8) allows a $s = (k_1/k_2)^3$-fold speedup for the scheme.

Let $\mathcal{B}_1$ denote the DL adversary referred to in Eq. (7), and $\mathcal{B}_2$ the 1-MBDL adversary referred to in (8). To use the equations, we now need estimates on their respective advantages. For this, we assume $\mathbb{G}$ is a group in which the security of discrete-log-related problems is captured by the bounds proven in the generic group model (GGM), as seems to be true, to best of our current understanding, for certain elliptic curve groups. We will ignore the simulation overhead in running time since the number of transcript queries of $\mathcal{A}$ reflects online executions of the identification protocol and should be considerably less than the running time of $\mathcal{A}$, so that we take the running times of both $\mathcal{B}_1$ and $\mathcal{B}_2$ to be about t, the running time of our IMP-PA adversary $\mathcal{A}$. Now the classical result of Shoup [47] says that $\mathbf{Adv}^{\text{dl}}_{\mathbb{G},g}(\mathcal{B}_1) \approx t^2/p$, and our Theorem 5.1 says that also $\mathbf{Adv}^{\text{mbdl}}_{\mathbb{G},g,1}(\mathcal{B}_2) \approx t^2/p$.

Here we pause to highlight that these two bounds being the same is a central attribute of the 1-MBDL assumption. That Theorem 4.1 (as per Figure 1) provides efficiency improvements stems not just from the reduction of Eq. (8) being tight, but also from that fact that the 1-MBDL problem is just as hard to solve as the DL problem, meaning $\mathbf{Adv}^{\text{mbdl}}_{\mathbb{G},g}(\mathcal{B}_2) \approx \mathbf{Adv}^{\text{dl}}_{\mathbb{G},g}(\mathcal{B}_1) \approx t^2/p$.

Continuing, putting together what we have so far gives two bounds on the IMP-PA advantage of $\mathcal{A}$, the first via Equations (7) and the second via Eq. (8), namely, dropping the $1/p$ terms,

$$\mathbf{Adv}^{\text{imp-pa}}_{\mathsf{ID}}(\mathcal{A}) \leq \epsilon_1(t) = \sqrt{\frac{t^2}{p}} = \frac{t}{\sqrt{p}} \tag{11}$$

$$\mathbf{Adv}^{\text{imp-pa}}_{\mathsf{ID}}(\mathcal{A}) \leq \epsilon_2(t) = \frac{t^2}{p}\ . \tag{12}$$

Recall our goal was to ensure that $\mathbf{Adv}^{\text{imp-pa}}_{\mathsf{SchID}[\mathbb{G},g]}(\mathcal{A}) \leq \epsilon$. We ask, what value of p, in either case, ensures this? Solving for p in the equations $\epsilon = \epsilon_1(t)$ and $\epsilon = \epsilon_2(t)$, we get two corresponding values, namely $p_1 \approx t^2/\epsilon^2$ and $p_2 \approx t^2/\epsilon$. We see that $p_1 > p_2$, meaning Theorem 4.1 guarantees the same security as Eq. (7) in groups of a smaller size. Finally, the ratio of representation sizes for group elements is

$$r \approx \frac{\log(p_1)}{\log(p_2)} \approx \frac{\log(t^2/\epsilon) + \log(1/\epsilon)}{\log(t^2/\epsilon)} = 1 + \frac{\log(1/\epsilon)}{\log(t^2/\epsilon)}\ .$$

Scheme algorithms employ exponentiation in the group and are thus cubic time, so the ratio of speeds is $s = r^3$, which we call the speedup factor, and we can now estimate it numerically. For a

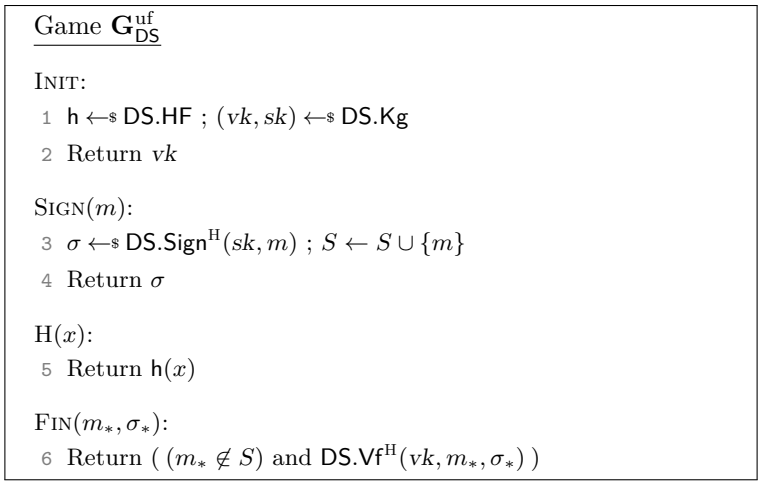


Figure 6: Game defining UF security of signature scheme DS.

---

few values of t, ϵ, Figure 1 shows the log of the group size p_i needed to ensure the desired security under prior results ($i = 1$) and ours ($i = 2$). Then it shows the speedup s. For example if we want attacks of time $t = 2^{64}$ to achieve advantage at most $\epsilon = 2^{-64}$, prior results would require a group of size p_1 satisfying $\log(p_1) \approx 256$, while our results allow it with a group of size $\log(p_2) \approx 192$, which yields a 2.4x speedup. Of course many more examples are possible.

Signature schemes. Towards results on the Schnorr signature scheme, we start by recalling definitions. A signature scheme DS specifies key generation algorithm DS.Kg, signing algorithm DS.Sign, deterministic verification algorithm DS.Vf and a set DS.HF of functions called the hash function space. Via $(vk, sk) \leftarrow\!\!\$\ \mathsf{DS.Kg}$ the signer generates a public verification key vk and secret signing key sk. Via $\sigma \leftarrow\!\!\$\ \mathsf{DS.Sign}^{\mathsf{h}}(sk, m)$ the signing algorithm takes sk and a message $m \in \{0,1\}^*$, and, with access to an oracle $\mathsf{h} \in \mathsf{DS.HF}$, returns a signature σ. Via $b \leftarrow \mathsf{DS.Vf}^{\mathsf{h}}(vk, m, \sigma)$, the verifier obtains a boolean decision $b \in \{\mathsf{true}, \mathsf{false}\}$ about the validity of the signature. The correctness requirement is that for all $\mathsf{h} \in \mathsf{DS.HF}$, all $(vk, sk) \in [\mathsf{DS.Kg}]$, all $m \in \{0,1\}^*$ and all $\sigma \in [\mathsf{DS.Sign}^{\mathsf{h}}(sk, m)]$ we have $\mathsf{DS.Vf}^{\mathsf{h}}(vk, m, \sigma) = \mathsf{true}$.

Game $\mathbf{G}^{\mathrm{uf}}$ in Fig. 6 captures UF (unforgeability under chosen-message attack) [30]. Procedure H is the random oracle [10], implemented as a function h chosen at random from DS.HF. We define the UF advantage of adversary $\mathcal{A}$ as $\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}}(\mathcal{A}) = \Pr[\mathbf{G}^{\mathrm{uf}}_{\mathsf{DS}}(\mathcal{A})]$.

Schnorr signatures. The Schnorr signature scheme $\mathsf{DS} = \mathsf{SchSig}[\mathbb{G}, g]$ is derived by applying the Fiat-Shamir transform [25] to the Schnorr identification scheme. Its algorithms are shown at the bottom right of Fig. 4. The set DS.HF consists of all functions $\mathsf{h} : \mathbb{G} \times \{0,1\}^* \to \mathbb{Z}_p$.

Our and prior results. We give a reduction, of the UF security of the Schnorr signature scheme to the 1-MBDL problem, that loses only a factor of the number of hash-oracle queries of the adversary. We start by recalling the following lemma from [1]. It derives the UF security of $\mathsf{SchSig}[G, g]$ from the IMP-PA security of $\mathsf{SchID}[G, g]$:

Lemma 4.2 [1] *Let $\mathbb{G}$ be a group of prime order $p = |\mathbb{G}|$, and let $g \in \mathbb{G}^*$ be a generator of $\mathbb{G}$. Let $\mathsf{ID} = \mathsf{SchID}[\mathbb{G}, g]$ and $\mathsf{DS} = \mathsf{SchID}[\mathbb{G}, g]$ be the Schnorr identification and signature schemes, respectively. Let $\mathcal{A}_{\mathrm{ds}}$ be an adversary attacking the uf-security of DS. Let $\alpha = (1 + \mathrm{Q}^{\mathrm{H}}_{\mathcal{A}_{\mathrm{ds}}} + \mathrm{Q}^{\mathrm{Sign}}_{\mathcal{A}_{\mathrm{ds}}})\mathrm{Q}^{\mathrm{Sign}}_{\mathcal{A}_{\mathrm{ds}}}$. Then we can construct an adversary $\mathcal{A}_{\mathrm{id}}$ such that*

$$\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}}(\mathcal{A}_{\mathrm{ds}}) \leq (1 + \mathrm{Q}^{\mathrm{H}}_{\mathcal{A}_{\mathrm{ds}}}) \cdot \mathbf{Adv}^{\text{imp-pa}}_{\mathsf{ID}}(\mathcal{A}_{\mathrm{id}}) + \frac{\alpha}{p} .$$

Additionally, $\mathrm{T}_{\mathcal{A}_{\mathrm{id}}} \approx \mathrm{T}_{\mathcal{A}_{\mathrm{ds}}}$ *and* $\mathrm{Q}^{\mathrm{Tr}}_{\mathcal{A}_{\mathrm{id}}} = \mathrm{Q}^{\mathrm{Sign}}_{\mathcal{A}_{\mathrm{ds}}}$.

Combining this with Theorem 4.1, we have:

Theorem 4.3 *Let* $\mathbb{G}$ *be a group of prime order* $p = |\mathbb{G}|$*, and let* $g \in \mathbb{G}^*$ *be a generator of* $\mathbb{G}$*. Let* $\mathsf{DS} = \mathsf{SchSig}[\mathbb{G}, g]$ *be the Schnorr signature scheme. Let* $\mathcal{A}$ *be an adversary attacking the uf security of* ID*. Let* $\beta = (1 + \mathrm{Q}^{\mathrm{H}}_{\mathcal{A}} + \mathrm{Q}^{\mathrm{Sign}}_{\mathcal{A}})\mathrm{Q}^{\mathrm{Sign}}_{\mathcal{A}} + (1 + \mathrm{Q}^{\mathrm{H}}_{\mathcal{A}})$*. Then we can construct an adversary* $\mathcal{B}$ *such that*

$$\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}}(\mathcal{A}) \le (1 + \mathrm{Q}^{\mathrm{H}}_{\mathcal{A}}) \cdot \mathbf{Adv}^{\mathrm{mbdl}}_{\mathbb{G},g,1}(\mathcal{B}) + \frac{\beta}{p} \ . \tag{13}$$

Additionally, $\mathrm{T}_{\mathcal{B}}$ *is roughly* $\mathrm{T}_{\mathcal{A}}$ *plus simulation overhead* $\mathcal{O}(\mathrm{Q}^{\mathrm{Sign}}_{\mathcal{A}} \cdot T^{\mathrm{exp}}_{\mathbb{G}})$.

Let's compare this to prior results. A simple proof of UF-security of DS from DL can be obtained by combining Lemma 4.2 with the classical DL-based security of ID as given by Eq. (7). For $\mathcal{A}$ an adversary attacking the UF security of DS, this would yield a discrete log adversary $\mathcal{B}$ such that

$$\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}}(\mathcal{A}) \le (1 + \mathrm{Q}^{\mathrm{H}}_{\mathcal{A}}) \cdot \sqrt{\mathbf{Adv}^{\mathrm{dl}}_{\mathbb{G},g}(\mathcal{B})} + \frac{\beta}{p} \ , \tag{14}$$

where β is as in Theorem 4.3 and $\mathrm{T}_{\mathcal{B}}$ is about $2\,\mathrm{T}_{\mathcal{A}}$ plus the same simulation overhead as above. This is however *not* the best prior bound. One can do better with a direct application of the general Forking Lemma of [6] as per [44]. For $\mathcal{A}$ an adversary attacking the UF security of DS, this would yield a discrete log adversary $\mathcal{B}$ such that

$$\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}}(\mathcal{A}) \le \sqrt{(1 + \mathrm{Q}^{\mathrm{H}}_{\mathcal{A}}) \cdot \mathbf{Adv}^{\mathrm{dl}}_{\mathbb{G},g}(\mathcal{B})} + \frac{\beta}{p} \ , \tag{15}$$

where β and $\mathrm{T}_{\mathcal{B}}$ are as above. The reason Eq. (15) is a better bound than Eq. (14) is that the $1 + \mathrm{Q}^{\mathrm{H}}_{\mathcal{A}}$ term has moved under the square root. Still we see that Eq. (13) is even better; roughly (neglecting the additive term), the bound in Eq. (13) is the square of the one in Eq. (15), and thus (always) smaller.

QUANTITATIVE COMPARISONS. Our numerical comparisons will be with the best prior bound, meaning that of Eq. (15). For a few values of t, q_h, ϵ with $t \ge q_h = \mathrm{Q}^{\mathrm{H}}_{\mathcal{A}}$, Figure 1 shows the speedup s from Eq. (13) over Eq. (15). The table shows that the speedup is a bit less than for Schnorr identification shown in the same Figure, but still significant. For example if we want attacks of time $t = 2^{64}$ to achieve advantage at most $\epsilon = 2^{-64}$, Theorem 4.3 is allowing group sizes to go down enough to yield a 5.4-fold speedup.

To derive these estimates, we use the same framework and setup as we did for identification. Let $\mathbb{G}$ be a group of prime order p with generator g. We take as goal to ensure that any adversary $\mathcal{A}$ with running time t, making q_h queries to H and q_s queries to SIGN, has advantage $\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}}(\mathcal{A}) \le \epsilon$ in violating UF security of $\mathsf{DS} = \mathsf{SchSig}[\mathbb{G}, g]$, where t, ϵ, q_h, q_s are parameters. We assume $q_s < q_h \le t$, as one expects in practice. Let $\mathcal{B}_1, \mathcal{B}_2$ be the adversaries of Equations (15) and (13), respectively. As before, assume $\mathbf{Adv}^{\mathrm{dl}}_{\mathbb{G},g}(\mathcal{B}_1) \approx t^2/p$ from [47], and also $\mathbf{Adv}^{\mathrm{mbdl}}_{\mathbb{G},g,1}(\mathcal{B}_2) \approx t^2/p$ from Theorem 5.1. Then

$$\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}}(\mathcal{A}) \le \epsilon_1(t, q_h) \approx \sqrt{\frac{q_h t^2}{p}}$$

$$\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}}(\mathcal{A}) \le \epsilon_2(t, q_h) \approx q_h \cdot \frac{t^2}{p} = \frac{q_h t^2}{p} \approx \epsilon_1(t, q_h)^2 \ .$$

In the estimates above, we have dropped the additive term, which has order $q_h q_s/p$, because this is negligible compared to the other term for reasonable parameter values, including the ones we

Game $\mathbf{G}^{\text{gg-mbdl}}_{G,n}$ // Set G is the range of the encoding, $|G|$ is prime.

INIT():
1 $p \leftarrow |G|$; $\mathrm{E} \leftarrow\!\!\$$ Bijections$(\mathbb{Z}_p, G)$ // Think " $\mathrm{E}(x) = g^x$ "
2 $\mathbb{1} \leftarrow \mathrm{E}(0)$; $g \leftarrow \mathrm{E}(1)$ // Think $\mathbb{1}$ the identity and g generator
3 $y \leftarrow\!\!\$\ \mathbb{Z}_p$; $Y \leftarrow \mathrm{E}(y)$
4 For $i = 1, \ldots, n$ do $x_i \leftarrow\!\!\$\ \mathbb{Z}_p^*$; $X_i \leftarrow \mathrm{E}(x_i)$
5 $\mathrm{GL} \leftarrow \{\mathbb{1}, g, Y, X_1, \ldots, X_n\}$
6 Return $\mathbb{1}, g, Y, X_1, \ldots, X_n$

OP(A, B, sgn): // $A, B \in G$ and $\mathsf{sgn} \in \{+, -\}$
7 If ($A \notin \mathrm{GL}$ or $B \notin \mathrm{GL}$) then return $\perp$
8 $c \leftarrow (\mathrm{E}^{-1}(A)\ \mathsf{sgn}\ \mathrm{E}^{-1}(B)) \bmod p$; $C \leftarrow \mathrm{E}(c)$; $\mathrm{GL} \leftarrow \mathrm{GL} \cup \{C\}$
9 Return C

DLO(i, W): // $i \in [1..n]$ and $W \in G$
10 If ($W \notin \mathrm{GL}$) then return $\perp$
11 $z \leftarrow x_i^{-1} \cdot \mathrm{E}^{-1}(W) \bmod p$ // x_i^{-1} is inverse of x_i mod p
12 Return z

FIN(y'):
13 Return $(y = y')$

Figure 7: Game defining n-MBDL problem in the generic group model.

consider. This leaves ϵ_1, ϵ_2 not depending on q_s, but recall the latter is expected to be (much) smaller than q_h. Then our bound ϵ_2 is about the square of the prior one, and thus always smaller.

We now ask what value of p ensures $\mathbf{Adv}^{\text{uf}}_{\mathsf{DS}}(\mathcal{A}) \le \epsilon$, in each case. Solving $\epsilon_1(t, q_h) \le \epsilon$ yields $p_1 \approx t^2 q_h/\epsilon^2$, and solving $\epsilon_2(t, q_h) \le \epsilon$ yields $p_2 \approx t^2 q_h/\epsilon$. As before we see that $p_2 < p_1$, meaning Theorem 4.1 guarantees security in groups of smaller size. The ratio of the representation-size of group elements is

$$r \approx \frac{\log(p_1)}{\log(p_2)} \approx \frac{\log(t^2 q_h/\epsilon) + \log(1/\epsilon)}{\log(t^2 q_h/\epsilon)} = 1 + \frac{\log(1/\epsilon)}{\log(t^2 q_h/\epsilon)} \ .$$

As before the ratio of speeds (speedup factor) is $s = r^3$, and we can now estimate it numerically. For a few values of t, ϵ, Figure 1 shows the log of the group size p_i needed to ensure the desired security under prior results ($i = 1$) and ours ($i = 2$). Then it shows the speedup s.

5 MBDL hardness in the Generic Group Model

With a new problem like MBDL it is important to give evidence of hardness. Here we provide this in the most common and accepted form, namely a proof of hardness in the generic group model (GGM).

The quantitative aspect of the result is just as important as the qualitative. Theorem 5.1 below says that the advantage of a GGM adversary $\mathcal{A}$ in breaking n-MBDL is $\mathcal{O}(q^2/p)$ where q is n plus the number of group operations (time) invested by $\mathcal{A}$, namely about the same as the ggm-dl-advantage of an adversary of the same resources. Reductions (to some problem) from MBDL that are tighter than ones from DL now bear fruit in justifying the secure use of smaller groups, which lowers costs.

The proof of Theorem 5.1 begins with a Lemma that characterizes the distribution of replies to the DLO query. A game sequence is then used to reduce bounding the adversary advantage to some static problems in linear algebra.

Some prior proofs in the GGM have been found to be wrong. (An example is that of [13] as pointed out by [33]. We note that the assumption was changed to fill the gap in [14].) Also we, at least, have often found GGM proofs imprecise and hard to verify. This has motivated us to try to be precise with definitions and to attend to details.

Starting with definitions, we associate to any encoding function E an explicit binary operation op_{E} that turns the range-set of E into a group. A random choice of E then results in the GGM, with the "generic group" being now explicitly defined as the group associated to E. The proof uses a game sequence and has been done at a level of detail that is perhaps unusual in this domain.

MBDL IN THE GGM. We start with definitions. Suppose G is a set whose size $p = |G|$ is a prime, and $\mathrm{E} : \mathbb{Z}_p \to G$ is a bijection, called the encoding function. For $A, B \in G$, define $A \,\mathsf{op}_{\mathrm{E}}\, B = \mathrm{E}(\mathrm{E}^{-1}(A) + \mathrm{E}^{-1}(B))$. Then G is a group under the operation op_{E} [51], with identity element $\mathrm{E}(0)$, and the encoding function becomes a group homomorphism: $\mathrm{E}(a+b) = \mathrm{E}(a)\,\mathsf{op}_{\mathrm{E}}\,\mathrm{E}(b)$ for all $a, b \in \mathbb{Z}_p$. The element $g = \mathrm{E}(1) \in G$ is a generator of this group, and $\mathrm{E}^{-1}(A)$ is then the discrete logarithm of $A \in G$ relative to g. We call op_{E} the group operation on G induced by E.

In the GGM, the encoding function E is picked at random and the adversary is given an oracle for the group operation op_{E} induced on G by E. Game $\mathbf{G}^{\text{gg-mbdl}}_{G,n}$ in Fig. 7 defines, in this way, the n-MBDL problem. The set G parameterizes the game, and the random choice of encoding function $\mathrm{E} : \mathbb{Z}_p \to G$ is shown at line 1. Procedure OP then implements either the group operation op_{E} on G induced by E (when sgn is $+$) or its inverse (when sgn is $-$). Lines 3,4 pick $y, x_1, \ldots, x_n$ and define the corresponding group elements $Y, X_1, \ldots, X_n$. Set GL holds all group elements generated so far. The new element here is the oracle DLO that takes $i \in [1..n]$ and $W \in G$ to return the discrete logarithm of W in base X_i. This being x_i^{-1} times the discrete logarithm of W in base g, the procedure returns $z \leftarrow x_i^{-1} \cdot \mathrm{E}^{-1}(W)$. The inverse and the operations here are modulo p. Only one query to this oracle is allowed, and the adversary wins if it halts with output y' that equals y. We let $\mathbf{Adv}^{\text{gg-mbdl}}_{G,n}(\mathcal{A}) = \Pr[\mathbf{G}^{\text{gg-mbdl}}_{G,n}(\mathcal{A})]$ be its ggm-mbdl-advantage.

RESULT. The following upper bounds the ggm-mbdl-advantage of an adversary $\mathcal{A}$ as a function of the number of its OP queries and n.

Theorem 5.1 *Let G be a set whose size $p = |G|$ is a prime. Let $n \geq 1$ be an integer. Let $\mathcal{A}$ be an adversary making $\mathrm{Q}^{\mathrm{OP}}_{\mathcal{A}}$ queries to its* OP *oracle and one query to its* DLO *oracle. Let $q = \mathrm{Q}^{\mathrm{OP}}_{\mathcal{A}} + n + 3$. Then*

$$\mathbf{Adv}^{\text{gg-mbdl}}_{G,n}(\mathcal{A}) \leq \frac{2 + q(q-1)}{p-1} \,. \tag{16}$$

PROOF FRAMEWORK AND LEMMA. Much of our work in the proof is over $\mathbb{Z}_p^{n+2}$ regarded as a vector space over $\mathbb{Z}_p$. We let $\vec{0} \in \mathbb{Z}_p^{n+2}$ be the all-zero vector, and $\vec{e}_i \in \mathbb{Z}_p^{n+2}$ the i-th basis vector, meaning it has a 1 in position i and zeros elsewhere. We let $\langle \vec{a}, \vec{b} \rangle = (\vec{a}[1]\vec{b}[1] + \cdots + \vec{a}[n+2]\vec{b}[n+2])$ denote the inner product of vectors $\vec{a}, \vec{b} \in \mathbb{Z}_p^{n+2}$, where the operations are modulo p.

In the GGM, the encoding function takes as input a point in $\mathbb{Z}_p$. The proof of GGM hardness of the DL problem [47] moved to a modified encoding function that took input a univariate polynomial, the variable representing the target discrete logarithm y. We extend this to have the modified encoding function take input a degree one polynomial in $n+1$ variables, these representing $x_1, \ldots, x_n, y$. The polynomial will be represented by the vector of its coefficients, so that representations, formally, are vectors in $\mathbb{Z}_p^{n+2}$. At some point, games in our proof will need to simulate the

Game $\mathbf{G}^{\mathrm{smp1}}_{p,n}$

INIT$(i, \vec{w})$: // $i \in [1..n]$ and $\vec{w} \in \mathbb{Z}_p^{n+2}$

1 $x_1, \ldots, x_n \leftarrow\!\!\$\ \mathbb{Z}_p^*$; $y \leftarrow\!\!\$\ \mathbb{Z}_p$; $w \leftarrow \vec{w}[i]$

2 $\vec{x}_i \leftarrow (x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n, 1, y)$

3 $z \leftarrow w + x_i^{-1} \cdot \langle \vec{w}, \vec{x}_i \rangle$; Return $(x_1, \ldots, x_n, y, z)$

Game $\mathbf{G}^{\mathrm{smp0}}_{p,n}$

INIT$(i, \vec{w})$: // $i \in [1..n]$ and $\vec{w} \in \mathbb{Z}_p^{n+2}$

1 $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n \leftarrow\!\!\$\ \mathbb{Z}_p^*$; $y \leftarrow\!\!\$\ \mathbb{Z}_p$; $w \leftarrow \vec{w}[i]$

2 $\vec{x}_i \leftarrow (x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n, 1, y)$

3 If ($\langle \vec{w}, \vec{x}_i \rangle \neq 0$) then $z \leftarrow\!\!\$\ \mathbb{Z}_p \setminus \{w\}$; $x_i \leftarrow (z - w)^{-1} \cdot \langle \vec{w}, \vec{x}_i \rangle$

4 Else $z \leftarrow w$; $x_i \leftarrow\!\!\$\ \mathbb{Z}_p^*$

5 Return $(x_1, \ldots, x_n, y, z)$

Figure 8: Games for Lemma 5.2.

reply to a DLO(i, W) query, meaning provide a reply z without knowing x_i. At this point, $W \in G$ will be represented by a vector $\vec{w} \in \mathbb{Z}_p^{n+2}$ that is known to the game and adversary. The natural simulation approach is to return a random $z \leftarrow\!\!\$\ \mathbb{Z}_p$ or $z \leftarrow\!\!\$\ \mathbb{Z}_p^*$, but these turn out to not perfectly mimic the true distribution of replies, because this distribution depends on $\vec{w}$. We start with a lemma that describes how to do a perfect simulation.

While the above serves as motivation for the Lemma, the Lemma itself is self-contained, making no reference to the DLO oracle. We consider the games of Figure 8. They are played with an adversary making a single INIT query whose arguments are an integer $i \in [1..n]$ and a vector $\vec{w} \in \mathbb{Z}_p^{n+2}$. The operations in the games, including inverses of elements in $\mathbb{Z}_p^*$, are in the field $\mathbb{Z}_p$. Game $\mathbf{G}^{\mathrm{smp1}}_{p,n}$ captures what, in our vector-representation, will be the "real" game, with z at line 3 computed correctly as a function of x_i. Game $\mathbf{G}^{\mathrm{smp0}}_{p,n}$ represents the simulation, first picking z and then defining x_i. Lines 3,4 show that there are two cases for how z, x_i are chosen in the simulation, depending on the value of $w = \vec{w}[i]$ and the inner product of $\vec{w}$ with $\vec{x}_i$. The games return all variables involved. The claim is that the outputs of the games are identically distributed, captured formally, in the statement of Lemma 5.2 below, as the condition that any adversary returns true with the same probability in the two games.

Lemma 5.2 *Let p be a prime and $n \geq 1$ an integer. Then for any adversary $\mathcal{A}$ we have*

$$\Pr[\mathbf{G}^{\mathrm{smp1}}_{p,n}(\mathcal{A})] = \Pr[\mathbf{G}^{\mathrm{smp0}}_{p,n}(\mathcal{A})] \ , \tag{17}$$

where the games are in Figure 8.

Proof of Lemma 5.2: With $i, \vec{w}$ being $\mathcal{A}$'s query to INIT, we can regard vector $\vec{x}_i = (x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n, 1, y)$ as fixed, since its constituents are chosen identically in the two games. Let $\alpha = \langle \vec{w}, \vec{x}_i \rangle$. Now consider two cases. The first is that $\alpha = 0$. Then, in both games, $z = w$, and x_i is chosen randomly from $\mathbb{Z}_p^*$. The second case is that $\alpha \neq 0$. For $x \in \mathbb{Z}_p^*$ let $Z_{w,\alpha}(x) = w + x^{-1} \cdot \alpha$, so that $z = Z_{w,\alpha}(x_i)$ at line 3 of game $\mathbf{G}^{\mathrm{smp1}}_{p,n}$. That $\alpha \neq 0$ implies $Z_{w,\alpha}(x) \neq w$, meaning the function $Z_{w,\alpha}$ maps as $Z_{w,\alpha} : \mathbb{Z}_p^* \to \mathbb{Z}_p \setminus \{w\}$. For $z \in \mathbb{Z}_p \setminus \{w\}$, let $X_{w,\alpha}(z) = \alpha \cdot (z - w)^{-1}$, so that $x_i = X_{w,\alpha}(z)$ at line 3 of game $\mathbf{G}^{\mathrm{smp0}}_{p,n}$. That $z \neq w$ and $\alpha \neq 0$ means $X_{w,\alpha}(z) \neq 0$, meaning the

INIT(): // Gm_0–Gm_3
1 $p \leftarrow |G|$; $\mathrm{E} \leftarrow\!\!\$\ \mathrm{Bijections}(\mathbb{Z}_p, G)$; $y \leftarrow\!\!\$\ \mathbb{Z}_p$
2 For $i = 1, \ldots, n$ do $x_i \leftarrow\!\!\$\ \mathbb{Z}_p^*$
3 $\vec{x} \leftarrow (x_1, \ldots, x_n, 1, y)$; $\vec{v} \leftarrow \vec{0}$
4 $\mathbb{1} \leftarrow \mathrm{VE}(\vec{0})$; $g \leftarrow \mathrm{VE}(\vec{e}_{n+1})$; $Y \leftarrow \mathrm{VE}(\vec{e}_{n+2})$
5 For $i = 1, \ldots, n$ do $X_i \leftarrow \mathrm{VE}(\vec{e}_i)$
6 Return $\mathbb{1}, g, Y, X_1, \ldots, X_n$

$\mathrm{VE}(\vec{t})$: // Gm_0. Here $\vec{t} \in \mathbb{Z}_p^{n+2}$.
7 If ($\mathrm{TV}[\vec{t}] \neq \perp$) then return $\mathrm{TV}[\vec{t}]$
8 $v \leftarrow \langle \vec{t}, \vec{x} \rangle$; $C \leftarrow \mathrm{E}(v)$; $\mathrm{TV}[\vec{t}] \leftarrow C$; $\mathrm{TI}[C] \leftarrow \vec{t}$; Return $\mathrm{TV}[\vec{t}]$

$\mathrm{VE}^{-1}(C)$: // Gm_0–Gm_3. Here $\mathrm{TI}[C] \neq \perp$.
9 Return $\mathrm{TI}[C]$

OP(A, B, sgn): // Gm_0–Gm_3. Here $\mathrm{TI}[A], \mathrm{TI}[B] \neq \perp$ and $\mathsf{sgn} \in \{+, -\}$
10 $\vec{c} \leftarrow \mathrm{VE}^{-1}(A)\ \mathsf{sgn}\ \mathrm{VE}^{-1}(B)$; $C \leftarrow \mathrm{VE}(\vec{c})$; Return C

DLO(i, W): // Gm_0. Here $i \in [n]$ and $\mathrm{TI}[W] \neq \perp$.
11 $\vec{w} \leftarrow \mathrm{VE}^{-1}(W)$; $z \leftarrow (x_i)^{-1} \cdot \langle \vec{w}, \vec{x} \rangle$; Return z

FIN(y'): // Gm_0–Gm_3
12 Return $(y = y')$

Figure 9: Game Gm_0 for the proof of Theorem 5.1. Some procedures will also be in later games, as marked.

---

function $X_{w,\alpha}$ maps as $X_{w,\alpha} : \mathbb{Z}_p \setminus \{w\} \to \mathbb{Z}_p^*$. The proof is complete if we show that these functions are inverses of each other, in particular showing that both are bijections. Indeed, for any $x \in \mathbb{Z}_p^*$ we have $X_{w,\alpha}(Z_{w,\alpha}(x)) = X_{w,\alpha}(w + x^{-1} \cdot \alpha) = \alpha \cdot (w + x^{-1} \cdot \alpha - w)^{-1} = \alpha \cdot x \cdot \alpha^{-1} = x$. ▮

Equipped with this lemma, we give the proof of Theorem 5.1.

Proof of Theorem 5.1: By $\mathrm{span}(\vec{v})$ we denote the span of a vector $\vec{v} \in \mathbb{Z}_p^{n+2}$, which simply means the set of all $a \cdot \vec{v}$ as a ranges over $\mathbb{Z}_p$. Beyond the procedures of game $\mathbf{G}^{\text{gg-mbdl}}_{G,n,m}$, some of our games define procedures VE and VE^{-1}, the vector-encoding and its inverse. These procedures are not exported, meaning can be called only by other game procedures, not by the adversary. Throughout, we assume the adversary $\mathcal{A}$ makes no trivial queries. By this we mean that the checks at lines 7 and 10 of game $\mathbf{G}^{\text{gg-mbdl}}_{G,n,m}$ are not triggered. In our games the consequence is that we assume $\mathrm{TI}[A], \mathrm{TI}[B] \neq \perp$ in any OP(A, B, sgn) query and, for a DLO(i, W) query, that $i \in [n]$, that $\mathrm{TI}[W] \neq \perp$ and that the number of queries to this oracle is exactly $m = 1$. (The table $\mathrm{TI}[\cdot]$ referred to here starts appearing in Game Gm_0 of Figure 9.)

We start with game Gm_0 of Figure 9, claiming that

$$\mathbf{Adv}^{\text{gg-mbdl}}_{G,n,m}(\mathcal{A}) = \Pr[\mathrm{Gm}_0(\mathcal{A})] \,. \tag{18}$$

We now explain the game and justify Eq. (18). At line 10, operation sgn is performed modulo p, and at line 11, the inverse and product in computing z are modulo p. The game picks $y, x_1, \ldots, x_n$ in the same way as game $\mathbf{G}^{\text{gg-mbdl}}_{G,n,m}$. At line 1, it also picks encoding function E in the same way as game $\mathbf{G}^{\text{gg-mbdl}}_{G,n,m}$, but does not use this function directly to do the encoding, instead calling VE, which we call the vector-encoding function, on the indicated vector arguments. This procedure

VE($\vec{t}$): // $\boxed{\mathrm{Gm}_1}$, Gm_2. Here $\vec{t} \in \mathbb{Z}_p^{n+2}$.
13 If ($\mathrm{TV}[\vec{t}] \neq \perp$) then return $\mathrm{TV}[\vec{t}]$
14 If ($\exists\, \vec{t}'$: ($\mathrm{TV}[\vec{t}'] \neq \perp$ and $\vec{t} - \vec{t}' \in \mathrm{span}(\vec{v})$)) then
15 $C \leftarrow \mathrm{TV}[\vec{t}']$; $\mathrm{TV}[\vec{t}] \leftarrow C$; $\mathrm{TI}[C] \leftarrow \vec{t}$; Return $\mathrm{TV}[\vec{t}]$
16 $C \leftarrow\!\!\$\ G \setminus \mathrm{GL}$
17 If ($\exists\, \vec{t}'$: ($\mathrm{TV}[\vec{t}'] \neq \perp$ and $\langle \vec{t}, \vec{x} \rangle = \langle \vec{t}', \vec{x} \rangle$)) then
18 bad $\leftarrow$ true ; $\boxed{C \leftarrow \mathrm{TV}[\vec{t}']}$
19 $\mathrm{TV}[\vec{t}] \leftarrow C$; $\mathrm{TI}[C] \leftarrow \vec{t}$; $\mathrm{GL} \leftarrow \mathrm{GL} \cup \{C\}$; Return $\mathrm{TV}[\vec{t}]$

DLO(i, W): // Gm_1, Gm_2. Here $i \in [n]$ and $\mathrm{TI}[W] \neq \perp$.
20 $\vec{w} \leftarrow \mathrm{VE}^{-1}(W)$; $z \leftarrow (x_i)^{-1} \cdot \langle \vec{w}, \vec{x} \rangle$; $\vec{v} \leftarrow \vec{w} - z \cdot \vec{e}_i$
21 Return z

Figure 10: Procedures for games $\mathrm{Gm}_1, \mathrm{Gm}_2, \mathrm{Gm}_3$ in the proof of Theorem 5.1, where Gm_1 includes the boxed code.

---

maintains tables $\mathrm{TV} : \mathbb{Z}_p^{n+2} \to G \cup \{\perp\}$ and $\mathrm{TI} : G \to \mathbb{Z}_p^{n+2} \cup \{\perp\}$ (the "I" stands for "inverse") that from the code can be seen to satisfy the following, where vector $\vec{x}$ is defined at line 3:

(1) If $\mathrm{TV}[\vec{t}] \neq \perp$ then $\mathrm{TV}[\vec{t}] = \mathrm{E}(\langle \vec{t}, \vec{x} \rangle)$
(2) If $\mathrm{TI}[C] \neq \perp$ then $\langle \mathrm{TI}[C], \vec{x} \rangle = \mathrm{E}^{-1}(C)$

This ensures Eq. (18) as follows. From line 4 and the above we have $g = \mathrm{TV}[\vec{e}_{n+1}] = \mathrm{E}(\langle \vec{e}_{n+1}, \vec{x} \rangle) = \mathrm{E}(1)$, and, similarly, we have $Y = \mathrm{E}(y)$ and $X_i = \mathrm{E}(x_i)$ for $i \in [1..n]$, meaning these quantities are as in game $\mathbf{G}^{\text{gg-mbdl}}_{G,n,m}$. Turning to OP, by linearity of the inner product and item (2) above, we have

$$\begin{aligned} \langle \vec{c}, \vec{x} \rangle &= \langle \mathrm{TI}[A] \ \mathsf{sgn}\ \mathrm{TI}[B], \vec{x} \rangle = \langle \mathrm{TI}[A], \vec{x} \rangle \ \mathsf{sgn}\ \langle \mathrm{TI}[B], \vec{x} \rangle \\ &= \mathrm{E}^{-1}(A) \ \mathsf{sgn}\ \mathrm{E}^{-1}(B) \ , \end{aligned}$$

so by item (1) we have $\mathrm{VE}(\vec{c}) = \mathrm{E}(\mathrm{E}^{-1}(A) \ \mathsf{sgn}\ \mathrm{E}^{-1}(B))$, as in game $\mathbf{G}^{\text{gg-mbdl}}_{G,n,m}$. Finally, for DLO, item (2) says that $\langle \vec{w}, \vec{x} \rangle = \mathrm{E}^{-1}(W)$, again as in game $\mathbf{G}^{\text{gg-mbdl}}_{G,n,m}$.

Games $\mathrm{Gm}_1, \mathrm{Gm}_2$ are formed by taking the indicated procedures of Figure 9 and adding those of Figure 10, with the former game including the boxed code, and the latter not. Procedure VE no longer invokes E, instead sampling it lazily. The vector $\vec{v}$ defined at line 20 satisfies $\langle \vec{v}, \vec{x} \rangle = \langle \vec{w} - z \cdot \vec{e}_i, \vec{x} \rangle = \langle \vec{w}, \vec{x} \rangle - z \cdot \langle \vec{e}_i, \vec{x} \rangle = \langle \vec{w}, \vec{x} \rangle - x_i^{-1} \cdot \langle \vec{w}, \vec{x} \rangle \cdot x_i = 0$. As a result, at any time, any vector $\vec{u} \in \mathrm{span}(\vec{v})$ satisfies $\langle \vec{u}, \vec{x} \rangle = 0$. Now we claim that

$$\Pr[\mathrm{Gm}_1(\mathcal{A})] = \Pr[\mathrm{Gm}_0(\mathcal{A})] \ . \tag{19}$$

Let us justify this. If the "If" statement at line 14 is true, we have, by the above, $\langle \vec{t} - \vec{t}', \vec{x} \rangle = 0$, or $\langle \vec{t}, \vec{x} \rangle = \langle \vec{t}', \vec{x} \rangle$, and so, as per line 8 of Figure 9, ought indeed to set $\mathrm{TV}[\vec{t}] = \mathrm{TV}[\vec{t}']$. The inclusion of the boxed code at line 18 further ensures consistency with line 8 of Figure 9. So VE is returning the same things in games $\mathrm{Gm}_1, \mathrm{Gm}_0$. While DLO defines some new quantities, what it returns does not change compared to game Gm_0. This concludes the justification of Eq. (19).

Games $\mathrm{Gm}_1, \mathrm{Gm}_2$ are identical-until-bad as defined in [11]. Let B_2 be the event that $\mathrm{Gm}_2(\mathcal{A})$ sets bad. Then by the Fundamental Lemma of Game Playing [11],

$$\Pr[\mathrm{Gm}_1(\mathcal{A})] \leq \Pr[\mathrm{Gm}_2(\mathcal{A}) \text{ and } \overline{\mathrm{B}}_2] + \Pr[\mathrm{B}_2] \ , \tag{20}$$

INIT(): // Gm_3–Gm_5, $\mathrm{Gm}_{\alpha,\beta}$.
1 $p \leftarrow |G|$; $\mathbb{1} \leftarrow \mathrm{VE}(\vec{0})$; $g \leftarrow \mathrm{VE}(\vec{e}_{n+1})$; $Y \leftarrow \mathrm{VE}(\vec{e}_{n+2})$
2 For $i = 1, \ldots, n$ do $X_i \leftarrow \mathrm{VE}(\vec{e}_i)$
3 Return $\mathbb{1}, g, Y, X_1, \ldots, X_n$

$\mathrm{VE}(\vec{t})$: // Gm_3–Gm_5, $\mathrm{Gm}_{\alpha,\beta}$. Here $\vec{t} \in \mathbb{Z}_p^{n+2}$.
4 If ($\mathrm{TV}[\vec{t}] \neq \perp$) then return $\mathrm{TV}[\vec{t}]$
5 $C \leftarrow\!\!\$\ G \setminus \mathrm{GL}$
6 If ($\exists\, \vec{t}'$: ($\mathrm{TV}[\vec{t}'] \neq \perp$ and $\vec{t} - \vec{t}' \in \mathrm{span}(\vec{v})$)) then $C \leftarrow \mathrm{TV}[\vec{t}']$
7 Else $k \leftarrow k + 1$; $\vec{t}_k \leftarrow \vec{t}$; $\mathrm{GL} \leftarrow \mathrm{GL} \cup \{C\}$
8 $\mathrm{TV}[\vec{t}] \leftarrow C$; $\mathrm{TI}[C] \leftarrow \vec{t}$; Return $\mathrm{TV}[\vec{t}]$

$\mathrm{VE}^{-1}(C)$: // Gm_3–Gm_5, $\mathrm{Gm}_{\alpha,\beta}$. Here $\mathrm{TI}[C] \neq \perp$.
9 Return $\mathrm{TI}[C]$

OP(A, B, sgn): // Gm_3–Gm_5, $\mathrm{Gm}_{\alpha,\beta}$. Here $\mathrm{TI}[A], \mathrm{TI}[B] \neq \perp$ and $\mathsf{sgn} \in \{+, -\}$
10 $\vec{c} \leftarrow \mathrm{VE}^{-1}(A)\ \mathsf{sgn}\ \mathrm{VE}^{-1}(B)$; $C \leftarrow \mathrm{VE}(\vec{c})$; Return C

DLO(i, W): // $\boxed{\mathrm{Gm}_3}$, Gm_4. Here $i \in [n]$ and $\mathrm{TI}[W] \neq \perp$.
11 $\vec{w} \leftarrow \mathrm{VE}^{-1}(W)$; $w \leftarrow \vec{w}[i]$
12 If $(\vec{w} - w \cdot \vec{e}_i = \vec{0})$ then return w
13 $z \leftarrow\!\!\$\ \mathbb{Z}_p \setminus \{w\}$; $y \leftarrow\!\!\$\ \mathbb{Z}_p$; $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n \leftarrow\!\!\$\ \mathbb{Z}_p^*$
14 $\vec{x}_i \leftarrow (x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n, 1, y)$; $x_i \leftarrow (z - w)^{-1} \cdot \langle \vec{w}, \vec{x}_i \rangle$
15 If ($\langle \vec{w}, \vec{x}_i \rangle = 0$) then $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{z \leftarrow w\ ;\ x_i \leftarrow\!\!\$\ \mathbb{Z}_p^*}$
16 $\vec{v} \leftarrow \vec{w} - z \cdot \vec{e}_i$; Return z

FIN(y'): // Gm_3, Gm_4.
17 $\vec{x} \leftarrow (x_1, \ldots, x_n, 1, y)$
18 Return ($(y = y')$ or $(\exists\, \alpha, \beta : 1 \leq \alpha < \beta \leq k$ and $\langle \vec{t}_\alpha - \vec{t}_\beta, \vec{x} \rangle = 0$))

Figure 11: Procedures for games $\mathrm{Gm}_3, \mathrm{Gm}_4$ in the proof of Theorem 5.1. Some procedures, as marked, will be used in later games.

---

where $\overline{\mathrm{B}}_2$ denotes the complement of event B_2. We claim that

$$\Pr[\mathrm{Gm}_2(\mathcal{A}) \text{ and } \overline{\mathrm{B}}_2] + \Pr[\mathrm{B}_2] \leq \Pr[\mathrm{Gm}_3(\mathcal{A})] \ , \tag{21}$$

where game Gm_3 is in Figure 11. It includes the boxed code, which game Gm_4 excludes. In these games, VE returns the same thing as in game Gm_2, but also indexes (keeps track of) vectors $\vec{t}$ that might set bad in Gm_2, so that it can refer to them in FIN. The achievement is that this procedure no longer refers to $\vec{x}$. Now we would like the same to be true for DLO. A natural approach would be to have DLO return a random $z \leftarrow\!\!\$\ \mathbb{Z}_p$. However, the true distribution of z is more complex, and instead we will use Lemma 5.2. Line 11 sets $w \in \mathbb{Z}_p$ to be the i-th coordinate of vector $\vec{w}$. Line 12 checks if $\vec{w}$ is 0 at all but its i-th coordinate, if so correctly returning w as the answer to the oracle query. At lines 13,14, the choices of z and x_i are made in accordance with one case of Lemma 5.2, with y, and the x_j for $j \neq i$, chosen correctly. Line 15 checks if it is the other case that happened, and, if so, game Gm_3 corrects the choices of z, x_i according to the Lemma. The Lemma thus implies that in game Gm_3, the returned z is distributed as it is in game Gm_2. FIN of game Gm_3 returns true if either $y = y'$, or game Gm_2 would set bad, justifying Eq. (21).

DLO(i, W): // Gm_5, $\mathrm{Gm}_{\alpha,\beta}$. Here $i \in [n]$ and $\mathrm{TI}[W] \neq \perp$.
19 $\vec{w} \leftarrow \mathrm{VE}^{-1}(W)$; $w \leftarrow \vec{w}[i]$
20 If $(\vec{w} - w \cdot \vec{e}_i = \vec{0})$ then return w
21 $z \leftarrow\!\!\$\ \mathbb{Z}_p \setminus \{w\}$; $\vec{v} \leftarrow \vec{w} - z \cdot \vec{e}_i$; Return z

FIN(y'): // Gm_5.
22 $y \leftarrow\!\!\$\ \mathbb{Z}_p$; Return $(y = y')$

FIN(y'): // $\mathrm{Gm}_{\alpha,\beta}$.
23 If (not $(1 \leq \alpha < \beta \leq k)$) then return false
24 $y \leftarrow\!\!\$\ \mathbb{Z}_p$; $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n \leftarrow\!\!\$\ \mathbb{Z}_p^*$
25 $\vec{x}_i \leftarrow (x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n, 1, y)$; $x_i \leftarrow (z - w)^{-1} \cdot \langle \vec{w}, \vec{x}_i \rangle$
26 $\vec{x} \leftarrow (x_1, \ldots, x_n, 1, y)$
27 Return ($\langle \vec{t}_\alpha - \vec{t}_\beta, \vec{x} \rangle = 0$)

Figure 12: Further procedures to define game Gm_5 and games $\mathrm{Gm}_{\alpha,\beta}$ $(1 \leq \alpha < \beta \leq q)$ in the proof of Theorem 5.1.

Games $\mathrm{Gm}_3, \mathrm{Gm}_4$ are identical-until-bad, so by the Fundamental Lemma of Game Playing [11],

$$\Pr[\mathrm{Gm}_3(\mathcal{A})] \leq \Pr[\mathrm{Gm}_4(\mathcal{A})] + \Pr[\mathrm{Gm}_4(\mathcal{A}) \text{ sets bad}] \ . \tag{22}$$

We claim

$$\Pr[\mathrm{Gm}_4(\mathcal{A}) \text{ sets bad}] \leq \frac{1}{p-1} \ . \tag{23}$$

That is, the probability that $\langle \vec{w}, \vec{x}_i \rangle = 0$ at line 15 is at most $1/(p-1)$. We now justify this. Line 12 tells us that, at line 15, there is some $j \in [1..n+2] \setminus \{i\}$ such that $\vec{w}[j] \neq 0$. Consider two cases. The first is that there is such a j satisfying $j \neq n+1$. If $j = n+2$, there is exactly one choice of $y \in \mathbb{Z}_p$ making $\langle \vec{w}, \vec{x}_i \rangle = 0$, while if $j \in [1..n] \setminus \{i\}$, there is at most one choice of $x_j \in \mathbb{Z}_p^*$ making $\langle \vec{w}, \vec{x}_i \rangle = 0$, so overall the probability that $\langle \vec{w}, \vec{x}_i \rangle = 0$ is at most $1/(p-1)$. The second case is that $\vec{w}[j] = 0$ for all $j \neq n+1$. But then the probability that $\langle \vec{w}, \vec{x}_i \rangle = 0$ is zero. This completes the justification of Eq. (23).

We now define a game Gm_5, and also a game $\mathrm{Gm}_{\alpha,\beta}$ for each $1 \leq \alpha < \beta \leq q$, where $q = \mathrm{Q}_{\mathcal{A}}^{\mathrm{OP}} + n + 3$. The DLO, FIN procedures of these games are shown in Figure 12, and the other procedures remain as in Figure 11. Since the boxed code is absent in DLO of game Gm_4, the only random choice it needs to make is z, yielding the simplified DLO procedure of Figure 12. The other random choices are delayed to FIN. The event resulting in game Gm_4 returning true is broken up in the new games so that, by the union bound,

$$\Pr[\mathrm{Gm}_4(\mathcal{A})] \leq \Pr[\mathrm{Gm}_5(\mathcal{A})] + \sum_{1 \leq \alpha < \beta \leq q} \Pr[\mathrm{Gm}_{\alpha,\beta}(\mathcal{A})] \ . \tag{24}$$

Clearly

$$\Pr[\mathrm{Gm}_5(\mathcal{A})] \leq \frac{1}{p} \ . \tag{25}$$

Now, fix any $1 \leq \alpha < \beta \leq q$. We assume wlog that k always equals q. In game $\mathrm{Gm}_{\alpha,\beta}$, let $\vec{d} = \vec{t}_\alpha - \vec{t}_\beta$, let $a = (z - w)^{-1}$ and let $\vec{u} = a \cdot \vec{d}[i] \cdot \vec{w} + \vec{d}$. Let Z be the event that $\langle \vec{d}, \vec{x} \rangle = 0$, and let S be the event that $\vec{d} \in \mathrm{span}(\vec{v})$. Then

$$\Pr[\mathrm{Gm}_{\alpha,\beta}(\mathcal{A})] = \Pr[\mathrm{Z}] = \Pr[\mathrm{Z} \text{ and } \overline{\mathrm{S}}] + \Pr[\mathrm{Z} \text{ and } \mathrm{S}]$$

$$\leq \Pr[\,\mathrm{Z}\,|\,\overline{\mathrm{S}}\,] + \Pr[\mathrm{S}]\ . \tag{26}$$

We will show that

$$\Pr[\,\mathrm{Z}\,|\,\overline{\mathrm{S}}\,] \leq \frac{1}{p-1} \tag{27}$$

$$\Pr[\mathrm{S}] \leq \frac{1}{p-1}\ . \tag{28}$$

We now justify Eq. (27). We have

$$\begin{aligned}\langle \vec{d}, \vec{x}\rangle &= x_i \cdot \vec{d}[i] + \langle \vec{d}, \vec{x}_i\rangle = a \cdot \langle \vec{w}, \vec{x}_i\rangle \cdot \vec{d}[i] + \langle \vec{d}, \vec{x}_i\rangle \\ &= \langle a \cdot \vec{d}[i] \cdot \vec{w} + \vec{d}, \vec{x}_i\rangle = \langle \vec{u}, \vec{x}_i\rangle\end{aligned}$$

Assume $\vec{d} \notin \mathrm{span}(\vec{v})$, meaning event $\overline{\mathrm{S}}$ happens. Then we claim (we will justify this in a bit) that there exists a $j \in [1..n+2] \setminus \{i, n+1\}$ such that $\vec{u}[j] \neq 0$. This means that the random choice of either x_j (if $j \in [1..n] \setminus \{i\}$) or y (if $j = n+2$) has probability at most $1/(p-1)$ of making $\langle \vec{u}, \vec{x}_i\rangle = 0$. To justify the claim, suppose to the contrary that for all $j \in [1..n+2] \setminus \{i, n+1\}$ we have $\vec{u}[j] = 0$. Since $\langle \vec{u}, \vec{x}_i\rangle = 0$, it must be that $\vec{u}[n+1] = 0$ as well. Let $b = -a \cdot \vec{d}[i]$, so that $\vec{d}[i] = -b \cdot a^{-1} = -b \cdot (z - w) = b \cdot (w - z)$. For $j \in [1..n+2] \setminus \{i\}$ we have $a \cdot \vec{d}[i] \cdot \vec{w}[j] + \vec{d}[j] = 0$, or $\vec{d}[j] = -a \cdot \vec{d}[i] \cdot \vec{w}[j] = b \cdot \vec{w}[j]$. Recalling that $\vec{v} = \vec{w} - z \cdot \vec{e}_i$ and $w = \vec{w}[i]$, we see that $\vec{d} = b \cdot \vec{v}$, which puts $\vec{d}$ in $\mathrm{span}(\vec{v})$, contradicting our assumption that $\vec{d} \notin \mathrm{span}(\vec{v})$. This concludes the justification of Eq. (27).

We turn to Eq. (28). Suppose $\vec{d} \in \mathrm{span}(\vec{v})$, meaning $\vec{d} = b \cdot \vec{v} = b \cdot \vec{w} - bz \cdot \vec{e}_i$ for some $b \in \mathbb{Z}_p^*$. By line 4 of Figure 11, $\vec{t}_\alpha \neq \vec{t}_\beta$, so $\vec{d} \neq \vec{0}$ so $b \neq 0$. So there is at most one $z \in \mathbb{Z}_p$ such that $\vec{d}[i] = bw - bz$, and our z chosen at random from $\mathbb{Z}_p \setminus \{w\}$ has probability at most $1/(p-1)$ of being this one.

Putting the above together we have

$$\begin{aligned}\mathbf{Adv}^{\text{gg-mbdl}}_{G,n,m}(\mathcal{A}) &\leq \frac{1}{p-1} + \frac{1}{p} + \frac{q(q-1)}{2}\frac{2}{p-1} \\ &= \frac{1 + q(q-1)}{p-1} + \frac{1}{p}\ .\end{aligned}$$

This concludes the proof. ∎

References

[1] M. Abdalla, J. H. An, M. Bellare, and C. Namprempre. From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In L. R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 418–433. Springer, Heidelberg, Apr. / May 2002. 3, 6, 10, 15

[2] M. Abe, M. Ohkubo, and K. Suzuki. 1-out-of-n signatures from a variety of keys. In Y. Zheng, editor, *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 415–432. Springer, Heidelberg, Dec. 2002. 3, 6

[3] A. Bagherzandi, J. H. Cheon, and S. Jarecki. Multisignatures secure under the discrete logarithm assumption and a generalized forking lemma. In P. Ning, P. F. Syverson, and S. Jha, editors, *ACM CCS 2008*, pages 449–458. ACM Press, Oct. 2008. 3

[4] J. Bartusek, F. Ma, and M. Zhandry. The distinction between fixed and random generators in group-based assumptions. In A. Boldyreva and D. Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 801–830. Springer, Heidelberg, Aug. 2019. 10

[5] M. Bellare, C. Namprempre, D. Pointcheval, and M. 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. 4, 7, 8

[6] M. Bellare and G. Neven. Multi-signatures in the plain public-key model and a general forking lemma. In A. Juels, R. N. Wright, and S. De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 390–399. ACM Press, Oct. / Nov. 2006. 3, 6, 16, 29

[7] M. Bellare and A. Palacio. GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In M. Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 162–177. Springer, Heidelberg, Aug. 2002. 3, 4, 6, 11, 28

[8] M. Bellare and A. Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 273–289. Springer, Heidelberg, Aug. 2004. 6, 7

[9] M. Bellare, B. Poettering, and D. Stebila. From identification to signatures, tightly: A framework and generic transforms. In J. H. Cheon and T. Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 435–464. Springer, Heidelberg, Dec. 2016. 11

[10] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In D. E. Denning, R. Pyle, R. Ganesan, R. S. Sandhu, and V. Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, Nov. 1993. 3, 15

[11] M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, Heidelberg, May / June 2006. 8, 12, 21, 23

[12] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang. High-speed high-security signatures. *Journal of cryptographic engineering*, 2(2):77–89, 2012. 3

[13] A. Boldyreva, C. Gentry, A. O'Neill, and D. H. Yum. Ordered multisignatures and identity-based sequential aggregate signatures, with applications to secure routing. In P. Ning, S. De Capitani di Vimercati, and P. F. Syverson, editors, *ACM CCS 2007*, pages 276–285. ACM Press, Oct. 2007. 18

[14] A. Boldyreva, C. Gentry, A. O'Neill, and D. H. Yum. Ordered multisignatures and identity-based sequential aggregate signatures, with applications to secure routing. Cryptology ePrint Archive, Report 2007/438, 2007. https://eprint.iacr.org/2007/438. 18

[15] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, Heidelberg, Aug. 2004. 7

[16] D. Boneh and M. K. Franklin. Identity based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003. 7

[17] C. Cachin, S. Micali, and M. Stadler. Computationally private information retrieval with polylogarithmic communication. In J. Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 402–414. Springer, Heidelberg, May 1999. 7

[18] R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003. 7

[19] I. Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In J. Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 445–456. Springer, Heidelberg, Aug. 1992. 6

[20] E. De Cristofaro and G. Tsudik. Practical private set intersection protocols with linear complexity. In R. Sion, editor, *FC 2010*, volume 6052 of *LNCS*, pages 143–159. Springer, Heidelberg, Jan. 2010. 4

[21] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976. 4

[22] M. Drijvers, K. Edalatnejad, B. Ford, E. Kiltz, J. Loss, G. Neven, and I. Stepanovs. On the security of two-round multi-signatures. In *2019 IEEE Symposium on Security and Privacy*, pages 1084–1101. IEEE Computer Society Press, May 2019. 4

[23] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31:469–472, 1985. 4

[24] U. Feige, A. Fiat, and A. Shamir. Zero-knowledge proofs of identity. *Journal of Cryptology*, 1(2):77–94, June 1988. 3, 10

[25] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, Aug. 1987. 3, 15, 29

[26] M. Fischlin and N. Fleischhacker. Limitations of the meta-reduction technique: The case of Schnorr signatures. In T. Johansson and P. Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 444–460. Springer, Heidelberg, May 2013. 4

[27] G. Fuchsbauer, E. Kiltz, and J. Loss. The algebraic group model and its applications. In H. Shacham and A. Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 33–62. Springer, Heidelberg, Aug. 2018. 4, 6

[28] G. Fuchsbauer, A. Plouviez, and Y. Seurin. Blind schnorr signatures and signed ElGamal encryption in the algebraic group model. In A. Canteaut and Y. Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 63–95. Springer, Heidelberg, May 2020. 6

[29] S. D. Galbraith and P. Gaudry. Recent progress on the elliptic curve discrete logarithm problem. *Designs, Codes and Cryptography*, 78(1):51–72, 2016. 9

[30] S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, Apr. 1988. 3, 15

[31] S. Hada and T. Tanaka. On the existence of 3-round zero-knowledge protocols. In H. Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 408–423. Springer, Heidelberg, Aug. 1998. 6

[32] J. Herranz and G. Sáez. Forking lemmas for ring signature schemes. In T. Johansson and S. Maitra, editors, *INDOCRYPT 2003*, volume 2904 of *LNCS*, pages 266–279. Springer, Heidelberg, Dec. 2003. 3

[33] J. Y. Hwang, D. H. Lee, and M. Yung. Universal forgery of the identity-based sequential aggregate signature scheme. In W. Li, W. Susilo, U. K. Tupakula, R. Safavi-Naini, and V. Varadharajan, editors, *ASIACCS 09*, pages 157–160. ACM Press, Mar. 2009. 18

[34] IANIX. Things that use Ed25519. https://ianix.com/pub/ed25519-deployment.html. 3

[35] M. J. Jacobson, N. Koblitz, J. H. Silverman, A. Stein, and E. Teske. Analysis of the xedni calculus attack. *Designs, Codes and Cryptography*, 20(1):41–64, 2000. 9

[36] E. Kiltz, D. Masny, and J. Pan. Optimal security proofs for signatures from identification schemes. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 33–61. Springer, Heidelberg, Aug. 2016. 3, 6, 7, 30

[37] C. Komlo and I. Goldberg. FROST: Flexible round-optimized schnorr threshold signatures. Cryptology ePrint Archive, Report 2020/852, 2020. https://eprint.iacr.org/2020/852. 3

[38] G. Maxwell, A. Poelstra, Y. Seurin, and P. Wuille. Simple schnorr multi-signatures with applications to bitcoin. Cryptology ePrint Archive, Report 2018/068, 2018. https://eprint.iacr.org/2018/068. 3

[39] S. Micali, K. Ohta, and L. Reyzin. Accountable-subgroup multisignatures: Extended abstract. In M. K. Reiter and P. Samarati, editors, *ACM CCS 2001*, pages 245–254. ACM Press, Nov. 2001. 3

[40] M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. *Journal of the ACM*, 51(2):231–262, 2004. 7

[41] K. Ohta and T. Okamoto. On concrete security treatment of signatures derived from identification. In H. Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 354–369. Springer, Heidelberg, Aug. 1998. 6

[42] T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In E. F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 31–53. Springer, Heidelberg, Aug. 1993. 6, 27

[43] P. Paillier and D. Vergnaud. Discrete-log-based signatures may not be equivalent to discrete log. In B. K. Roy, editor, *ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 1–20. Springer, Heidelberg, Dec. 2005. 4

[44] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, June 2000. 3, 6, 16

[45] C.-P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, Jan. 1991. 3, 11

[46] I. Semaev. Summation polynomials and the discrete logarithm problem on elliptic curves. Cryptology ePrint Archive, Report 2004/031, 2004. https://eprint.iacr.org/2004/031. 9

[47] V. Shoup. Lower bounds for discrete logarithms and related problems. In W. Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, Heidelberg, May 1997. 3, 4, 6, 14, 16, 18

[48] J. H. Silverman. The xedni calculus and the elliptic curve discrete logarithm problem. *Designs, Codes and Cryptography*, 20(1):5–40, 2000. 9

[49] J. H. Silverman and J. Suzuki. Elliptic curve discrete logarithms and the index calculus. In K. Ohta and D. Pei, editors, *ASIACRYPT'98*, volume 1514 of *LNCS*, pages 110–125. Springer, Heidelberg, Oct. 1998. 9

[50] D. R. Stinson and R. Strobl. Provably secure distributed Schnorr signatures and a (t, n) threshold scheme for implicit certificates. In V. Varadharajan and Y. Mu, editors, *ACISP 01*, volume 2119 of *LNCS*, pages 417–434. Springer, Heidelberg, July 2001. 3, 6

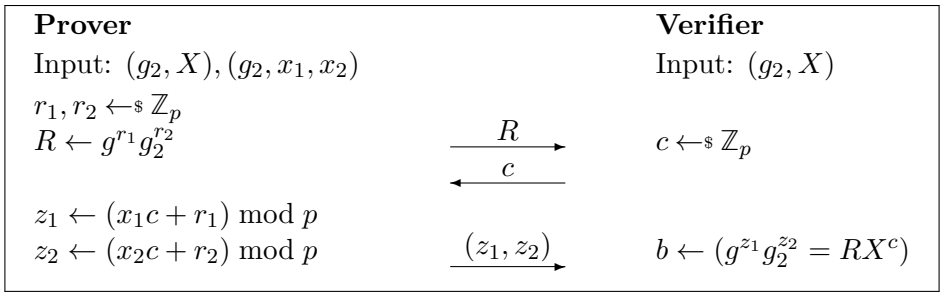
[51] A. Yun. Generic hardness of the multiple discrete logarithm problem. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 817–836. Springer, Heidelberg, Apr. 2015. 18

A Okamoto Identification and Signatures from MBDL

In this section, we give a *tight* reduction of the IMP-PA security of the Okamoto identification scheme to the 1-MBDL problem and derive a corresponding improvement for Okamoto signatures.

OKAMOTO IDENTIFICATION SCHEME AND PRIOR RESULTS. Let $\mathbb{G}$ be a group of prime order $p = |\mathbb{G}|$, and $g \in \mathbb{G}^*$ a generator of $\mathbb{G}$. We recall the Okamoto identification scheme [42] $\mathsf{ID} = \mathsf{OkaID}[\mathbb{G}, g]$ in Fig. 13. The public key has the form $vk = (g_2, X) \in \mathbb{G}^2$ where g_2 is a generator and $X = g^{x_1} g_2^{x_2}$, where the secret key is $sk = (g_2, x_1, x_2) \in \mathbb{Z}_p^3$. The commitment is $R = g^{r_1} g_2^{r_2} \in \mathbb{G}$, and (r_1, r_2) is returned as the prover state by the commitment algorithm. Challenges are drawn from $\mathsf{ID.Ch} = \mathbb{Z}_p$, and the response z and decision b are computed as shown.

Given an IMP-PA adversary $\mathcal{A}$ against $\mathsf{ID} = \mathsf{OkaID}[\mathbb{G}, g]$, the classical proof of [42] builds a DL-adversary $\mathcal{B}$, as follows. On input a target point Y whose discrete-log it wants to compute, $\mathcal{B}$ sets $g_2 = Y$. It then itself picks x_1, x_2 and sets $X = g^{x_1} g_2^{x_2}$, so that (x_1, x_2) is what's called a representation of X. Now $\mathcal{B}$ runs $\mathcal{A}$ on public key (g_2, X). Knowing the secret key (g_2, x_1, x_2), it is easy for $\mathcal{B}$ to simulate the Tr oracle. When $\mathcal{A}$ makes its impersonation attempt, rewinding is used, as usual, to obtain two accepting conversation transcripts with the same commitment R_*. From these, $\mathcal{B}$ can compute another representation of X, namely some a_1, a_2 such that $X = g^{a_1} g_2^{a_2}$. The witness indistinguishability property of the protocol says that $(a_1, a_2) \neq (x_1, x_2)$, except with



ID.Kg:

1 $g_2 \leftarrow\!\!\$\ \mathbb{G}^*$
2 $x_1, x_2 \leftarrow\!\!\$\ \mathbb{Z}_{|\mathbb{G}|}$; $X \leftarrow g^{x_1} g_2^{x_2}$
3 Return $((g_2, X), (g_2, x_1, x_2))$

ID.Cmt$((g_2, X))$:

4 $r_1, r_2 \leftarrow\!\!\$\ \mathbb{Z}_{|\mathbb{G}|}$; $R \leftarrow g^{r_1} g_2^{r_2}$
5 Return $(R, (r_1, r_2))$

ID.Rsp$((g_2, x_1, x_2), c, (r_1, r_2))$:

6 $z_1 \leftarrow (x_1 c + r_1) \mod |\mathbb{G}|$
7 $z_2 \leftarrow (x_2 c + r_2) \mod |\mathbb{G}|$
8 Return (z_1, z_2)

ID.Vf$(X, R, c, (z_1, z_2))$:

9 $b \leftarrow (g^{z_1} g_2^{z_2} = X^c R)$; Return b

DS.Kg:

1 $g_2 \leftarrow\!\!\$\ \mathbb{G}^*$
2 $x_1, x_2 \leftarrow\!\!\$\ \mathbb{Z}_{|\mathbb{G}|}$; $X \leftarrow g^{x_1} g_2^{x_2}$
3 Return $((g_2, X), (g_2, x_1, x_2))$

DS.Sign$^{\mathrm{H}}((g_2, x_1, x_2), m)$:

4 $r_1, r_2 \leftarrow\!\!\$\ \mathbb{Z}_{|\mathbb{G}|}$; $R \leftarrow g^{r_1} g_2^{r_2}$
5 $c \leftarrow \mathrm{H}(R, m)$
6 $z_1 \leftarrow (x_1 c + r_1) \mod |\mathbb{G}|$
7 $z_2 \leftarrow (x_2 c + r_2) \mod |\mathbb{G}|$
8 Return $(R, (z_1, z_2))$

DS.Vf$^{\mathrm{H}}((g_2, X), m, \sigma)$:

9 $(R, (z_1, z_2)) \leftarrow \sigma$
10 $c \leftarrow \mathrm{H}(R, m)$
11 Return $(g^{z_1} g_2^{z_2} = X^c R)$

Figure 13: Let $\mathbb{G}$ be a group of prime order $p = |\mathbb{G}|$ and let $g \in \mathbb{G}^*$ be a generator of $\mathbb{G}$. The Okamoto ID scheme $\mathsf{ID} = \mathsf{OkaID}[\mathbb{G}, g]$ is shown pictorially at the top and algorithmically at the bottom left. At the bottom right is the Okamoto signature scheme $\mathsf{DS} = \mathsf{OkaSig}[\mathbb{G}, g]$, using $\mathrm{H} : \mathbb{G} \times \{0,1\}^* \to \mathbb{Z}_p$.

---

probability $1/p$. Finally, from the two distinct representations of X, adversary $\mathcal{B}$ can compute $\mathsf{DL}_{\mathbb{G},g}(g_2)$. Again the simplest analysis is via the Reset Lemma of [7], which says that

$$\mathbf{Adv}^{\text{imp-pa}}_{\mathsf{ID}}(\mathcal{A}) \le \sqrt{\mathbf{Adv}^{\text{dl}}_{\mathbb{G},g}(\mathcal{B})} + \frac{2}{p}\ , \tag{29}$$

the extra $1/p$ term compared to Equation (7) being due to the probability that the two representations are equal. The running time $\mathrm{T}_{\mathcal{B}}$ of $\mathcal{B}$ is roughly $2\,\mathrm{T}_{\mathcal{A}}$ plus simulation overhead $\mathcal{O}(\mathrm{Q}^{\mathrm{Tr}}_{\mathcal{A}} \cdot T^{\exp}_{\mathbb{G}})$, where $T^{\exp}_{\mathbb{G}}$ is the time for an exponentiation in $\mathbb{G}$.

OUR RESULT. We show that the IMP-PA-security of the Okamoto identification scheme reduces *tightly* to the 1-MBDL problem. As with Schnorr, the reduction does not use rewinding.

Theorem A.1 *Let $\mathbb{G}$ be a group of prime order $p = |\mathbb{G}|$, and let $g \in \mathbb{G}^*$ be a generator of $\mathbb{G}$. Let $\mathsf{ID} = \mathsf{OkaID}[\mathbb{G}, g]$ be the Okamoto identification scheme. Let $\mathcal{A}$ be an adversary attacking the imp-pa security of ID. Then we can construct an adversary $\mathcal{B}$ (shown explicitly in Figure 14) such that*

$$\mathbf{Adv}^{\text{imp-pa}}_{\mathsf{ID}}(\mathcal{A}) \le \mathbf{Adv}^{\text{mbdl}}_{\mathbb{G},g,1}(\mathcal{B}) + \frac{1}{p}\ . \tag{30}$$

Additionally, $\mathrm{T}_{\mathcal{B}}$ is roughly $\mathrm{T}_{\mathcal{A}}$ plus simulation overhead $\mathcal{O}(\mathrm{Q}^{\mathrm{Tr}}_{\mathcal{A}} \cdot T^{\exp}_{\mathbb{G}})$.

Adversary $\mathcal{B}^{\mathrm{DLO}}$:

1 $(Y, X) \leftarrow\!\!\$\ \textsc{Init}()$; $w \leftarrow\!\!\$\ \mathbb{Z}_p^*$; $g_2 \leftarrow g^w$

2 $(z_1, z_2) \leftarrow\!\!\$\ \mathcal{A}^{\textsc{Ch},\textsc{Tr}}((g_2, X))$

3 Return $z_1 + wz_2$

$\textsc{Ch}(R_*)$:

4 $W \leftarrow R_*^{-1} \cdot Y$; $c_* \leftarrow \mathrm{DLO}(1, W)$; Return c_*

Tr:

5 $z_1, z_2 \leftarrow\!\!\$\ \mathbb{Z}_p$; $c \leftarrow\!\!\$\ \mathbb{Z}_p$; $R \leftarrow g^{z_1} g_2^{z_2} \cdot X^{-c}$; Return $(R, c, (z_1, z_2))$

Figure 14: MBDL adverary $\mathcal{B}$ for Theorem A.1, based on IMP-PA adversary $\mathcal{A}$.

Proof of of Theorem A.1: Our reduction from MBDL deviates from the prior one discussed above. It does not set g_2 to the target point Y, instead picking w and setting $g_2 = g^w$. It sets X to a base under which it can take a discrete logarithm. When adversary $\mathcal{A}$ provides R_* in its impersonation attempt, adversary $\mathcal{B}$ picks c_* so that $Y = R_* X^{c_*}$. Then, from $\mathcal{A}$, it gets (z_1, z_2) satisfying $g^{z_1} g_2^{z_2} = R_* X^{c_*} = Y$. Using w, adversary $\mathcal{B}$ then finds $\mathsf{DL}_{\mathbb{G},g}(Y)$. It simulates the Tr oracle using the zero-knowledge simulator. Thus, while in the prior approach the reduction knows the secret key but not $\mathsf{DL}_{\mathbb{G},g}(g_2)$, in ours the reduction does not know the secret key but knows $\mathsf{DL}_{\mathbb{G},g}(g_2)$.

For the formal proof, we claim that the adversary $\mathcal{B}$, shown in Fig. 14, satisfies Equation (30). Since the analysis is similar to that in the proof of Theorem 4.1, we will be brief. The X provided by $\mathcal{B}$ to $\mathcal{A}$ is a generator. In the scheme, $X = g^{x_1 + wx_2}$ fails to be generator iff $x_1 + wx_2 = 0$, which happens with probability $1/p$, accounting for this additive term in the bound. Adversary $\mathcal{B}$ simulates the transcript oracle correctly by the usual zero-knowledge method. If $\mathcal{A}$ succeeds, we have $g^{z_1} g_2^{z_2} = R_* X^{c_*}$. But $g^{z_1} g_2^{z_2} = g^{z_1 + wz_2}$ and $R_* X^{c_*} = Y$, so $z_1 + wz_2$ can be returned as the discrete log of Y. ∎

Okamoto signatures. The Okamoto signature scheme $\mathsf{DS} = \mathsf{OkaSig}[\mathbb{G}, g]$ is derived by applying the Fiat-Shamir transform [25] to the Okamoto identification scheme. Its algorithms are shown at the bottom right of Fig. 13. The set $\mathsf{DS.HF}$ consists of all functions $\mathsf{h} : \mathbb{G} \times \{0,1\}^* \to \mathbb{Z}_p$.

Combining Lemma 4.2 with Theorem A.1, we get the following reduction, of the UF security of the Okamoto signature scheme to the 1-MBDL problem, that loses only a factor of the number of hash-oracle queries of the adversary.

Theorem A.2 *Let $\mathbb{G}$ be a group of prime order $p = |\mathbb{G}|$, and let $g \in \mathbb{G}^*$ be a generator of $\mathbb{G}$. Let $\mathsf{DS} = \mathsf{OkaSig}[\mathbb{G}, g]$ be the Okamoto signature scheme. Let $\mathcal{A}$ be an adversary attacking the uf security of ID. Let $\beta = (1 + \mathrm{Q}^{\mathrm{H}}_{\mathcal{A}} + \mathrm{Q}^{\mathrm{Sign}}_{\mathcal{A}})\mathrm{Q}^{\mathrm{Sign}}_{\mathcal{A}} + (1 + \mathrm{Q}^{\mathrm{H}}_{\mathcal{A}})$. Then we can construct an adversary $\mathcal{B}$ such that*

$$\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}}(\mathcal{A}) \le (1 + \mathrm{Q}^{\mathrm{H}}_{\mathcal{A}}) \cdot \mathbf{Adv}^{\mathrm{mbdl}}_{\mathbb{G},g,1}(\mathcal{B}) + \frac{\beta}{p} \,. \tag{31}$$

Additionally, $\mathrm{T}_{\mathcal{B}}$ is roughly $\mathrm{T}_{\mathcal{A}}$ plus simulation overhead $\mathcal{O}(\mathrm{Q}^{\mathrm{Sign}}_{\mathcal{A}} \cdot T^{\mathrm{exp}}_{\mathbb{G}})$.

As before, the best prior result, obtained via the general Forking Lemma of [6], said that given an adversary $\mathcal{A}$ attacking the UF security of DS, one can construct a discrete log adversary $\mathcal{B}$ such

that

$$\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}}(\mathcal{A}) \leq \sqrt{(1+\mathrm{Q}^{\mathrm{H}}_{\mathcal{A}}) \cdot \mathbf{Adv}^{\mathrm{dl}}_{\mathbb{G},g}(\mathcal{B})} + \frac{\beta}{p}\,, \tag{32}$$

where β and $\mathrm{T}_{\mathcal{B}}$ are as above. Roughly the bound in Eq. (31) is the square of the one in Eq. (32), and thus (always) smaller.

B Ratio-based tightness

KMP [36] claims a tight reduction between passive impersonation security of Schnorr identification and discrete log. Their results are claimed to be tight when evaluated under time-to-sucesss ratio. We show here why their result does not give bounds that are as good as ours.

Let ID be the Schnoor identification scheme defined in Section 4. Let $\mathcal{A}$ be an adversary against the IMP-PA security of ID with running time $\mathrm{T}_{\mathcal{A}}$. For any given parameter $N \geq 1$, KMP [36] [Lemma 3.5] construct a DL adversary $\mathcal{D}_N$ such that

$$\sqrt{\mathbf{Adv}^{\mathrm{dl}}_{\mathbb{G},g}(\mathcal{D}_N)} \geq 1 - \left[1 - \left(\mathbf{Adv}^{\text{imp-pa}}_{\mathsf{ID}}(\mathcal{A}) - \frac{1}{p}\right)\right]^N\,, \tag{33}$$

and $\mathrm{T}_{\mathcal{D}_N} = 2N \cdot \mathrm{T}_{\mathcal{A}}$. Notice that when $N = 1$, this is identical to Eq. (7), meaning there is no improvement in that case. Next, KMP [36] pick a *specific* value of N that we call N^*. This value is $N^* = (\mathbf{Adv}^{\text{imp-pa}}_{\mathsf{ID}}(\mathcal{A}) - 1/p)^{-1}$. So the term on the right hand side of Eq. (33) becomes

$$1 - \left[1 - \left(\mathbf{Adv}^{\text{imp-pa}}_{\mathsf{ID}}(\mathcal{A}) - \frac{1}{p}\right)\right]^{N^*} \approx 1 - \frac{1}{e} \approx 0.63\,, \tag{34}$$

a constant close to 1. Let $\mathcal{B}^* = \mathcal{D}_{N^*}$ be the DL adversary for this parameter choice. Then, neglecting $1/p$ as being essentially 0, one has

$$\mathbf{Adv}^{\mathrm{dl}}_{\mathbb{G},g}(\mathcal{B}^*) \geq \left(1 - \frac{1}{e}\right)^2 \approx 0.4 \tag{35}$$

$$\mathrm{T}_{\mathcal{B}^*} = 2N^* \cdot \mathrm{T}_{\mathcal{A}} \approx \frac{\mathrm{T}_{\mathcal{A}}}{\mathbf{Adv}^{\text{imp-pa}}_{\mathsf{ID}}(\mathcal{A})}\,. \tag{36}$$

Dividing, they obtain the ratio tightness

$$\frac{\mathbf{Adv}^{\text{imp-pa}}_{\mathsf{ID}}(\mathcal{A})}{\mathrm{T}_{\mathcal{A}}} \leq \frac{\mathbf{Adv}^{\mathrm{dl}}_{\mathbb{G},g}(\mathcal{B}^*)}{\mathrm{T}_{\mathcal{B}^*}}\,. \tag{37}$$

"Tightness" is claimed because the time-to-success ratio is preserved. However, we will show that one cannot use the above to instantiate parameters that as competitive as the ones guaranteed by our bounds. This is because the running time $\mathrm{T}_{\mathcal{B}^*}$ from Eq. (36) is in general much larger than $\mathrm{T}_{\mathcal{A}}$ and the ratio tightness only holds when the running time of the DL adversary is increased in this way to make its advantage a constant as per Eq. (35).

As before, let us the GGM bound for breaking DL, i.e. $\mathbf{Adv}^{\mathrm{dl}}_{\mathbb{G},g}(\mathcal{B}^*) \leq \mathrm{T}^2_{\mathcal{B}^*}/p$. Then, from Eq. (35) one has $\mathrm{T}_{\mathcal{B}^*} \approx \sqrt{0.4 \cdot p}$, so

$$\frac{\mathbf{Adv}^{\text{imp-pa}}_{\mathsf{ID}}(\mathcal{A})}{\mathrm{T}_{\mathcal{A}}} \leq \frac{0.4}{\sqrt{0.4 \cdot p}}\,, \tag{38}$$

which means that one would need a group of size

$$p \approx \left(\frac{\mathrm{T}_{\mathcal{A}}}{\mathbf{Adv}^{\text{imp-pa}}_{\mathsf{ID}}(\mathcal{A})}\right)^2\,. \tag{39}$$

This is exactly the same requirement as dictated by the prior results, namely Equation (7) and

Equation (11). Hence, the guarantee by the results of KMP is the same as offered by prior results in Fig. 1.