The Abe-Okamoto Partially Blind Signature Scheme Revisited

Julia Kastner[1*], Julian Loss[2**], and Jiayu Xu[3***]

[1] Department of Computer Science, ETH Zurich, Zurich, Switzerland
julia.kastner@inf.ethz.ch
[2] CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
loss@cispa.de
[3] School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA xujiay@oregonstate.edu

Abstract. Partially blind signatures, an extension of ordinary blind signatures, are a primitive with wide applications in e-cash and electronic voting. One of the most efficient schemes to date is the one by Abe and Okamoto (CRYPTO 2000), whose underlying idea — the OR-proof technique — has served as the basis for several works.
We point out several subtle flaws in the original proof of security, and provide a new detailed and rigorous proof, achieving similar bounds as the original work. We believe our insights on the proof strategy will find useful in the security analyses of other OR-proof-based schemes.

1 Introduction

Blind signatures, first introduced by Chaum [13], are a fundamental cryptographic primitive. They allow two parties, a *signer* who holds the secret key and a *user* who holds the message, to jointly generate a signature. Roughly speaking, security requires that the signer learns nothing about the message nor the signature (*blindness*), and the user cannot forge a signature that does not result from its interaction with the signer (*one-more unforgeability*). Blind signatures have found extensive applications in settings where anonymity is of great concern, such as e-cash [13, 15, 21, 42] and electronic voting [14, 20].

However, in a blind signature scheme, the signer has absolutely no control over the message it signs. This leads to various shortcomings in practice. First, in an e-cash system where a bank uses blind signatures to issue its coins, to avoid the double spending problem, the bank has to keep record of all coins that have been spent; to prevent the ledger from growing unlimitedly, old coins need to expire after a period of time, so that the corresponding entries in the ledger can be deleted. Second, there is no way to inscribe the value or expiration date of a coin. Thus, the bank has to use a different public key for each value/expiration

* Work done while supported by ERC Project PREP-CRYPTO 724307
** Work done while at University of Maryland
*** Work done while at George Mason University

date, and anyone who spends or receives these coins has to maintain a list of all public keys, which has to evolve over time when old coins expire and are replaced by new ones. Similarly, in electronic voting, voters have to download a new public key for each election.

To address these issues, Abe & Fujisaki [2] proposed an extension called *partially* blind signatures, which allow a signer to explicitly include some common information (called the *tag*) in the signature. The tag is agreed upon by the signer and the user in advance and remains unblinded throughout the signing procedure; for example, it can be the date of issue or the value of the electronic coin. Setting the tag to the empty string yields an ordinary blind signature scheme. Informally, a partially blind signature scheme is secure if it satisfies (1) *partial blindness*: for multiple signatures that use the same tag, an adversarial signer cannot link these signatures to the signing sessions they originate from; and (2) *one-more-unforgeability*, or *OMUF security*: an adversarial user that interacts with the signer in at most ℓ many sessions, cannot output more than ℓ valid message-signature pairs.

Despite 25 years of research, there have been very few partially blind signature schemes ever proposed. The most efficient scheme up to date is the one proposed by Abe and Okamoto (AO) [4], which involves only 2 group (multi-) exponentiations for the signer and 4 (multi-)exponentiations for the user. The scheme is based on the classical OR-proof technique for obtaining witness indistinguishable protocols by Cramer *et al.* [17], and its security proof involves an intricate rewinding argument. The ideas behind both the scheme and its security proof repeatedly appear in blind signatures [1, 5, 7, 40].

Unfortunately, close scrutiny shows that there are a number of critical issues with the proof of one-more-unforgeability in AO and in some other subsequent works. In particular, the analysis of the reduction's success probability is based on a problematic counting argument. In this paper, we revisit the AO partially blind signature scheme and present a new comprehensive analysis of its one-more-unforgeability, which addresses *all issues in the original security proof*. (The proof of partial blindness in AO is correct and is not the focus of this paper.) The contributions of this paper are two-fold. First, we identify the flaws in the proof of AO, which we elaborate on in Section 1.1. Second, we overcome these issues by resorting to a more involved and rigorous counting argument. Our insights lead to new proof techniques and a much better understanding of AO's ideas. While we focus on the AO partially blind signature scheme, we believe that our techniques are applicable to other blind signature schemes based on the OR-proof technique.

1.1 Technical Overview

In this section we provide an overview of our security proof of the AO partially blind signature scheme, and explain the issues in the original work [4]. Similar to AO, our proof is done in two steps. First we consider the simplified case where there is only a single tag. This is the most technically involved part of the entire security analysis, and contains essential modifications to the proof in AO. Then

we generalize it to the multi-tag case. This part of the proof is straightforward and mostly follows [4]. For simplicity, we only discuss the case of a single tag in this technical overview.

Forking: A Recap. The reduction in our security proof uses the forking technique to rewind the adversary and solve the discrete logarithm problem [36]. As is standard in a forking argument, we first define what we call a *deterministic wrapper* which provides a simplified, non-interactive interface to the reduction. More precisely, the wrapper takes as input an *instance* $\mathbf{I}$ (containing a public key and the internal values used to generate the signer's first messages of all signing sessions), a *random tape* rand (containing the random tape of the actual adversary), and a random *hash vector* $\overrightarrow{h}$ (to be used as outputs of random oracle queries). The reduction forks the wrapper instead of forking the adversary directly. In more concrete terms, this means that the reduction runs the wrapper once on inputs $\mathbf{I}, \mathsf{rand}, \overrightarrow{h}$ and obtains an output which implicitly defines an index $J \in [|\overrightarrow{h}|]$. It then generates a vector $\overrightarrow{h}'$ by resampling the vector $\overrightarrow{h}$ uniformly at random from position J, and keeping the first $J-1$ entries the same. It reruns the wrapper on inputs $\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'$, which will generate a run that is identical up the point where the reduction answers the J-th random oracle query. In particular, the input to this query remains identical in both runs. The goal of the reduction is to infer some equality from these relations so as to solve a discrete logarithm instance that it suitably embeds in its interaction with the adversary (see below).

Dealing with OR-Proofs in Forking. The AO scheme uses the classical OR-proof strategy of [17] to combine two Schnorr-style signatures into one. The witness for one branch of the proof is the actual secret key x of the scheme; the other branch corresponds to the *tag key* $\mathbf{z}$ which is obtained through hashing the tag info. On the signer's side, the protocol is a witness indistinguishable (WI) proof of knowledge of at least one witness, either the secret key x or the discrete logarithm of the tag key $\operatorname{dlog} \mathbf{z}$. This gives rise to the following proof strategy, which was also used in [7]: The reduction can choose these tag keys such that it knows a witness and sign without knowing the secret key (so it can embed a discrete logarithm challenge in the public key), or it can embed its discrete logarithm challenge in a tag key and sign using the actual secret key. The intuitive idea here is that for each run of the protocol, the witness used internally by the reduction is perfectly hidden from the adversary (due to WI). Thus, the probability that the reduction is able to extract the "opposing" witness (i.e., the one it is not using itself for answering signing queries) from two forking runs of the adversary should be high.

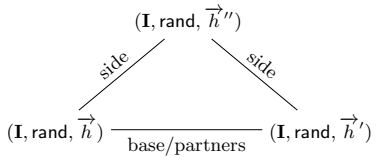
Unfortunately, this intuition proves incorrect upon closer inspection. While WI perfectly hides the witness during any *single* run of the protocol, the transcripts of two executions of the protocol with the adversary (as performed by the reduction) can depend on the witness internally used by the reduction. Therefore, arguing that the reduction indeed extracts the opposing witness from two runs of the adversary turns out to be highly non-trivial.

Partnering Runs. We now describe the general idea for proving that the reduction has a significant probability to extract the witness it needs. For now we fix an instance $\mathbf{I}$ and a random tape rand, and consider the hash vector $\overrightarrow{h}$ as the only varying parameter of the reduction. Using a simple counting argument, one can show that for a significant portion of pairs $\mathbf{I}, \mathsf{rand}$, there must exist two hash vectors $\overrightarrow{h}$, $\overrightarrow{h}'$ that lead to the same transcript between the wrapper and the adversary when the wrapper is run on $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ or $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$, respectively. Borrowing the terminology from [4], we refer to such triples $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ as *partners*. The key observation is that the witness extracted from partnering runs is independent of which witness was used by the reduction as part of the instance $\mathbf{I}$, and thus the reduction has a significant probability of extracting the desired witness (i.e., the witness not used by the reduction).[4] Unfortunately, given $\mathbf{I}, \mathsf{rand}$, finding a pair of partners $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ and $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ might not be efficiently possible, as in general, only few of them may exist. Hence it requires an additional argument to ensure that the reduction produces forks from which the desired witness can be efficiently extracted.

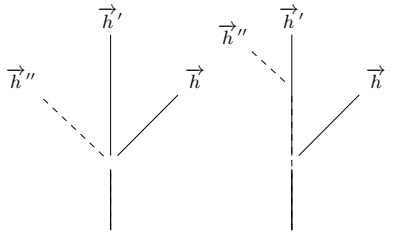
From Partners to Triangles. The next step in our chain of reasoning is to apply the strategy of AO for "amplifying" the number of forking runs from which the desired witness can be extracted. Thus, analogous to AO, we define *triangles* as follows. The corners of a triangle will be three triples $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'')$, which produce successful runs for the wrapper. In addition, $\overrightarrow{h}, \overrightarrow{h}', \overrightarrow{h}''$ all share a common prefix of some $i-1$ entries and start to fork from each other at the i-th entry. The most important property of a triangle, however, is that $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ and $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ be partnering runs, i.e., produce the same transcript for the wrapper. (AO refer to the pair of partnering runs as the "triangle base" and to the remaining pairs of triples as the "triangle sides".) We illustrate this in fig. 1. As observed by AO, if the forked runs corresponding to $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ and $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ yield the desired witness (i.e., the one not stored inside $\mathbf{I}$), then either of the forked runs $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'')$ or $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'')$ yield the same witness. Their key insight is that the number of triangles should be far greater than the number of triangle bases formed by partnering runs $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ and $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$. Intuitively, this is the case because a *single pair* of triples $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ can serve as the base in *many different* triangles.

A Gap in AO. The next step in the analysis of AO is to count the number of triangles for which at least one side yields the desired witness. (We call such

[4] Due to the WI property of the scheme, for any $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$, there exists a corresponding triple $(\mathbf{I}', \mathsf{rand}, \overrightarrow{h})$ that contains the other witness and produces the same transcript as $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$. This means that the same witness w could have been extracted from a pair of partnering runs $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$, or from $(\mathbf{I}', \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}', \mathsf{rand}, \overrightarrow{h}')$, where one of $\mathbf{I}$ and $\mathbf{I}'$ contains w, and the other instance does not.



(a) A triangle consists of a pair of partners (the base) and one additional tuple (the top). A pair consisting of the top and one of the base corners is called a side.



(b) Left: forking as in a triangle (solid lines are the base, dashed lines are the top); right: not a triangle (forking at wrong point).

Fig. 1: Triangles

triangle sides "successful".) Recall that we keep $\mathbf{I}, \mathsf{rand}$ fixed throughout this counting argument, and argue only about the number of successful hash vectors associated with runs using $\mathbf{I}, \mathsf{rand}$. If we can show that there are enough of triangles with a successful side, we might hope that when sampling a random pair $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'')$ during forking, the reduction will hit a successful triangle side, from which the desired witness can be extracted.

This is the point where our analysis diverges significantly from [4]. As noted above, many triangles may share a base; that is, for any given base, there exist many possible triangle tops. This makes it possible to "amplify" the extractability of the desired witness from a single base to extracting it from many possible triangle sides which are adjacent to this base in some triangle. (Recall that if a triangle base is successful, then at least one of the two sides must also be successful.) However, we observe that *many triangles may also share a side*. If many triangles overlap on successful sides (but not on unsuccessful sides), it might happen that the total number of successful sides is *much smaller* than the total number of unsuccessful sides.[5]

Indeed, this is where the most crucial gap occurs in [4]. First, for each triangle base corner $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$, they assign this corner a partner $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ using the mapping Prt (so $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') = Prt(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ forms a triangle base together with $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$; see [4, p. 284]). It is, however, unclear if this is intended to be

[5] We stress that simply replacing a triple $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ with an indistinguishable triple $(\mathbf{I}', \mathsf{rand}, \overrightarrow{h})$ is not sufficient to solve this problem. Indeed, one might hope that since the adversary can not detect this change, an unsuccessful side may become successful when switching from $\mathbf{I}$ to $\mathbf{I}'$, as the desired witness would flip. However, a successful forking pair $((\mathbf{I}', \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}', \mathsf{rand}, \overrightarrow{h}'))$ need only exist if $((\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'))$ is a base. The same is not true, in general, for sides, as their endpoints may not (and generally do not) yield the same transcript. Because of this, an unsuccessful side $((\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'))$ might not even be part of a triangle side when switching witnesses from $\mathbf{I}$ to $\mathbf{I}'$.

an injective assignment (i.e., no two base corners can share the same partner). If so, there is a gap as to why this assignment is possible, i.e., why there are enough such partners for each of the base corners to find a *different* partner. In fact, we provide an argument in our analysis for why such partners — which we call *opposing base corners* — are *not much fewer* than original base corners, but they do not necessarily need to be equal in size.

On the other hand, if the assignment Prt is not injective, then different triangles may share the same side. This is also problematic, as we explain now. [4] proceeds to claim that if (for a fixed pair $\mathbf{I}, \mathsf{rand}$) at least $\frac{4}{5}$ of triangle sides are unsuccessful, then at least $\frac{3}{5}$ of triangle bases are also unsuccessful, i.e., they yield the undesired witness that is used by the reduction. (See the proof of the last claim on [4, p. 284].) Although this claim is not explicitly argued, the underlying reasoning seems to be as follows: since every triangle has two sides and one base, if $\frac{4}{5}$ of all sides are unsuccessful, then at least $\frac{4}{5}+\frac{4}{5}-1=\frac{3}{5}$ fraction of triangles have two unsuccessful sides, which implies that their bases must also be unsuccessful. However, this argument *implicitly assumes that no triangles ever share a base or a side*, which, as we have mentioned, is not necessarily the case.

Concrete Counterexamples and Additional Issues. We now provide concrete counterexamples to show why the claim above is false if triangles may share sides, or even just bases. For triangles sharing sides, consider the example in the middle of fig. 2, where 8 out of the 10 triangle sides are unsuccessful, yet only 2 out of the 6 triangle bases are unsuccessful. For triangles sharing only bases (recall that in this case there is already a gap as to why there exists an assignment Prt such that triangles do not share sides), the claim is also untrue: see the rightmost part of fig. 2 for an example where 5 out of the 6 triangle sides are unsuccessful, yet only 1 out of the 2 triangle bases are unsuccessful.

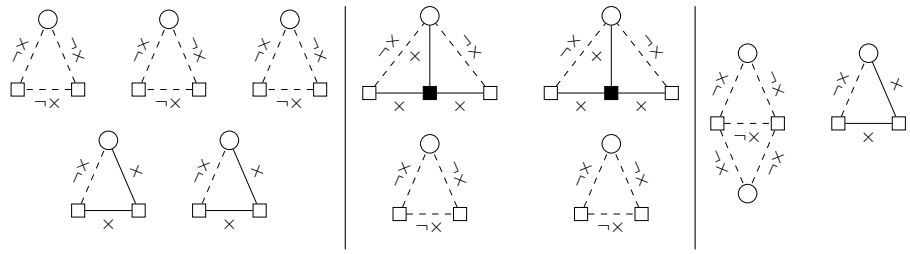


Fig. 2: Claim in [4] that if at least $\frac{4}{5}$ of triangle sides are unsuccessful (i.e., yield the undesirable witness $\neg\times$), then at least $\frac{3}{5}$ of bases (incident to two square nodes) also yield this witness. This holds for non-overlapping triangles (left), but not for triangles overlapping in sides (middle) or in bases (right).

We further note that there are some relatively minor gaps that are easier to fix. In particular, it is incorrect to assert that the probability to extract either

witness from a triangle base is close to $\frac{1}{2}$ — we refer here to the last sentence in the proof of the last claim on [4, p. 284]:

> *Since the information of a base,* $(\overrightarrow{\varepsilon}, \overrightarrow{\varepsilon}')[(\overrightarrow{h}, \overrightarrow{h}')]$*, is independent of the witness the simulator already has as a part of* Ω $[\mathbf{I}, \mathsf{rand}]$*, this contradicts that a biased result should occur with probability (over* Ω $[\mathbf{I}, \mathsf{rand}]$*) less than* $1/2 + 1/\mathsf{poly}(n)$ *for any polynomial* poly*.*

(The expressions in brackets are a translation to our notation.)

To see why this claim is incorrect, imagine a computationally unbounded adversary that finds the secret key $x = \mathrm{dlog}\,\mathbf{y}$ by brute force, and wins the OMUF game by running the real signer's code. Then the signatures produced by this adversary — including pairs of signatures obtained from triangle bases — will always yield the same witness (the secret key x), rather than yielding either witness with probability close to $\frac{1}{2}$. Our approach to deal with this issue is to define a "majority witness" $\times$ which can be extracted from many triangle bases (for a suitable definition of many). We then show that it is possible to extract $\times$, using a suitable counting argument.

Resolving the issues from earlier works. We now provide an overview of our strategy to bridge the gaps in [4], achieving the same result. We first recall that for any triple $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$, there is a corresponding instance $\mathbf{I}'$ that contains the other branch of witness, such that $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ and $(\mathbf{I}', \mathsf{rand}, \overrightarrow{h})$ yield the same transcript. This naturally leads to the concept of *both-sided triangle bases*, namely triangle bases $((\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'))$ that are also the base of some triangle when $\mathbf{I}$ is replaced by $\mathbf{I}'$. Using several counting arguments, we show that the set of both-sided base corners must be large. While our counting arguments are more detailed and rigorous, they are in the same spirit as those of [4].

We now bridge the gap in [4], by showing that there cannot be too large of an overlap between triangle sides such that the absolute amount of successful triangle sides would get small. We define *good base corners* as triples that are incident to many successful both-sided triangle bases, as well as many successful triangle sides. We further require that these triangle sides and bases must exist at the good base corner's *maximum branching index* — the index at which the largest number of partners fork from it. Similarly, we define *good opposing corners* that are incident to a successful both-sided triangle base and many successful sides, but the triangle base and sides are located at the maximum branching index of the triple *at the other end of the base.*

Our crucial observation is that *if there are not too many good base corners, then there must be many good opposing corners.* To see this, consider a base corner $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ that is not good, and consider all triples $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ that are partners of $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$. Let i denote the maximum branching index of $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$; by definition, a significant portion of these partners $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ fork with $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ at index i. Recall that if a triangle base is successful, then at least one of its sides must also be successful. Since most of the triangle sides

involving $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ are unsuccessful at index i, this means that many of the *other* triangle sides, i.e., those involving the partners $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$, are successful at index i. In other words, a significant portion of a non-good base corner $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$'s partners, are good opposing corners. (In the formal proof, we will also rule out the possibility that different non-good base corners' corresponding good opposing corners overlap too much.)

The above conclusion means that, when the reduction samples the triple for the first forking run, with significant probability the triple is either a good base corner or a good opposing corner. Then, due to the definitions of these good triangle corners, it is not hard to show that with significant probability the reduction hits a successful triangle side while sampling the second triple — that is, the desired witness can be extracted from the two forking runs.

Finally, we remark that our reduction guesses in advance which hash values the adversary will actually use to produce its signatures. This introduces a loss of $\binom{Q_h}{\ell+1}$ in the reduction's advantage, where Q_h is the number of the adversary's hash queries and ℓ is the number of signing sessions closed. This step is necessary in our analysis as we need all possible forking indices to have a signature attached to them in order to lower-bound the set of good opposing base corners. (See Remark 2 in Section 4.4.) We notice that a loss in this order of magnitude seems inherent due to the recent polynomial-time ROS-attack [8], and that we achieve comparable bounds to the original work of Abe & Okamoto [4].[6]

Other OR-Proof Based Blind Signature Schemes. While our focus is on proving one-more unforgeability of the scheme by Abe & Okamoto [4], here we briefly discuss other schemes that use a similar approach. One example is the scheme by Abe [1] which uses a similar approach to AO, but in such a way that the "tag key" is blinded by the user, yielding a ordinary blind signature scheme under the DDH assumption. However, as later pointed out by Ohkubo & Abe [32], the forking-based proof in the original work contains a flaw when it comes to arguing why the desired witness can be extracted, and up to this point, security analyses were given only in the generic group model [32] and the algebraic group model [26]. In appendix F we provide a sketch of how to apply our proof technique to Abe's scheme, and leave its full analysis to future work.

[7] proposed a blind signature scheme with attributes, whose construction idea is similar to [1]. Using our terminology, in their security analysis *only one triangle* is considered for which there is at least one successful side. Their argument critically relies on the claim that it must be equally as likely to sample a successful side as it is to sample an unsuccessful one. While this holds true for a *single* triangle, as discussed above, the overall set of unsuccessful sides may be far larger and thus far more likely to be sampled at random.

Another scheme that uses the OR-proof technique is the BlindOR scheme by Alkadri *et al.* [5] (also see the full version [6]), based on the module learning with error (MLWE) assumption. [6, Lemma 3.5] argues that the desired witness can

[6] See the top of [4, p. 285], where the reduction's advantage includes a term η_1^2, where $\eta_1 = \eta/2Q_h^{\ell+1}$ and η is the adversary's advantage.

be extracted with a significant probability: An MLWE distinguisher flips a coin to decide which branch to use for generating signatures, and embeds its MLWE challenge in the opposing branch. Then, if the opposing witness can be extracted from two forking runs of the adversary, the MLWE challenge must have been a yes-instance; otherwise the distinguisher flips another coin as its output, as either the MLWE challenge was a no-instance, or the adversary happened to use the same branch as the distinguisher for generating its signatures.

However, this argument seems circular, as it already assumes that if both branches contain a yes-instance, then the opposing witness can be extracted from a significant portion of forking runs, which is the statement of the lemma. We leave it as future work to explore the application of our method to the BlindOR scheme.

The above discussion illustrates that the security analyses of OR-proof based (partially) blind signature schemes are extremely hard and subtle, and require large amounts of rigor and caution.

1.2 Related Work

Partially blind signatures were introduced in [2], which also presented a scheme based on a non-standard RSA-type assumption. Cao *et al.* [11] proposed another construction based on the RSA assumption, but their scheme was cryptoanalyzed in [31]. Zhang *et al.* [43], as well as Chow *et al.* [16], proposed schemes based on bilinear pairings, and Papachristoudis *et al.* [34] proposed a scheme based on lattice assumptions. Okamoto [33] proposed a theoretical construction that does not rely on the random oracle model. Finally, Maitland & Boyd [30] considered a *restrictive* partially blind signature scheme, where the user's choice of messages must follow certain rules.

There is a rich literature on (ordinary) blind signatures and their applications. Its security notion was formalized by Pointcheval & Stern [35] and Juels *et al.* [25], and later strengthened by Schröder & Unruh [39]. Fischlin [18] and Abe & Ohkubo [3] considered security definitions in the universal composability (UC) framework. Camenisch *et al.* [10] and Fischlin & Schröder [19] considered a stronger notion of blindness called *selective-failure blindness*. There are a large number of blind signature schemes based on various assumptions and in various models; a very incomplete list includes [1, 9, 12, 22–24, 27–29, 33, 35, 37, 38].

We notice that the security analyses of (partially) blind signature schemes are usually extremely involved, with the original security proofs sometimes being flawed. Apart from the schemes already discussed, we give two additional examples here. The security of the Schnorr blind signature [38] relies on the hardness of the ROS problem, which was recently shown to be easy [8]; a new security proof in the weaker sequential setting appears in [26]. For partially blind signature schemes, the aforementioned Zhang *et al.* scheme [43] has an issue in its security proof, and the full analysis came much later [41]. This paper can be seen as yet another attempt of spotting and fixing issues in previous works; however, we stress that the underlying OR-proof technique of the Abe-Okamoto

scheme is widely used in blind signatures, and we believe that our techniques will find applications in the security analyses of other schemes as well.

2 Preliminaries

2.1 Notation

We denote by $[\ell] \coloneqq \{1, \ldots, \ell\}$. For a vector $\overrightarrow{h}$, its i-th entry is denoted by h_i, and the vector of its first i entries is denoted by $\overrightarrow{h}_{[i]}$. We denote by $x \xleftarrow{\$} X$ that x is sampled uniformly at random from set X. For a vector $\overrightarrow{x} \in X^n$, we denote by $\overrightarrow{x}' \xleftarrow{\$} X^n_{|\overrightarrow{x}_{[i]}}$ that $\overrightarrow{x}'$ is sampled uniformly at random from $\{\overrightarrow{x}' \in X^n | \overrightarrow{x}'_{[i]} = \overrightarrow{x}_{[i]}\}$. For an algorithm A, we use t_{A} to denote its running time.

2.2 Computational Problems

Definition 1 (Discrete Logarithm Problem). *For public parameters* $\mathsf{pp} = (\mathbb{G}, q, \mathbf{g})$ *for a group* $\mathbb{G}$ *with order* q *and generator* $\mathbf{g}$*, we describe the discrete logarithm game* $\mathbf{DLOG}_{\mathbb{G}}$ *with adversary* A *as follows:*

Setup. *Sample* $x \xleftarrow{\$} \mathbb{Z}_q$ *and set* $\mathbf{y} := \mathbf{g}^x$*. Output* $(\mathsf{pp}, \mathbf{y})$ *to* A.

Output Determination. *When* A *outputs* $x' \in \mathbb{Z}_q$*, return* 1 *if* $g^{x'} = \mathbf{y}$ *and* 0 *otherwise.*

We define the advantage of A *as*

$$\mathrm{adv}^{\mathbf{DLOG}_{\mathbb{G}}}_{\mathsf{A}} \coloneqq \Pr[\mathbf{DLOG}^{\mathsf{A}}_{\mathbb{G}} = 1]$$

where the probability goes over the randomness of the game as well as the randomness of the adversary A*. We say that the discrete logarithm problem is* (t, ϵ)-hard *in* $\mathbb{G}$ *if for any adversary* A *that runs in time at most* t*, it holds that*

$$\mathrm{adv}^{\mathbf{DLOG}_{\mathbb{G}}}_{\mathsf{A}} \leq \epsilon.$$

(When it is clear from context, we may omit $\mathbb{G}$ *and only write* **DLOG** *for the game.)*

2.3 Partially Blind Signatures

The definitions in this section mostly follow [4].

Definition 2 (Partially Blind Signature scheme). *A three-move* partially blind signature scheme $\mathsf{PBS} = (\mathsf{KeyGen}, \mathsf{Sign} = (\mathsf{Sign}_1, \mathsf{Sign}_2), \mathsf{User} = (\mathsf{User}_1, \mathsf{User}_2), \mathsf{Verify})$ *consists of the following* PPT *algorithms:*

Key Generation. *On input public parameters* pp*, the probabilistic algorithm* KeyGen *outputs a public key* pk *and a secret key* sk*. Henceforth we assume that* pp *is provided to all parties (including the adversary) as an input, and do not explicitly write it.*

Signer: *The interactive signer* $\mathsf{Sign} = (\mathsf{Sign}_1, \mathsf{Sign}_2)$ *has two phases:*

- Sign_1**:** *On input a tag* info *and a secret key* sk*, the probabilistic algorithm* Sign_1 *outputs an internal signer state* $\mathsf{st}_{\mathsf{Sign}}$*, and a response* R*.*
- Sign_2**:** *On input the secret key* sk*, a challenge value* e*, and the corresponding internal state* $\mathsf{st}_{\mathsf{Sign}}$*, the deterministic algorithm* Sign_2 *outputs a response* S*.*

User. *The interactive user* $\mathsf{User} = (\mathsf{User}_1, \mathsf{User}_2)$ *has two phases:*

- User_1**:** *On input a public key* pk*, a tag* info*, a message* m*, and a* Sign_1 *response* R*, the probabilistic algorithm* User_1 *outputs a challenge value* e *and an internal user state* $\mathsf{st}_{\mathsf{User}}$*.*
- User_2**:** *On input a public key* pk*, a* Sign_2 *response* S*, and the corresponding internal user state* $\mathsf{st}_{\mathsf{User}}$*, the deterministic algorithm* User_2 *outputs a signature* sig *on message* m *along with the tag* info*.*

Verification. *On input a public key* pk*, a message* m*, a signature* sig*, and a tag* info*, the deterministic algorithm* Verify *outputs either* 1 *or* 0*, where* 1 *indicates that the signature is valid, and* 0 *that it is not.*

We say a partially blind signature scheme PBS *is* (perfectly) correct *if for all* $\mathsf{pk}, m, \mathsf{sig}, \mathsf{info}$ *that result from an honest interaction between signer and user,* $\mathsf{Verify}(\mathsf{pk}, m, \mathsf{sig}, \mathsf{info}) = 1$*.*

We now define the one-more-unforgeability of a partially blind signature scheme. We do not focus on partial blindness in this paper; we include the definition in Appendix A for completeness, and for a proof that the Abe-Okamoto scheme is partially blind, see the original paper [4].

Definition 3 (One-more-unforgeability). *For a three-move partially blind signature scheme* PBS*, we define the* ℓ*-one more unforgeability (*ℓ*-OMUF) game* ℓ-$\mathbf{OMUF}_{\mathsf{PBS}}$ *with an adversary* U *(in the role of the user) as follows:*

Setup. *Sample a pair of keys* $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{PBS.KeyGen}(\mathsf{pp})$*. Initialize* $\ell_{\mathtt{closed}} := 0$ *and run* U *on input* pk*.*

Online Phase. U *is given access to oracles* sign_1 *and* sign_2*, which behave as follows.*

- **Oracle** sign_1**:** *On input* info*, the oracle samples a fresh session identifier* sid*. It sets* $\mathsf{open}_{\mathsf{sid}} := \mathtt{true}$ *and generates* $(R_{\mathsf{sid}}, \mathsf{st}_{\mathsf{sid}}) \xleftarrow{\$} \mathsf{PBS.Sign}_1(\mathsf{sk}, \mathsf{info})$*. Then it returns the response* R_{sid} *together with* sid *to* U*.*
- **Oracle** sign_2**:** *If* $\ell_{\mathtt{closed}} < \ell$*, the oracle takes as input a challenge* e *and a session identifier* sid*. If* $\mathsf{open}_{\mathsf{sid}} = \mathtt{false}$*, it returns* $\perp$*. Otherwise, it sets* $\ell_{\mathtt{closed}}{+}{+}$ *and* $\mathsf{open}_{\mathsf{sid}} := \mathtt{false}$*. Then it computes the response* $S \xleftarrow{\$} \mathsf{PBS.Sign}_2(\mathsf{sk}, \mathsf{st}_{\mathsf{sid}}, e)$ *and returns* S *to* U*.*

Output Determination. *When* U *outputs distinct tuples* $(m_1, \mathsf{sig}_1, \mathsf{info}_1), \ldots, (m_k, \mathsf{sig}_k, \mathsf{info}_k)$*, return* 1 *if* $k \geq \ell_{\mathtt{closed}} + 1$ *and for all* $i \in [k]$: $\mathsf{PBS.Verify}(\mathsf{pk}, \sigma_i, m_i, \mathsf{info}_i) = 1$*. Otherwise, return* 0*.*

We define the advantage of U *as*

$$\mathrm{adv}_{\mathsf{U}}^{\ell\text{-}\mathbf{OMUF}_{\mathsf{PBS}}} = \Pr\left[\ell\text{-}\mathbf{OMUF}_{\mathsf{PBS}}^{\mathsf{U}} = 1\right]$$

where the probability goes over the randomness of the game as well as the randomness of the adversary U. *We say the scheme* PBS *is* (t, ϵ, ℓ)-one-more unforgeable *if for any adversary* U *that runs in time at most* t *and makes at most* ℓ *queries to* sign_2,

$$\mathrm{adv}_{\mathsf{U}}^{\ell\text{-}\mathbf{OMUF}_{\mathsf{PBS}}} \leq \epsilon.$$

If U *always queries the same tag to oracle* sign_1, *we denote the game as* ℓ-1-info-$\mathbf{OMUF}_{\mathsf{PBS}}$ *and say that* PBS *is* (t, ϵ, ℓ)-single-tag one-more unforgeable.

3 The Abe-Okamoto Partially Blind Signature Scheme

In this section we describe the partially blind signature scheme by Abe & Okamoto [4]. The idea of the scheme relies on the OR-Proof technique by Cramer *et al.* [17]. It runs a proof of knowledge that the signer knows either the secret key x or the discrete logarithm of the so-called *tag key* $\mathbf{z}$, which is obtained through hashing the tag info. In this way we obtain a *witness indistinguishable* scheme: an honest signer does not know $\mathrm{dlog}\,\mathbf{z}$ and is forced to use x for issuing signatures; while the reduction may program the random oracle so that it knows the $\mathrm{dlog}\,\mathbf{z}$ and can then simulate the signer without knowing the secret key x.

Key Generation. On input public parameters $\mathsf{pp} = (\mathbb{G}, \mathbf{g}, q, \mathsf{H}^*, \mathsf{H})$ (where H^* and H are random oracles with ranges $\mathbb{G}$ and $\mathbb{Z}_q$, respectively), KeyGen samples $x \xleftarrow{\$} \mathbb{Z}_q$ and sets $\mathbf{y} := \mathbf{g}^x$. It then outputs $(\mathsf{pk}, \mathsf{sk}) := (\mathbf{y}, x)$.

Signer. $\mathsf{Sign} = (\mathsf{Sign}_1, \mathsf{Sign}_2)$ behaves as follows:

- Sign_1**:** On input info and sk, Sign_1 computes the tag key $\mathbf{z} := \mathsf{H}^*(\mathsf{info})$ and samples $u, s, d \xleftarrow{\$} \mathbb{Z}_q$. It then computes the *commitments* $\mathbf{a} := \mathbf{g}^u, \mathbf{b} := \mathbf{g}^s \cdot \mathbf{z}^d$. It outputs the response $(\mathbf{a}, \mathbf{b})$ to the user and an internal state $\mathsf{st}_{\mathsf{Sign}} := (u, s, d)$.
- Sign_2**:** On input $e \in \mathbb{Z}_q, \mathsf{st}_{\mathsf{Sign}} = (u, s, d), \mathsf{sk} = x$, Sign_2 computes $c := e - d$ and $r := u - cx$. It outputs the response (r, c, s, d) to the user.

User. $\mathsf{User} = (\mathsf{User}_1, \mathsf{User}_2)$ behaves as follows:

- User_1**:** On input $\mathsf{pk}, m, \mathsf{info}, \mathbf{a}, \mathbf{b}$, User_1 computes the tag key $\mathbf{z} := \mathsf{H}^*(\mathsf{info})$ and samples $t_1, t_2, t_3, t_4 \xleftarrow{\$} \mathbb{Z}_q$. It then computes $\boldsymbol{\alpha} := \mathbf{g}^{t_1} \cdot \mathbf{y}^{t_2} \cdot \mathbf{a}$ and $\boldsymbol{\beta} := \mathbf{g}^{t_3} \cdot \mathbf{z}^{t_4} \cdot \mathbf{b}$, queries $h := \mathsf{H}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{z}, m)$ for the message m it wants to sign, and computes the blinded challenge $e := h - t_2 - t_4$. It outputs e to the signer and an internal state $\mathsf{st}_{\mathsf{User}} := (t_1, t_2, t_3, t_4)$.
- User_2**:** On input $\mathsf{pk}, (r, c, s, d), \mathsf{st}_{\mathsf{User}} = (t_1, t_2, t_3, t_4)$, User_1 computes $\rho := r + t_1$, $\omega := c + t_2$, $\sigma := s + t_3$, and $\delta := d + t_4$. It then verifies that $\omega + \delta = \mathsf{H}(\mathbf{g}^\rho \cdot \mathbf{y}^\omega, \mathbf{g}^\sigma \cdot \mathbf{z}^\delta, \mathbf{z}, m)$; if so, it outputs the signature $(\rho, \omega, \sigma, \delta)$. (Otherwise, it outputs $\perp$.)

Verification. On input $\mathbf{y}, m, \mathsf{info}, (\rho, \omega, \sigma, \delta)$, Verify computes $\mathbf{z} := \mathsf{H}^*(\mathsf{info})$. It outputs 1 if $\omega + \delta = \mathsf{H}(\mathbf{g}^\rho \cdot \mathbf{y}^\omega, \mathbf{g}^\sigma \cdot \mathbf{z}^\delta, \mathbf{z}, m)$ and 0 otherwise.

For a graphic illustration of the scheme, see Figure 6 on page 50.

4 Computing the probability for extracting the "good" witness

As mentioned in the introduction, our analysis of the Abe-Okamoto scheme is done in two steps. In this section, we deal with the case that the adversary U only uses a *single* tag, i.e., U plays the ℓ-1-info-$\mathbf{OMUF}_{\mathsf{AO}}$ game.

4.1 The Deterministic OMUF Wrapper

Restricting the adversary to making $\ell+1$ hash queries. Suppose that the adversary U makes ℓ queries to sign_2 (henceforth "signing queries") and Q_h queries to H (henceforth "hash queries"), and uses a single tag $\overline{\mathsf{info}}$. Below we assume w.l.o.g. that U never makes the same query to H twice.

We say that a message-signature pair $(m, (\rho, \omega, \sigma, \delta))$ *corresponds to* an index $i \in [Q_h]$, or corresponds to the adversary U's i-th hash query, if this query was $\mathsf{H}(\mathbf{y}^{\omega}\mathbf{g}^{\rho}, \mathbf{z}^{\delta}\mathbf{g}^{\sigma}, \mathbf{z}, m)$. (When the message m is clear from context, we may say that the signature $(\rho, \omega, \sigma, \delta)$ corresponds to index i.) We remark that we can further assume w.l.o.g. that there exist $\ell+1$ hash queries of U, each of which corresponds to a distinct message-signature pair in the output of U (in particular, $Q_h \geq \ell+1$). This is because otherwise one of the following must hold (assuming that U succeeds):

- There exists a pair $(m, (\omega, \rho, \delta, \sigma))$ that does not correspond to any hash query, i.e., $\mathsf{H}(\mathbf{y}^{\omega}\mathbf{g}^{\rho}, \mathbf{z}^{\delta}\mathbf{g}^{\sigma}, \mathbf{z}, m)$ has never been queried. In this case, U can be turned into another adversary U' that runs the code of U and additionally makes such a hash query; obviously U and U' have the same advantage.
- There exist two distinct pairs $(m_1, (\omega_1, \rho_1, \delta_1, \sigma_1)), (m_2, (\omega_2, \rho_2, \delta_2, \sigma_2))$ that correspond to the same hash query. In this case, we have that $m_1 = m_2$, $\mathbf{y}^{\omega_1}\mathbf{g}^{\rho_1} = \mathbf{y}^{\omega_2}\mathbf{g}^{\rho_2}$, and $\mathbf{z}^{\delta_1}\mathbf{g}^{\sigma_1} = \mathbf{z}^{\delta_2}\mathbf{g}^{\sigma_2}$. Then a reduction to the discrete logarithm problem can easily compute both x and w as $x = (\omega_1 - \omega_2)^{-1} \cdot (\rho_2 - \rho_2)$ and $w = (\delta_1 - \delta_2)^{-1} \cdot (\sigma_2 - \sigma_1)$.

It is not hard to see that any adversary U can be turned into another adversary that makes *exactly* $\ell+1$ hash queries, with a factor of $\binom{Q_h}{\ell+1}$ loss in advantage. Formally, we define an adversary $\mathsf{M} := \mathsf{M}^{\mathsf{U}}$ that works as follows. M, on input of a public key pk, chooses a random subset I of $[Q_h]$ with $|I| = \ell+1$, and invokes $\mathsf{U}(\mathsf{pk})$. For U's i-th query to H, if $i \notin I$, M responds with a random integer in $\mathbb{Z}_q$. For any other query (including queries to signing oracles, queries to H^*, and the i-th query to H for $i \in I$), M forwards it to the corresponding oracle of M's own challenger, and forwards the response back to U. When U outputs a set of $\ell+1$ message-signature pairs, M checks if every pair $(m, (\rho, \omega, \sigma, \delta))$ corresponds to some index $i \in I$, that is, U's i-th hash query was $\mathsf{H}(\mathbf{y}^{\omega}\mathbf{g}^{\rho}, \mathbf{z}^{\delta}\mathbf{g}^{\sigma}, \mathbf{z}, m)$. If so, M copies U's output (and outputs $\perp$ otherwise).

Lemma 1. *For* M *described above, we have that*

$$\mathrm{adv}_{\mathsf{M}}^{\ell\text{-1-}\mathsf{info}\text{-}\mathbf{OMUF}_{\mathsf{AO}}} \geq \frac{\mathrm{adv}_{\mathsf{U}}^{\ell\text{-1-}\mathsf{info}\text{-}\mathbf{OMUF}_{\mathsf{AO}}}}{\binom{Q_h}{\ell+1}}.$$

Proof. It is straightforward that M simulates the OMUF game to U perfectly. Assume that U succeeds. By our assumption on U, there is a set of indices $I^* \subset [Q_h]$ corresponding to the message-signature pairs in U's output, with $|I^*| = \ell + 1$. If $I^* = I$, then M also succeeds. Since I is a random subset of size $\ell + 1$ of $[Q_h]$, the probability that $I^* = I$ is $\frac{1}{\binom{Q_h}{\ell+1}}$. The lemma follows. □

The lemma above implies that it is sufficient to consider an adversary that makes exactly $\ell + 1$ (distinct) hash queries, since an upper bound of the adversary's advantage in this specific case immediately translates to such an upper bound in the general case. Below we simply assume that the adversary makes $\ell + 1$ hash queries.

The deterministic wrapper. For any adversary M that makes exactly $\ell + 1$ distinct hash queries, we define a deterministic *wrapper* A that, given the witness and random coin tosses for one side, simulates the view of M. The wrapper uses either the $\mathbf{y}$-side witness (i.e., the secret key) x or the $\mathbf{z}$-side witness $w = \operatorname{dlog} \mathbf{z}$ to respond to sign_2 queries, and simulates the other side of the OR-proof using fixed values. We begin with the formal definition of an instance:

Definition 4 (Instances). *For the deterministic wrapper simulating the OMUF-game to the adversary we define two types of* instances $\mathbf{I}$. *A* $\mathbf{y}$-side *(a.k.a. honest)* instance *consists of the following components:*

$b = 0$**:** *bit indicating that the secret key x will be used for simulation*
x**:** *the secret key, also referred to as the* $\mathbf{y}$*-side witness*
$\mathbf{z}$**:** *the tag key, to be returned by oracle* H^* *for requested* $\overline{\mathsf{info}}$
d_i, s_i**:** *simulator choices for* $\mathbf{z}$*-side part corresponding to the i-th signing session*
u_i**:** *discrete logarithm of the* $\mathbf{y}$*-side commitment* $\mathbf{a}_i$ *in the i-th signing session*

A $\mathbf{z}$-side instance *consists of the following components:*

$b = 1$**:** *bit indicating that the tag witness w will be used for simulation*
$\mathbf{y}$**:** *the public key*
w**:** *the discrete logarithm of the tag key* $\mathbf{z}$ *as above*
c_i, r_i**:** *simulator choices for* $\mathbf{y}$*-side part corresponding to the i-th signing session*
v_i**:** *discrete logarithm of the* $\mathbf{z}$*-side commitment* $\mathbf{b}_i$ *in the i-th signing session*

Let $\overrightarrow{h}$ be the vector of responses returned by random oracle H (so $\left|\overrightarrow{h}\right| = \ell+1$), rand be the randomness used by the adversary M, and $\overline{\mathsf{info}}$ be the tag used in the OMUF game. We define a deterministic wrapper $\mathsf{A} := \mathsf{A}^{\mathsf{M}}_{\overline{\mathsf{info}}}$ that runs on $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ as shown in Figure 3. The wrapper allows us to argue about which $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ tuples cause the adversary to succeed.

A has two simulation modes. For $b = 0$, it runs the honest signer's algorithm to simulate both sign_1 and sign_2 oracle queries; for H^* queries, it responds with $\mathbf{z}$ if the input is $\overline{\mathsf{info}}$ and $\perp$ for all other inputs. In mode $b = 1$, A knows w and not x. It therefore runs the so-called $\mathbf{z}$-side signer (see Figure 7 in Appendix E), which is the honest signer's algorithm except that w is treated as the secret key.

A responds to queries to H^* with $\mathbf{g}^w$ for $\overline{\mathsf{info}}$ and $\perp$ otherwise. In both modes, A responds to queries to H using entries in the hash vector $\overrightarrow{h}$. Finally, upon receiving M's output message-signature pairs, A checks if they are all valid, and if so, A copies M's output (and outputs $\perp$ otherwise).

It is easy to see that

$$t_{\mathsf{A}} = t_{\mathsf{M}} + \mathrm{O}(\ell) = t_{\mathsf{U}} + \mathrm{O}(\ell) + \mathrm{O}({Q_h}^2) = t_{\mathsf{U}} + \mathrm{O}(\ell) + \mathrm{O}({Q_h}^2),$$

where the term $\mathrm{O}(\ell)$ comes from verifying $\ell + 1$ signatures, and $\mathrm{O}({Q_h}^2)$ comes from identifying the hash indices that correspond to signatures.

$\mathsf{A}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$
00 parse b from $\mathbf{I}$
01 if $b = 0$
02 parse $(b, x, \mathbf{z}, \overrightarrow{d}, \overrightarrow{s}, \overrightarrow{u}) := \mathbf{I}$
03 $\mathsf{pk} := \mathbf{g}^x$
04 else
05 parse $(b, \mathbf{y}, w, \overrightarrow{c}, \overrightarrow{r}, \overrightarrow{v}) := \mathbf{I}$
06 $\mathsf{pk} := \mathbf{y}$
07 $\mathsf{sid} := 0$
08 $j := 0$
09 $(m_i, (\rho_i, \omega_i, \sigma_i, \delta_i))_{i=1}^{\ell+1} := \mathsf{M}^{\mathsf{sign}_1, \mathsf{sign}_2, \mathsf{H}, \mathsf{H}^*}(\mathsf{pk}; \mathsf{rand})$
10 if $\forall i$: $\mathsf{Verify}(\mathsf{pk}, m_i, (\rho_i, \omega_i, \sigma_i, \delta_i))$
11 return $(m_i, (\rho_i, \omega_i, \sigma_i, \delta_i))_{i=1}^{\ell+1}$
12 else
13 return $\perp$

$\mathsf{H}(\xi)$
14 j++
15 return h_j

$\mathsf{H}^*(\mathsf{info})$
16 if $\mathsf{info} = \overline{\mathsf{info}}$
17 if $b = 0$ return $\mathbf{z}$
18 else return $\mathbf{g}^w$
19 else return $\perp$

$\mathsf{sign}_1(\mathsf{info})$
20 if $\mathsf{info} = \overline{\mathsf{info}}$
21 sid++
22 $\mathtt{open}(\mathsf{sid}) := \mathtt{true}$
23 if $b = 0$
24 $\mathbf{a}_{\mathsf{sid}} := \mathbf{g}^{u_{\mathsf{sid}}}$
25 $\mathbf{b}_{\mathsf{sid}} := \mathbf{g}^{s_{\mathsf{sid}}} \cdot \mathbf{z}^{d_{\mathsf{sid}}}$
26 else
27 $\mathbf{a}_{\mathsf{sid}} := \mathbf{g}^{r_{\mathsf{sid}}} \cdot \mathbf{y}^{c_{\mathsf{sid}}}$
28 $\mathbf{b}_{\mathsf{sid}} := \mathbf{g}^{v_{\mathsf{sid}}}$
29 return $(\mathsf{sid}, \mathbf{a}_{\mathsf{sid}}, \mathbf{b}_{\mathsf{sid}})$
30 else return $\perp$

$\mathsf{sign}_2(\mathsf{sid}, e_{\mathsf{sid}})$
31 if $\mathtt{open}(\mathsf{sid})$
32 if $b = 0$
33 $c_{\mathsf{sid}} := e_{\mathsf{sid}} - d_{\mathsf{sid}}$
34 $r_{\mathsf{sid}} := u_{\mathsf{sid}} - c_{\mathsf{sid}} \cdot x$
35 else
36 $d_{\mathsf{sid}} := e_{\mathsf{sid}} - c_{\mathsf{sid}}$
37 $s_{\mathsf{sid}} := v_{\mathsf{sid}} - d_{\mathsf{sid}} \cdot w$
38 else
39 return $\perp$
40 $\mathtt{open}(\mathsf{sid}) := \mathtt{false}$
41 return $(c_{\mathsf{sid}}, r_{\mathsf{sid}}, d_{\mathsf{sid}}, s_{\mathsf{sid}})$

Fig. 3: Wrapper A that simulates the OMUF game to the adversary M

The set of successful tuples. Let

$$\mathsf{Succ} := \{(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) | \mathsf{A}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \neq \perp\}$$

be the set of all "successful" input tuples to the wrapper A. For a pair of instance and randomness $\mathbf{I}, \mathsf{rand}$, it is also useful to define $\mathsf{Succ}_{\mathbf{I},\mathsf{rand}}$ as the set of successful input tuples with instance $\mathbf{I}$ and randomness rand, i.e.,

$$\mathsf{Succ}_{\mathbf{I},\mathsf{rand}} := \left\{ (\mathbf{I}', \mathsf{rand}', \overrightarrow{h}) \in \mathsf{Succ} \middle| \begin{matrix} \mathbf{I}' = \mathbf{I} \\ \mathsf{rand}' = \mathsf{rand} \end{matrix} \right\}.$$

In the following we further denote by $\mathcal{I}$ the set of all possible instances, by $\mathcal{R}$ the set of all possible randomness of A, and by ϵ the success probability of A,

i.e.,

$$\epsilon := \frac{|\mathsf{Succ}|}{\left|\mathcal{I} \times \mathcal{R} \times \mathbb{Z}_q^{\ell+1}\right|}$$

We show in Lemma 2 below (in Section 4.3) that the simulation using the $\mathbf{z}$-side witness is perfectly indistinguishable from the real execution where the $\mathbf{y}$-side witness is used (this is called the *witness indistinguishability* of the scheme), i.e., A simulates the OMUF game to M perfectly. Furthermore, if M succeeds, then so does A, since A copies M's output in this case (see lines 10–11 of Figure 3). Therefore,

$$\epsilon = \mathrm{adv}_{\mathsf{M}}^{\ell\text{-1-info-}\mathbf{OMUF}_{\mathsf{AO}}}.$$

4.2 Basic Definitions

We first define some concepts related to the wrapper A's input tuple $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$, that will be used throughout the security proof.

Transcripts. We begin with the definition of the *query transcript*, which consists of the adversary's signing queries:

Definition 5 (Query Transcript). *Consider the wrapper* A *running on input tuple* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$*. The* query transcript*, denoted* $\overrightarrow{e}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$*, is the vector of queries* e_{sid} *made to the* sign_2 *oracle (simulated by* A*) by the adversary* M*, ordered by* sid*.*

Next, we define (full) interaction *transcripts* between adversary M and wrapper A. These contain, in addition to $\overrightarrow{e}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$, also M's sign_1 queries and the signatures from the output of M. This will be useful to argue about A's behavior on different inputs $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$. Looking ahead, we will see that it is possible to deterministically transform $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ into a dual input $\Phi_{\mathsf{rand}, \overrightarrow{h}}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ that results in the same behavior as $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ (i.e., produces the same full transcript as $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$), but inverts the type of the witness $\mathbf{I}$ from $\mathbf{y}$-side to $\mathbf{z}$-side (or vice-versa).

Definition 6 (Full Transcripts). *Consider the wrapper* A *running on input tuple* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$*. We denote by* $\mathsf{tr}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ *the transcript produced between* A *and the adversary* M*, i.e., all messages sent between the user (played by* M*) and the signer (played by* A*). Concretely,*

$$\mathsf{tr}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = \left(\mathsf{info}, (\overrightarrow{\mathbf{a}}, \overrightarrow{\mathbf{b}}), \overrightarrow{e}, (\overrightarrow{c}, \overrightarrow{r}, \overrightarrow{d}, \overrightarrow{s}), \mathsf{sig}_1, \ldots \mathsf{sig}_{\ell+1}\right),$$

where $\mathsf{sig}_1, \ldots, \mathsf{sig}_{\ell+1}$ *are the signatures output by* M*. (If* M *aborts at any point during the protocol or outputs fewer than* $\ell + 1$ *signatures, we consider any undefined entry to be* $\perp$*.)*

Forking, partners, and triangles. We next define what it means for two input tuples to *fork* successfully — this corresponds to all cases where the reduction would be able to compute at least one of the two witnesses from the resulting signatures. However, without further work, the witness that can be computed might be the one that the reduction already knows.

Definition 7 (Successful forking). *We say two successful input tuples* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in \mathsf{Succ}$ fork *from each other at index* $i \in [\ell+1]$ *if* $\overrightarrow{h}_{[i-1]} = \overrightarrow{h}'_{[i-1]}$ *but* $h_i \neq h_i$. *We denote the set of hash vector pairs* $(\overrightarrow{h}, \overrightarrow{h}')$ *such that* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ *fork at index* i *as* $F_i(\mathbf{I}, \mathsf{rand})$.

We now define *partners*, which will play a key role in our analysis. Informally, two tuples $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ and $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ are partners at some index i if they fork from this index and produce the same query transcript (but not necessarily the same full transcript).

Definition 8 (Partners). *We say two (successful) tuples* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ *are* partners *at index* $i \in [\ell+1]$ *if the followings hold:*

- $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ *and* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ *fork at index* i
- $\overrightarrow{e}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = \overrightarrow{e}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$

We denote the set of $(\overrightarrow{h}, \overrightarrow{h}')$ *such that* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ *and* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ *are partners at index* i *by* $\mathrm{prt}_i(\mathbf{I}, \mathsf{rand})$. *We further denote by* $P_{\mathbf{I},\mathsf{rand}}$ *the following set:*

$$P_{\mathbf{I},\mathsf{rand}} = \left\{ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \mathsf{Succ}_{\mathbf{I},\mathsf{rand}} \middle| \exists \overrightarrow{h}', i \in [\ell+1] \colon (\overrightarrow{h}, \overrightarrow{h}') \in \mathrm{prt}_i(\mathbf{I}, \mathsf{rand}) \right\}$$

We define *triangles* in order to extend the nice properties of partners to more general forking tuples. Informally, a triangle consists of three vectors $\overrightarrow{h}, \overrightarrow{h'}, \overrightarrow{h''}$ which all fork from each other at the same index, and also have the property that $\overrightarrow{h}$ and $\overrightarrow{h'}$ are partners at this index. This way, it is natural to view these vectors as corners of the triangle and any pair of two vectors as the sides.

Definition 9 (Triangles). *A* triangle *at index* $i \in [\ell+1]$ *with respect to* $\mathbf{I}, \mathsf{rand}$ *is a tuple of three (successful) tuples in the following set:*

$$\triangle_i(\mathbf{I}, \mathsf{rand}) = \left\{ \begin{array}{l} ((\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), \\ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'), \\ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'')) \end{array} \middle| \begin{array}{l} (\overrightarrow{h}, \overrightarrow{h}') \in \mathrm{prt}_i(\mathbf{I}, \mathsf{rand}) \\ (\overrightarrow{h}, \overrightarrow{h}'') \in F_i(\mathbf{I}, \mathsf{rand}) \\ (\overrightarrow{h}', \overrightarrow{h}'') \in F_i(\mathbf{I}, \mathsf{rand}) \end{array} \right\}$$

For a triangle $((\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'')) \in \triangle_i(\mathbf{I}, \mathsf{rand})$, *we call the pair of tuples* $((\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'))$ *the* base, *and* $((\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}''))$ *and* $((\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}''))$ *the* sides. *We further refer to the tuples* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'')$ *as* corners, *where the two corners incident to the base are called* base corners, *and the third corner is called the* top. *We will sometimes write* $(\overrightarrow{h}, \overrightarrow{h}', \overrightarrow{h}'') \in \triangle_i(\mathbf{I}, \mathsf{rand})$ *for compactness.*

Maximum branching index and set. In the following we define two important characteristics of partner tuples. We begin by defining the *maximum branching index*, which is the index at which a partner tuple $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in P_{\mathbf{I},\mathsf{rand}}$ has *the most partners.*

Definition 10 (Maximum Branching Index). *Fix a pair* $\mathbf{I}, \mathsf{rand}$*. The* maximum branching index *of a partner tuple* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in P_{\mathbf{I},\mathsf{rand}}$ *is the index at which* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ *has the most partners, i.e.,*

$$\mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = \mathrm{argmax}_{i \in [\ell+1]} \left| \left\{ \overrightarrow{h}' \middle| (\overrightarrow{h}, \overrightarrow{h}') \in \mathrm{prt}_i(\mathbf{I}, \mathsf{rand}) \right\} \right|.$$

In case of ties, we pick the lowest such index.

The maximum branching index naturally defines a partition of any non-empty set of partnered tuples $P_{\mathbf{I},\mathsf{rand}}$, where the i-th set of the partition contains all tuples with maximum branching index i. We define the *maximum branching set* as the largest part of this partition, i.e., the largest subset of tuples that share a common maximum branching index.

Definition 11 (Maximum Branching Set). *For a pair* $\mathbf{I}, \mathsf{rand}$*, consider the partition of partner tuples according to their maximal branching indices:*

$$B_i(\mathbf{I}, \mathsf{rand}) = \left\{ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \middle| \mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = i \right\}.$$

The maximum branching set *of* $\mathbf{I}, \mathsf{rand}$ *is defined as the largest set among them, i.e.,*

$$B_{max}(\mathbf{I}, \mathsf{rand}) = B_{i_{max}(\mathbf{I},\mathsf{rand})}(\mathbf{I}, \mathsf{rand}),$$

where

$$i_{max}(\mathbf{I}, \mathsf{rand}) = \mathrm{argmax}_{i \in [\ell+1]} \left| B_i(\mathbf{I}, \mathsf{rand}) \right|.$$

In case of ties, we pick the lowest such index.

Note in particular that $B_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})}(\mathbf{I}, \mathsf{rand})$ (henceforth $B_{\mathrm{Br}_{\max}}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ for simplicity) is the set of all tuples $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ which have the same maximum branching index as $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ (so $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in B_{\mathrm{Br}_{\max}}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$).

4.3 The Mapping Φ

For any successful tuple $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$, we now define the mapping $\Phi_{\mathsf{rand},\overrightarrow{h}}$ and prove its transcript preserving properties in Lemma 2. We remark that this mapping is not efficiently computable and will merely serve as a technical tool in our analysis.

Definition 12 (Mapping instances via transcript). *For* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \mathsf{Succ}$*, we define* $\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I})$ *as follows. For a* $\mathbf{y}$*-side instance* $\mathbf{I} = (1, w, \mathbf{y}, \overrightarrow{c}, \overrightarrow{r}, \overrightarrow{u})$*,* $\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I})$ *is a* $\mathbf{z}$*-side instance that consists of*

$$b = 0 \qquad x = \mathrm{dlog}\, \mathbf{y} \qquad \mathbf{z} = \mathbf{g}^w \qquad \forall i \in [\ell] \colon d_i = e_i - c_i$$

$$\forall i \in [\ell]\colon s_i = u_i - d_i \cdot w \qquad \forall i \in [\ell]\colon v_i = c_i \cdot x + r_i$$

For a $\mathbf{z}$*-side instance* $\mathbf{I} = (0, x, \mathbf{z}, d, s, v)$*,* $\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I})$ *is a* $\mathbf{y}$*-side instance that consists of*

$$b = 1 \qquad w = \operatorname{dlog} \mathbf{z} \qquad \mathbf{y} = \mathbf{g}^x \qquad \forall i \in [\ell]\colon c_i = e_i - d_i$$
$$\forall i \in [\ell]\colon r_i = v_i - c_i \cdot x \qquad \forall i \in [\ell]\colon u_i = d_i \cdot w + s_i$$

(where $\overrightarrow{e}$ *is the query vector produced by* $\mathsf{rand}, \overrightarrow{h}$ *using instance* $\mathbf{I}$*). We will sometimes use the notation* $\Phi_{\overrightarrow{e}}$ *instead of* $\Phi_{\mathsf{rand},\overrightarrow{h}}$ *for a given* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$*. We also define* $\Phi(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = (\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h})$*.*

Lemma 2 ($\Phi_{\mathsf{rand},\overrightarrow{h}}$ is a bijection that preserves transcripts). *Fix* $\mathsf{rand}, \overrightarrow{h}$*. For all tuples* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \mathsf{Succ}$*,* $\Phi_{\mathsf{rand},\overrightarrow{h}}$ *is a self-inverse bijection and*

$$\mathsf{tr}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = \mathsf{tr}(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h})$$

The proof is deferred to Appendix C.3.

The lemma above shows that the Abe-Okamoto scheme is *witness indistinguishable*, i.e., a simulator that uses the $\mathbf{z}$-side witness to sign (see Figure 7 in Appendix E) creates a view identical to the real view to the adversary. In particular, this implies that the wrapper A simulates the ℓ-OMUF game to the adversary M perfectly.

Corollary 1. $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \mathsf{Succ} \Leftrightarrow (\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h}) \in \mathsf{Succ}$*.*

We look into the effect of the transcript mapping function on partner tuples. We have proven that $\Phi_{\mathsf{rand},\overrightarrow{h}}$ preserves the transcript (and hence success) of $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$. However, note that this does not (by itself) imply that partnering tuples $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ and $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ result in partnering tuples $(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h})$ and $(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h}')$, or $(\Phi_{\mathsf{rand},\overrightarrow{h}'}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h})$ and $(\Phi_{\mathsf{rand},\overrightarrow{h}'}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h}')$, respectively. Lemma 3 asserts that this is indeed the case.

Lemma 3 (Partners stay partners through Φ). *For all* $\mathbf{I}, \mathsf{rand}$*, and vectors* $\overrightarrow{h}, \overrightarrow{h}'$*,*

$$(\overrightarrow{h}, \overrightarrow{h}') \in \mathrm{prt}_i(\mathbf{I}, \mathsf{rand}) \Leftrightarrow (\overrightarrow{h}, \overrightarrow{h}') \in \mathrm{prt}_i(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand})$$
$$\Leftrightarrow (\overrightarrow{h}, \overrightarrow{h}') \in \mathrm{prt}_i(\Phi_{\mathsf{rand},\overrightarrow{h}'}(\mathbf{I}), \mathsf{rand})$$

Proof. Suppose $(\overrightarrow{h}, \overrightarrow{h}') \in \mathrm{prt}_i(\mathbf{I}, \mathsf{rand}) \subset F_i(\mathbf{I}, \mathsf{rand})$; we have that $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$, $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in \mathsf{Succ}$. Then by Corollary 1, $(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h})$, $(\Phi_{\mathsf{rand},\overrightarrow{h}'}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h}') \in \mathsf{Succ}$.

Furthermore, as $\overrightarrow{h}, \overrightarrow{h}'$ are partners for $\mathbf{I}, \mathsf{rand}$, they produce the same query vector $\overrightarrow{e}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = \overrightarrow{e}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$. Thus $\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}) = \Phi_{\overrightarrow{e}}(\mathbf{I}) = \Phi_{\mathsf{rand},\overrightarrow{h}'}(\mathbf{I})$. Using this fact, we obtain

$$\begin{aligned}&\overrightarrow{e}(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h}) = \overrightarrow{e}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = \overrightarrow{e}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')\\ &= \overrightarrow{e}(\Phi_{\mathsf{rand},\overrightarrow{h}'}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h}') = \overrightarrow{e}(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h}')\end{aligned}$$

as follows. The first equality follows because $\overrightarrow{e}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ is contained in $\mathsf{tr}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ and by Lemma 2, we have that $\mathsf{tr}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = \mathsf{tr}(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h})$. The second equality holds because $\overrightarrow{h}$ and $\overrightarrow{h}'$ are partners. The third equality follows because $\overrightarrow{e}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ is contained in $\mathsf{tr}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ and from Lemma 2, we have that $\mathsf{tr}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') = \mathsf{tr}(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h}')$. The fourth equality holds by another application of Lemma 2 which yields $\mathsf{tr}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') = \mathsf{tr}(\Phi_{\mathsf{rand},\overrightarrow{h}'}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h}') = \mathsf{tr}(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h}')$.

Combining the two paragraphs above, we get $(\overrightarrow{h}, \overrightarrow{h}') \in \mathrm{prt}_i(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand})$. Using a similar argument, $(\overrightarrow{h}, \overrightarrow{h}') \in \mathrm{prt}_i(\Phi_{\mathsf{rand},\overrightarrow{h}'}(\mathbf{I}), \mathsf{rand})$. The inverse direction follows from the self-inverse property of $\Phi_{\mathsf{rand},\overrightarrow{h}}$. □

Corollary 2. $\mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = \mathrm{Br}_{\max}(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h})$.

4.4 Extracting a Witness from a Fork

Witness Extraction. We briefly recall how the reduction can compute a witness from two signatures from forking runs of the wrapper A. We say a signature $(\rho, \omega, \sigma, \delta)$ on a message m in the output of A on input $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ *corresponds to* a hash value h_i, if $\mathsf{H}(\mathbf{g}^{\rho}\mathbf{y}^{\omega}, \mathbf{g}^{\sigma}\mathbf{z}^{\delta}, \mathbf{z}, m)$ was the i-th hash query made to the random oracle H in this run of A. Informally we say that a witness *can be extracted* from $\mathbf{I}, \mathsf{rand}$, and a pair of forking hash vectors $(\overrightarrow{h}, \overrightarrow{h}') \in F_i(\mathbf{I}, \mathsf{rand})$, if it can be efficiently computed from the two signatures corresponding to h_i and h'_i. We make this formal in the following definition.

Definition 13 (Witness Extraction). *Fix* $\mathbf{I}, \mathsf{rand}$ *and let* $(\overrightarrow{h}, \overrightarrow{h}') \in F_i(\mathbf{I}, \mathsf{rand})$ *for some* $i \in [\ell + 1]$. *Moreover, denote* $\mathsf{sig}_i, \mathsf{sig}'_i$ *the signatures that correspond to* h_i *and* h'_i, *respectively. Consider the two witness extraction algorithms* $\mathsf{E}_{\mathbf{y}}, \mathsf{E}_{\mathbf{z}}$ *as described in Figure 4. For* $\times \in \{\mathbf{y}, \mathbf{z}\}$, *we say that the* $\times$*-side witness* can be extracted from $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ and $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ *at index* i *if* $\mathsf{E}_{\times}$ *on input* $(\mathsf{sig}_i, \mathsf{sig}'_i)$ *does not return* $\perp$.

Lemma 4. *Let* $\mathbf{I}, \mathsf{rand}, i, (\overrightarrow{h}, \overrightarrow{h}') \in F_i(\mathbf{I}, \mathsf{rand}), \mathsf{sig}_i, \mathsf{sig}'_i$, *and algorithms* $\mathsf{E}_{\mathbf{y}}, \mathsf{E}_{\mathbf{z}}$ *be as in Definition 13. Then at least one of* $\mathsf{E}_{\mathbf{y}}$ *and* $\mathsf{E}_{\mathbf{z}}$ *outputs a correct witness on input the two signatures* $\mathsf{sig}_i = (\rho_i, \omega_i, \sigma_i, \delta_i)$ *and* $\mathsf{sig}'_i = (\rho'_i, \omega'_i, \sigma'_i, \delta'_i)$ *corresponding to* h_i *and* h'_i. *More specifically,* $\mathsf{E}_{\mathbf{y}}$ *outputs the* $\mathbf{y}$*-side witness if and only if* $\omega_i \neq \omega'_i$, *otherwise* $\mathsf{E}_{\mathbf{z}}$ *outputs the* $\mathbf{z}$*-side witness.*

The proof is a standard forking argument and is deferred to Appendix C.5.

$\underline{\mathsf{E}_{\mathbf{y}}((\rho_i,\omega_i,\sigma_i,\delta_i),(\rho_i',\omega_i',\sigma_i',\delta_i'))}$	$\underline{\mathsf{E}_{\mathbf{z}}((\rho_i,\omega_i,\sigma_i,\delta_i),(\rho_i',\omega_i',\sigma_i',\delta_i'))}$
42 if $(\omega_i \neq \omega_i')$	46 if $(\delta_i \neq \delta_i')$
43 return $x := \frac{\rho_i - \rho_i'}{\omega_i' - \omega_i}$	47 return $w := \frac{\sigma_i - \sigma_i'}{\delta_i' - \delta_i}$
44 else	48 else
45 return $\perp$	49 return $\perp$

Fig. 4: The two witness extraction algorithms from Definition 13

Remark 1. We note that the witness may be contained in the instance $\mathbf{I}$, in which case the witness can be trivially extracted. For the purposes of the lemma we only consider the more interesting case that the witness can be computed from the two signatures directly, regardless of which witness was used for simulating the signing oracles.

Witnesses in triangles. We now show that if a witness can be extracted from the base of a triangle, it can also be extracted from at least one of the sides. This was previously shown in [4].

Corollary 3. *Fix* $\mathbf{I}, \mathsf{rand}$ *and let* $(\overrightarrow{h}, \overrightarrow{h}', \overrightarrow{h}'') \in \triangle_i(\mathbf{I}, \mathsf{rand})$ *for some* $i \in [\ell + 1]$*. Moreover, suppose that the* $\mathbf{y}$*-side witness can be extracted from the base* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ *of the triangle at index* i*. Then the* $\mathbf{y}$*-side witness can also be extracted from at least one of the sides* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'')$ *or* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'')$ *at index* i*. An analogous statement holds for the* $\mathbf{z}$*-side witness.*

Proof. Toward a contradiction, suppose that the $\mathbf{y}$-side witness can be extracted from the base $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ at index i, but can not be extracted at index i for either of the sides $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'')$ or $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'')$. Then, by Lemma 4, $\omega_i = \omega_i''$ and $\omega_i' = \omega_i''$, so $\omega_i = \omega_i'$. By Lemma 4 again, the $\mathbf{y}$-side witness can not be extracted from $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$, a contradiction. An analogous argument can be made for the $\mathbf{z}$-side. □

We now define *both-sided triangle base corners* as triangle base corners $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ which remain base corners of some triangle at their maximal branching index when mapped via $\Phi_{\mathsf{rand}, \overrightarrow{h}}$. (Recall that by Corollary 2, the maximal branching index is preserved under Φ.) On top of this, if $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ is a both-sided triangle base corner, and forms a triangle base with $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ at index $\mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$, then $(\Phi_{\mathsf{rand}, \overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h})$ and $(\Phi_{\mathsf{rand}, \overrightarrow{h}}, \mathsf{rand}, \overrightarrow{h}')$ also form a triangle base.

For every such tuple $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$, we further define the set $D_i^{\mathbf{y}}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ of tuples that form a both-sided triangle base with $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ at index i from which the $\mathbf{y}$-side witness can be extracted, and an analogous set $D_i^{\mathbf{z}}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ for the $\mathbf{z}$-side witness. This allows us to then define sets $B_T^{\mathbf{y}}$ and $B_T^{\mathbf{z}}$ that contain tuples where the majority of both-sided triangle bases incident to the tuple allow for extraction of the $\mathbf{y}$-side or $\mathbf{z}$-side witness, respectively.

Definition 14 (Both-sided Triangle Base Corners). *We call elements of the set*

$$B_T := \left\{ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \;\middle|\; \begin{array}{c} \exists \overrightarrow{h}', \\ \overrightarrow{h}'', \overrightarrow{h}''' : \end{array} \begin{array}{c} (\overrightarrow{h}, \overrightarrow{h}', \overrightarrow{h}'') \in \triangle_{\mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})}(\mathbf{I}, \mathsf{rand}) \\ (\overrightarrow{h}, \overrightarrow{h}', \overrightarrow{h}''') \in \triangle_{\mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})}(\Phi_{\mathsf{rand}, \overrightarrow{h}}(\mathbf{I}), \mathsf{rand}) \end{array} \right\}$$

both-sided triangle base corners. *For any index $i \in [\ell + 1]$, we define sets*

$$D_i^{\mathbf{y}}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) := \left\{ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \;\middle|\; \begin{array}{c} \exists \overrightarrow{h}'', \\ \overrightarrow{h}''' : \end{array} \begin{array}{c} (\overrightarrow{h}, \overrightarrow{h}', \overrightarrow{h}'') \in \triangle_i(\mathbf{I}, \mathsf{rand}) \\ (\overrightarrow{h}, \overrightarrow{h}', \overrightarrow{h}''') \in \triangle_i(\Phi_{\mathsf{rand}, \overrightarrow{h}}(\mathbf{I}), \mathsf{rand}) \\ \text{The } \mathbf{y}\text{-side witness can be} \\ \text{extracted from } (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), \\ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \text{ at index } i \end{array} \right\}$$

and $B_T^{\mathbf{y}} \subset B_T$ as

$$B_T^{\mathbf{y}} := \left\{ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \;\middle|\; \begin{array}{l} D^{\mathbf{y}}_{\mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \neq \emptyset \\ \left| D^{\mathbf{y}}_{\mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \right| \\ \geq \left| D^{\mathbf{z}}_{\mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \right| \end{array} \right\}$$

We define sets $D_i^{\mathbf{z}}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ and $B_T^{\mathbf{z}}$ analogously.

Lemma 5 (Both-sided triangle bases produce the same witness on both sides).

1. *$\Phi(B_T^{\mathbf{y}}) = B_T^{\mathbf{y}}$ and $\Phi(B_T^{\mathbf{z}}) = B_T^{\mathbf{z}}$;*
2. *$B_T^{\mathbf{y}} \cup B_T^{\mathbf{z}} = B_T$.*

We defer the proof to Appendix C.5.

We define $B_T^{\times}$ as the larger set of $B_T^{\mathbf{y}}$ and $B_T^{\mathbf{z}}$. By the second item of Lemma 5, $\left|B_T^{\times}\right| \geq \frac{1}{2}\left|B_T\right|$.

Let $B_{T,\mathbf{y}}^{\times}$ (resp. $B_{T,\mathbf{z}}^{\times}$) be the subset of $B_T^{\times}$ with $\mathbf{y}$-side instances (resp. $\mathbf{z}$-side instances). We stress that $B_T^{\mathbf{y}}$ and $B_{T,\mathbf{y}}^{\times}$ are two different sets: $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in B_T^{\mathbf{y}}$ means that more both-sided triangle bases (with $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ as one of its corners) allow for extracting the $\mathbf{y}$-side witness than the $\mathbf{z}$-side witness; whereas $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in B_{T,\mathbf{y}}^{\times}$ means that $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in B_T^{\times}$ and $\mathbf{I}$ is a $\mathbf{y}$-side witness.

Lemma 6. $\left|B^{\times}_{T,\mathbf{y}}\right| = \left|B^{\times}_{T,\mathbf{z}}\right| = \frac{1}{2}\left|B^{\times}_{T}\right|$.

Proof. By the first item of Lemma 5, Φ is a bijection within $B^{\times}_{T}$, and since Φ maps a tuple with a $\mathbf{y}$-side instance to a tuple with a $\mathbf{z}$-side instance (and vice versa), we know that Φ is a bijection between $B^{\times}_{T,\mathbf{y}}$ and $B^{\times}_{T,\mathbf{z}}$; therefore, $\left|B^{\times}_{T,\mathbf{y}}\right| = \left|B^{\times}_{T,\mathbf{z}}\right|$. Since $B^{\times}_{T,\mathbf{y}}$ and $B^{\times}_{T,\mathbf{z}}$ form a partition of $B^{\times}_{T}$, we know that $\left|B^{\times}_{T,\mathbf{y}}\right| + \left|B^{\times}_{T,\mathbf{z}}\right| = \left|B^{\times}_{T}\right|$, and the lemma follows. □

We now give a lower bound of the size of $B^{\times}_{T}$. Let $\epsilon_{B^{\times}_{T}}$ be the probability of getting a tuple in $B^{\times}_{T}$ while sampling uniformly at random, i.e.,

$$\epsilon_{B^{\times}_{T}} \coloneqq \frac{\left|B^{\times}_{T}\right|}{\left|\mathcal{I} \times \mathcal{R} \times {\mathbb{Z}_q}^{\ell+1}\right|}.$$

Lemma 7 (Lower-bounding the size of $B^{\times}_{T}$). *Assume* $\epsilon \geq \frac{432\left(1-\frac{1}{(\ell+1)^2}\right)}{q}$. *Then*

$$\epsilon_{B^{\times}_{T}} \geq \frac{\epsilon}{96}.$$

We defer the proof to Appendix C.5.

Finding triangle tops. In order for our security proof to go through, a key step is to compute the probability that the reduction hits a triangle side from which the $\times$-side witness can be extracted when forking the wrapper, independently of the witness that is being used by the reduction. This event is crucial in our proof because, assuming that the reduction samples one of these sides, it is likely that it did so with the witness opposite of $\times$, meaning that it extracts the witness $\times$ it *does not already know* with significant probability, hence solving the discrete logarithm problem. In order to lower bound the probability of the event above, we first define *relevant triangle tops* for a both-sided triangle base corner $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in B^{\times}_{T}$. These are all the tuples $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'')$ such that $(\overrightarrow{h}, \overrightarrow{h}', \overrightarrow{h}'')$ forms triangles at index i (where $\overrightarrow{h}'$ is as in the definition of both-sided triangle tops (Definition 14)).

Definition 15 (Relevant triangle tops). *For a tuple* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$, *define its* relevant triangle tops *at index* i *as tuples in the following set:*

$$T^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \coloneqq \left\{ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'') \;\middle|\; \exists\, \overrightarrow{h}' \colon \begin{array}{c} (\overrightarrow{h}, \overrightarrow{h}', \overrightarrow{h}'') \in \triangle_i(\mathbf{I}, \mathsf{rand}) \\ \text{The } \times\text{-side witness} \\ \text{can be extracted from } (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), \\ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \text{ at } i \end{array} \right\}$$

We will mostly consider relevant triangle tops at the maximum branching index $\mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ and we thus define $T_T^{\times}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \coloneqq T^{\times}_{T,\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$.

What remains to be shown is that many elements of $B_T^{\times}$ actually have many relevant triangle tops, regardless of whether they reside in $B^{\times}_{T,\mathbf{y}}$ or $B^{\times}_{T,\mathbf{z}}$, i.e., independently of the witness that they store. This ensures that when the reduction samples and then (partially) resamples the vectors during the forking process, it will hit a side from which the desired witness can be extracted with significant probability, as explained above.

Lemma 8 (There are enough relevant triangle tops). *There exists a subset $G_{\mathbf{y}} \subset B^{\times}_{T,\mathbf{y}}$ with $|G_{\mathbf{y}}| \geq \frac{3}{8}\left|B^{\times}_{T,\mathbf{y}}\right|$ such that for each $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in G_{\mathbf{y}}$,*

$$\left|T_T^{\times}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})\right| \geq \frac{\epsilon_{B_T^{\times}}}{16(\ell+1)} \cdot q^{\ell-\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})+2} - 2q^{\ell-\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})+1}.$$

An analogous statement holds for $B^{\times}_{T,\mathbf{z}}$.

The proof is deferred to Appendix C.5.

Corollary 4. *Let $G_{\mathbf{y}}$ be as in Lemma 8. Then*

$$\Pr_{\substack{(\mathbf{I},\mathsf{rand},\overrightarrow{h})\xleftarrow{\$}\mathcal{I}\times\mathcal{R}\times\mathbb{Z}_q{}^{\ell+1} \\ i\xleftarrow{\$}[\ell+1],\overrightarrow{h}'\xleftarrow{\$}\mathbb{Z}_q{}^{\ell+1}_{|\overrightarrow{h}_{[i-1]}}}}\left[(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in T_T^{\times}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \,\middle|\, \begin{matrix}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in G_{\mathbf{y}} \\ \mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = i\end{matrix}\right]$$
$$\geq \frac{\epsilon_{B_T^{\times}}}{16(\ell+1)} - \frac{2}{q}.$$

An analogous statement holds for $G_{\mathbf{z}}$.

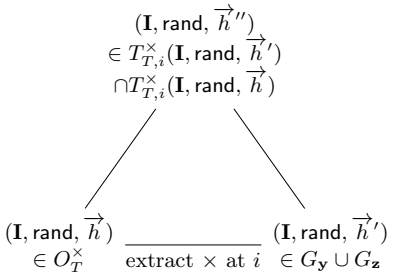
Proof. Suppose $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in G_{\mathbf{y}}$ and $\mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = i$. Note that $\left|\mathbb{Z}_q{}^{\ell+1}_{|\overrightarrow{h}_{[i-1]}}\right| = q^{\ell-i+2}$. Therefore, the probability of sampling an $\overrightarrow{h}'$ such that $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in T_T^{\times}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ is

$$\frac{\left|T_T^{\times}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})\right|}{q^{\ell-i+2}} \geq \frac{\frac{\epsilon_{B_T^{\times}}}{16(\ell+1)} \cdot q^{\ell-i+2} - 2q^{\ell-i+1}}{q^{\ell-i+2}} = \frac{\epsilon_{B_T^{\times}}}{16(\ell+1)} - \frac{2}{q}.$$

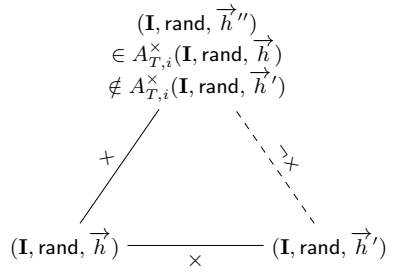
□

Opposing base corners. By Corollary 3 we know that each triangle with a relevant base has at least one relevant side. We now want to consider the probability of finding such a relevant side in the forking proof.

To this end, we consider *opposing base corners* — corners of relevant bases whose partners are in $G_{\mathbf{y}}$ or $G_{\mathbf{z}}$. See Figure 5a for a graphic illustration. (Keep in mind that the sets $G_{\mathbf{y}}$ and $G_{\mathbf{z}}$ are the sets of both sided triangle base corners for which there exist many triangle tops.)



(a) Definition of $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in O^{\times}_T$ where $i = \mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$.



(b) A useful triangle top for a base corner is one where the $\times$-side witness can be extracted. The $\times$ (resp. $\neg\times$) on an edge means that the $\times$-side witness can (resp. cannot) be extracted at index i.

Fig. 5: Opposing base corners and useful triangle tops

Definition 16 (Opposing base corners).

$$O^{\times}_T \coloneqq \left\{ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \;\middle|\; \exists \overrightarrow{h}' \colon \begin{array}{c} (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in G_{\mathbf{y}} \cup G_{\mathbf{z}} \\ (\overrightarrow{h}, \overrightarrow{h}') \in \mathrm{prt}_{\mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')}(\mathbf{I}, \mathsf{rand}) \\ \text{the } \times\text{-side witness can be} \\ \text{extracted from } (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), \\ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \text{ at } \mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \end{array} \right\}$$

Good corners with useful tops. For each tuple $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ in $O^{\times}_T$ or $B^{\times}_T$ we define *useful triangle tops* — triangle tops that allow for extraction of the $\times$-side witness when combined with the base corner $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ (see Figure 5b for a graphic illustration):

Definition 17 (Useful triangle tops). *For any* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in O^{\times}_T \cup B^{\times}_T$*, define*

$$A^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \coloneqq \left\{ \begin{array}{c} (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'') \\ \in T^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \end{array} \;\middle|\; \begin{array}{c} \text{the } \times\text{-side witness can be} \\ \text{extracted from } (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), \\ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'') \text{ at index } i \end{array} \right\}$$

Recall that relevant base corners — those in $G_{\mathbf{y}}$ or $G_{\mathbf{z}}$ — are tuples in $B^{\times}_T$ for which many triangle tops are relevant (i.e., the corresponding $T^{\times}_T$ set is large). We now consider a subset of these relevant base corners for which a lot of the relevant triangle tops are useful (i.e., the corresponding $A^{\times}_T$ set is large). We call these base corners *good*.

Definition 18 (Good base corners). *We say that a base corner in $G_{\mathbf{y}} \cup G_{\mathbf{z}}$ is* good *if it lies within the following set:*

$$\widehat{B_T^{\times}} \coloneqq \left\{ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in G_{\mathbf{y}} \cup G_{\mathbf{z}} \;\middle|\; \begin{array}{l} \left|A_T^{\times}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})\right| \\ \geq \frac{1}{2}\left|T_T^{\times}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})\right| \\ -q^{\ell - \mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})+1} \end{array} \right\}$$

We now want to show that if the set of good base corners is small, then there exist a lot of opposing base corners — which we call *good opposing base corners* — that fulfill a property analogous to good base corners.

Definition 19 (Good opposing base corners).

$$\widehat{O_T^{\times}} \coloneqq \left\{ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \;\middle|\; \exists \overrightarrow{h}' \colon \begin{array}{c} (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in G_{\mathbf{y}} \cup G_{\mathbf{z}} \\ (\overrightarrow{h}, \overrightarrow{h}') \in \mathrm{prt}_{\mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')}(\mathbf{I}, \mathsf{rand}) \\ \text{the } \times\text{-side witness can be} \\ \text{extracted from } (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), \\ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \text{ at } \mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \\ \left|A^{\times}_{T, \mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})\right| \\ \geq \frac{1}{2}\left|T^{\times}_{T, \mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})\right| \\ -q^{\ell - \mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')+1} \end{array} \right\}$$

Let $\widehat{B^{\times}_{T,\mathbf{y}}} \subset \widehat{B_T^{\times}}$ and $\widehat{O^{\times}_{T,\mathbf{y}}} \subset \widehat{O_T^{\times}}$ be analogous to $B^{\times}_{T,\mathbf{y}} \subset B_T^{\times}$, i.e., the subset of tuples with $\mathbf{y}$-side instances. We define $\widehat{B^{\times}_{T,\mathbf{z}}}$ and $\widehat{O^{\times}_{T,\mathbf{z}}}$ similarly.

Lemma 9. *If* $\left|\widehat{B^{\times}_{T,\mathbf{y}}}\right| < \frac{1}{2}|G_{\mathbf{y}}|$*, then* $\left|\widehat{O^{\times}_{T,\mathbf{y}}}\right| \geq \frac{1}{8(\ell+1)}|G_{\mathbf{y}}|$*. An analogous statement holds for* $\mathbf{z}$*.*

Proof. Let $F = G_{\mathbf{y}} \setminus \widehat{B^{\times}_{T,\mathbf{y}}}$ (so $|F| \geq \frac{1}{2}|G_{\mathbf{y}}|$). Consider any $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in F$, and let $i = \mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$. Then

$$\left|A^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')\right| < \frac{1}{2}\left|T^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')\right| - q^{\ell - i + 1}.$$

By Corollary 3, for any $(\overrightarrow{h}, \overrightarrow{h}', \overrightarrow{h}'') \in \triangle_i(\mathbf{I}, \mathsf{rand})$ such that the $\times$-side witness can be extracted from the base $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$, if the $\times$-side witness cannot be extracted from $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'')$, then it can be extracted from $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'')$. (All extractions mentioned above are at index i.) Therefore,

$$\left|A^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})\right| + \left|A^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')\right| \geq \left|T^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')\right|.$$

We note that all but $q^{\ell-i+1}$ elements of $T^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ are also elements of $T^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$. This is because $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}^*) \in T^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \backslash T^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ implies that $(\overrightarrow{h}, \overrightarrow{h}^*) \in F_i(\mathbf{I}, \mathsf{rand})$ but $(\overrightarrow{h}', \overrightarrow{h}^*) \notin F_i(\mathbf{I}, \mathsf{rand})$, which means that $\overrightarrow{h}^*$ must share its first i entries with $\overrightarrow{h}'$ (recall that $\overrightarrow{h}$ and $\overrightarrow{h}'$ share the first $i-1$ entries), so there are at most $q^{\ell-i+1}$ such vectors. We get that

$$\left|T^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')\right| \geq \left|T^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})\right| - q^{\ell-i+1}.$$

Combining all inequalities above, we get

$$\begin{aligned}
\left|A^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})\right| &\geq \left|T^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')\right| - \left|A^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')\right| \\
&> \left|T^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')\right| - \left(\frac{1}{2}\left|T^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')\right| - q^{\ell-i+1}\right) \\
&= \frac{1}{2}\left|T^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')\right| + q^{\ell-i+1} \\
&\geq \frac{1}{2}\left(\left|T^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})\right| - q^{\ell-i+1}\right) + q^{\ell-i+1} \\
&> \frac{1}{2}\left|T^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})\right| - q^{\ell-i+1}
\end{aligned}$$

I.e., if $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in F$, then all of its partners $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ at index i with which it forms triangle bases from which the $\times$-side witness can be extracted, are in $\widehat{O^{\times}_{T,\mathbf{y}}}$.

We now lower-bound the number of such partners $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$. Define the set of tuples that yield the same query transcript with $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ as

$$E(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') = \{(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}^{\star}) | \overrightarrow{e}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}^{\star}) = \overrightarrow{e}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')\}.$$

Note that $E(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ is the set of partners of $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ at any index. Consider a subset $E_i(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ of all tuples that fork from $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ at index i, i.e., $E_i(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') = \{(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}^{\star}) | (\overrightarrow{h}^{\star}, \overrightarrow{h}') \in \mathrm{prt}_i(\mathbf{I}, \mathsf{rand})\}$. Recall that $i = \mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$. By the definition of maximum branching index, we have

$$\left|E_i(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')\right| \geq \frac{1}{\ell+1}\left(\left|E(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')\right| - 1\right) \geq \frac{1}{2(\ell+1)}\left|E(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')\right|$$

(where the -1 comes from excluding $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ itself). As $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in B^{\times}_{T}$, it holds that at least half of the tuples in $E_i(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$, together with $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$, allow for the extraction of the $\times$-side witness. This means that at least half of the tuples in $E_i(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ are in $\widehat{O^{\times}_{T,\mathbf{y}}}$.

We have shown that for any $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in F$, at least $\frac{1}{4(\ell+1)}$ of tuples in $E(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ are in $\widehat{O^{\times}_{T,\mathbf{y}}}$. Further note that for any $(\mathbf{I}_1, \mathsf{rand}_1, \overrightarrow{h}_1)$ and

$(\mathbf{I}_2, \mathsf{rand}_2, \overrightarrow{h}_2)$, either $E(\mathbf{I}_1, \mathsf{rand}_1, \overrightarrow{h}_1) = E(\mathbf{I}_2, \mathsf{rand}_2, \overrightarrow{h}_2)$ or $E(\mathbf{I}_1, \mathsf{rand}_1, \overrightarrow{h}_1) \cap E(\mathbf{I}_2, \mathsf{rand}_2, \overrightarrow{h}_2) = \emptyset$. [7] Summing over all $E(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ for some $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in F$, we get

$$\begin{aligned}
\left|O_T^{\times}\right| &\geq \frac{1}{4(\ell+1)} \sum_{\substack{E \text{ s.t. } E=E(\mathbf{I},\mathsf{rand},\overrightarrow{h}') \\ \text{for some } (\mathbf{I},\mathsf{rand},\overrightarrow{h}')\in F}} |E| \geq \frac{1}{4(\ell+1)} \sum_{\substack{E \text{ s.t. } E=E(\mathbf{I},\mathsf{rand},\overrightarrow{h}') \\ \text{for some } (\mathbf{I},\mathsf{rand},\overrightarrow{h}')\in F}} |E \cap F| \\
&= \frac{1}{4(\ell+1)} \left| \bigcup_{\substack{E \text{ s.t. } E=E(\mathbf{I},\mathsf{rand},\overrightarrow{h}') \\ \text{for some } (\mathbf{I},\mathsf{rand},\overrightarrow{h}')\in F}} (E \cap F) \right| \\
&= \frac{1}{4(\ell+1)} |F| \geq \frac{1}{8(\ell+1)} |G_{\mathbf{y}}| .
\end{aligned}$$

□

Remark 2. We point out that it is at this point that we need to require the adversary to make *exactly* $\ell + 1$ hash queries (and thus lose a $\binom{Q_h}{\ell+1}$ factor in advantage). The proof of Lemma 9 would not go through with $Q_h > \ell + 1$ hash queries, as hash vectors in this case may fork at arbitrary indices that do not have a corresponding signature. Therefore, not every tuple in an E-set would also be a partner of every other tuple in the same E-set (with the definition of partners adapted to this setting, i.e., two tuples can only be partners if they both have a signature at their forking index).

In the following, we want to avoid the case distinction of whether triangle corners come from the B-sets or the O-sets. We therefore define *good triangle corners*:

Definition 20. *Let* $\widehat{G_{\mathbf{y}}}$ *be the larger set of* $\widehat{B^{\times}_{T,\mathbf{y}}}$ *and* $\widehat{O^{\times}_{T,\mathbf{y}}}$*. Furthermore, for a tuple* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \widehat{G_{\mathbf{y}}}$*, let* $t(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ *be an index at which many relevant triangle tops exist, i.e.,*

$$t(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = \begin{cases} \mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) & \text{(if } \widehat{G_{\mathbf{y}}} = \widehat{B^{\times}_{T,\mathbf{y}}}\text{)} \\ \mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') & \text{(if } \widehat{G_{\mathbf{y}}} = \widehat{O^{\times}_{T,\mathbf{y}}}\text{)} \end{cases}$$

(where $\overrightarrow{h}'$ *is as in the definition of* $\widehat{O^{\times}_{T,\mathbf{y}}}$*). If multiple such* $\overrightarrow{h}'$ *(and thus multiple choices for* t*) exist, choose one that results in the smallest value of* t*. Define set* $\widehat{G_{\mathbf{z}}}$ *analogously, and for a tuple* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \widehat{G_{\mathbf{z}}}$*, define* $t(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ *analogously.*

[7] This is because $E(\mathbf{I}_1, \mathsf{rand}_1, \overrightarrow{h}_1) \cap E(\mathbf{I}_2, \mathsf{rand}_2, \overrightarrow{h}_2) \neq \emptyset$ implies that $\mathbf{I}_1 = \mathbf{I}_2$, $\mathsf{rand}_1 = \mathsf{rand}_2$, and $\overrightarrow{e}(\mathbf{I}_1, \mathsf{rand}_1, \overrightarrow{h}_1) = \overrightarrow{e}(\mathbf{I}_2, \mathsf{rand}_2, \overrightarrow{h}_2)$, which in turn implies that $E(\mathbf{I}_1, \mathsf{rand}_1, \overrightarrow{h}_1) = E(\mathbf{I}_2, \mathsf{rand}_2, \overrightarrow{h}_2)$.

It is easy to see that for a good opposing base corner, the number of triangle tops is the same as for the corresponding tuple from $G_{\mathbf{y}} \cup G_{\mathbf{z}}$. We state this as a lemma.

Lemma 10.

$$\Pr_{\substack{b \xleftarrow{\$} \{0,1\} \\ (\mathbf{I},\mathsf{rand},\overrightarrow{h}) \xleftarrow{\$} \mathcal{I}_b \times \mathcal{R} \times \mathbb{Z}_q{}^{\ell+1} \\ i \xleftarrow{\$} [\ell+1], \overrightarrow{h}' \xleftarrow{\$} \mathbb{Z}_q{}^{\ell+1}_{\mid \overrightarrow{h}_{[i-1]}}}} \left[\overrightarrow{h}' \in T^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \;\middle|\; \begin{array}{l} (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \widehat{G_{\mathbf{y}}} \\ t(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = i \end{array} \right] \geq \frac{\epsilon_{B^{\times}_T}}{16(\ell+1)} - \frac{2}{q}$$

An analogous statement holds for $\widehat{G_{\mathbf{z}}}$.

Proof. If $\widehat{G_{\mathbf{y}}} = \widehat{B^{\times}_{T,\mathbf{y}}}$, then the lower bound is implied by Corollary 4. If $\widehat{G_{\mathbf{y}}} = \widehat{O^{\times}_{T,\mathbf{y}}}$, setting the partner from the proof of Lemma 8 to the triangle corner from $\widehat{O^{\times}_{T,\mathbf{y}}}$ yields this lower bound. □

We furthermore note the following regarding the probability of sampling a tuple in $\widehat{G_{\mathbf{y}}}$ and $\widehat{G_{\mathbf{z}}}$:

Lemma 11.

$$\Pr_{\substack{b \xleftarrow{\$} \{0,1\} \\ (\mathbf{I},\mathsf{rand},\overrightarrow{h}) \xleftarrow{\$} \mathcal{I}_b \times \mathcal{R} \times \mathbb{Z}_q{}^{\ell+1} \\ i \xleftarrow{\$} [\ell+1], \overrightarrow{h}' \xleftarrow{\$} \mathbb{Z}_q{}^{\ell+1}_{\mid \overrightarrow{h}_{[i-1]}}}} \Pr\left[(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \widehat{G_{\mathbf{y}}} \right] \geq \frac{3}{128(\ell+1)} \epsilon_{B^{\times}_T}$$

The same holds for $\widehat{G_{\mathbf{z}}}$.

Proof. We prove the lemma for $\widehat{G_{\mathbf{y}}}$; the argument for $\widehat{G_{\mathbf{z}}}$ is analogous. By Lemma 9, either $\left|\widehat{B^{\times}_{T,\mathbf{y}}}\right| \geq \frac{1}{2}\left|G_{\mathbf{y}}\right|$ or $\left|\widehat{O^{\times}_{T,\mathbf{y}}}\right| \geq \frac{1}{8(\ell+1)}\left|G_{\mathbf{y}}\right|$, so $\left|\widehat{G_{\mathbf{y}}}\right| = \max\left\{\left|\widehat{B^{\times}_{T,\mathbf{y}}}\right|, \left|\widehat{O^{\times}_{T,\mathbf{y}}}\right|\right\} \geq \frac{1}{8(\ell+1)}\left|G_{\mathbf{y}}\right|$. By Lemma 8, $\left|G_{\mathbf{y}}\right| \geq \frac{3}{8}\left|B^{\times}_{T,\mathbf{y}}\right|$; by Lemma 6, $\left|B^{\times}_{T,\mathbf{y}}\right| = \frac{1}{2}\left|B^{\times}_{T}\right|$. Combining these three inequalities yields

$$\left|\widehat{G_{\mathbf{y}}}\right| \geq \frac{3}{128(\ell+1)}\left|B^{\times}_{T}\right|,$$

and the lemma follows. □

We will use the sets $\widehat{G_{\mathbf{y}}}$ and $\widehat{G_{\mathbf{z}}}$ for simplicity in the forking proof to avoid case distinctions over whether $\widehat{B^{\times}_{T,\mathbf{y}}}$ or $\widehat{O^{\times}_{T,\mathbf{y}}}$ (or $\widehat{B^{\times}_{T,\mathbf{z}}}$ or $\widehat{O^{\times}_{T,\mathbf{z}}}$) are larger.

4.5 Forking Proof for concurrent OMUF

In this section, we show that the Abe-Okamoto partially blind signature scheme AO is single-tag one-more unforgeable. We extend the proof to multiple tags in Section 4.6.

Theorem 1 (OMUF security for single-tag adversaries). *For all $\ell \in \mathbb{N}$, if there exists an adversary U that makes Q_h hash queries to random oracle H and $(t_{\mathsf{U}}, \epsilon_{\mathsf{U}}, \ell)$-breaks 1-$\mathsf{info}$-**OMUF**$_{\mathsf{AO}}$ with $\epsilon_{\mathsf{U}} \geq \frac{432\left(1-\frac{1}{(\ell+1)^2}\right)}{q} \cdot \binom{Q_h}{\ell+1}$, then there exists an algorithm B that $\left(t_{\mathsf{B}} = 2t_{\mathsf{U}} + \mathrm{O}({Q_h}^2), \epsilon_{\mathsf{B}} \approx \frac{3\epsilon_{\mathsf{U}}^2}{75423744 \cdot \binom{Q_h}{\ell+1}^2 \cdot (\ell+1)^3}\right)$-breaks* **DLOG**.

Proof. We use the wrapper A as described in Figure 3. We now construct a reduction B that plays the **DLOG** game as follows.

After B receives its discrete logarithm challenge $\mathbf{U}$, it samples a bit $b \xleftarrow{\$} \{0,1\}$. It then samples an instance $\mathbf{I}$ of type b where it sets $\mathbf{z} := \mathbf{U}$ if $b = 0$ and $\mathbf{y} := \mathbf{U}$ if $b = 1$, and all other items uniformly at random from $\mathbb{Z}_q$. Furthermore, B samples a random tape rand for A and a random hash vector $\overrightarrow{h}$. After that, B runs A on $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$. If A returns a set of $\ell+1$ valid message-signature pairs, B chooses a random index $i \xleftarrow{\$} [\ell+1]$. B then re-samples the vector $\overrightarrow{h}' \xleftarrow{\$} \mathbb{Z}_q^{\ell+1}{}_{|\overrightarrow{h}[i-1]}$ and runs A on $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$. If A outputs a second set of $\ell+1$ valid message-signature pairs, B identifies the signature matching the hash value h_i and h_i' respectively in both pair (it aborts if there exists no such signature for h_i'). Denote the corresponding signature components to the ith hash query by $\rho_i, \rho_i', \omega_i, \omega_i', \sigma_i, \sigma_i', \delta_i, \delta_i'$ (see Figure 6 on page 50).

If $\omega_i \neq \omega_i'$ and $b = 1$, B computes

$$x := (\omega_i - \omega_i')^{-1} \cdot (\rho_i' - \rho_i)$$

as its output; if $\delta_i \neq \delta_i'$ and $b = 0$, B computes

$$w := (\delta_i - \delta_i')^{-1} \cdot (\sigma_i' - \sigma_i)$$

as its output. Otherwise B aborts. (If A fails to return a set of $\ell+1$ valid message-signature pairs either time, B also aborts.)

B runs A twice, and performs $\Theta(\ell)$ additional computation (in particular, B verifies up to $2(\ell+1)$ signatures). Plugging in $t_{\mathsf{A}} = t_{\mathsf{U}} + \mathrm{O}({Q_h}^2)$, we get that

$$t_{\mathsf{B}} = 2t_{\mathsf{U}} + \mathrm{O}({Q_h}^2).$$

We now analyze the advantage of reduction B. Let ϵ_{U} be the advantage of U in the OMUF game, and ϵ be the probability that A outputs $\ell+1$ valid message-signature pairs. By Lemma 1 and subsequent analysis in Section 4.1,

$$\epsilon \geq \frac{\epsilon_{\mathsf{U}}}{\binom{Q_h}{\ell+1}}.$$

We can see that B internally runs the witness extracting algorithm $\mathsf{E}_{\mathbf{y}}$ or $\mathsf{E}_{\mathbf{z}}$ in Definition 13. Therefore, by Lemma 4, we have that

$$\begin{aligned}
\mathrm{adv}_{\mathsf{B}}^{\mathbf{DLOG}} = & \Pr_{\substack{b \xleftarrow{\$} \{0,1\} \\ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \xleftarrow{\$} \mathcal{I}_b \times \mathcal{R} \times \mathbb{Z}_q^{[\ell+1]} \\ i \xleftarrow{\$} [\ell+1], \overrightarrow{h}' \xleftarrow{\$} \mathbb{Z}_q^{[\ell+1]}_{|\overrightarrow{h}_{[i-1]}}}} \left[\begin{array}{c} (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \mathsf{Succ} \wedge (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in \mathsf{Succ} \\ (b = 0 \wedge \delta_i \neq \delta_i') \vee (b = 1 \wedge \omega_i \neq \omega_i') \end{array} \right] \\
\geq & \Pr \left[\begin{array}{c} (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \widehat{G_{\mathbf{y}}} \cup \widehat{G_{\mathbf{z}}} \\ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in T_{T,i}^{\times}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \\ t(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = i \\ \big((b = 0 \wedge \delta_i \neq \delta_i') \\ \vee (b = 1 \wedge \omega_i \neq \omega_i')\big) \end{array} \right] \geq \Pr \left[\begin{array}{c} (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \widehat{G_{\mathbf{y}}} \cup \widehat{G_{\mathbf{z}}} \\ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in T_{T,i}^{\times}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \\ t(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = i \\ \big((b = 0 \wedge \times = \mathbf{z} \wedge \delta_i \neq \delta_i') \\ \vee (b = 1 \wedge \times = \mathbf{y} \wedge \omega_i \neq \omega_i')\big) \end{array} \right] \\
= & \Pr \left[\begin{array}{c} (b = 0 \wedge \times = \mathbf{z} \wedge \delta_i \neq \delta_i') \\ \vee (b = 1 \wedge \times = \mathbf{y} \wedge \omega_i \neq \omega_i') \end{array} \middle| \begin{array}{c} (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \widehat{G_{\mathbf{y}}} \cup \widehat{G_{\mathbf{z}}} \\ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in T_{T,i}^{\times}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \\ t(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = i \end{array} \right] \\
& \cdot \Pr \left[\begin{array}{c} (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \widehat{G_{\mathbf{y}}} \cup \widehat{G_{\mathbf{z}}} \\ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in T_{T,i}^{\times}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \\ t(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = i \end{array} \right]
\end{aligned}$$

We now lower-bound the first term, where we abbreviate the event $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \widehat{G_{\mathbf{y}}} \cup \widehat{G_{\mathbf{z}}} \wedge (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in T_{T,i}^{\times}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \wedge t(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = i$ as $\mathbf{E}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$:

$$\begin{aligned}
& \Pr_{\substack{b \xleftarrow{\$} \{0,1\} \\ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \xleftarrow{\$} \mathcal{I}_b \times \mathcal{R} \times \mathbb{Z}_q^{[\ell+1]} \\ i \xleftarrow{\$} [\ell+1], \overrightarrow{h}' \xleftarrow{\$} \mathbb{Z}_q^{[\ell+1]}_{|\overrightarrow{h}_{[i-1]}}}} \left[\begin{array}{c} (b = 0 \wedge \times = \mathbf{z} \wedge \delta_i \neq \delta_i') \\ \vee (b = 1 \wedge \times = \mathbf{y} \wedge \omega_i \neq \omega_i') \end{array} \middle| \mathbf{E}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \right] \\
= & \Pr \left[\begin{array}{c} b = 1 \wedge \times = \mathbf{y} \\ \omega_i \neq \omega_i' \end{array} \middle| \mathbf{E}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \right] + \Pr \left[\begin{array}{c} b = 0 \wedge \times = \mathbf{z} \\ \delta_i \neq \delta_i' \end{array} \middle| \mathbf{E}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \right] \\
= & \Pr[b = 1] \cdot \Pr \left[\times = \mathbf{y} \wedge \omega_i \neq \omega_i' \middle| \ b = 1 \wedge \mathbf{E}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \right] \\
& + \Pr[b = 0] \cdot \Pr \left[\times = \mathbf{z} \wedge \delta_i \neq \delta_i' \middle| \ b = 0 \wedge \mathbf{E}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \right] \\
= & \frac{1}{2} \Big(\Pr \left[\times = \mathbf{y} \wedge \omega_i \neq \omega_i' \middle| \ b = 1 \wedge \mathbf{E}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \right] \\
& + \Pr \left[\times = \mathbf{z} \wedge \delta_i \neq \delta_i' \middle| \ b = 0 \wedge \mathbf{E}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \right] \Big) \\
= & \frac{1}{2} \Big(\Pr[\times = \mathbf{y}] \cdot \Pr \left[\omega_i \neq \omega_i' \middle| \ b = 1 \wedge \times = \mathbf{y} \wedge \mathbf{E}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \right] \\
& + \Pr[\times = \mathbf{z}] \cdot \Pr \left[\delta_i \neq \delta_i' \middle| \ b = 0 \wedge \times = \mathbf{z} \wedge \mathbf{E}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \right] \Big) \\
= & \frac{1}{2} \left(\Pr[\times = \mathbf{y}] \cdot \Pr \left[(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in A_{T,i}^{\times}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \middle| \begin{array}{c} b = 1 \wedge \times = \mathbf{y} \\ \wedge \mathbf{E}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \end{array} \right] \right.
\end{aligned}$$

$$
\begin{aligned}
&+\Pr[\times=\mathbf{z}]\cdot\Pr\left[(\mathbf{I},\mathsf{rand},\overrightarrow{h}')\in A^{\times}_{T,i}(\mathbf{I},\mathsf{rand},\overrightarrow{h})\,\middle|\,\begin{matrix}b=0\wedge\times=\mathbf{z}\\ \wedge\mathbf{E}(\mathbf{I},\mathsf{rand},\overrightarrow{h})\end{matrix}\right]\Bigg)\\
\geq&\frac{1}{2}\left(\Pr[\times=\mathbf{y}]\cdot\left(\frac{1}{2}-\frac{1}{q}\right)+\Pr[\times=\mathbf{z}]\cdot\left(\frac{1}{2}-\frac{1}{q}\right)\right)\\
=&\left(\frac{1}{4}-\frac{1}{2q}\right)\cdot(\Pr[\times=\mathbf{y}]+\Pr[\times=\mathbf{z}])=\frac{1}{4}-\frac{1}{2q},
\end{aligned}
$$

where the inequality is due to the following: since $(\mathbf{I},\mathsf{rand},\overrightarrow{h})\in\widehat{G_{\mathbf{y}}}\cup\widehat{G_{\mathbf{z}}}$, we have that

$$\left|A^{\times}_{T,i}(\mathbf{I},\mathsf{rand},\overrightarrow{h})\right|\geq\frac{1}{2}\left|T^{\times}_{T,i}(\mathbf{I},\mathsf{rand},\overrightarrow{h})\right|-q^{\ell-i+1}.$$

Since we conditioned on $(\mathbf{I},\mathsf{rand},\overrightarrow{h}')\in T^{\times}_{T,i}(\mathbf{I},\mathsf{rand},\overrightarrow{h})$, the probability in question is

$$\frac{\left|A^{\times}_{T,i}(\mathbf{I},\mathsf{rand},\overrightarrow{h})\right|}{\left|T^{\times}_{T,i}(\mathbf{I},\mathsf{rand},\overrightarrow{h})\right|}\geq\frac{1}{2}-\frac{q^{\ell-i+1}}{\left|T^{\times}_{T,i}(\mathbf{I},\mathsf{rand},\overrightarrow{h})\right|}\geq\frac{1}{2}-\frac{q^{\ell-i+1}}{q^{\ell-i+2}}=\frac{1}{2}-\frac{1}{q}.$$

In the following we denote by $\widehat{G}_b$ the set $\widehat{G_{\mathbf{y}}}$ if $b=1$ and the set $\widehat{G_{\mathbf{z}}}$ if $b=0$. Plugging the result back into the previous lower bound of B's advantage yields

$$
\begin{aligned}
\mathrm{adv}^{\mathbf{DLOG}}_{\mathsf{B}}&\geq\left(\frac{1}{4}-\frac{1}{2q}\right)\cdot\Pr\left[\begin{matrix}(\mathbf{I},\mathsf{rand},\overrightarrow{h})\in\widehat{G}_b\\ (\mathbf{I},\mathsf{rand},\overrightarrow{h}')\in T^{\times}_{T,i}(\mathbf{I},\mathsf{rand},\overrightarrow{h})\\ t(\mathbf{I},\mathsf{rand},\overrightarrow{h})=i\end{matrix}\right]\\
&=\left(\frac{1}{4}-\frac{1}{2q}\right)\cdot\Pr\left[(\mathbf{I},\mathsf{rand},\overrightarrow{h}')\in T^{\times}_{T,i}(\mathbf{I},\mathsf{rand},\overrightarrow{h})\,\middle|\,\begin{matrix}(\mathbf{I},\mathsf{rand},\overrightarrow{h})\in\widehat{G}_b\\ t(\mathbf{I},\mathsf{rand},\overrightarrow{h})=i\end{matrix}\right]\\
&\quad\cdot\Pr\left[(\mathbf{I},\mathsf{rand},\overrightarrow{h})\in\widehat{G}_b\right]\cdot\Pr\left[t(\mathbf{I},\mathsf{rand},\overrightarrow{h})=i\middle|(\mathbf{I},\mathsf{rand},\overrightarrow{h})\in\widehat{G}_b\right]\\
&\geq\left(\frac{1}{4}-\frac{1}{2q}\right)\cdot\left(\frac{\epsilon_{B^{\times}_T}}{16(\ell+1)}-\frac{2}{q}\right)\cdot\frac{3\epsilon_{B^{\times}_T}}{128(\ell+1)}\cdot\frac{1}{\ell+1}
\end{aligned}
$$

(where the last inequality is due to Lemma 10 and Lemma 11). Plugging in $\epsilon_{B^{\times}_T}\geq\frac{\epsilon}{96}$ for $\epsilon\geq\frac{432\left(1-\frac{1}{(\ell+1)^2}\right)}{q}$ (see Lemma 7) and $\epsilon=\frac{\epsilon_{\mathsf{U}}}{\binom{Q_h}{\ell+1}}$ yields the theorem statement. □

4.6 Extension to multiple tags

Theorem 2. *Let* U *be an adversary against* ℓ-$\mathbf{OMUF}_{\mathsf{AO}}$ *that runs in time* t_{U}*, closes at most* ℓ_{info} *signing sessions per tag* info*, closes at most* ℓ *signing sessions in total, and queries at most* Q_{info} *tags* info *to oracle* H^**. Let* $\mathrm{adv}^{\mathbf{OMUF}_{\mathsf{AO}}}_{Q_{\mathsf{info}},\ell_{\mathsf{info}},\mathsf{U}}$ *be* U*'s*

advantage. Then there exists a reduction B *against* 1-info-$\mathbf{OMUF}_{\mathsf{AO}}$ *that runs in time* $t_{\mathsf{B}} \approx t_{\mathsf{U}}$ *and makes at most* ℓ_{info} *signing queries and has advantage*

$$\mathrm{adv}^{\ell_{\mathsf{info}}\text{-1-}\mathsf{info}\text{-}\mathbf{OMUF}_{\mathsf{AO}}}_{\mathsf{B}} \geq \frac{\mathrm{adv}^{\ell-\mathbf{OMUF}_{\mathsf{AO}}}_{Q_{\mathsf{info}},\ell_{\mathsf{info}},\mathsf{A}} - \frac{\ell}{q}}{Q_{\mathsf{info}}}.$$

The proof of this theorem mostly follows that in [4]. We provide it in Appendix D for completeness.

References

1. Abe, M. *A Secure Three-Move Blind Signature Scheme for Polynomially Many Signatures* in *Advances in Cryptology – EUROCRYPT 2001* (Springer, Heidelberg, Germany, 2001), 136–151.
2. Abe, M. & Fujisaki, E. *How to Date Blind Signatures* in *Advances in Cryptology – ASIACRYPT'96* (Springer, Heidelberg, Germany, 1996), 244–251.
3. Abe, M. & Ohkubo, M. *A Framework for Universally Composable Non-committing Blind Signatures* in *Advances in Cryptology – ASIACRYPT 2009* (Springer, Heidelberg, Germany, 2009), 435–450.
4. Abe, M. & Okamoto, T. *Provably Secure Partially Blind Signatures* in *Advances in Cryptology – CRYPTO 2000* (Springer, Heidelberg, Germany, 2000), 271–286.
5. Alkadri, N. A., Harasser, P. & Janson, C. *BlindOR: An Efficient Lattice-Based Blind Signature Scheme from OR-Proofs* in *CANS 21International Conference on Cryptology and Network Security* (Springer, Heidelberg, Germany, 2021), 95–115.
6. Alkeilani Alkadri, N., Harasser, P. & Janson, C. *BlindOR: An Efficient Lattice-Based Blind Signature Scheme from OR-Proofs* Cryptology ePrint Archive, Report 2021/1385. 2021.
7. Baldimtsi, F. & Lysyanskaya, A. *Anonymous credentials light* in *ACM CCS 2013: 20th Conference on Computer and Communications Security* (ACM Press, 2013), 1087–1098.
8. Benhamouda, F., Lepoint, T., Loss, J., Orrù, M. & Raykova, M. *On the (in)security of ROS* in *Advances in Cryptology – EUROCRYPT 2021, Part I* (Springer, Heidelberg, Germany, 2021), 33–53.
9. Camenisch, J., Piveteau, J.-M. & Stadler, M. *Blind Signatures Based on the Discrete Logarithm Problem (Rump Session)* in *Advances in Cryptology – EUROCRYPT'94* (Springer, Heidelberg, Germany, 1995), 428–432.
10. Camenisch, J., Neven, G. & shelat, a. *Simulatable Adaptive Oblivious Transfer* in *Advances in Cryptology – EUROCRYPT 2007* (Springer, Heidelberg, Germany, 2007), 573–590.
11. Cao, T., Lin, D. & Xue, R. A randomized RSA-based partially blind signature scheme for electronic cash. *Computers & Security,* 1 (2005).
12. Chairattana-Apirom, R., Hanzlik, L., Loss, J., Lysyanskaya, A. & Wagner, B. *PI-Cut-Choo and Friends: Compact Blind Signatures via Parallel Instance Cut-and-Choose and More* in *CRYPTO, to appear* (2022).
13. Chaum, D. *Blind Signatures for Untraceable Payments* in *Advances in Cryptology – CRYPTO'82* (Plenum Press, New York, USA, 1982), 199–203.
14. Chaum, D. *Elections with Unconditionally-Secret Ballots and Disruption Equivalent to Breaking RSA* in *Advances in Cryptology – EUROCRYPT'88* (Springer, Heidelberg, Germany, 1988), 177–182.

15. Chaum, D., Fiat, A. & Naor, M. *Untraceable Electronic Cash* in *Advances in Cryptology – CRYPTO'88* (Springer, Heidelberg, Germany, 1990), 319–327.
16. Chow, S. S. M., Hui, L. C. K., Yiu, S.-M. & Chow, K. P. *Two Improved Partially Blind Signature Schemes from Bilinear Pairings* in *ACISP 05: 10th Australasian Conference on Information Security and Privacy* (Springer, Heidelberg, Germany, 2005), 316–328.
17. Cramer, R., Damgård, I. & Schoenmakers, B. *Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols* in *Advances in Cryptology – CRYPTO'94* (Springer, Heidelberg, Germany, 1994), 174–187.
18. Fischlin, M. *Round-Optimal Composable Blind Signatures in the Common Reference String Model* in *Advances in Cryptology – CRYPTO 2006* (Springer, Heidelberg, Germany, 2006), 60–77.
19. Fischlin, M. & Schröder, D. *Security of Blind Signatures under Aborts* in *PKC 2009: 12th International Conference on Theory and Practice of Public Key Cryptography* (Springer, Heidelberg, Germany, 2009), 297–316.
20. Fujioka, A., Okamoto, T. & Ohta, K. *A Practical Secret Voting Scheme for Large Scale Elections* in *Advances in Cryptology – AUSCRYPT'92* (Springer, Heidelberg, Germany, 1993), 244–251.
21. Hanatani, Y., Komano, Y., Ohta, K. & Kunihiro, N. *Provably Secure Electronic Cash Based on Blind Multisignature Schemes* in *FC 2006: 10th International Conference on Financial Cryptography and Data Security* (Springer, Heidelberg, Germany, 2006), 236–250.
22. Hauck, E., Kiltz, E. & Loss, J. *A Modular Treatment of Blind Signatures from Identification Schemes* in *Advances in Cryptology – EUROCRYPT 2019, Part III* (Springer, Heidelberg, Germany, 2019), 345–375.
23. Hauck, E., Kiltz, E., Loss, J. & Nguyen, N. K. *Lattice-Based Blind Signatures, Revisited* in *Advances in Cryptology – CRYPTO 2020, Part II* (Springer, Heidelberg, Germany, 2020), 500–529.
24. Hazay, C., Katz, J., Koo, C.-Y. & Lindell, Y. *Concurrently-Secure Blind Signatures Without Random Oracles or Setup Assumptions* in *TCC 2007: 4th Theory of Cryptography Conference* (Springer, Heidelberg, Germany, 2007), 323–341.
25. Juels, A., Luby, M. & Ostrovsky, R. *Security of Blind Digital Signatures (Extended Abstract)* in *Advances in Cryptology – CRYPTO'97* (Springer, Heidelberg, Germany, 1997), 150–164.
26. Kastner, J., Loss, J. & Xu, J. *On Pairing-Free Blind Signature Schemes in the Algebraic Group Model* in *PKC 2022: 25th International Conference on Theory and Practice of Public Key Cryptography* (Springer, Heidelberg, Germany, 2022), 468–497.
27. Katsumata, S. & del Pino, R. *A New Framework For More Efficient Round-Optimal Lattice-Based (Partially) Blind Signature via Trapdoor Sampling* in *CRYPTO, to appear* (2022).
28. Katsumata, S., Nishimaki, R., Yamada, S. & Yamakawa, T. *Round-Optimal Blind Signatures in the Plain Model from Classical and Quantum Standard Assumptions* in *Advances in Cryptology – EUROCRYPT 2021, Part I* (Springer, Heidelberg, Germany, 2021), 404–434.
29. Katz, J., Loss, J. & Rosenberg, M. *Boosting the Security of Blind Signature Schemes* in *Advances in Cryptology – ASIACRYPT 2021, Part IV* (Springer, Heidelberg, Germany, 2021), 468–492.

30. Maitland, G. & Boyd, C. *A Provably Secure Restrictive Partially Blind Signature Scheme* in *PKC 2002: 5th International Workshop on Theory and Practice in Public Key Cryptography* (Springer, Heidelberg, Germany, 2002), 99–114.
31. Martinet, G., Poupard, G. & Sola, P. *Cryptanalysis of a Partially Blind Signature Scheme or How to Make $100 Bills with $1 and $2 Ones* in *FC 2006: 10th International Conference on Financial Cryptography and Data Security* (Springer, Heidelberg, Germany, 2006), 171–176.
32. Ohkubo, M. & Abe, M. *Security of Some Three-move Blind Signature Schemes Reconsidered* The 2003 Symposium on Cryptography and Information Security. 2003.
33. Okamoto, T. *Efficient Blind and Partially Blind Signatures Without Random Oracles* in *TCC 2006: 3rd Theory of Cryptography Conference* (Springer, Heidelberg, Germany, 2006), 80–99.
34. Papachristoudis, D., Hristu-Varsakelis, D., Baldimtsi, F. & Stephanides, G. *Leakage-Resilient Lattice-Based Partially Blind Signatures* 2019.
35. Pointcheval, D. & Stern, J. *Provably Secure Blind Signature Schemes* in *Advances in Cryptology – ASIACRYPT'96* (Springer, Heidelberg, Germany, 1996), 252–265.
36. Pointcheval, D. & Stern, J. Security Arguments for Digital Signatures and Blind Signatures. *Journal of Cryptology,* 361–396 (2000).
37. Rückert, M. *Lattice-Based Blind Signatures* in *Advances in Cryptology – ASIACRYPT 2010* (Springer, Heidelberg, Germany, 2010), 413–430.
38. Schnorr, C.-P. *Security of Blind Discrete Log Signatures against Interactive Attacks* in *ICICS 01: 3rd International Conference on Information and Communication Security* (Springer, Heidelberg, Germany, 2001), 1–12.
39. Schröder, D. & Unruh, D. *Security of Blind Signatures Revisited* in *PKC 2012: 15th International Conference on Theory and Practice of Public Key Cryptography* (Springer, Heidelberg, Germany, 2012), 662–679.
40. Tessaro, S. & Zhu, C. *Short Pairing-Free Blind Signatures with Exponential Security* Cryptology ePrint Archive, Report 2022/047. 2022.
41. Tyagi, N. *et al. A Fast and Simple Partially Oblivious PRF, with Applications* Cryptology ePrint Archive, Report 2021/864. 2021.
42. Yi, X. & Lam, K.-Y. *A New Blind ECDSA Scheme for Bitcoin Transaction Anonymity* in *ASIACCS 19: 14th ACM Symposium on Information, Computer and Communications Security* (ACM Press, 2019), 613–620.
43. Zhang, F., Safavi-Naini, R. & Susilo, W. *Efficient Verifiably Encrypted Signature and Partially Blind Signature from Bilinear Pairings* in *Progress in Cryptology - INDOCRYPT 2003: 4th International Conference in Cryptology in India* (Springer, Heidelberg, Germany, 2003), 191–204.

Supplementary Material

A Definition of Partial Blindness

Definition 21 (Partial Blindness). *For a three-move partially blind signature scheme* PBS*, we define the partial blindness game* $\mathbf{PBLIND}_{\mathsf{PBS}}$ *with an adversary* S *(in the role of the signer) as follows:*

Setup. *The game samples* $b \xleftarrow{\$} \{0,1\}$*. It then runs* S *on input* pp*.*

Online Phase. *When* S *outputs messages* $\widetilde{m}_0$ *and* $\widetilde{m}_1$*, a tag* info*, and a public key* pk*, the game checks if* pk *is a valid public key if so, it assigns* $m_0 := \widetilde{m}_b$*,* $m_1 := \widetilde{m}_{1-b}$*. If* pk *is not a valid public key, the game aborts and outputs* 0*.* S *is given access to oracles* user_1 *and* user_2*, which behave as follows.*

- **Oracle** user_1**:** *On input a bit* b' *and a* Sign_1 *response* R*, if the session* b' *is not yet open, the oracle marks session* b' *as open and generates a state and a challenge as* $(\mathsf{st}_{b'}, e) \xleftarrow{\$} \mathsf{PBS.User}_1(\mathsf{pk}, m_{b'}, R, \mathsf{info})$*. It returns* e *to* S*. Otherwise, it returns* $\perp$*.*
- **Oracle** user_2**:** *On input of a* Sign_2 *response* S *and a bit* b'*, if the session* b' *is open, the oracle computes the signature* $\mathsf{sig}_{b'} := \mathsf{PBS.User}_2(\mathsf{pk}, \mathsf{st}_{b'}, R)$*. It marks session* b' *as closed and saves* $\mathsf{sig}_{b'}$*. If both sessions are closed and produced signatures, the oracle outputs the two signatures* $\mathsf{sig}_0, \mathsf{sig}_1$ *to* S*.*

Output Determination. *If both sessions are closed and produced signatures, the game outputs* 1 *iff* S *outputs a bit* b^* *s.t.* $b^* = b$*. Otherwise, it outputs* 0*.*

We define the advantage of S *as*

$$\mathrm{adv}^{\mathbf{PBLIND}_{\mathsf{PBS}}}_{\mathsf{S}} = \left| \Pr\left[\mathbf{PBLIND}^{\mathsf{S}}_{\mathsf{PBS}} = 1\right] - \frac{1}{2} \right|$$

where the probability goes over the randomness of the game as well as the randomness of the adversary S*. We say the scheme* PBS *is* (t, ϵ)-$\mathsf{partially\ blind}$ *if for any adversary* S *running in time at most* t*,*

$$\mathrm{adv}^{\mathbf{PBLIND}_{\mathsf{PBS}}}_{\mathsf{S}} \leq \epsilon.$$

B Important Lemmata

In this section, we review the classical splitting lemma and introduce what we call the bucket lemma, which will help facilitate our proofs in Appendix C.

Lemma 12 (Splitting Lemma [36]). *Let* $A \subset X \times Y$ *such that*

$$\Pr_{(x,y) \xleftarrow{\$} X \times Y}[(x,y) \in A] \geq \epsilon.$$

For any $\alpha \in [0, \epsilon)$ define

$$B = \left\{ (x, y) \in X \times Y \middle| \Pr_{y' \xleftarrow{\$} Y}[(x, y') \in A] \geq \epsilon - \alpha \right\}.$$

(B is sometimes called the heavy row *of A.) Then the following statements hold:*

1. $\Pr_{(x,y) \xleftarrow{\$} X \times Y}[(x, y) \in B] \geq \alpha$
2. $\forall (x, y) \in B\colon \Pr_{y' \xleftarrow{\$} Y}[(x, y') \in A] \geq \epsilon - \alpha$
3. $\Pr_{(x,y) \xleftarrow{\$} X \times Y}[(x, y) \in B | (x, y) \in A] \geq \frac{\alpha}{\epsilon}$

Lemma 13 (Bucket Lemma). *Let X be a finite set, $b \in \mathbb{Z}^+$, and let $B_1, \ldots, B_b \subset X$ s.t. $\bigcup_{i=1}^{b} B_i = X$. Then for all $\alpha \in (0, 1)$ there exists a set $G_\alpha \subset X$ such that*

1. *$|G_\alpha| > (1 - \alpha) \cdot |X|$.*
2. *For all $x \in G_\alpha$, there exists $i \in [b]$ s.t. $x \in B_i$ and $|B_i| \geq \alpha \cdot \frac{|X|}{b}$.*

Proof. Fix α. Let $F_\alpha \subset X$ be the set of elements that do not belong to *any* B_i ($i \in [b]$) with $|B_i| \geq \alpha \cdot \frac{|X|}{b}$. It therefore holds that $F_\alpha \subset \bigcup_{B_i : |B_i| < \alpha \cdot \frac{|X|}{b}} B_i$. We now compute an upper bound for the size of F_α as

$$\begin{aligned} |F_\alpha| &\leq \left| \bigcup_{i : |B_i| < \alpha \cdot \frac{|X|}{b}} B_i \right| \leq \sum_{i : |B_i| < \alpha \cdot \frac{|X|}{b}} |B_i| \\ &< \sum_{i : |B_i| < \alpha \cdot \frac{|X|}{b}} \alpha \cdot \frac{|X|}{b} \leq b \cdot \left(\alpha \cdot \frac{|X|}{b} \right) = \alpha \cdot |X| \end{aligned}$$

Setting $G_\alpha = X \setminus F_\alpha$ yields the statement. □

C Deferred Proofs and Definitions from the Main body

C.1 Additional Definitions

We define the *heavy row* of the set of successful tuples Succ as

$$HR(\mathsf{Succ}) \coloneqq \left\{ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \mathsf{Succ} \middle| |\mathsf{Succ}_{\mathbf{I}, \mathsf{rand}}| \geq \frac{\epsilon}{2} \cdot q^{\ell+1} \right\}$$

By Lemma 12, $|HR(\mathsf{Succ})| \geq \frac{1}{2} |\mathsf{Succ}|$.

In the following we define a subset $P \subset HR(\mathsf{Succ})$ of "partner tuples" which have a partner at some index. We also define a "good" subset P_G of P and its "bad" complement P_B. Intuitively, P_G consists of those tuples $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ in P which have many partnering tuples at at least one index, i.e., for which $B_{\mathrm{Br}_{\max}}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ is large (relative to the number of all tuples $(\mathbf{I}, \mathsf{rand}, \cdot)$ in P).

Definition 22 (Partner Tuples). *We call* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in HR(\mathsf{Succ})$ *a* partner tuple *if* $\overrightarrow{h}$ *has a partner with respect to* $\mathbf{I}, \mathsf{rand}$ *(at any index* $i \in [\ell+1]$*), i.e., if* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in P$ *where*

$$P = \left\{ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in HR(\mathsf{Succ}) \middle| (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in P_{\mathbf{I},\mathsf{rand}} \right\}.$$

We further define the set of good partner tuples *as*

$$P_G = \left\{ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in P \middle| \left| B_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})} \right| \geq \frac{1}{(\ell+1)^3} |P \cap P_{\mathbf{I},\mathsf{rand}}| \right\}.$$

The set of bad partner tuples *is defined as* $P_B = P \setminus P_G$.

Finally, we introduce the notion of *S-suffixes* (at some index j). For a successful tuple $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in S \subset \mathsf{Succ}_{\mathbf{I},\mathsf{rand}}$ we consider all hash vectors that share a j-prefix (i.e., up to index $j-1$) with $\overrightarrow{h}$ and also lie in S. We define the set $\Gamma_{j,S}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ of its S-suffixes at index j as the set of all the j-th entries h^* of such vectors.

Definition 23 (S-Suffixes). *Fix* $\mathbf{I}, \mathsf{rand}$ *and some* $S \subset \mathsf{Succ}_{\mathbf{I},\mathsf{rand}}$*. For a hash vector* $\overrightarrow{h}$ *with* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in S$ *and all* $j \in [\ell+1]$*, we define its set of* S-suffixes *at index* j *as*

$$\Gamma_{j,S}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) := \left\{ h^* \middle| \exists\, \overrightarrow{h}' \colon \begin{array}{c} (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in S \\ \overrightarrow{h}'_{[j-1]} = \overrightarrow{h}_{[j-1]} \\ h'_j = h^* \end{array} \right\}$$

C.2 Counting Partners and Triangles

Having defined our basic objects of interest, we now move to lower bounding their numbers. We start by considering the sizes of sets P and P_G.

The following lemma asserts that if the set $\mathsf{Succ}_{\mathbf{I},\mathsf{rand}}$ is sufficiently large for some fixed $\mathbf{I}, \mathsf{rand}$ (i.e., many different vectors $\overrightarrow{h}$ lead to success together with $\mathbf{I}, \mathsf{rand}$), then the set $P_{\mathbf{I},\mathsf{rand}}$ of tuples $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ that have some partner $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$, is large.

Lemma 14. *For* $\mathbf{I}, \mathsf{rand}$ *such that* $|\mathsf{Succ}_{\mathbf{I},\mathsf{rand}}| > q^\ell$*, there exist at least* $|\mathsf{Succ}_{\mathbf{I},\mathsf{rand}}| - q^\ell + 1$ *hash vectors* $\overrightarrow{h}$ *such that* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in P_{\mathbf{I},\mathsf{rand}}$.

Proof. There are at most q^ℓ possible query transcripts. Thus, by pigeon hole-principle, for $|\mathsf{Succ}_{\mathbf{I},\mathsf{rand}}| > q^\ell$ there can be at most $q^\ell - 1$ hash vectors that do not have a partner. This yields the statement. □

We have proven above that the number of partner tuples with respect to a sufficiently good pair $\mathbf{I}, \mathsf{rand}$ (i.e., one for which many $\overrightarrow{h}$ lead to success) is large. The following simple corollaries combine the above with the properties of $HR(\mathsf{Succ})$ to ensure that the set P of heavy-row partner tuples (i.e., over all pairs $\mathbf{I}, \mathsf{rand} \in HR(\mathsf{Succ})$) is also large.

Corollary 5. *For* $\mathbf{I}, \mathsf{rand}$ *such that* $\exists \overrightarrow{h}: (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in HR(\mathsf{Succ})$ *and* $\epsilon \geq \frac{4}{q}$*, it holds that* $|P_{\mathbf{I},\mathsf{rand}}| \geq \frac{1}{2} |\mathsf{Succ}_{\mathbf{I},\mathsf{rand}}|$.

Corollary 6. *For* ϵ *as in Corollary 5,* $|P| \geq \frac{1}{4} |\mathsf{Succ}|$.

Proof. By Corollary 5, $|P_{\mathbf{I},\mathsf{rand}}| \geq \frac{1}{2} |\mathsf{Succ}_{\mathbf{I},\mathsf{rand}}|$ for $\mathbf{I}, \mathsf{rand}$ with $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in HR(\mathsf{Succ})$ for some $\overrightarrow{h}$. Summing over all such $(\mathbf{I}, \mathsf{rand})$ pairs yields that $|P| \geq \frac{1}{2} |HR(\mathsf{Succ})|$. As $|HR(\mathsf{Succ})| \geq \frac{1}{2} |\mathsf{Succ}|$, the statement follows. □

Next, we also show that the subset P_G of good tuples is large within P.

Lemma 15 (Many partner tuples are good).

$$|P_G| \geq \left(1 - \frac{1}{(\ell+1)^2}\right) |P| .$$

Proof. Fix $\mathbf{I}, \mathsf{rand}$ such that $P_{\mathbf{I},\mathsf{rand}} \cap P \neq \emptyset$. For all $i \in [\ell+1]$, let $B_i = B_i(\mathbf{I}, \mathsf{rand})$ (as in Definition 11) and $\alpha = \frac{1}{(\ell+1)^2}$. We note here that $P \cap P_{\mathbf{I},\mathsf{rand}} = P_{\mathbf{I},\mathsf{rand}}$ for $\mathbf{I}, \mathsf{rand}$ as above and thus the B_i are a partition of $P \cap P_{\mathbf{I},\mathsf{rand}}$. By Lemma 13, there exists a subset $G(\mathbf{I}, \mathsf{rand})$ of size at least $\left(1 - \frac{1}{(\ell+1)^2}\right) |P \cap P_{\mathbf{I},\mathsf{rand}}|$, such that all tuples $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in G(\mathbf{I}, \mathsf{rand})$ lie in a set B_i of size at least $|B_i| \geq \frac{1}{(\ell+1)^3} |P \cap P_{\mathbf{I},\mathsf{rand}}|$, where by definition $i = \mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$. By definition of P_G, $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in P_G$. Since this holds for any $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in G(\mathbf{I}, \mathsf{rand})$, we have that $G(\mathbf{I}, \mathsf{rand}) \subset P_G$. Hence,

$$\bigcup_{\substack{\mathbf{I},\mathsf{rand}: \\ P \cap P_{\mathbf{I},\mathsf{rand}} \neq \emptyset}} G(\mathbf{I}, \mathsf{rand}) \subset P_G,$$

and since the sets $G(\mathbf{I}, \mathsf{rand}) \subset P_{\mathbf{I},\mathsf{rand}}$ are disjoint for distinct $\mathbf{I}, \mathsf{rand}$,

$$\begin{aligned} |P_G| &\geq \sum_{\substack{\mathbf{I},\mathsf{rand}: \\ P \cap P_{\mathbf{I},\mathsf{rand}} \neq \emptyset}} |G(\mathbf{I}, \mathsf{rand})| \\ &\geq \sum_{\substack{\mathbf{I},\mathsf{rand}: \\ P \cap P_{\mathbf{I},\mathsf{rand}} \neq \emptyset}} \left(1 - \frac{1}{(\ell+1)^2}\right) |P \cap P_{\mathbf{I},\mathsf{rand}}| \\ &= \left(1 - \frac{1}{(\ell+1)^2}\right) |P| . \end{aligned}$$

□

We now want to argue that for sufficiently large sets of vectors, there must be a sufficiently large set of possible suffixes for many vectors within the set. In particular, this will help us find triangles.

Lemma 16 (Lower-bounding the amount of possible suffixes). *Fix* $\mathbf{I}$, rand *and* $\zeta \in [0,1]$. *Let* $H \subset \mathbb{Z}_q^{\ell+1}$ *with* $|H| \geq \zeta \cdot q^{\ell+1}$, *such that for all* $\overrightarrow{h} \in H$, $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in S \subset \mathsf{Succ}$ *for some set* $S \supset H$. *Then for any constant* $c \in (0,1)$ *the following holds: for each index* $j \in [\ell+1]$, *there exists a subset* $H_j \subset H$ *with* $|H_j| > c \cdot |H|$, *such that for any* $\overrightarrow{h} \in H_j$,

$$\left|\Gamma_{j,S}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})\right| \geq (1-c) \cdot \zeta \cdot q$$

Proof. Assume toward a contradiction that for some $c \in (0,1)$ and index $j \in [\ell+1]$, no such $H_j \subset H$ exists. This can be rephrased as: there exists a subset $F \subset H$ such that $|F| > (1-c) \cdot |H|$ and for all $\overrightarrow{h} \in F$, $\left|\Gamma_{j,S}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})\right| < (1-c) \cdot \zeta \cdot q$. For any $\overrightarrow{h} \in F$, consider all successful vectors $\overrightarrow{h}'$ with $\overrightarrow{h}'_{[j-1]} = \overrightarrow{h}_{[j-1]}$. The j-th entry of $\overrightarrow{h}'$ takes $\left|\Gamma_{j,S}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})\right|$ possible values, and all of the remaining $\ell - j + 1$ entries take (up to) q possible values. Therefore,

$$\left|\left\{\overrightarrow{h}' \in {\mathbb{Z}_q}^{\ell+1} \;\middle|\; \begin{matrix}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in \mathsf{Succ} \\ \overrightarrow{h}'_{[j-1]} = \overrightarrow{h}_{[j-1]}\end{matrix}\right\}\right| \leq \left|\Gamma_{j,S}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})\right| \cdot q^{\ell-j+1} < (1-c) \cdot \zeta \cdot q^{\ell-j+2}$$

Then we have

$$\begin{aligned}
|F| &\leq \sum_{\substack{\overrightarrow{h}_{[j-1]} \\ \text{s.t. } \overrightarrow{h} \in F}} \left|\left\{\overrightarrow{h}' \in {\mathbb{Z}_q}^{\ell+1} \;\middle|\; \begin{matrix}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in \mathsf{Succ} \\ \overrightarrow{h}'_{[j-1]} = \overrightarrow{h}_{[j-1]}\end{matrix}\right\}\right| \\
&\leq q^{j-1} \cdot \max_{\overrightarrow{h} \in F} \left|\left\{\overrightarrow{h}' \in {\mathbb{Z}_q}^{\ell+1} \;\middle|\; \begin{matrix}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in \mathsf{Succ} \\ \overrightarrow{h}'_{[j-1]} = \overrightarrow{h}_{[j-1]}\end{matrix}\right\}\right| \\
&< (1-c) \cdot \zeta \cdot q^{(\ell+1)} \leq (1-c) \cdot |H|\,,
\end{aligned}$$

which is a contradiction to the assumption that $|F| > (1-c) \cdot |H|$. □

We apply the lower bound for suffixes from above to lower bound the number of triangle base corners that lie within P_G. We begin by proving the following technical lemma.

Lemma 17 (Many good partner tuples are triangle base corners). *Assume* $\epsilon \geq \frac{72(\ell+1)^3}{q}$ *and fix* $\mathbf{I}, \mathsf{rand}$ *such that* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in HR(\mathsf{Succ})$ *for some* $\overrightarrow{h}$. *Then at least* $\frac{5}{6}$ *of tuples in* $P_G \cap P_{\mathbf{I},\mathsf{rand}}$ *are triangle base corners at index* $\mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$. *That is, there exists a subset* $T \subset P_G \cap P_{\mathbf{I},\mathsf{rand}}$ *with* $|T| \geq \frac{5}{6}\left|P_G \cap P_{\mathbf{I},\mathsf{rand}}\right|$ *such that all tuples* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in T$ *are base corners of a triangle at index* $\mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$.

Proof. Take any $\mathbf{I}, \mathsf{rand}$ as in the lemma statement. Then

$$|P \cap P_{\mathbf{I},\mathsf{rand}}| \geq \frac{1}{2}\left|\mathsf{Succ}_{\mathbf{I},\mathsf{rand}}\right| \geq \frac{\epsilon}{4} \cdot q^{\ell+1},$$

where the first inequality is due to Corollary 5, and the second inequality is due to the definition of $HR(\mathsf{Succ})$. (Corollary 5 requires that $\epsilon \geq \frac{4}{q}$, which is implied by our assumption on ϵ here.)

Consider any index ν for which $B_\nu(\mathbf{I}, \mathsf{rand}) \cap P_G \neq \emptyset$. Then, by definition of P_G it holds that

$$|B_\nu(\mathbf{I}, \mathsf{rand})| \geq \frac{1}{(\ell+1)^3} |P \cap P_{\mathbf{I},\mathsf{rand}}| \geq \frac{\epsilon}{4(\ell+1)^3} \cdot q^{\ell+1}.$$

Applying Lemma 16 with $H = S = B_\nu(\mathbf{I}, \mathsf{rand})$, $\zeta = \frac{\epsilon}{4(\ell+1)^3}$, and $c = \frac{5}{6}$, we get: for any index $j \in [\ell+1]$, there exists a subset $T_j(\mathbf{I}, \mathsf{rand}) \subset B_\nu(\mathbf{I}, \mathsf{rand})$ with $|T_j(\mathbf{I}, \mathsf{rand})| \geq \frac{5}{6} |B_\nu|$ such that for all $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in T_j(\mathbf{I}, \mathsf{rand})$,

$$\left| \Gamma_{j,B_\nu}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \right| \geq \left(1 - \frac{5}{6}\right) \cdot \frac{\epsilon}{4(\ell+1)^3} \cdot q \geq 3.$$

The set $T_\nu(\mathbf{I}, \mathsf{rand})$ yields a set of triangle corners at index ν, which can be seen as follows. First, for any tuple $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in T_\nu(\mathbf{I}, \mathsf{rand})$, there is a partner tuple $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ at index ν (by definition of $B_\nu(\mathbf{I}, \mathsf{rand})$). Hence, $h_j, h_j' \in \Gamma_{j,B_\nu}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = \Gamma_{j,B_\nu}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$. As $\left|\Gamma_{j,B_\nu}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})\right| \geq 3$, there exists at least one further entry h_j'' which lies in $\Gamma_{j,B_\nu}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$. Thus, $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$, $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$, $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'')$ mutually fork from each other at index ν. Moreover, the first two among them are partners and at least one of them lies in $T_\nu(\mathbf{I}, \mathsf{rand})$. Hence, the three of them satisfy the definition of a triangle at index ν and at least one of the triangle base corners lies in $T_\nu(\mathbf{I}, \mathsf{rand})$. Finally, by definition of the set $B_\nu(\mathbf{I}, \mathsf{rand})$, $\mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = \nu$ as required.

Now define

$$T \coloneqq \bigcup_{\nu\colon B_\nu(\mathbf{I},\mathsf{rand}) \cap P_G \neq \emptyset} T_\nu(\mathbf{I}, \mathsf{rand}).$$

Using that the sets $B_\nu(\mathbf{I}, \mathsf{rand})$ s.t. $B_\nu(\mathbf{I}, \mathsf{rand}) \cap P_G$ form a partition of $P_{\mathbf{I},\mathsf{rand}} \cap P_G$

$$\begin{aligned} |T| &= \left| \bigcup_{\nu\colon B_\nu(\mathbf{I},\mathsf{rand}) \cap P_G \neq \emptyset} T_\nu \right| = \sum_{\nu\colon B_\nu(\mathbf{I},\mathsf{rand}) \cap P_G \neq \emptyset} |T_\nu| \\ &\geq \sum_{\nu\colon B_\nu(\mathbf{I},\mathsf{rand}) \cap P_G \neq \emptyset} \frac{5}{6} |B_\nu(\mathbf{I}, \mathsf{rand})| \geq \frac{5}{6} |P_{\mathbf{I},\mathsf{rand}} \cap P_G| , \end{aligned}$$

where the second equality follows from disjointness of the sets T_ν. This yields the statement of the Lemma. □

Corollary 7. *Assume ϵ as in Lemma 17. Then at least $\frac{5}{6}$ of all tuples in P_G are triangle base corners.*

Proof. Consider $\mathbf{I}, \mathsf{rand}$ such that $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in P_G$ for some $\overrightarrow{h}$. By definition, $P_G \subset HR(\mathsf{Succ})$, so $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in HR(\mathsf{Succ})$ for some $\overrightarrow{h}$. By Lemma 17, at

least $\frac{5}{6}$ of vectors in $P_G \cap \mathsf{Succ}_{\mathbf{I},\mathsf{rand}}$ are triangle base corners. Summing over all such $(\mathbf{I}, \mathsf{rand})$ pairs yields the result. □

Lemma 18. *If* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$,$(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ *are a triangle base at index* i*, and* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'')$ *is a partner of* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ *at index* i*, then* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$*,* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'')$ *are also a triangle base at index* i*.*

Proof. We distinguish between two cases:

Case $h'_i = h''_i$**:** Let $\overrightarrow{h}'''$ be such that $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$, $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$, $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}''')$ form a triangle at index i (such $\overrightarrow{h}'''$ must exist because $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$, $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ form a triangle base at index i). Then $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$, $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'')$, $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}''')$ also form a triangle at index i.

Case $h'_i \neq h''_i$**:** in this case $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$, $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}'')$, $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ form a triangle at index i where $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ takes the role of the triangle top.

□

C.3 Deferred Proof from Section 4.3

Proof (of Lemma 2). We show the lemma for a $\mathbf{z}$-side instance $\mathbf{I} = (1, w, \mathbf{y}, \overrightarrow{c}, \overrightarrow{r}, \overrightarrow{u})$; the argument for $\mathbf{y}$-side instances works analogously.

Let $(0, x, \mathbf{z}, \overrightarrow{d}, \overrightarrow{s}, \overrightarrow{v}) = \Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I})$ and $\overrightarrow{e}$ be the vector of queries to Sign_2 made by the adversary U on input $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$. We first show that $\mathbf{I}$ and $\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I})$ produce the same transcript. The bit b is not used actively in the simulation and thus does not affect the transcript. As the public key $\mathbf{y} = \mathbf{g}^x$ is the same in both instances from the view of U, and U is running on the same randomness rand, the info used will be the same for both instances. The tag key $\mathbf{z} = \mathbf{g}^w$ is the same in both instances. We now look at a single session of the protocol. For any i it holds that: The commitments $\mathbf{a}_i, \mathbf{b}_i$ are computed as $\mathbf{a}_i = \mathbf{g}^{v_i} = \mathbf{g}^{r_i + c \cdot x} = \mathbf{g}^{r_i} \cdot \mathbf{y}^{c_i}$ and $\mathbf{b}_i = \mathbf{g}^{s_i} \cdot \mathbf{z}^{d_i} = \mathbf{g}^{u_i - d_i \cdot w} \cdot \mathbf{g}^{d_i \cdot w} = \mathbf{g}^{u_i}$ which are the same group elements for both instances. We now use induction on the signing sessions in the order of the Sign_2 requests. Let therefore i_k be the session index of the kth closed session. As the instances provide the same response to Sign_1, the view up until the first query to Sign_2 is identical for U and thus it makes the same first Sign_2 query in both runs. Analogously, if the transcript is identical up to the kth request to Sign_2, the kth query e_{i_k} will also be identical. We now argue that for the kth closed session, if the views have been identical before, the kth response to Sign_2 is also identical. Thus, U makes the same query e_{i_k} to Sign_2. As $d_{i_k} = e_{i_k} - c_{i_k}$, it holds that $c'_{i_k} = e_{i_k} - (e_{i_k} - c_{i_k}) = c_{i_k}$, where c'_{i_k} is the c_{i_k} computed in the run with $\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I})$. Thus $r'_{i_k} = v_{i_k} - c_{i_k} \cdot x = (r_{i_k} + c_{i_k} \cdot x) - c_{i_k} \cdot x = r_{i_k}$ where r'_{i_k} is the r_{i_k} computed in the run with $\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I})$. Thus, the response $r_{i_k}, c_{i_k}, s_{i_k}, d_{i_k}$ of the oracle Sign_2 is identical. As the view as the adversary is identical for the entire run of the protocol, it must also output the same signatures in both runs. Thus, the two transcripts are identical.

We thus use $\overrightarrow{e}$ to denote the queries to Sign_2 in both runs. We now show that $\Phi_{\mathsf{rand},\overrightarrow{h}}$ is a self-inverse bijection. For an instance $\mathbf{I}$, we show that $\Phi_{\mathsf{rand},\overrightarrow{h}}(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I})) = \mathbf{I}$ (denote with ' the components of $\Phi_{\mathsf{rand},\overrightarrow{h}}(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}))$):

- $w' = \operatorname{dlog} \mathbf{z} = \operatorname{dlog} \mathbf{g}^w = w$
- $\mathbf{y}' = \mathbf{g}^x = \mathbf{y}$
- $\forall i \in [\ell]\colon c'_i = e_i - d_i = e_i - (e_i - c_i) = c_i$
- $\forall i \in [\ell]\colon r'_i = v_i - c_i \cdot x = (c_i \cdot x + r_i) - c_i \cdot x = r_i$
- $\forall i \in [\ell]\colon u'_i = d_i \cdot w + s_i = (e_i - c_i) \cdot w + [u_i - (e_i - c_i) \cdot w] = u_i$

Thus, $\Phi_{\mathsf{rand},\overrightarrow{h}}$ is a bijection and its own inverse. □

C.4 Counting The Image of Φ

In the following, we consider the image of the set P_G of all partners under Φ[8]. Recall that (roughly speaking) we defined P_G as the set of all 'good' partner tuples, i.e., tuples with many partner tuples. The goal of the next lemma is to lower bound the number of 'doubly good' tuples in P_G who retain a large number of partners after being mapped with Φ, i.e., good partner tuples whose image under Φ remains 'good'. Below, we implicitly use the fact that $\Phi(P)$ yields a set of partner tuples (due to Lemma 3).

Lemma 19 (Many good partner tuples have a good image). *Let*

$$P'_G = \left\{ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \Phi(P) \middle| \left| B_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})} \cap \Phi(P) \right| \geq \frac{1}{(\ell+1)^3} \left| \Phi(P) \cap \mathsf{Succ}_{\mathbf{I},\mathsf{rand}} \right| \right\}.$$

Then

$$\left| \Phi(P_G) \cap P'_G \right| \geq \left(1 - \frac{2}{(\ell+1)^2} \right) |P| .$$

Proof. Fix $\mathbf{I}, \mathsf{rand}$ with $\Phi(P) \cap \mathsf{Succ}_{\mathbf{I},\mathsf{rand}} \neq \emptyset$. Then it holds that each $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \mathsf{Succ}_{\mathbf{I},\mathsf{rand}} \cap \Phi(P)$ lies in one set $B_i \cap \Phi(P)$ (namely $i = \mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$). As there are $(\ell+1)$ hash queries, there are at most $(\ell+1)$ such sets. Thus, by lemma 13 with $\alpha = \frac{1}{(\ell+1)}$, there exists a set G_α such that for all $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in G_\alpha$ it holds that $\left| B_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})} \cap \Phi(P) \right| \geq \frac{1}{(\ell+1)^3} \left| \Phi(P) \cap \mathsf{Succ}_{\mathbf{I},\mathsf{rand}} \right|$ and $|G_\alpha| \geq \left(1 - \frac{1}{(\ell+1)}\right) \cdot |\mathsf{Succ}_{\mathbf{I},\mathsf{rand}}| \cap \Phi(P)$. Taking the G_α of all $\mathsf{Succ}_{\mathbf{I},\mathsf{rand}}$ with $\mathsf{Succ}_{\mathbf{I},\mathsf{rand}} \cap \Phi(P) \neq \emptyset$ yields that $|P'_G| \geq \left(1 - \frac{1}{(\ell+1)^2}\right) \cdot |P|$. Since $\Phi(P_G) \subset \Phi(P)$ and $P'_G \subset \Phi(P)$, it holds, using lemma 15 and the inclusion-exclusion principle that

$$\left| \Phi(P_G) \cap P'_G \right| \geq \left(1 - \frac{2}{(\ell+1)^2} \right) \cdot |P| .$$

□

[8] We have defined $\Phi(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = (\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h})$, hence $\Phi(P)$ is well-defined.

We now turn to lower bounding the number of triangle base corner within the set P'_G from Lemma 19. Together with the fact that P_G has many triangle base-corners, we will then be able to conclude that the images of many triangle base-corners remain base-corners in some other triangle at the same index.

Lemma 20 (Many images of good partner tuples are triangle base corners). *Assume* $\epsilon \geq \frac{3\cdot 144\cdot \frac{(\ell+1)^2-1}{(\ell+1)^2}}{q}$ *as well as* ϵ *as in Corollary 5 (whichever is larger). Then at least* $\frac{11}{18}$ *of tuples* $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in P'_G$ *are base corners of a triangle at* $\mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$.

Proof. Let $\epsilon_{P'_G} = \frac{|P'_G|}{|\mathcal{I}\times\mathcal{R}\times\mathbb{Z}_q{}^{(\ell+1)}|}$ be the probability of getting a tuple in P'_G when sampling a tuple uniformly at random. Then

$$\epsilon_{P'_G} \geq \left(1 - \frac{1}{(\ell+1)^2}\right)\cdot \epsilon_P \geq \left(1 - \frac{1}{(\ell+1)^2}\right)\cdot \frac{\epsilon}{4}, \qquad (*)$$

where the first inequality is due to the fact that $|P'_G| \geq \left(1 - \frac{1}{(\ell+1)^2}\right)|P|$ (see the proof of Lemma 19), and the second inequality is due to Corollary 6.

Define the heavy row of P'_G as

$$HR(P'_G) = \left\{(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in P'_G \middle| |P'_G \cap \mathsf{Succ}_{\mathbf{I},\mathsf{rand}}| \geq \frac{\epsilon_{P'_G}}{3}\cdot q^{(\ell+1)}\right\}.$$

By Lemma 12, $|HR(P'_G)| \geq \frac{2}{3}|P'_G|$. Now consider any tuple $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in HR(P'_G)$. From the definition of P'_G it follows that

$$\begin{aligned}\left|B_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})} \cap P'_G\right| &\geq \frac{1}{(\ell+1)^3}\,|\Phi(P) \cap \mathsf{Succ}_{\mathbf{I},\mathsf{rand}}| \\ &\overset{P'_G\subset\Phi(P)}{\geq} \frac{1}{(\ell+1)^3}\,|P'_G \cap \mathsf{Succ}_{\mathbf{I},\mathsf{rand}}| \geq \frac{\epsilon_{P'_G}}{3(\ell+1)^3}\cdot q^{(\ell+1)}.\end{aligned}$$

Similar to the proof of Lemma 17, we apply Lemma 16 with $H = B_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})} \cap P'_G$, $S = S_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})} \cap P'_G$, $\zeta = \frac{\epsilon_{P'_G}}{3(\ell+1)^3}$, and $c = \frac{11}{12}$. This yields that for all indices $j \in [\ell+1]$, there exists a subset $H_j(\mathbf{I}, \mathsf{rand}) \subset B_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})} \cap P'_G$ with $|H_j(\mathbf{I}, \mathsf{rand})| \geq \frac{11}{12}\left|B_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})} \cap P'_G\right|$ such that for all $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in H_j(\mathbf{I}, \mathsf{rand})$,

$$\left|\Gamma_{j,S}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')\right| \geq \left(1 - \frac{11}{12}\right)\cdot\frac{\epsilon_{P'_G}}{3(\ell+1)^3}\cdot q \overset{(*)}{\geq} \left(1 - \frac{1}{(\ell+1)^2}\right)\cdot\frac{\epsilon}{144(\ell+1)^3}\cdot q \geq 3$$

where the last step is obtained by plugging in ϵ as in the lemma statement. Setting $j = \mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$, we obtain a subset $H_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})}(\mathbf{I}, \mathsf{rand})$ of triangle base corners at index $\mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ by a similar argument as in

lemma 17. (Henceforth we simplify it to $H_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})}$.) Let T be the union of all such sets, i.e.,

$$T = \bigcup_{(\mathbf{I},\mathsf{rand},\overrightarrow{h})\in HR(P'_G)} H_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})}$$

Then all vectors in T are triangle base corners, and

$$|T| = \left|\bigcup_{(\mathbf{I},\mathsf{rand},\overrightarrow{h})\in HR(P'_G)} H_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})}\right| \geq \frac{11}{12}\left|\bigcup_{(\mathbf{I},\mathsf{rand},\overrightarrow{h})} \left(B_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})} \cap P'_G\right)\right|$$
$$\overset{(**)}{\geq} \frac{11}{12}\left|HR(P'_G)\right| \geq \frac{11}{12}\cdot\frac{2}{3}\left|P'_G\right| = \frac{11}{18}\left|P'_G\right|,$$

where $(**)$ is because for any $(\mathbf{I},\mathsf{rand},\overrightarrow{h}\,')\in HR(P'_G)$ it holds that $(\mathbf{I},\mathsf{rand},\overrightarrow{h}\,')\in B_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h}\,')}\cap P'_G \subset \bigcup_{(\mathbf{I},\mathsf{rand},\overrightarrow{h})\in HR(P'_G)}\left(B_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})}\cap P'_G\right)$, hence $HR(P'_G)\subset \bigcup_{(\mathbf{I},\mathsf{rand},\overrightarrow{h})\in HR(P'_G)}\left(B_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})}\cap P'_G\right)$. □

Having bounded the number of triangle corners within both P_G and P'_G, we now compute their overlap. More precisely, we show that there is a large set T such that all tuples $(\mathbf{I},\mathsf{rand},\overrightarrow{h})\in T$ are triangle base corners and, moreover, $\Phi(\mathbf{I},\mathsf{rand},\overrightarrow{h})$ is also a triangle base-corner at the same index.

Lemma 21. *Assume ϵ as in Lemma 20. Then there exists a set $T\subset P$ with $|T|\geq\frac{1}{12}|P|$ such that for all $(\mathbf{I},\mathsf{rand},\overrightarrow{h})\in T$ it holds that both $(\mathbf{I},\mathsf{rand},\overrightarrow{h})$ and $\Phi(\mathbf{I},\mathsf{rand},\overrightarrow{h})$ are triangle base-corners at index $\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})$.*[9]

Proof. Let C denote the set of triangle base corners at their maximal branching index, i.e.,

$$C \coloneqq \left\{(\mathbf{I},\mathsf{rand},\overrightarrow{h})\middle|\exists\overrightarrow{h}\,',\overrightarrow{h}\,'' \colon (\overrightarrow{h},\overrightarrow{h}\,',\overrightarrow{h}\,'')\in\triangle_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})}\right\}.$$

Then

$$|P'_G\cap C| \overset{\text{Lemma 20}}{\geq} \frac{11}{18}|P'_G| \overset{\text{Lemma 19}}{\geq} \frac{11}{18}\cdot\left(1-\frac{1}{(\ell+1)^2}\right)|P| \overset{\ell\geq 1}{\geq} \frac{11}{24}|P|;$$

$$|P_G\cap C| \overset{\text{Lemma 17}}{\geq} \frac{5}{6}|P_G| \overset{\text{Lemma 15}}{\geq} \left(1-\frac{1}{(\ell+1)^2}\right)\cdot\frac{5}{6}|P| \overset{(\ell\geq 1}{\geq} \frac{5}{8}|P|.$$

Let $T=\Phi(P_G\cap C)\cap(P'_G\cap C)$. Clearly $T\subset C\cap\Phi(C)$, implying that the tuples in T satisfy the requirements of the lemma. Moreover, $T\subset\Phi(P_G)\cap P'_G\subset P$. By inclusion-exclusion, this yields

$$\begin{aligned}|T| &\geq |\Phi(P_G\cap C)| + |P'_G\cap C| - |P| = |P_G\cap C| + |P'_G\cap C| - |P|\\ &\geq \frac{5}{8}|P| + \frac{11}{24}|P| - |P| = \frac{1}{12}|P|\end{aligned}$$

□

[9] Note that $\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h}) = \mathrm{Br}_{\max}(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}),\mathsf{rand},\overrightarrow{h})$ due to corollary 2.

C.5 Proofs from Section 4.4

Proof (of Lemma 7).

$$\left|B_T^{\times}\right| \geq \frac{1}{2}\left|B_T\right| \geq \frac{1}{2}\cdot\frac{1}{12}\left|P\right| \geq \frac{1}{2}\cdot\frac{1}{12}\cdot\frac{1}{4}\left|\mathsf{Succ}\right| = \frac{\epsilon}{96}\left|\mathcal{I}\times\mathcal{R}\times{\mathbb{Z}_q}^{\ell+1}\right|,$$

where the steps follow (in this order) from Lemma 5, Corollary 8, Corollary 6, and the definition of ϵ. Thus, $\epsilon_{B_T^{\times}} = \frac{\left|B_T^{\times}\right|}{\left|\mathcal{I}\times\mathcal{R}\times{\mathbb{Z}_q}^{\ell+1}\right|} \geq \frac{\epsilon}{96}$. □

Proof (of Lemma 4). Suppose $\omega_j \neq \omega_j'$. Let A make two runs, one on $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ and one on $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$. As the two runs were identical up to the point when A makes its j-th query to H, this query $\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j, \mathbf{z}_j, m_j$ was also identical (note that rand is fixed and thus A is deterministic). Since $(\overrightarrow{h}, \overrightarrow{h}') \in F_j(\mathbf{I}, \mathsf{rand})$, we know that $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in \mathsf{Succ}$, i.e., A outputs $\ell+1$ valid signatures in both runs. This means that the two sigma protocol transcripts $(\boldsymbol{\alpha}_j, \omega_j, \rho_j)$ and $(\boldsymbol{\alpha}_j, \omega_j', \rho_j')$ are both accepting, so we have $\boldsymbol{\alpha}_j = \mathbf{g}^{\rho_j}\cdot\mathbf{g}^{\omega_j\cdot x} = \mathbf{g}^{\rho_j'}\cdot\mathbf{g}^{\omega_j'\cdot x}$ Thus, x can be computed as $x = (\omega_j' - \omega_j)^{-1}\cdot(\rho_j - \rho_j')$. Now suppose that $\omega_j = \omega_j'$. In this case, since $\omega_j + \delta_j = h_j \neq h_j' = \omega_j' + \delta_j'$, $\delta_j \neq \delta_j'$. For $\delta_j \neq \delta_j'$ we have $\boldsymbol{\beta}_j = \mathbf{g}^{\sigma_j}\cdot\mathbf{g}^{\delta_j\cdot w} = \mathbf{g}^{\sigma_j'}\cdot\mathbf{g}^{\delta_j'\cdot w}$ and thus $w = (\delta_j' - \delta_j)^{-1}\cdot(\sigma_j - \sigma_j')$. □

Proof (of Lemma 5).

1. We show the equation for $B_T^{\mathbf{y}}$; the one for $B_T^{\mathbf{z}}$ can be proved similarly. By definition of B_T and the self-inverse property of Φ, it follows that $\Phi(B_T) = B_T$ and thus $\Phi(B_T^{\mathbf{y}}) \subset B_T$. Fix any $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in B_T^{\mathbf{y}}$. Further, fix a vector $\overrightarrow{h}'$ with $(\overrightarrow{h}, \overrightarrow{h}') \in \mathrm{prt}_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})}(\mathbf{I}, \mathsf{rand})$ for which there exist $\overrightarrow{h}''$, $\overrightarrow{h}'''$ with $(\overrightarrow{h}, \overrightarrow{h}', \overrightarrow{h}'') \in \triangle_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})}(\mathbf{I}, \mathsf{rand})$, $(\overrightarrow{h}, \overrightarrow{h}', \overrightarrow{h}''') \in \triangle_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})}(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand})$ and such that the $\mathbf{y}$-side witness can be extracted from $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ and $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ at $\mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ (i.e., as guaranteed the definition of $B_T^{\mathbf{y}}$).

By Lemma 2, the signatures resulting from the tuple $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ and $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ are the same as the signatures resulting from $(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h})$ and $(\Phi_{\mathsf{rand},\overrightarrow{h}'}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h}')$, respectively. As $\overrightarrow{h}$ and $\overrightarrow{h}'$ are partners, Lemma 3 implies that $(\Phi_{\mathsf{rand},\overrightarrow{h}'}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h}') = (\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h}')$ and by Corollary 2, $\mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = \mathrm{Br}_{\max}(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h})$. Hence, the signatures that result from $(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h}), (\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h}')$ are the same signatures that result from $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ and $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$. As the witness that can be extracted from $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ and $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$ at index $\mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ is completely determined by the signatures corresponding to $h_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})}$ and $h'_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})}$, the same witness can be extracted from $(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h}), (\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h}')$

at index i. By assumption, $(\overrightarrow{h}, \overrightarrow{h}', \overrightarrow{h}''') \in \triangle_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})}(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand})$ and $(\overrightarrow{h}, \overrightarrow{h}', \overrightarrow{h}'') \in \triangle_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})}(\mathbf{I}, \mathsf{rand}) = \triangle_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})}(\Phi_{\mathsf{rand},\overrightarrow{h}}(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I})), \mathsf{rand})$, where we have applied the self-inverse property of $\Phi_{\mathsf{rand},\overrightarrow{h}}$. So $(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h})$ meets the requirements of the definition of $B_T^{\mathbf{y}}$, and thus $(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h}) \in B_T^{\mathbf{y}}$. As $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in B_T^{\mathbf{y}}$ was chosen arbitrarily, we obtain that $\Phi(B_T^{\mathbf{y}}) \subset B_T^{\mathbf{y}}$. Using the self-inverse property of the bijection Φ once more, we immediately obtain the converse inclusion $B_T^{\mathbf{y}} \subset \Phi(B_T^{\mathbf{y}})$. Thus $\Phi(B_T^{\mathbf{y}}) = B_T^{\mathbf{y}}$.

2. Consider any tuple $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in B_T$. By Lemma 4, at least one witness can be extracted from each triangle base.Let $i = \mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$. For any $\overrightarrow{h}'$ as in Definition 14, we know that $(\overrightarrow{h}, \overrightarrow{h}') \in F_i(\mathbf{I}, \mathsf{rand})$. By Lemma 4, at least one witness can be extracted from $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ and $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$, so at least one of $D_i^{\mathbf{y}}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}), D_i^{\mathbf{z}}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ is not $\emptyset$. Suppose $D_i^{\times}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ is the larger of the two sets; then $D_i^{\times}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \neq \emptyset$ and thus $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in B_T^{\times}$. This shows that any tuple in B_T is in $B_T^{\mathbf{y}}$ or $B_T^{\mathbf{z}}$, so $B_T^{\mathbf{y}} \cup B_T^{\mathbf{z}} = B_T$. □

We now relate the sets T from Lemma 21 and B_T. Recall that elements of the set T are triangle base-corners $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ at $\mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ s.t. $(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h})$ remains a triangle base-corner at index $\mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$. Concretely, we show that $T \subset B_T$. This establishes, for one, that B_T is large (because T is large, as we have shown).

Lemma 22. *Let T be as in Lemma 21. Then $T \subset B_T$.*

Proof. Fix some $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in T$. Then there exist $\overrightarrow{h}', \overrightarrow{h}'', \overrightarrow{h}''', \overrightarrow{h}''''$ such that $(\overrightarrow{h}, \overrightarrow{h}', \overrightarrow{h}''') \in \triangle_{\mathrm{Br}_{\max}(\mathbf{I},\mathsf{rand},\overrightarrow{h})}(\mathbf{I}, \mathsf{rand})$ and $(\overrightarrow{h}, \overrightarrow{h}'', \overrightarrow{h}'''') \in \triangle_{\mathrm{Br}_{\max}(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}),\mathsf{rand},\overrightarrow{h})}(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand})$. By Corollary 2, $\mathrm{Br}_{\max}(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand}, \overrightarrow{h}) = \mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$; let this index be i. In the following, we also use that $(\overrightarrow{h}, \overrightarrow{h}') \in \mathrm{prt}_i(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand})$ which follows from Lemma 3.

- If $h_i' = h_i''$, then we can replace $\overrightarrow{h}''$ by $\overrightarrow{h}'$ in the triangle $(\overrightarrow{h}, \overrightarrow{h}'', \overrightarrow{h}'''')$ as $h_i' = h_i'' \neq h_i''''$. Since $(\overrightarrow{h}, \overrightarrow{h}') \in \mathrm{prt}_i(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand})$, $(\overrightarrow{h}, \overrightarrow{h}', \overrightarrow{h}'''') \in \triangle_i(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand})$ and thus, $\overrightarrow{h}, \overrightarrow{h}', \overrightarrow{h}''$ and $\overrightarrow{h}''''$ meet the definition of B_T. Hence, $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in B_T$.
- If $h_i' \neq h_i''$, then, since $(\overrightarrow{h}, \overrightarrow{h}') \in \mathrm{prt}_i(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand})$, $(\overrightarrow{h}', \overrightarrow{h}'') \in \mathrm{prt}_i(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand})$, and $h_i \neq h_i''$, it must also be the case that $(\overrightarrow{h}, \overrightarrow{h}'') \in \mathrm{prt}_i(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand})$. This implies that $(\overrightarrow{h}, \overrightarrow{h}', \overrightarrow{h}'') \in \triangle_i(\Phi_{\mathsf{rand},\overrightarrow{h}}(\mathbf{I}), \mathsf{rand})$ and thus $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in B_T$.

Either way, $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in B_T$, so $T \subset B_T$. □

Corollary 8. $|B_T| \geq \frac{1}{12}|P|$.

Proof (of Lemma 8). We show this for $G_{\mathbf{y}}$; the proof for $G_{\mathbf{z}}$ works analogously. By Lemma 6 it holds that $\left|B^{\times}_{T,\mathbf{y}}\right| = \frac{1}{2}\left|B^{\times}_{T}\right|$.

For $i \in [\ell+1]$ we define a subset of $B^{\times}_{T,\mathbf{y}}$ that are both-sided triangle base corners at index i as follows:

$$G_{i,\mathbf{y}} \coloneqq \left\{(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in B^{\times}_{T,\mathbf{y}} \middle| i = \mathrm{Br}_{\max}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})\right\}$$

It is easy to see that $G_{i,\mathbf{y}}$ ($i \in [\ell+1]$) form a partition of $B^{\times}_{T,\mathbf{y}}$. We note that membership in $G_{i,\mathbf{y}}$ is symmetrical, i.e., the other base corner of a both-sided triangle is always also contained in $G_{i,\mathbf{y}}$.

Denote the set of indices $i \in [\ell+1]$ such that $|G_{i,\mathbf{y}}| \geq \frac{1}{4(\ell+1)}\left|B^{\times}_{T,\mathbf{y}}\right|$ as $\mathcal{J}$. We now apply Lemma 13 with $B_i = G_{i,\mathbf{y}}$, $b = \ell+1$, $\alpha = \frac{1}{4}$, and $X = B^{\times}_{T,\mathbf{y}}$. Due to Lemma 13, at least $\frac{3}{4}$ of the tuples $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ in $B^{\times}_{T,\mathbf{y}}$ has the property that there exists $i \in \mathcal{J}$ such that $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in G_{i,\mathbf{y}}$. For each $G_{i,\mathbf{y}}$ with $i \in \mathcal{J}$, define

$$HR(G_{i,\mathbf{y}}) = \left\{(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in G_{i,\mathbf{y}} \middle| \left|G_{i,\mathbf{y}} \cap \mathsf{Succ}_{\mathbf{I},\mathsf{rand},\overrightarrow{h}_{[i-1]}}\right| \geq \frac{1}{8} \cdot \frac{1}{\ell+1}\epsilon_{B^{\times}_{T,\mathbf{y}}} \cdot q^{\ell-i+2}\right\}$$

where $\epsilon_{B^{\times}_{T,\mathbf{y}}} \coloneqq \frac{\left|B^{\times}_{T,\mathbf{y}}\right|}{\left|\mathcal{I}\times\mathcal{R}\times\mathbb{Z}_q{}^{\ell+1}\right|} = \frac{1}{2}\epsilon_{B^{\times}_{T}}$. Then, by Lemma 12, $|HR(G_{i,\mathbf{y}})| \geq \frac{1}{2}|G_{i,\mathbf{y}}|$ for each $G_{i,\mathbf{y}}$ with $i \in \mathcal{J}$. Now, fix some arbitrary $i \in \mathcal{J}$ and $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in HR(G_{i,\mathbf{y}}) \subset G_{i,\mathbf{y}}$. Furthermore, fix a partner $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in D^{\times}_{i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})'$. Then, there exist at most $q^{\ell+1-i}$ vectors in $G_{i,\mathbf{y}} \cap \mathsf{Succ}_{\mathbf{I},\mathsf{rand},\overrightarrow{h}_{[i-1]}}$ that share the first i entries with $\overrightarrow{h}$ and at most $q^{\ell+1-i}$ vectors in $G_{i,\mathbf{y}} \cap \mathsf{Succ}_{\mathbf{I},\mathsf{rand},\overrightarrow{h}_{[i-1]}}$ that share the first i entries with its designated partner $\overrightarrow{h}'$. These vectors do not form triangles at index i with $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ and $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$. We denote this set of non-triangle-tops by $N(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}, \overrightarrow{h}')$ and by the above reasoning, $\left|N(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}, \overrightarrow{h}')\right| \leq 2 \cdot q^{\ell+1-i}$. We note that $\mathsf{Succ}_{\mathbf{I},\mathsf{rand},\overrightarrow{h}_{[i-1]}} \setminus N(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}, \overrightarrow{h}') \subset T^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$. Thus, the amount of triangle tops for $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ is at least

$$\begin{aligned}
\left|T^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})\right| &\geq \left|\mathsf{Succ}_{\mathbf{I},\mathsf{rand},\overrightarrow{h}_{[i-1]}} \setminus N(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}, \overrightarrow{h}')\right| \\
&\geq \left|\mathsf{Succ}_{\mathbf{I},\mathsf{rand},\overrightarrow{h}_{[i-1]}}\right| - \left|N(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}, \overrightarrow{h}')\right| \\
&\geq \frac{1}{8} \cdot \frac{1}{\ell+1}\epsilon_{B^{\times}_{T,\mathbf{y}}} \cdot q^{\ell+1-i+1} - 2 \cdot q^{\ell+1-i} \\
&\geq \frac{1}{16} \cdot \frac{1}{\ell+1}\epsilon_{B^{\times}_{T}} \cdot q^{\ell+1-i+1} - 2 \cdot q^{\ell+1-i}
\end{aligned}$$

Since $(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in HR(G_{i,\mathbf{y}})$ was arbitrarily chosen, taking the union over all $HR(G_{i,\mathbf{y}})$ s.t. $i \in \mathcal{J}$ yields the statement.

□

D Proof of Theorem 2

Proof. Without loss of generality, assume that U's queries to H^* are all distinct. We first describe a game-hop.

$\mathbf{G}_0$: This is the original ℓ-**OMUF**$_{\mathsf{AO}}$ game.
$\mathbf{G}_1$: This game outputs 0 if U outputs a valid tuple $(m, \mathsf{sig}, \mathsf{info})$ where info has never been queried to H^*.

Claim. $\left|\Pr[\mathbf{G}_0^{\mathsf{U}} = 1] - \Pr[\mathbf{G}_1^{\mathsf{U}} = 1]\right| \leq \frac{\ell}{q}$.

Proof. Let Valid be the event that U outputs a valid tuple $(m, \mathsf{sig} = (\rho, \omega, \sigma, \delta), \mathsf{info})$ where info has never been queried to H^*; $\mathbf{G}_0$ and $\mathbf{G}_1$ are identical unless Valid happens. For each output $(m, \mathsf{sig}, \mathsf{info})$ of U, if $H^*(\mathsf{info})$ has never been queried, $\mathbf{z} = H^*(\mathsf{info})$ is a random element in $\mathbb{G}$ independent of all other random variables in U's view. Hence, $H(\mathbf{g}^{\rho}\mathbf{y}^{\omega}, \mathbf{g}^{\sigma}\mathbf{z}^{\delta}, \mathbf{z}, m)$ is a random integer in $\mathbb{Z}_q$, and the probability that it equals $\omega + \delta$ is $1/q$. Since there are at most ℓ such output tuples in total, we have that

$$\Pr[\mathsf{Valid}] \leq \frac{\ell}{q},$$

and the claim follows. □

The reduction B against 1-info-**OMUF**$_{\mathsf{AO}}$ behaves as follows.

Setup: On input a public key $\mathsf{pk} = \mathbf{y}$, B forwards it to U and samples $J \xleftarrow{\$} [Q_{\mathsf{info}}]$ (a guess that U's J-th H^* query is part of its final output).
Online Phase: B answers signing and hash queries as follows.
- **Queries to H^*:** For the J-th query info_J to H^*, B forwards the query to its challenger and forwards the response back to U. For any other query info, B lazily samples $w_{\mathsf{info}} \xleftarrow{\$} \mathbb{Z}_q$ and sets $H^*(\mathsf{info}) := \mathbf{g}^{w_{\mathsf{info}}}$.
- **Queries to H:** B forwards these queries to its challenger and forwards the responses back to U.
- **Queries to sign_1:** On input info, if $\mathsf{info} = \mathsf{info}_J$, B forwards the query to its challenger and forwards the response back to U. Otherwise B behaves as the $\mathbf{z}$-side signer. That is, it increments the session id sid, sets $\mathsf{info}_{\mathsf{sid}} := \mathsf{info}$, samples $c_{\mathsf{sid}}, r_{\mathsf{sid}}, v_{\mathsf{sid}} \xleftarrow{\$} \mathbb{Z}_q$, and sets $\mathbf{a} := \mathbf{y}^{c_{\mathsf{sid}}} \cdot \mathbf{g}^{r_{\mathsf{sid}}}$ and $\mathbf{b} := \mathbf{g}^{v_{\mathsf{sid}}}$. It saves the internal state $c_{\mathsf{sid}}, r_{\mathsf{sid}}, v_{\mathsf{sid}}$ and outputs $\mathsf{sid}, \mathbf{a}, \mathbf{b}$ to U.
- **Queries to sign_2:** On input (sid, e), B checks if $\mathsf{info}_{\mathsf{sid}} = \mathsf{info}_J$. If so, it forwards the query to its challenger. Otherwise it computes $d_{\mathsf{sid}} := e - c_{\mathsf{sid}}$ and $s_{\mathsf{sid}} := v_{\mathsf{sid}} - d_{\mathsf{sid}} w_{\mathsf{info}}$. It outputs $c_{\mathsf{sid}}, r_{\mathsf{sid}}, d_{\mathsf{sid}}, s_{\mathsf{sid}}$ to U.

Output determination: When U outputs a list of signatures $(m_i, \mathsf{sig}_i, \mathsf{info}_i)_{i=1}^{\ell+1}$, B checks that all info were queried to H^* by U. If so, B outputs all $(m_i, \mathsf{sig}_i, \mathsf{info}_i)$ tuples with $\mathsf{info}_i = \mathsf{info}_J$. Otherwise B aborts.

We now analyze the advantage of the reduction B. Due to the witness-indistinguishability of the scheme (see Lemma 2 in Section 4.3), B simulates

game $\mathbf{G}_1$ perfectly to U. If U wins $\mathbf{G}_1$, there must be one tag for which U has output more signatures than closed signing sessions. By the definition of $\mathbf{G}_1$, this tag was queried to H^* by U. Therefore, with probability $\frac{1}{Q_{\mathsf{info}}}$, this tag is info_J.

We conclude that

$$\mathrm{adv}^{\ell_{\mathsf{info}}\text{-1-info-}\mathbf{OMUF}_{\mathsf{AO}}}_{\mathsf{B}} \geq \frac{\Pr[\mathbf{G}_1^{\mathsf{U}} = 1]}{Q_{\mathsf{info}}} \geq \frac{\Pr[\mathbf{G}_0^{\mathsf{U}} = 1] - \frac{\ell}{q}}{Q_{\mathsf{info}}} = \frac{\mathrm{adv}^{\ell-\mathbf{OMUF}_{\mathsf{AO}}}_{Q_{\mathsf{info}},\ell_{\mathsf{info}},\mathsf{A}} - \frac{\ell}{q}}{Q_{\mathsf{info}}}.$$

□

E Deferred Figures from the Main body

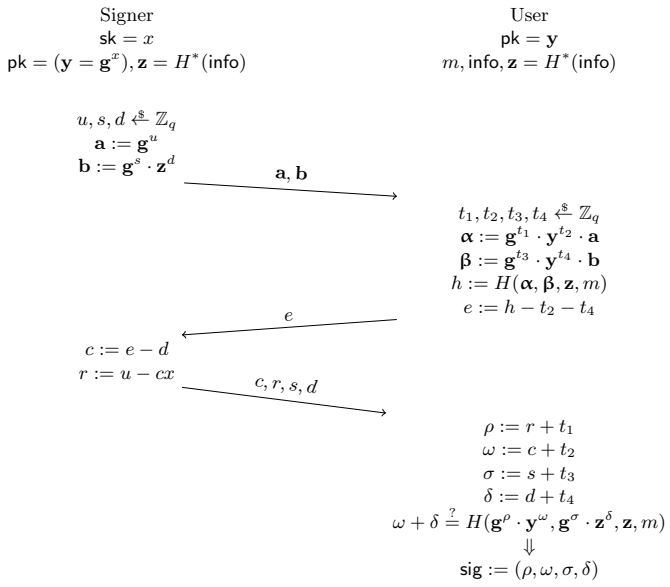


Fig. 6: The Abe-Okamoto scheme

F Applying our Techniques to Abe's Blind Signature Scheme

In this section, we briefly sketch how the technique described in the main body can be applied to the blind signature scheme by Abe [1]. We note that Abe's blind signature scheme is believed to be immune to the ROS-attack, and thus could be secure even in a setting with polynomially many signing sessions, but our security reduction yields a large loss in that setting.

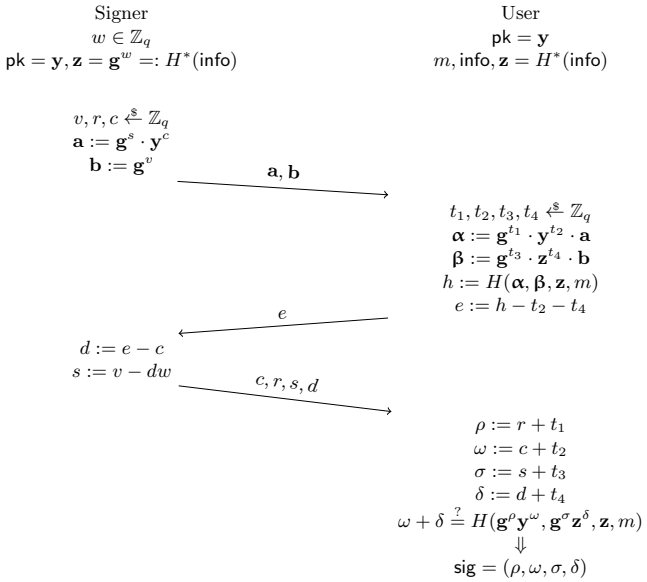


Fig. 7: The Abe-Okamoto scheme using the $\mathbf{z}$-side witness w to sign

F.1 Abe's Blind Signature Scheme

We recall how the blind signature scheme from [1] works.

Key Generation. On input pp, KeyGen samples $\mathbf{h} \xleftarrow{\$} \mathbb{G}$, $x \xleftarrow{\$} \mathbb{Z}_q$ and sets $\mathbf{y} \leftarrow \mathbf{g}^x$, It sets $\mathbf{z} \leftarrow H_1(\mathbf{g}, \mathbf{h}, \mathbf{y})$. It sets $\mathsf{sk} \leftarrow x$, $\mathsf{pk} \leftarrow (\mathbf{g}, \mathbf{h}, \mathbf{y}, \mathbf{z})$ and returns $(\mathsf{sk}, \mathsf{pk})$.

Signer. The signer $\mathsf{Sign} = (\mathsf{Sign}_1, \mathsf{Sign}_2)$ behaves as follows:

Sign_1**:** On input sk, Sign_1 samples $\mathrm{rnd} \xleftarrow{\$} \{0,1\}^\lambda$ and $u, d, s_1, s_2 \xleftarrow{\$} \mathbb{Z}_q$. It computes $\mathbf{z}_1 \leftarrow H_2(\mathrm{rnd})$, $\mathbf{z}_2 \leftarrow \mathbf{z}/\mathbf{z}_1$, $\mathbf{a} \leftarrow \mathbf{g}^u$, $\mathbf{b_1} \leftarrow \mathbf{g}^{s_1} \cdot \mathbf{z}_1^d$, $\mathbf{b_2} \leftarrow \mathbf{h}^{s_2} \cdot \mathbf{z}_2^d$. It returns a commitment $(\mathrm{rnd}, \mathbf{a}, \mathbf{b_1}, \mathbf{b_2})$ and a state $\mathsf{st}_S = (u, d, s_1, s_2)$.

Sign_2**:** On input a secret key sk, a challenge e, and state $\mathsf{st}_S = (u, d, s_1, s_2)$, Sign_2 computes $c \leftarrow e - d \mod q$, $r \leftarrow u - c \cdot \mathsf{sk} \mod q$ and returns the response (c, d, r, s_1, s_2).

User. The user $\mathsf{User} = (\mathsf{User}_1, \mathsf{User}_2)$ behaves as follows:

User_1**:** On input a public key pk and a commitment $(\mathrm{rnd}, \mathbf{a}, \mathbf{b_1}, \mathbf{b_2})$, and message m, User_1 does the following. It samples $\gamma \xleftarrow{\$} \mathbb{Z}_q^*$ and $\tau, t_1, t_2, t_3, t_4, t_5 \xleftarrow{\$} \mathbb{Z}_q$. Then, it computes $\mathbf{z}_1 \leftarrow H_2(\mathrm{rnd})$, $\alpha \leftarrow \mathbf{a} \cdot \mathbf{g}^{t_1} \cdot \mathbf{y}^{t_2}$, $\zeta \leftarrow \mathbf{z}^\gamma$, $\zeta_1 \leftarrow \mathbf{z}_1^\gamma$, $\zeta_2 \leftarrow \zeta/\zeta_1$. Next, it sets $\beta_1 \leftarrow \mathbf{b}_1^\gamma \cdot \mathbf{g}^{t_3} \cdot \zeta_1^{t_4}$, $\beta_2 \leftarrow \mathbf{b}_2^\gamma \cdot \mathbf{h}^{t_5} \cdot \zeta_2^{t_4}$, $\eta \leftarrow \mathbf{z}^\tau$, and $h \leftarrow H_3(\zeta, \zeta_1, \alpha, \beta_1, \beta_2, \eta, m)$. Finally, it computes a challenge $e \leftarrow h - t_2 - t_4 \mod q$, the state $St_U \leftarrow (\gamma, \tau, t_1, t_2, t_3, t_4, t_5, m)$ and returns e, St_U.

User_2**:** On input a public key pk, a response (c, d, r, s_1, s_2) and a state $(\gamma, \tau, t_1, t_2, t_3, t_4, t_5, m)$, User_2 first computes $\rho \leftarrow r + t_1$, $\omega \leftarrow c + t_2$, $\sigma_1 \leftarrow \gamma \cdot s_1 + t_3$, $\sigma_2 \leftarrow \gamma \cdot s_2 + t_5$, and $\delta \leftarrow d + t_4$. Then, it computes

$\mu \leftarrow \tau - \delta \cdot \gamma$ and $h \leftarrow H_3(\zeta, \zeta_1, \mathbf{g}^{\rho}\mathbf{y}^{\omega}, \mathbf{g}^{\sigma_1}\zeta_1^{\delta}, \mathbf{h}^{\sigma_2}\zeta_2^{\delta}, \mathbf{z}^{\mu}\zeta^{\delta}, m)$. It returns the signature $\sigma \leftarrow (\zeta, \zeta_1, \omega, \delta, \rho, \sigma_1, \sigma_2, \mu)$ if $\delta + \omega = h$; otherwise, it returns $\perp$.

Verification. On input a public key pk, a signature $(\zeta, \zeta_1, \omega, \delta, \rho, \sigma_1, \sigma_2, \mu)$ and a message m, Verify computes first $h := H_3(\zeta, \zeta_1, \mathbf{g}^{\rho}\mathbf{y}^{\omega}, \mathbf{g}^{\sigma_1}\zeta_1^{\delta}, \mathbf{h}^{\sigma_2}\zeta_2^{\delta}, \mathbf{z}^{\mu}\zeta^{\delta}, m)$. It returns 1 if $\delta + \omega = h$; otherwise, it returns 0.

F.2 The deterministic wrapper

We describe the reduction strategy.

Restricting the hash queries to $\ell + 1$. For an adversary U that makes ℓ signing queries (i.e. closes ℓ signing sessions) and Q_h hash queries we make use of the same hash query guessing strategy as before, i.e. we use a wrapper M that restricts the adversary to exactly $\ell + 1$ hash queries and introduces a loss of $\frac{1}{\binom{Q_h}{\ell+}}$.

The Deterministic Wrapper We describe the inputs of the deterministic wrapper A First, we define $\mathbf{y}$-side instances, i.e. instances that use the $\mathbf{y}$-side witness x for simulation.

- $b = 0$
- $x \in \mathbb{Z}_q$
- $\mathbf{h} \in \mathbb{G}$
- $w_1, w_2 \in \mathbb{Z}_q$
- $\mathbf{z} = \mathbf{g}^{w_1} \cdot \mathbf{h}^{w_2}$
- for $i \in [\ell]$: $\mathrm{rnd}_i \in \{0,1\}^{\lambda}$
- for $i \in [\ell]$: $\mathbf{z}_{1,i} = \mathbf{g}^{w_{1,i}}$ with $w_{1,i} \in \mathbb{Z}_q$
- for $i \in [\ell]$: $u_i, d_i, s_{1,i}, s_{2,i} \in \mathbb{Z}_q$

We further describe the $\mathbf{z}$-side instances:

- $b = 1$
- $\mathbf{y} \in \mathbb{G}$
- $w, w_1, w_2 \in \mathbb{Z}_q$
- $w_0 \coloneqq w_1 + w_2 \cdot w \in \mathbb{Z}_q$
- for $i \in [\ell]$: $\mathrm{rnd}_i \in \{0,1\}^{\lambda}$
- for $i \in [\ell]$: $w_{1,i} \in \mathbb{Z}_q$ implicitly defines $w_{2,i} \coloneqq \frac{w_0 - w_{1,i}}{w}$
- for $i \in [\ell]$: $c_i, r_i, v_{1,i}, v_{2,i} \in \mathbb{Z}_q$

The wrapper additionally takes as input a set of random coins $\mathsf{rand} = (\mathsf{r_B}, \mathsf{r_A})$ as well as a vector of hash responses $\overrightarrow{h} \in {\mathbb{Z}_q}^{Q_h}$. We show the wrapper as pseudocode in fig. 8

Analogous to before, we can define forking tuples, partners, and triangles in an analogous way to before. We can use the same definition for the maximum branching index and apply the same counting arguments to count the set P_G and show that many of the tuples in P_G are triangle base corners.

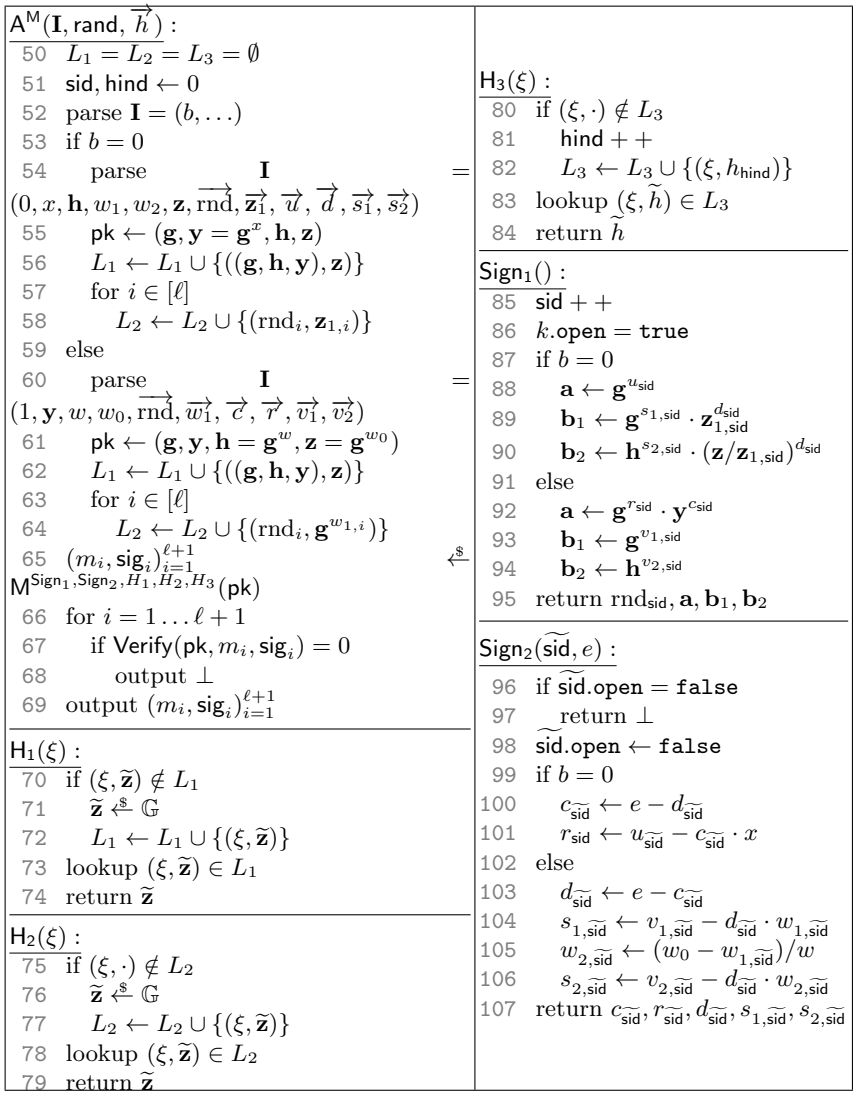


Fig. 8: Deterministic wrapper for Abe's blind signature scheme

F.3 The Transcript Mapping Function and Its Image

We describe the transcript mapping function Φ for Abe's scheme:

Definition 24 (Mapping Instances of Abe's scheme). *The transcript mapping function Φ is defined like this. For an instance $\mathbf{I} = (0, x, \mathbf{h}, w_1, w_2, \mathbf{z}, \overrightarrow{\mathrm{rnd}}, \overrightarrow{\mathbf{z}_1}, \overrightarrow{u}, \overrightarrow{d}, \overrightarrow{s}_1, \overrightarrow{s}_2)$, mapping $\Phi(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ with the query transcript $\overrightarrow{e}$ generated by running the wrapper $\mathsf{A}^{\mathsf{U}}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$ does the following:*

- $b \leftarrow 1$
- $\mathbf{y} \leftarrow \mathbf{g}^x$
- $w \leftarrow \mathrm{dlog}_{\mathbf{g}} \mathbf{h}$
- $w_1' \leftarrow w_1$
- $w_2' \leftarrow w_2$
- $\overrightarrow{\mathrm{rnd}}' \leftarrow \overrightarrow{\mathrm{rnd}}$
- $\forall i \in [\ell]\colon w_{1,i} \leftarrow \mathrm{dlog}_{\mathbf{g}} \mathbf{z}_{1,i}$
- $\forall i \in [\ell]\colon c_i \leftarrow e_i - d_i$
- $\forall i \in [\ell]\colon r_i \leftarrow u_i - c_i \cdot x$
- $\forall i \in [\ell]\colon v_{1,i} \leftarrow w_{1,i} \cdot d_i + s_{1,i}$
- $\forall i \in [\ell]\colon v_{2,i} \leftarrow w_{2,i} \cdot d_i + s_{2,i}$

For an instance $\mathbf{I} = (1, \mathbf{y}, w, w_1, w_2, \overrightarrow{\mathrm{rnd}}, \overrightarrow{w}_1, \overrightarrow{c}, \overrightarrow{r}, \overrightarrow{v_1}, \overrightarrow{v_2})$ the mapping does the following:

- $b \leftarrow 0$
- $x \leftarrow \mathrm{dlog}_{\mathbf{g}} \mathbf{y}$
- $\mathbf{h} \leftarrow \mathbf{g}^w$
- $w_1' \leftarrow w_1$
- $w_2' \leftarrow w_2$
- $\overrightarrow{\mathrm{rnd}}' \leftarrow \overrightarrow{\mathrm{rnd}}$
- $\forall i \in [\ell]\colon \mathbf{z}_{1,i} \leftarrow \mathbf{g}^{w_{1,i}}$
- $\forall i \in [\ell]\colon u_i \leftarrow c_i \cdot x + r_i$
- $\forall i \in [\ell]\colon d_i \leftarrow e_i - c_i$
- $\forall i \in [\ell]\colon s_{1,i} \leftarrow v_{1,i} - d_i \cdot w_{1,i}$
- $\forall i \in [\ell]\colon s_{2,i} \leftarrow v_{2,i} - d_i \cdot w_{2,i}$

Analogously to before, we can show that Φ is a self-inverse bijection that preserves the partner relation. We can then lower-bound the sizes of relevant sets in the image of Φ to finally show that there is a large enough set of both-sided triangle corners.

Using the analogous definition of $B_T^{\times}$ and $\widehat{B_T^{\times}}$, as well as $\widehat{O_T^{\times}}$, we can obtain that there must be a "good set" G for which forking is likely to result in the desired witness.

F.4 Forking Reduction

Theorem 3 (OMUF security of Abe's scheme). *For all $\ell \in \mathbb{N}$, if there exists an adversary U that makes Q_h hash queries to random oracle H_3 and $(t_{\mathsf{U}}, \epsilon_{\mathsf{U}}, \ell)$-breaks* $\mathbf{OMUF}_{\mathsf{Abe}}$ *with $\epsilon_{\mathsf{U}} \geq \frac{432\left(1-\frac{1}{(\ell+1)^2}\right)}{q} \cdot \binom{Q_h}{\ell+1}$, then there exists an algorithm B that $\left(t_{\mathsf{B}} \approx 2t_{\mathsf{U}} + \mathrm{O}(\ell+1) + \mathrm{O}({Q_h}^2), \epsilon_{\mathsf{B}} \approx \frac{3\epsilon_{\mathsf{U}}^2}{75423744 \cdot \binom{Q_h}{\ell+1}^2 \cdot (\ell+1)^3}\right)$-breaks* **DLOG**.

Proof. We give a sketch of the main parts of the proof that work slightly different from the AO scheme.

We describe the reduction R. On input of a discrete logarithm challenge $\mathbf{U}$, R first samples a bit b. If $b = 0$ it samples a $\mathbf{y}$-side instance $\mathbf{I}$ with $\mathbf{h} = \mathbf{U}$, otherwise it samples a $\mathbf{z}$-side instance $\mathbf{I}$ with $\mathbf{y} = \mathbf{U}$. It furthermore samples rand and a hash vector $\overrightarrow{h} \xleftarrow{\$} {\mathbb{Z}_q}^{\ell+1}$. It then runs the wrapper $\mathsf{B}^{\mathsf{M}}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h})$. Where M is the hash query reduction wrapper around U as described in the previous subsection.

R then samples $i \xleftarrow{\$} [\ell+1]$ and re-samples $\overrightarrow{h}' \xleftarrow{\$} {\mathbb{Z}_q}^{\ell+1}_{|\overrightarrow{h}_{[i]}}$. The reduction R then re-runs $\mathsf{B}^{\mathsf{A}}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}')$.

We denote by $(\zeta_i, \zeta_{1,i}, \omega_i, \delta_i, \rho_i, \sigma_{1,i}, \sigma_{2,i}, \mu_i)$ and $(\zeta'_i, \zeta'_{1,i}, \omega'_i, \delta'_i, \rho'_i, \sigma'_{1,i}, \sigma'_{2,i}, \mu'_i)$ the signature at hash index i in the first and second run respectively. If both runs are successful and produced a signature for index i, the reduction attempts to solve its discrete logarithm challenge as follows:

If the reduction chose $b = 0$: First compute

$$\mathrm{dlog}_{\mathbf{g}} \zeta_{1,i} = \mathrm{dlog}_{\mathbf{g}} \zeta'_{1,i} = \frac{\sigma'_{1,i} - \sigma_{1,i}}{\delta_i - \delta'_i}$$

and

$$\mathrm{dlog}_{\mathbf{h}} \zeta_{2,i} = \mathrm{dlog}_{\mathbf{h}} \zeta'_{2,i} = \frac{\sigma'_{2,i} - \sigma_{2,i}}{\delta_i - \delta'_i}$$

and

$$\mathrm{dlog}_{\mathbf{z}} \zeta_i = \mathrm{dlog}_{\mathbf{z}} \zeta'_i = \frac{\mu'_i - \mu_i}{\delta_i - \delta'_i}.$$

Then, compute

$$\mathrm{dlog}_{\mathbf{g}} \mathbf{z}_1^* = \frac{\mathrm{dlog}_{\mathbf{g}} \zeta_{1,i}}{\mathrm{dlog}_{\mathbf{z}} \zeta_i}$$

and

$$\mathrm{dlog}_{\mathbf{h}} \mathbf{z}_2^* = \frac{\mathrm{dlog}_{\mathbf{h}} \zeta_{2,i}}{\mathrm{dlog}_{\mathbf{z}} \zeta_i}$$

to finally obtain

$$\mathrm{dlog}_{\mathbf{g}} \mathbf{h} = \frac{w_1 - \mathrm{dlog}_{\mathbf{g}} \mathbf{z}_1^*}{\mathrm{dlog}_{\mathbf{h}} \mathbf{z}_2^* - w_2}$$

If the reduction chose $b = 1$ it merely computes

$$\mathrm{dlog}_{\mathbf{g}}\,\mathbf{y} = \frac{\rho_i' - \rho_i}{\omega_i' - \omega_i}.$$

We note that if $b = 0$, extraction works if $\delta_i \neq \delta_i'$ and if $\mathrm{dlog}_{\mathbf{h}}\,\mathbf{z}_2^* \neq w_2$. We denote the event that $\mathrm{dlog}_{\mathbf{h}}\,\mathbf{z}_2^* = w_2$ with F. We further note that this event is only dependent on the first run, as this run fixes $\zeta_i, \zeta_{1,i}$, and $\zeta_{2,i}$ already (it is only that the reduction needs the second run to compute)

In particular, this event can occur and is well-defined regardless of whether $b = 0$ or $b = 1$, as w_2 is also included in the $b = 1$ instance type, it is just not relevant for extraction in that case.

Claim 1.

$$\Pr\left[F \,\middle|\, \begin{array}{c}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \widehat{G}_b \\ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in T^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \\ t(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = i\end{array}\right] \leq \frac{\ell + 1}{q}$$

Proof. This follows from the fact that w_1, w_2 are information-theoretically hidden from the adversary even in forking runs. □

In the case that $b = 1$, the reduction succeeds in solving its dlog challenge if $\omega_i \neq \omega_i'$.

We can thus apply a similar analysis as for the Abe-Okamoto scheme and obtain the following.

$$\begin{aligned}
\mathrm{adv}^{\mathbf{DLOG}}_{\mathsf{R}} &\geq \left(\frac{1}{4} - \frac{1}{2q}\right) \cdot \Pr\left[\begin{array}{c}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \widehat{G}_b \\ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in T^{\times}_{T,i}\mathbf{I}, \mathsf{rand}, \overrightarrow{h} \\ t(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = i \\ \neg F\end{array}\right] \\
&= \left(\frac{1}{4} - \frac{1}{2q}\right) \cdot \Pr\left[\neg F \,\middle|\, \begin{array}{c}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \widehat{G}_b \\ (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in T^{\times}_{T,i}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \\ t(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = i\end{array}\right] \\
&\quad \cdot \Pr\left[(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}') \in T^{\times}_{T,i}\mathbf{I}, \mathsf{rand}, \overrightarrow{h} \,\middle|\, \begin{array}{c}(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \widehat{G}_b \\ t(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = i\end{array}\right] \\
&\quad \cdot \Pr\left[(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \widehat{G}_b\right] \cdot \Pr\left[t(\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) = i \,\middle|\, (\mathbf{I}, \mathsf{rand}, \overrightarrow{h}) \in \widehat{G}_b\right] \\
&\geq \left(\frac{1}{4} - \frac{1}{2q}\right) \cdot \left(1 - \frac{\ell + 1}{q}\right) \left(\frac{\epsilon_{B_T^{\times}}}{16(\ell + 1)} - \frac{2}{q}\right) \cdot \frac{3\epsilon_{B_T^{\times}}}{128(\ell + 1)} \cdot \frac{1}{\ell + 1}
\end{aligned}$$

(where the last inequality is due to Claim 1, Lemma 10, and Lemma 11). Plugging in $\epsilon_{B_T^{\times}} \geq \frac{\epsilon_{\mathsf{M}}}{96}$ for $\epsilon_{\mathsf{M}} \geq \frac{432\left(1 - \frac{1}{(\ell+1)^2}\right)}{q}$ and $\epsilon_{\mathsf{M}} = \frac{\epsilon_{\mathsf{U}}}{\binom{Q_h}{\ell+1}}$ (see Lemma 7) yields the theorem statement. □

Discussion. We note that Abe's scheme is believed to be immune to the ROS-attack and thus the bounds induced by the recent polynomial-time ROS-solver [8] do not apply to this scheme. This means that unlike for the Abe-Okamoto scheme, the gap between our proof of security and the best possible bound one could hope for is rather large. In fact, Abe's scheme can be proven secure in the AGM for *polynomially many concurrent* signing sessions [26] whereas our proof of security only allows for a fairly small number of signing sessions. It therefore remains an interesting open question whether this gap can be bridged using the random oracle model alone.