Black-Box Knowledge Extraction Revisited*

Universal Approach with Precise Bounds

Emilia Käsper[1], Sven Laur[2], and Helger Lipmaa[3]

[1] Katholieke Universiteit Leuven, Belgium
[2] Helsinki University of Technology, Finland
[3] University College London, UK

Abstract. Rewinding techniques form the essence of many security reductions including proofs for identification and signature schemes. We propose a simple and modular approach for the construction of such proofs. Straightforward applications of our central result include, but are not limited to, the security of identification schemes, generic signatures and ring signatures. These results are well known, however, we generalise them in such a way that our technique can be used off-the-shelf for future applications. We note that less is more: as a side-effect of our less complex analysis, all our proofs are more precise; for example, we get a new proof of the forking lemma that is 2^{15} times more precise than the original result by Pointcheval and Stern. Finally, we give the first precise security analysis of Blum's coin flipping protocol with k-bit strings, as yet another example of the strength of our results.

Keywords. Coin flipping, forking lemma, generic signature schemes, matrix algorithms, proofs of knowledge, special soundness.

1 Introduction

Many security proofs incorporate complicated black-box rewinding techniques to force certain behaviour from the adversarial algorithm. Most notably, rewinding is the core part of protocol soundness proofs, for example, it is used to prove the security of many signature schemes. Here, the task is to somehow force the adversary to reveal the secret key, thus showing that nobody can create a valid signature without knowing the secret. In this paper, our goal is to clarify different black-box rewinding techniques and provide simple and exact proofs. We provide a universal approach together with sharp bounds on running times and success probabilities of rewinding algorithms. Our main contribution is a straightforward off-the-shelf technique that is applicable in many contexts.

To provide a gentle introduction, we first explain our approach on simple examples and then gradually generalise until we have covered all target rewinding scenarios. As an introductory example, we consider the security of Schnorr identification protocol [Sch91].

Then, we move on to more sophisticated rewinding techniques and prove the security of generic signature schemes. The corresponding result is already known as the forking lemma [PS00]; on the other hand, our approach provides a straightforward and natural proof with better time bounds. We prove one central result (Theorem 3) that allows us to immediately conclude security results for generic signatures and generic ring signatures. We avoid both the complex transform from strict non-uniform time to expected uniform time, and the use of imprecise auxiliary combinatorial results such as the so-called "splitting lemma". Although efficiency gain is not the main advantage of our approach, it is worth noting that our results are nearly

* *** Draft, October 20, 2006 ***

optimal for the best known reduction technique (forking): a more specialised algorithm can be less than two times faster. Thus, we sacrifice almost no efficiency for the sake of simplicity.

In the second part of this paper, we address the issue of *soundness* versus *security*. Namely, the results obtained so far consider the security of a protocol w.r.t. a fixed key. In this context it is not possible to obtain a true reduction to the underlying hard problem, say, the discrete logarithm problem, since such problems are known to be hard only on average. We resolve this issue by proving the second central result (Theorem 8) which quantifies the success probability of the rewinding algorithm, taken over the random choice of the secret. Our result allows trade-offs between the running time and the success probability ε of the adversary; for the best trade-off point, the time-success ratio of our reduction is a factor $\sqrt{\varepsilon}$ better than previous results.

Finally, we show the strength of our results by applying them in a completely different context. Our final target is the security analysis of Blum's coin flipping protocol [Blu81], which plays a central role in many zero-knowledge proofs: it allows to guard honest-verifier zero-knowledge protocols against malicious verifiers. The protocol uses computationally secure string commitments to guarantee the randomness of the protocol output. However, security properties of Blum's protocol are not directly implied by classic definitions of binding and hiding if k-bit string commitments are used instead of bit commitments. We again use rewinding techniques to reduce the randomness guarantee on the protocol output to the binding property of the commitment scheme. To sum up, we remark that similar collision finding problems arise in different areas including timestamping [BS04,BL06] and manual authentication protocols [Vau05b,PV06,LN06].

Road Map. In Sect. 2, we introduce basic notions needed to analyse the security of generic signature schemes. Sect. 3 is devoted to rewinding techniques. Our starting point is the security of Schnorr identification protocol (Sect. 3.1). In Sect. 3.2, we prove our first main result (Thm. 3) and in the next subsection, we give two examples on how to apply the theorem to specific protocols such as signature schemes. We then make an intermediate summary of our results. In Sect. 4, we fill the gap between soundness and security and provide means to reduce the security of the protocol to the underlying hard problem (Thm. 8). As an additional treat, we apply the result to prove the security of Blum's coin flipping protocol. Finally, Sect. 5 discusses some open problems. Many technical results are included as appendices, since they are straightforward but tedious to prove. We encourage to read them, as they give formal proofs to many intuitively understandable claims.

2 Preliminaries

Notation. For a positive integer i, let $[i] = \{1, \ldots, i\}$. For an $(s+1)$-dimensional array A, let $\mathsf{A}(\omega_0, \ldots, \omega_s)$ denote its corresponding element. Let $\sharp\mathcal{S}$ be the cardinality of set $\mathcal{S}$. Finally, to avoid common confusion about convex and concave functions, we call a function f convex-cup if $f(\frac{x+y}{2}) \leq \frac{f(x)+f(y)}{2}$ and convex-cap if $f(\frac{x+y}{2}) \geq \frac{f(x)+f(y)}{2}$.
Schnorr Identification Protocol. In the Schnorr identification protocol, a user $\mathcal{P}$ proves to a verifier $\mathcal{V}$ that she knows the discrete logarithm of a public key $y = g^x$ where g is a generator of a q-element group $\mathbb{G}$. The corresponding protocol is the following: $\mathcal{P}$ chooses $a \leftarrow \mathbb{Z}_q$ and sends $\alpha = g^a$ to $\mathcal{V}$. Verifier $\mathcal{V}$ sends a challenge $\beta \leftarrow \mathbb{Z}_q$ and $\mathcal{P}$ replies with $\gamma = a + \beta x \pmod q$. Verifier $\mathcal{V}$ accepts the transcript if $g^\gamma = g^{a+\beta x} = \alpha \cdot y^\beta$.
Special Soundness and Knowledge Extraction. The security of a protocol is proved by showing that if a prover can provide accepting transcripts, then she indeed knows the secret

x. In the case of Schnorr identification, it is straightforward to show that given two accepting transcripts $(\alpha, \beta_1, \gamma_1)$ and $(\alpha, \beta_2, \gamma_2)$ with $\beta_1 \neq \beta_2$, we can compute $x = (\beta_2 - \beta_1)^{-1}(\gamma_2 - \gamma_1) \pmod q$. In general, such a property is known as *special soundness*. If a prover $\mathcal{P}$ succeeds with probability higher than $1/q$, then the complete listing of all protocol transcripts contains such colliding triples and these collisions can be found efficiently by rewinding $\mathcal{P}$ with different challenges β; this search is one example of *knowledge extraction* techniques.

Fiat-Shamir Heuristic and Schnorr Signatures. The interactive challenge-response identification protocol described above can be converted into a non-interactive signature scheme using the so-called Fiat-Shamir heuristic [FS86]: the prover computes the random challenge of the verifier herself by means of a one-way hash function h. In particular, Schnorr signatures are quadruples $(m, \alpha, \beta, \gamma)$ where $\beta = h(m, \alpha)$ and (α, β, γ) is an accepting transcript of the corresponding identification scheme with public key $\mathsf{pk} = y$ and secret key $\mathsf{sk} = x$. Heuristically, if the hash function is "cryptographically strong", then substituting β with $h(m, \alpha)$ should not significantly decrease security.

Generic Signature Schemes and the Random Oracle Model. Many important signature schemes can be formalised similarly; the corresponding generalisation is known as a *generic signature scheme*. Namely, a generic signature is a tuple $(m, \alpha, \beta, \gamma)$, where $m \in \mathcal{M}$ is the signed message, α is a random nonce chosen uniformly from some set $\mathcal{D}$, $\beta = h(m, \alpha) \in \mathcal{T}$ for some fixed function h, and γ is a verification value that is computed from m, α and h using the secret key sk. Verification consists of two steps: (a) the validity of γ is verified by evaluating a pk-dependent predicate $\mathsf{Verify}_{\mathsf{pk}}(m, \alpha, \beta, \gamma)$; (b) the connection between the signature and the message is assured by testing that $\beta \stackrel{?}{=} h(m, \alpha)$. However, a more formal security analysis is accessible only in the random oracle model [BR93], where h is modelled as a black-box function chosen uniformly from the set of all functions $h : \mathcal{M} \times \mathcal{D} \to \mathcal{T}$. To compute $h(m, \alpha)$, one has to make explicit oracle calls.

Security of Signature Schemes. A signature scheme is secure, if it is infeasible to obtain a signature without knowledge of the secret key (a *forgery*). It is important to distinguish existential and universal forgeries. In the case of *universal forgery*, an adversary must produce a valid signature for a given message m, whereas for *existential forgery*, it is sufficient to produce a single but new valid signature. A signature scheme is (t, ε)-secure against existential forgeries if any t-time adversary succeeds in existential forgery with probability less than ε, where the probability is taken over the coin tosses of all relevant algorithms including the key generation algorithm.

Generic Ring Signature Schemes. In a *ring signature scheme*, any user belonging to a ring can compute a signature on behalf of the entire ring, using only her private key and the public keys of other members. An example of such a scheme is an extension of the single-user Schnorr signature scheme [HS03]. However, we omit here the details how ring signatures are constructed and only describe the generic model. A generic ring signature in a ring of n members is a tuple $(m, \boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma)$, where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ such that $\alpha_i \neq \alpha_j$ for $i \neq j$; and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_n)$ such that $\beta_i = h(m, \alpha_i)$. The special soundness property of ring signatures then translates to the following: given two valid signatures $(m, \boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma)$, $(m, \boldsymbol{\alpha}, \boldsymbol{\beta}', \gamma')$ such that $\beta_i \neq \beta'_i$ for exactly one index $i \in [n]$, we can compute the secret key of one ring member. Schnorr ring signatures are specially sound.

3 Black-Box Knowledge Extraction

It is common to model knowledge extraction as a randomised search. In this section, we propose a relatively simple randomised algorithm Rewind (Alg. 1 and 2) that provides solutions to many knowledge extraction tasks. Recall that the idea behind knowledge extraction is to find related protocol transcripts that reveal the secret. Thus, we abstract away from protocol details as much as possible and simply label protocol transcripts in such a way that two random transcripts with coinciding labels allow to extract the secret with high probability. Moreover, as a transcript is completely determined by the random choices of the adversarial prover $\mathcal{A}$ and the verifier $\mathcal{V}$, we may identify each transcript with the underlying randomness. Throughout this paper, we denote by $\boldsymbol{\omega}$ the randomness used in the protocol, and by $\mathsf{A}(\boldsymbol{\omega})$ the label on the corresponding transcript.

For example, in the case of Schnorr identification scheme, $\boldsymbol{\omega} = (r, c)$ represents the random choices of the adversarial prover and the verifier, respectively. We set $\mathsf{A}(\boldsymbol{\omega}) = 1$ if the transcript is accepting and $\mathsf{A}(\boldsymbol{\omega}) = 0$ otherwise, so A is simply a binary matrix. Then we can skip implementation details and we are left with the problem of finding two ones in the same row of matrix A. In the general case, we allow an arbitrary set of labels, but we always reserve the label 0 to denote failure. Then, if $\mathcal{A}$ achieves advantage ε, an ε-fraction of transcripts $\boldsymbol{\omega}$ have nonzero labels $\mathsf{A}(\boldsymbol{\omega}) \neq 0$.

Rewinding. In a nutshell, knowledge extraction techniques considered in this paper run the adversarial prover multiple times with different challenges from the verifier; or experiment with different random oracles in the non-interactive case. More formally, let $\boldsymbol{\omega} = (\omega_0, \ldots, \omega_s)$ denote the random variables as they are requested in the protocol. Then, *rewinding* is a search strategy where $\omega_0, \ldots, \omega_{i-1}$ are fixed but the remaining values are altered; index i is the corresponding *rewinding point.* The crucial observation is that if we fix $\omega_0, \ldots, \omega_{i-1}$, then all protocol messages sent before some participant requests ω_i are the same, regardless of the values of $\omega_i, \ldots, \omega_s$. Obviously, different knowledge extraction problems require different labelling, search and rewinding criteria. Therefore, the general form of Rewind is quite complex for the reader that does not know enough context. Hence, we derive the corresponding algorithm gradually starting from simple examples and introduce new ideas until we reach the final form.

3.1 Security of the Schnorr Identification Protocol

In order to familiarise the reader with our concept, we start by proving the security of Schnorr identification protocol. The result itself is not novel, but it helps us to introduce the notation. Recall that the security of Schnorr identification protocol hinges on the special soundness property: given two accepting transcripts $(\alpha, \beta_1, \gamma_1)$ and $(\alpha, \beta_2, \gamma_2)$ with $\beta_1 \neq \beta_2$, we can efficiently derive the secret key x. Now, let $\mathcal{A}$ be a τ-time adversary that achieves advantage ε against the scheme, i.e., $\mathcal{A}$ acting as a prover manages to create α, γ so that an honest verifier $\mathcal{V}$ accepts the transcript with probability ε. Let r be the randomness used by $\mathcal{A}$ and let $c = \beta$ be the randomness used by $\mathcal{V}$. Then we can label all protocol transcripts by 0 if $\mathcal{V}$ does not accept and 1 otherwise. Let A be the corresponding matrix with entries $\mathsf{A}(r, c)$. W.l.o.g. we can assume that $r \in [m]$ and $c \in [n]$ are uniformly distributed. Then it is trivial to note that for fixed r, $\mathcal{A}$

Algorithm $\mathsf{Rewind\text{-}Basic}(\mathsf{A})$:

1. Probe random entries $\mathsf{A}(r, c)$ until $\mathsf{A}(r, c) \neq 0$. Store $\boldsymbol{\omega} = (r, c)$.
2. Fix the row r. Probe random entries $\mathsf{A}(r, c')$ in that row until $\mathsf{A}(r, c') \neq 0$. Store $\boldsymbol{\omega}' = (r, c')$.
3. Try to extract the secret from $\boldsymbol{\omega}$, $\boldsymbol{\omega}'$.
   – If the extraction is successful return the secret x else return $\perp$.

Algorithm $\mathsf{Rewind\text{-}Basic\text{-}Exp}(\mathsf{A})$:

- Repeat $\mathsf{Rewind\text{-}Basic}(\mathsf{A})$ until it succeeds and returns x.

Algorithm 1: Rewinding algorithm Rewind-Basic and its extension Rewind-Basic-Exp.

always outputs the same message α and we have to find two different non-zero entries $\mathsf{A}(r, c)$, $\mathsf{A}(r, c')$ in the same row[1]. The most obvious way to search such elements is the following:

– Probe random entries of $\mathsf{A}(r, c)$ until $\mathsf{A}(r, c) = 1$.
– Probe random entries $\mathsf{A}(r, c')$ in the same row until $\mathsf{A}(r, c') = 1$.
– Extract secret x using r, c, c' if possible.

By random probing we mean that $r \in [m]$ and $c \in [n]$ are chosen uniformly. In particular, we do not require $c \neq c'$ for otherwise this algorithm would never finish if it stumbled upon a row with only one accepting transcript. This implies that the basic algorithm always has failure probability $p_{\text{fail}} > 0$, so we may have to repeat the process.

The two algorithms are formalised as Algorithms Rewind-Basic and Rewind-Basic-Exp (see Alg. 1). Probing one entry in matrix A corresponds to a single execution of the protocol. To probe a second entry in the same row, we have to rewind $\mathcal{A}$ to the point where it receives β. Let probes_1 and probes_2 denote the number of probed elements during Step 1 and Step 2. First, we compute the expected running time of Rewind-Basic. Note that by construction, ε denotes the success probability of $\mathcal{A}$.

Lemma 1. *For any matrix* A *with an* ε*-fraction of nonzero entries,* Rewind-Basic *makes on average* $\mathbf{E}[\mathsf{probes}_1] = 1/\varepsilon$ *probes in the first and* $\mathbf{E}[\mathsf{probes}_2] \leq 1/\varepsilon$ *probes in the second step.*

Proof. Let A be an $m \times n$ matrix and let $\mathsf{nz}(r)$ be the number of nonzero entries in its rth row, so $\varepsilon mn = \mathsf{nz}(1) + \cdots + \mathsf{nz}(m)$. Since Step 1 and Step 2 sample elements until the first success (i.e., according to a binary distribution), we get $\mathbf{E}[\mathsf{probes}_1] = 1/\varepsilon$ and for any row r with $\mathsf{nz}(r) > 0$, $\mathbf{E}[\mathsf{probes}_2 | r] = n/\mathsf{nz}(r)$. Thus,

$$\mathbf{E}[\mathsf{probes}_2] = \sum_{r : \mathsf{nz}(r) > 0} \Pr[r] \cdot \mathbf{E}[\mathsf{probes}_2 | r] = \sum_{r : \mathsf{nz}(r) > 0} \frac{\mathsf{nz}(r)}{\varepsilon mn} \cdot \frac{n}{\mathsf{nz}(r)}$$
$$= \frac{1}{\varepsilon} \cdot \frac{\sharp \{r : \mathsf{nz}(r) > 0\}}{m} \leq \frac{1}{\varepsilon} \ .$$

□

[1] We could try to find colliding transcripts for different rows but then our chances are significantly smaller. In fact, the adversary $\mathcal{A}$ may output a different α for each row. Therefore, the search criterion is optimal for black-box reductions, as the search criterion must be independent of $\mathcal{A}$.

Failure probability and knowledge error. Although the transcripts found by $\mathsf{Rewind\text{-}Basic}$ are accepting by construction, we sometimes cannot extract the secret, for example if $\omega_1 = \omega_1'$. While this is the only possible cause of failure for the Schnorr protocol, in general there could be other failing tuples. Let $\mathsf{bad}(\boldsymbol{\omega}) = \mathsf{bad}(r, c)$ denote the number of vectors $\boldsymbol{\omega}' = (r, c')$ that lead to failure and let $\mathsf{nz}(\boldsymbol{\omega}) = \mathsf{nz}(r, c) = \sharp\{c' : \mathsf{A}(r, c') \neq 0\}$ be the number of accepting transcripts in a given row r. For the Schnorr protocol, $\mathsf{bad}(\boldsymbol{\omega}) = 1$, as extraction fails only if $c = c'$, but $\mathsf{nz}(\boldsymbol{\omega})$ depends on the adversary. Now, it is straightforward to compute the failure probability of $\mathsf{Rewind\text{-}Basic}$

$$p_{\mathrm{fail}}(\varepsilon) = \sum_{\mathsf{A}(\boldsymbol{\omega})\neq 0} \Pr\left[\boldsymbol{\omega} = (r, c)\right] \cdot \Pr\left[\mathsf{Rewind\text{-}Basic} = \perp | \boldsymbol{\omega}\right] = \frac{1}{\varepsilon m n} \cdot \sum_{\mathsf{A}(\boldsymbol{\omega})\neq 0} \frac{\mathsf{bad}(\boldsymbol{\omega})}{\mathsf{nz}(\boldsymbol{\omega})} \enspace ,$$

as the first tuple (r, c) is chosen uniformly among the nonzero entries of A. Let us define

$$\kappa = \frac{1}{mn} \cdot \max_{\mathsf{A}} \sum_{\mathsf{A}(\boldsymbol{\omega})\neq 0} \frac{\mathsf{bad}(\boldsymbol{\omega})}{\mathsf{nz}(\boldsymbol{\omega})} \enspace , \tag{1}$$

where the maximum is taken over all possible matrices for any ε. Then we get an upper bound on the failure probability $p_{\mathrm{fail}} \leq \kappa/\varepsilon$. For the Schnorr protocol, $\mathsf{bad}(\boldsymbol{\omega}) = 1$ and thus

$$\kappa = \frac{1}{mn} \cdot \max_{\mathsf{A}} \sum_{\mathsf{A}(\boldsymbol{\omega})=1} \frac{1}{\mathsf{nz}(\boldsymbol{\omega})} = \frac{1}{n} \enspace ,$$

as the last sum counts the number of rows with nonzero entries. Observe that a lower bound on κ is determined by the maximal fraction κ_0 of non-zero elements in A such that extraction always fails, i.e., there exists an adversary with success probability κ_0 that does not "know" the secret. For many problems, $\kappa_0 = \kappa$ and thus it is appropriate to call κ a *knowledge error*. Given κ, we can now estimate the average running-time of $\mathsf{Rewind\text{-}Basic\text{-}Exp}$.

Theorem 1. *For any matrix A with an ε-fraction of nonzero entries and for knowledge error κ, $\mathsf{Rewind\text{-}Exp}$ makes on average $\mathbf{E}[\mathsf{probes}] \leq 2/(\varepsilon - \kappa)$ if $\kappa < \varepsilon$. Asymptotically, the expected number of probes behaves $\mathbf{E}[\mathsf{probes}] \leq (2 + o(1))/\varepsilon$ in the process $\varepsilon/\kappa \to \infty$.*

Proof. As all runs of $\mathsf{Rewind\text{-}Basic}$ are independent and the success probability of a single finished run is $1 - p_{\mathrm{fail}}(\varepsilon)$, we get $\mathbf{E}[\mathsf{probes}] \leq (2/\varepsilon)/(1 - p_{\mathrm{fail}}(\varepsilon)) = 2/(\varepsilon - \kappa)$. As $2/(\varepsilon - \kappa) = 2 \cdot (1 + \kappa/(\varepsilon - \kappa))/\varepsilon$, the second claim follows. □

If the challenge space of Schnorr identification protocol has size $n = q$, then we say that $k = \log_2 q$ is the *security parameter* of the protocol. We conclude the following result.

Corollary 1. *Consider Schnorr identification protocol with security parameter k. Let $\mathcal{A}$ be a τ-time forger whose input consists only of public data. If $\mathcal{A}$ produces an accepting transcript with probability $\varepsilon > 2^{-k}$, then there exists a knowledge extractor which extracts the secret in expected time $t \leq 2\tau/(\varepsilon - 2^{-k})$.*

Lower bounds on average running-time. Note that Rewind-Exp discards some information as it can probe some entries more than once. Still, the next theorem shows that the expected number of probes, achieved by Rewind-Exp, is almost optimal when ε is reasonably large. Already for $\varepsilon > 3\kappa$, Rewind is less than a factor of 2 away from the optimal time bound.

Theorem 2. *For any black-box searching strategy $\mathcal{S}$, there exists a status matrix A with εmn nonzero entries such that $\mathcal{S}$ makes $\mathbf{E}[\mathsf{probes}] \geq 2(mn+1)/(\varepsilon mn+1) = 2(1-o(1))/\varepsilon$ probes.*

Proof. See App. A.

3.2 Rewinding in the General Case

In the previous example, the rewinding strategy was straightforward. In general, knowledge extraction can be more complex if there are many possible rewinding points and some choices lead to the execution of different sub-protocols. As an illustrative toy example, we consider an extension of Schnorr identification protocol where the prover has d chances to convince the verifier, i.e., d protocol instances are run sequentially and the verifier accepts if at least one sub-protocol leads to acceptance. Such a setting is closely connected to Schnorr signatures but is somewhat easier to grasp. Now, the protocol transcript is a tuple $(\alpha_1, \beta_1, \gamma_1, \ldots, \alpha_d, \beta_d, \gamma_d)$ and we have to find two accepting sub-transcripts $(\alpha_i, \beta_i, \gamma_i)$ and $(\alpha_i, \beta_i', \gamma_i')$ such that $\beta_i \neq \beta_i'$ for some $i \in \{1, \ldots, d\}$.

Let $\boldsymbol{\omega} = (\omega_0, \omega_1, \ldots, \omega_d)$, where ω_0 is the randomness used by the malicious prover $\mathcal{A}$ and ω_i is the ith challenge β_i. Clearly, it makes sense to group accepting transcripts into equivalence classes. Let $\mathsf{A}(\boldsymbol{\omega}) = i$ if the ith sub-transcript is the first accepting transcript and $\mathsf{A}(\boldsymbol{\omega}) = 0$ if no transcripts are accepting. To find suitable transcripts $(\alpha_i, \beta_i, \gamma_i)$ and $(\alpha_i, \beta_i', \gamma_i')$ we have to fix all random coins that are used before β_i is queried. The latter leads us to the following simple algorithm:

- Probe random entries of $\mathsf{A}(\boldsymbol{\omega})$ until $\mathsf{A}(\boldsymbol{\omega}) \neq 0$. Set $i = \mathsf{A}(\boldsymbol{\omega})$.
- Probe random entries $\mathsf{A}(\boldsymbol{\omega}')$ with $\omega_0 = \omega_0', \ldots, \omega_{i-1} = \omega_{i-1}'$ until $\mathsf{A}(\boldsymbol{\omega}) = \mathsf{A}(\boldsymbol{\omega}')$.
- Restore protocol transcripts and extract secret x using $\boldsymbol{\omega}$ and $\boldsymbol{\omega}'$.

Moreover, the above algorithm can be generalised to any protocol that satisfies the following two requirements:

1. Transcripts can be labelled in such a way that two random transcripts with coinciding labels allow to extract the secret with high probability.
2. For all transcripts with the same label the set of reasonable rewinding points is the same.

Indeed, as we shall see in several examples, finding a proper labelling is the crucial point in proving the desired result. The formalisation of the above idea is given as Alg. 2. Note that all proofs trivially generalise to the case where the the rewinding point is chosen at random from a set of reasonable candidates. For our example protocol, the rewinding point coincides with the label: $f(a) = a$. We also emphasize that all proofs go through if the adversary halts with success or failure before receiving all d challenges; or if the number of random bits she queries at any time is unknown ahead of time. In this case, A is not an array, rather it is just a tree which is not necessarily balanced; such a generalisation is considered in [Vau05a, p. 270]. Nevertheless,

Algorithm $\mathsf{Rewind}(\mathsf{A})$:

1. Probe random entries $\mathsf{A}(\boldsymbol{\omega})$ until $\mathsf{A}(\boldsymbol{\omega}) \neq 0$.
   Store the corresponding label $a = \mathsf{A}(\boldsymbol{\omega})$. Fix the rewinding point $i = f(a)$.
   Set $r = (\omega_0, \dots, \omega_{i-1})$ and $c = (\omega_i, \dots, \omega_d)$.
2. Probe random entries $\mathsf{A}(\boldsymbol{\omega}') = \mathsf{A}(r, c')$ until labels coincide $\mathsf{A}(\boldsymbol{\omega}') = \mathsf{A}(\boldsymbol{\omega})$.
3. Try to extract the secret from $\boldsymbol{\omega} = (\omega_0, \dots, \omega_{i-1}, \omega_i, \dots, \omega_d)$ and $\boldsymbol{\omega}' = (\omega_0, \dots, \omega_{i-1}, \omega_i', \dots, \omega_d')$.
   – If reconstruction is successful return the secret x else return $\perp$.

Algorithm $\mathsf{Rewind\text{-}Exp}(\mathsf{A})$:

- **repeat** $\mathsf{Rewind}(\mathsf{A})$ **until** succeeds and returns x.

Algorithm 2: Rewinding algorithm Rewind and its extension Rewind-Exp.

we can (a) fix the length of the queries by filling the gaps with random bits[2]; and (b) complete the tree with random values so that it is balanced. We are now back at the original situation.

Lemma 2. *Let $\{0, \dots, d\}$ be the set of labels. For any array* A *with an ε-fraction of nonzero entries,* Rewind *takes on average $\mathbf{E}[\mathsf{probes}] \leq (d+1)/\varepsilon$ probes.*

Proof. Fix a label $a > 0$ and consider Step 2 under the constraint $\mathsf{A}(\boldsymbol{\omega}) = a$. As $\boldsymbol{\omega}$ is chosen uniformly from accepting transcripts, then $\boldsymbol{\omega}$ is chosen uniformly under the condition $\mathsf{A}(\boldsymbol{\omega}) = a$ and the algorithm behaves exactly like Rewind-Basic on the matrix where all entries $\mathsf{A}(\boldsymbol{\omega}) \neq a$ are set to zeroes. Hence, Lemma 1 yields $\mathbf{E}[\mathsf{probes}_2 | \mathsf{A}(\boldsymbol{\omega}) = a] \leq 1/\varepsilon_a$ where ε_a is the fraction of a-labelled entries in A and

$$\mathbf{E}[\mathsf{probes}_2] = \sum_{a=1}^{d} \Pr\left[\mathsf{A}(\boldsymbol{\omega}) = a\right] \cdot \mathbf{E}[\mathsf{probes}_2 | \mathsf{A}(\boldsymbol{\omega}) = a] \leq \sum_{a=1}^{d} \frac{\varepsilon_a}{\varepsilon} \cdot \frac{1}{\varepsilon_a} = \frac{d}{\varepsilon} \enspace .$$

As $\mathbf{E}[\mathsf{probes}_1] = 1/\varepsilon$, the claim follows. □

Aggregated knowledge error. The failure probability $p_{\mathrm{fail}}(\varepsilon)$ of Rewind is also averaged over different labels. Let $p^{a}_{\mathrm{fail}}(\varepsilon_a)$ denote the failure probability of Rewind-Basic in the matrix where there are ε_a-fraction of nonzero entries and the rewinding point is chosen as $f(a)$. Let κ_a be the corresponding knowledge error. Then the summary failure probability of Rewind is

$$p_{\mathrm{fail}}(\varepsilon) = \max_{\varepsilon_1 + \dots + \varepsilon_d = \varepsilon} \left\{ \sum_{a=1}^{d} \frac{\varepsilon_a}{\varepsilon} \cdot p^{a}_{\mathrm{fail}}(\varepsilon_a) \right\} \leq \sum_{a=1}^{d} \frac{\varepsilon_a}{\varepsilon} \cdot \frac{\kappa_a}{\varepsilon_a} \leq \frac{\kappa_1 + \dots + \kappa_d}{\varepsilon} \enspace .$$

In other words, we can decompose the complex analysis of initial failure probabilities into simple sub-cases. Let $\kappa = \kappa_1 + \dots + \kappa_d$ be the *aggregated knowledge error.* Now we can state the first central theorem for knowledge extraction.

Theorem 3 (Dynamic Search). *Let $\{0, \dots, d\}$ be the set of labels and let κ be the aggregated knowledge error. For any array* A *with an ε-fraction of nonzero entries,* Rewind-Exp *makes on average $\mathbf{E}[\mathsf{probes}] \leq (d+1)/(\varepsilon - \kappa)$ probes if $\varepsilon > \kappa$.*

[2] If there is no other bound, the number of random bits queried is certainly bounded by the running time of the adversary.

Proof. The proof coincides with the proof of Thm. 1. □

We conclude this subsection by showing how to compute the aggregated knowledge error in the simple case where all transcripts $\boldsymbol{\omega}$, $\boldsymbol{\omega}'$ such that $\mathsf{A}(\boldsymbol{\omega}) = \mathsf{A}(\boldsymbol{\omega}')$ and $\omega_i \neq \omega_i'$ (where i is the rewinding point) allow to extract the secret. For this, we need to compute the knowledge error κ_a for each submatrix with only a-labelled nonzero entries. Let $\omega_0 \in [m]$ and $\omega_i \in [2^k]$ for $1 \leq i \leq d$. For $\boldsymbol{\omega} = (\omega_0, \dots, \omega_d)$ with rewinding point $f(a) = i$, extraction fails only if $\omega_i' = \omega_i$, so we get $\mathsf{bad}(\boldsymbol{\omega}) = \sharp \left\{ \boldsymbol{\omega}' = (\omega_0, \dots, \omega_{i-1}, \omega_i, \omega_{i+1}' \dots, \omega_d') \right\} = 2^{(d-i)k}$, so

$$\kappa_a = \frac{1}{m2^{dk}} \cdot \max_{\mathsf{A}} \sum_{\mathsf{A}(\boldsymbol{\omega})=a} \frac{\mathsf{bad}(\boldsymbol{\omega})}{\mathsf{nz}(\boldsymbol{\omega})} = \frac{2^{(d-i)k}}{m2^{nk}} \cdot \max_{\mathsf{A}} \sum_{\mathsf{A}(\boldsymbol{\omega})=a} \frac{1}{\mathsf{nz}(\boldsymbol{\omega})} = 2^{-k} \quad ,$$

as the last sum counts rows with a-labelled entries and there are $m2^{(i-1)k}$ rows. Hence, the aggregated knowledge error is

$$\kappa = \kappa_1 + \ldots + \kappa_d = d \cdot 2^{-k} \quad . \tag{2}$$

3.3 Universal Proofs for Forking-Lemma Type Knowledge Extractors

Security of Generic Signature Schemes. Next, we turn our attention to signature schemes. Our results from the previous section allow us to obtain a straightforward proof of the infamous forking lemma [PS00] that provides the security of generic signature schemes in the random oracle model. We are going to (re)prove the strongest result, namely, security against existential forgery. That is, we show that there exists no efficient adversary who is capable of producing a valid new signature on *any* message m of her choice without knowing the secret x. Let $\mathcal{A}$ be an adversary that tries to output a forged signature. To do so, she is allowed to query q_h hash queries $h(m_i, \alpha_i)$ from the random oracle before outputting a signature[3] $(m, \alpha, \beta, \gamma)$. The goal of the knowledge extractor is then to rewind $\mathcal{A}$ to produce two valid signatures $(m, \alpha, \beta, \gamma)$ and $(m, \alpha, \beta', \gamma')$ such that $\beta \neq \beta'$, since this allows to extract the secret x. Analogously to identification schemes, we say that k is the security parameter of the scheme if $k = \log_2 |\mathcal{T}|$ quantifies the size of the hash function tag space $\mathcal{T}$. The following theorem bounds the expected running time of the knowledge extractor.

Theorem 4 (Forking Lemma). *Consider a generic signature scheme with security parameter k. Let $\mathcal{A}$ be a τ-time forger whose input consists only of public data. Let q_h be the number of queries that $\mathcal{A}$ can ask from the random oracle. If $\mathcal{A}$ produces a valid signature with probability at least $\varepsilon > \kappa$, then there exists a knowledge extractor which extracts the secret key in expected time $t \leq \tau(q_h+1)/(\varepsilon - \kappa)$, where $\kappa = (q_h + 1) \cdot 2^{-k}$ is the knowledge error.*

Proof. Let $\boldsymbol{\omega} = (\omega_0, \omega_1, \dots, \omega_{q_h})$ be the used randomness, where ω_0 is the public input of $\mathcal{A}$ and ω_i is the oracle's reply to the ith query $h(m_i, \alpha_i)$. Let $\mathsf{A}(\boldsymbol{\omega}) = i$ if $\mathcal{A}$ outputs a valid signature on the ith query (m_i, α_i) and let $\mathsf{A}(\boldsymbol{\omega}) = 0$ otherwise. Notice that we set $\mathsf{A}(\boldsymbol{\omega}) = 0$ also if $\mathcal{A}$ outputs a valid signature without querying the corresponding hash value, so the fraction of nonzero entries in A is $\varepsilon' < \varepsilon$. Now, we can set the rewinding point to $f(a) = a$ for $a \neq 0$

[3] We assume that $\mathcal{A}$ never queries the same hash value twice. The assumption is perfectly reasonable as $\mathcal{A}$ gets no additional knowledge from repeating queries.

and apply the rewinding strategy of Alg. 2. In order to apply Thm. 3, we need to compute ε' and κ. Notice that due to the randomness of oracle outputs, the probability that $\mathcal{A}$ outputs a valid signature $(m, \alpha, \beta, \gamma)$ without querying $h(m, \alpha)$ is at most 2^{-k}, i.e., the probability of guessing the hash $h(m, \alpha)$ at random[4]. Thus, the fraction of entries with nonzero labels $\mathsf{A}(\boldsymbol{\omega})$ is $\varepsilon' \geq \varepsilon - 2^{-k}$. From (2), we get $\kappa = q_h \cdot 2^{-k}$. We now have everything we need to apply Thm. 3, so we conclude

$$t \leq \tau \cdot \frac{q_h + 1}{\varepsilon' - q_h 2^{-k}} \leq \tau \cdot \frac{q_h + 1}{\varepsilon - (q_h + 1)2^{-k}} \enspace .$$

□

Security of Generic Ring Signature Schemes. Recall that a ring signature is a tuple $(m, \boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma)$ and given two signatures $(m, \boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma)$, $(m, \boldsymbol{\alpha}', \boldsymbol{\beta}', \gamma)$ such that $\beta_i \neq \beta_i'$ for exactly one index i, it is possible to extract the secret x_i of ring member i. As in the case of single-signer signatures, we allow the adversary $\mathcal{A}$ to ask q_h queries from the random oracle and construct from $\mathcal{A}$ a knowledge extractor that computes two suitably related signatures that reveal the secret of some member. Apart from a more clever labelling and rewinding function f, the proof is similar to that of Thm. 4 and quite straightforward, so we omit some details.

Theorem 5 (Forking Lemma for Ring Signatures). *Consider a generic ring signature scheme with n ring members and security parameter k. Let $\mathcal{A}$ be a τ-time forger whose input consists only of public data. Let $q_h \geq n$ be the number of queries that $\mathcal{A}$ can ask from the random oracle. If $\mathcal{A}$ produces a valid ring signature with probability at least $\varepsilon > \kappa$, then there exists a knowledge extractor which extracts the secret of some ring member in expected time $t \leq \tau(V(q_h, n) + 1)/(\varepsilon - \kappa)$, where $V(q_h, n) = q_h(q_h - 1) \cdots (q_h - n + 1)$ is the number of n-permutations of q_h elements and $\kappa = (V(q_h, n) + n) \cdot 2^{-k}$ is the knowledge error.*

Proof. As above, let $\boldsymbol{\omega} = (\omega_0, \omega_1, \ldots, \omega_{q_h})$ be the used randomness, where ω_0 is the public input of $\mathcal{A}$ and ω_i is the oracle's reply to the ith query $h(m_i, \alpha_i)$. Let $\mathsf{A}(\boldsymbol{\omega}) = a = (a_1, \ldots, a_n)$ if $\mathcal{A}$ outputs a valid signature $(m, \alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n, \gamma)$ such that she queried (m, α_j) as the a_jth query; and let $\mathsf{A}(\boldsymbol{\omega}) = 0$ otherwise. The number of different labels is then $V(q_h, n) = q(q-1) \cdots (q - n + 1)$. For $\mathsf{A}(\boldsymbol{\omega}) \neq 0$, set the rewinding point to $i = f(a_1, \ldots, a_n) = \max_j \{a_j\}$ and apply the rewinding strategy of Alg. 2. Then the adversary indeed queries all (m, α_j) again, as her view is unchanged before making the last query (m, α_i). The probability that $\mathcal{A}$ outputs a valid signature without querying even one of the hashes $h(m, \alpha_j)$ is at most $n \cdot 2^{-k}$ by the union bound, so the fraction of transcripts with nonzero labels $\mathsf{A}(\boldsymbol{\omega}) \neq 0$ is $\varepsilon' \geq \varepsilon - n \cdot 2^{-k}$. Equation (2) again helps to find the aggregated knowledge error $\kappa = V(q_h, n) \cdot 2^{-k}$ and Thm. 3 gives the claimed bound. □

3.4 From Average-Case Complexity to Strict Time-Bounds

Theorems 1 and 3 only bound the *average* running-time of the knowledge extractor. Conversely, most security definitions are stated using strict time-bounds. Therefore, we use a standard methodology to get knowledge extractors that run in strict time with negligible failure probability $\sigma > 0$. Our construction is again *uniform*, i.e., the user does not need to specify the *a priori unknown* success probability ε, see Alg. 3.

[4] We may formalise this as follows: if $\mathcal{A}$ indeed outputs a signature $(m, \alpha, \beta, \gamma)$ such that $h(m, \alpha)$ was never queried, the verifier can query the oracle himself and reject immediately if the oracle replies $h(m, \alpha) \neq \beta$.

```
Rewind-Uni(A, σ):
for i = 1, 2, 3, ... do
   repeat at most ⌈log_2 1/σ⌉ times
      Execute Rewind-Exp(A) and stop if it executes more 2^i probes.
      if Rewind-Exp(A) succeeds in extracting the secret x halt with the output x.
   continue
continue
```

Algorithm 3: Rewinding algorithm Rewind-Uni.

Theorem 6 (Uniform Knowledge Extraction in Strict Time). *Let $\{0, 1, \dots, d\}$ be the set of labels. For any array* A *with an ε-fraction of nonzero entries and aggregated knowledge error κ, the algorithm* Rewind-Uni(A, σ) *fails in the first $8(d+1)(\log_2(1/\sigma)+1)/(\varepsilon-\kappa)$ probes with probability less than σ.*

Proof. Note that if $2^i \geq 2(d+1)/(\varepsilon-\kappa) = 2 \cdot \mathbf{E}[\mathsf{probes}]$ then Markov inequality $\Pr[X > \alpha] \leq \mathbf{E}[X]/\alpha$ assures that Rewind-Exp outputs a collision with probability at least $1/2$ in the repeat cycle. For $i_0 = \lceil \log_2(2(d+1)/(\varepsilon-\kappa)) \rceil$, the single repeat cycle makes at most $4(d+1)(\log_2(1/\sigma)+1)/(\varepsilon-\kappa)$ probes and fails with probability less than σ. As the number of probes in the ith repeat cycle is 2^i, the total complexity of the first i_0 invocations is at most $8(d+1)(\log_2(1/\sigma)+1)/(\varepsilon-\kappa)$. □

3.5 Significance of Our Results

Specially sound proofs of knowledge. The security of specially sound proofs of knowledge like Schnorr identification scheme is relatively well studied [Sch91,DF02,OO98]. Yet, we are the first to provide a knowledge-extraction algorithm that runs for all adversaries for whom knowledge extraction is possible at all. That is, we remove the artificial constraints $\varepsilon > c\kappa$ for some constant c and prove the result for all $\varepsilon > \kappa$. The expected running-time of our algorithm $2/(\varepsilon - \kappa)$ is approximately equal to the true lower bound when $\varepsilon \gg \kappa$. Previous analyses have used so called "heavy row" techniques to obtain the results and are thus inherently imprecise. In [DF02], the "heavy row" techniques lead to the bound $\mathbf{E}[\mathsf{probes}] \leq 56/\varepsilon$ when $\varepsilon \geq 4\kappa$, whereas we get the bound $\mathbf{E}[\mathsf{probes}] \leq 8/(3 \cdot \varepsilon)$. The main reason behind the factor 21 drop in this estimate is the simpler form of the new algorithm, which allows for a precise analysis.

Forking lemmata. Originally, forking lemma was derived in [PS00], where the authors managed to analyse the case where $\varepsilon \geq 7q \cdot 2^{-k}$ and proved that their algorithm requires $\mathbf{E}[\mathsf{probes}] \leq 84480 \cdot q/\varepsilon$ probes on average. Our analysis gives a bound $\mathbf{E}[\mathsf{probes}] \leq 2.8 \cdot q/\varepsilon$ that is several orders of magnitude better. However, the main advantage is the conceptual simplicity. The forking lemma itself is a simple conclusion of Thm. 3 where the only non-trivial steps are the choice of labels and the computation of aggregate knowledge error. The proof of Thm. 3 itself is straightforward compared to the original treatment in [PS00] that uses an auxiliary "splitting lemma". Moreover, Thm. 3 itself is a universal result, therefore, we do not need to repeat the same technical steps to prove slightly different results. Rather, we only have to choose proper labelling and estimate the aggregate knowledge error, which is in many cases a simple function of the scheme security parameters and the number of labels. Therefore, we believe that our approach provides a flexible and universal solution to many, not to say to all, knowledge extraction problems that require rewinding. For example, Thm. 3 can also be applied to prove the security of multi-signatures [BN06].

Finally, we explain why we have not considered security against an adversary that is also allowed to make signature queries. This is not because our approach does not allow for such proofs; on the contrary, existing results can be used to fill the gap easily and obtain new and sharper bounds. We have omitted the details mainly because the missing step deals with simulation and does not shed any new light to the rewinding problem.

Strict Time Bounds. Authors of the previous reductions always follow the same path: first, they construct a non-uniform knowledge extractor that runs in some strict time with fixed constant success probability. Then, they apply a complex transform to get a uniform knowledge extractor with bounded expected running-time. We, on the other hand, take a different approach and immediately construct a uniform expected-time algorithm. We can then apply a simple transform to achieve a uniform knowledge extractor with strict running time for *any desired success probability*. Of course, such a transform is possible also for the previous uniform knowledge extractors, but each additional transform comes at a cost of degradation in the precision of bounds.

4 From Soundness to Security

The forking lemma and its counterparts provide strong soundness guarantees: if somebody can generate signatures with non-negligible probability, then she must know the secret key. However, this result alone does not prove that the corresponding signature scheme is secure, as it might be easy to find the secret key from public parameters. Of course, we cannot choose the public parameters to attack the sub-primitive, e.g. the public key of the Schnorr signature is generated only once and we cannot resample it. The underlying hard problem for Schnorr signatures is the discrete logarithm problem, which is hard only on average. Thus, we want to show security for a key *chosen at random*. To close the gap, we must show that if there exists an efficient algorithm $\mathcal{A}$ that can forge signatures with probability ε, then there exists an efficient algorithm $\mathcal{B}$ that can solve discrete logarithm with high enough success probability. More precisely, if $\mathbb{G} = \langle g \rangle$ is a (τ, ε_0)-secure DL-group, then we need to construct for any sufficiently successful adversary $\mathcal{A}$, a τ-time adversary $\mathcal{B}$ that attacks the discrete logarithm problem with success probability $\Pr\left[x \leftarrow \mathbb{Z}_q, y \leftarrow g^x : \mathcal{B}(g, y) = x\right] > \varepsilon_0$.

Many authors [BP02,BN06] have addressed this problem by constructing from a t-time adversary $\mathcal{A}$ another adversary $\mathcal{B}$ that runs in time $2t$; the corresponding result is known as the reset lemma [BP02, p. 168]. In what follows, we give a large variety of similar knowledge extraction techniques that provide tradeoffs in terms of required security parameters (τ, ε_0) from the DL-group. For the best trade-off point, our reduction is a factor of $\sqrt{\varepsilon}$ sharper than the reset lemma. Moreover, our reduction is applicable in different areas including time-stamping schemes [BS04,BL06] and data-authentication [Vau05b,LN06].

In the formal security proof, one defines a success matrix A similarly to previous examples so that two coinciding labels $\mathsf{A}(\boldsymbol{\omega}) = \mathsf{A}(\boldsymbol{\omega}')$ allow to extract the secret x with high probability. In particular, we want to find a collision fast: approximately in $\ell = o\left(\frac{1}{\varepsilon - \kappa}\right)$ probes. For such a small number of probes, we cannot use our results for $\mathsf{Rewind\text{-}Uni}$, so we have to start from scratch. Also, the probability of finding even one nonzero element is $o(1)$, so we derive bounds only for $\mathsf{Rewind\text{-}Basic}$ and Rewind.

As before, we start with the simple case where $\mathsf{A}(r,c)$ is a zero-one matrix. Let probes_1 and probes_2 denote the number of probes in Step 1 and Step 2 of Rewind-Basic. Then the distribution of probes_1 depends only on the fraction of nonzero entries ε. However, the distribution of probes_2 depends on how the nonzero elements are distributed. Let p_r denote the probability of probing a nonzero element in the rth row. Although $p_r \in \{\frac{0}{n}, \dots, \frac{n}{n}\}$, we consider *relaxed* configurations where $p_r \in [0,1]$. We assume only that the probability of getting a nonzero element in the rth row is p_r and that $p_1 + \cdots + p_m = \varepsilon m$. Intuitively, if p_r varies over rows, then rows with high p_r are selected more likely. Hence, on average, the suitable nonzero element is found faster compared to the uniform configuration $p_r = \varepsilon$ for all rows. Indeed, the configuration $p_r = \varepsilon$ maximizes the failure probability after the first $\ell \leq 1/\varepsilon$ probes.

Lemma 3. *Let $\varepsilon \in (0,1]$ and $\ell \leq 1/\varepsilon - 1$. Then Step 2 of algorithm* Rewind-Basic *fails to finish in ℓ steps with probability at most* $\Pr[\mathsf{probes}_2 > \ell] \leq (1-\varepsilon)^\ell$.

Proof. To prove the claim, we have to show that $\Pr[\mathsf{probes}_2 > \ell]$ as a function of $\boldsymbol{p} = (p_1, \dots, p_m)$ is maximised at $\boldsymbol{p}_\circ = (\varepsilon, \dots, \varepsilon)$. The proof itself is straightforward but technical and thus the complete proof is given in Appendix B. □

Thm. 7 quantifies the probability that Rewind-Basic is successful, provided that a malicious adversary achieves advantage ε and we can freely choose the rows.

Theorem 7. *Let $\varepsilon \in (0,1]$ and let $\Pr[\mathsf{probes} \leq \ell \wedge \mathsf{success}]$ denote the probability that* Rewind-Basic *returns the secret in ℓ steps. Let κ be the knowledge error. If $\ell \leq 1/\varepsilon$ then* $\Pr[\mathsf{probes} \leq \ell \wedge \mathsf{success}] \geq \frac{1}{6}\ell(\ell-1)\varepsilon(\varepsilon-\kappa)$. *If $\ell \geq 1/\varepsilon$, then* $\Pr[\mathsf{probes} \leq \ell \wedge \mathsf{success}] \geq \frac{1}{4}(1-\kappa/\varepsilon)$.

Proof. Consider the failure probability of the Rewind-Basic algorithm after the first ℓ probes. There are $\ell+1$ disjoint events that can cause failure: either we find $\mathsf{A}(r,c) \neq 0$ at the ith probe and then fail to reveal the second $\mathsf{A}(r,c')$, or we cannot find the first nonzero element at all. Thus,

$$\Pr[\mathsf{probes} > \ell] = \sum_{i=1}^{\ell} \Pr[\mathsf{probes}_1 = i] \Pr[\mathsf{probes}_2 > \ell - i] + \Pr[\mathsf{probes}_1 > \ell] \enspace .$$

Note that probing in Step 1 is uniform and each probe is successful with probability ε. For $\ell \leq 1/\varepsilon$, $\ell - i \leq \ell - 1 \leq 1/\varepsilon - 1$ and Lemma 3 assures that

$$\Pr[\mathsf{probes} > \ell] \leq \sum_{i=1}^{\ell} (1-\varepsilon)^{i-1}\varepsilon(1-\varepsilon)^{\ell-i} + (1-\varepsilon)^\ell = \ell\varepsilon(1-\varepsilon)^{\ell-1} + (1-\varepsilon)^\ell \enspace .$$

A lower bound based on the third order Taylor expansion gives $1 - \Pr[\mathsf{probes} > \ell] \geq \frac{1}{6}\ell(\ell-1)\varepsilon^2$; see Appendix D and Lemma 5 for the corresponding proof. If $\ell \geq 1/\varepsilon$ then observe a new relaxed configuration $p'_r = \frac{1}{\ell\varepsilon} \cdot p_r$. Then the corresponding average $\varepsilon' = 1/\ell$ and Rewind chooses the rows r with the same probabilities as before. Since failure probabilities $\Pr[\mathsf{probes}_1 > \ell | \varepsilon']$ and $\Pr[\mathsf{probes}_2 > \ell | r, p'_r]$ only increase, we have obtained

$$\Pr[\mathsf{probes} \leq \ell | \varepsilon] \geq \Pr\left[\mathsf{probes} \leq \ell | \varepsilon'\right] \geq 1 - \left(1 - \tfrac{1}{\ell}\right)^\ell - \left(1 - \tfrac{1}{\ell}\right)^{\ell-1} =: g(\ell) \enspace ,$$

as Lemma 3 holds for ε'. Since $(1-\frac{1}{\ell})^{2\ell-1} \leq e^{-2}$, or equivalently, $2+(2\ell-1)\ln(1-\frac{1}{\ell}) \leq 0$, we get $g'(\ell) = -(1-\frac{1}{\ell})^{\ell}/(\ell-1) \cdot ((2\ell-1)\ln(1-\frac{1}{\ell})+2) \geq 0$ and $\Pr[\mathsf{probes} \leq \ell] \geq g(2) = 1/4$ for $\ell \geq 1/\varepsilon$. As $p_{\mathrm{fail}}(\varepsilon) \leq \varepsilon/\kappa$, the inequalities about success probabilities follow. □

Now, we have a non-trivial bound for the success probability of the rewinding algorithm for a small number of steps. The next theorem provides an adequate solution for reductions that take into account the average advantage over the public key. Namely, in the following, let ϵ_i denote the advantage of $\mathcal{A}$ for a given public key g^i, and let ϵ be the averaged advantage over all keys.

Theorem 8. *Let* $\mathsf{A}_1, \ldots, \mathsf{A}_u$ *be a collection of* $m \times n$ *matrices and let* ε_i *be the fraction of nonzero entries in* A_i*. Let* κ *be the knowledge error and let* $\varepsilon = \frac{1}{u} \cdot (\varepsilon_1 + \cdots + \varepsilon_u)$ *be the average fraction of nonzero elements. If* $2 \leq \ell \leq 1/\varepsilon$ *then for randomly chosen* $i \in [u]$, $\mathsf{Rewind\text{-}Basic}(\mathsf{A}_i)$ *succeeds with probability* $\Pr[i \leftarrow [u] : \mathsf{probes} \leq \ell \wedge \mathsf{success}] \geq (\varepsilon - \kappa) \cdot \min\left\{\frac{1}{8}, \frac{1}{6} \cdot \ell(\ell-1)\varepsilon\right\}$.

Proof. To prove the claim, we find the configuration of ε_i that minimises the average success probability $\mathsf{Adv} = \Pr[i \leftarrow [u] : \mathsf{probes} \leq \ell \wedge \mathsf{success}]$. More precisely, we use Thm. 7 to find a lower bound for each ε_i. As usual, we allow relaxed matrix configurations, since this can only decrease the lower bounds.

For a fixed ℓ, there are three types of matrices. If $\varepsilon_i \leq \kappa$ then Thm. 7 provides no guarantees. If $\varepsilon_i > 1/\ell$ then the lower bound to success given in Thm. 7 is convex-cap and for remaining cases the lower bound is convex-cup w.r.t. ε_i. Let $\mathcal{I}_a = \{i : \varepsilon_i < \kappa\}$, $\mathcal{I}_b = \{i : \kappa \leq \varepsilon_i < 1/\ell\}$, $\mathcal{I}_c = \{i : \varepsilon_i = 1/\ell\}$ and $\mathcal{I}_d = \{i : \varepsilon_i > 1/\ell\}$ be the corresponding index sets. Since the bounds of Thm. 7 are increasing w.r.t. ε_i, we may assume that $\mathcal{I}_a = \emptyset$, for otherwise we could decrease the lower bound by infinitesimally increasing ε_i for $i \in \mathcal{I}_a$. Secondly, for the index set $\mathcal{I}_c \cup \mathcal{I}_d$, the lower bound $\frac{1}{4}(1-\frac{\kappa}{\varepsilon_i})$ is convex-cap and it is straightforward to verify that the minimising configuration for $\mathcal{I}_c \cup \mathcal{I}_d$ consists of t_c values $\varepsilon_i = 1/\ell$, t_d values of $\varepsilon_i = 1$ and possibly from a single value $\varepsilon_{i^*} \in (1/\ell, 1)$. If we relax constraints and allow also fractional counts for t_c and t_d, then in the optimal configuration, $\varepsilon_i < 1/\ell$ with probability p_b, $\varepsilon_i = 1/\ell$ with probability p_c and $\varepsilon_i = 1$ with probability p_d. Finally, as $\varepsilon(\varepsilon - \kappa)$ is convex-cup w.r.t. ε, Jensen's inequality together with Thm. 7 gives

$$\Pr[\varepsilon_i \leq 1/\ell \wedge \mathsf{probes} \leq \ell \wedge \mathsf{success}] \geq \frac{(p_b + p_c)\ell(\ell-1)\varepsilon_2(\varepsilon_2 - \kappa)}{6} \enspace ,$$

where $\varepsilon_2 \leq 1/\ell$ is the weighted average of the $\varepsilon_i \leq 1/\ell$. As a result, the minimising configuration consists of $p_1 = p_d$ fraction of values $\varepsilon_i = 1$ and $p_2 = p_b + p_c$ fraction of values $\varepsilon_i = \varepsilon_2$, so the lower bound for Adv can be found as a minimising task $f(p_1, p_2, \varepsilon_2) = \frac{1}{4}p_1(1-\kappa) + \frac{1}{6}p_2\ell(\ell-1)\varepsilon_2(\varepsilon_2-\kappa) \to \min$ w.r.t. $p_1 + p_2 = 1$, $p_1 + p_2\varepsilon_2 = \varepsilon$, $\kappa \leq \varepsilon_2 \leq 1/\ell$, $p_1, p_2 \geq 0$. By applying Lemma 4 with $c = \frac{2\ell(\ell-1)}{3(1-\kappa)}$ and $\varepsilon_\circ = \frac{1}{\ell} \geq \varepsilon$, we get the desired result

$$\mathsf{Adv} \geq \tfrac{1}{4}(1-\kappa) \cdot \min\left\{\tfrac{1}{2(1-\kappa)}, c\varepsilon(\varepsilon-\kappa)\right\} = (\varepsilon-\kappa) \cdot \min\left\{\tfrac{1}{8}, \tfrac{1}{6} \cdot \ell(\ell-1)\varepsilon\right\} \enspace .$$

□

Note 1. One can verify that bounds derived in Lemmata 4 and 5 are at most three times away from the true bounds and Thm. 8 underestimates the worst case probability at most three times. Also, the Reset Lemma is a special case of Thm. 8 with $\ell = 2$.

It is now relatively straightforward to generalise the result from the basic algorithm to algorithm Rewind with multiple labels[5].

Corollary 2. *Let $\{0, 1, \ldots, d\}$ be the set of labels. Let $\mathsf{A}_1, \ldots, \mathsf{A}_u$ be a collection of arrays and let ε_i be the fraction of nonzero entries in A_i. Let κ be the maximal knowledge error over labels and let $\varepsilon = \frac{1}{u} \cdot (\varepsilon_1 + \cdots + \varepsilon_u)$ be the average fraction of nonzero elements. If $2 \leq \ell \leq 1/\varepsilon$ then for randomly chosen $i \in [u]$, $\mathsf{Rewind}(\mathsf{A}_i)$ succeeds with probability* $\Pr[i \leftarrow [u] : \mathsf{probes} \leq \ell \wedge \mathsf{success}] \geq (\frac{\varepsilon}{d} - \kappa) \cdot \min\left\{\frac{1}{8}, \frac{1}{6} \cdot \ell(\ell-1)\frac{\varepsilon}{d}\right\}$.

Proof. First, note that we can consider doubly indexed zero-one arrays $\mathsf{A}_{i,a}$ where $\mathsf{A}_{i,a}(\boldsymbol{\omega}) = 1$ iff $\mathsf{A}_i(\boldsymbol{\omega}) = a$. For fixed i, Rewind chooses matrix $\mathsf{A}_{i,a}$ with probability $\frac{\varepsilon_{i,a}}{\varepsilon_i}$ where $\varepsilon_{i,a}$ is the fraction of a labels in A_i. When Rewind has chosen the $\mathsf{A}_{i,a}$ it behaves exactly like Rewind-Basic. Since the dynamic change of the rewinding point does not change the analysis of Thm. 8 and we know that for all $\mathsf{A}_{i,a}$ the corresponding knowledge error is smaller than κ, we can apply Thm. 8 for the collection of arrays $\{\mathsf{A}_{i,a}\}$. However, we need to recalculate the average success probability for the collection $\{\mathsf{A}_{i,a}\}$

$$\varepsilon' = \mathbf{E}(\varepsilon_{i,a}) = \frac{1}{u} \cdot \sum_{i=1}^{u} \sum_{a=1}^{d} \frac{\varepsilon_{i,a}^2}{\varepsilon_i} \geq \frac{1}{u} \cdot \sum_{i=1}^{u} \frac{\varepsilon_i}{d} = \frac{\varepsilon}{d}$$

as $\sum_{a=1}^{d} \varepsilon_{i,a}^2 \geq \varepsilon_i^2/d$ by Jensen's inequality. The claim follows, if we substitute ε with ε' in Thm. 8. $\square$

Application to Schnorr Identification Scheme. To show the strength of Thm. 8, we again consider the exact security of the Schnorr identification scheme. More precisely, let $\mathcal{A}$ be an adversary that achieves ε average advantage against the protocol, where the public key is chosen randomly, i.e., $y = g^x$ for $x \leftarrow \mathbb{Z}_q$.

Theorem 9. *Let $\mathcal{A}$ be a t-time adversary that achieves average advantage ε against the Schnorr identification scheme over a q-element group $\mathbb{G}$. Then there exists a τ-time algorithm $\mathcal{B}$ that computes discrete logarithm with success probability ε_0 where*

$$\begin{cases} \tau &= \ell t, \\ \varepsilon_0 &= (\varepsilon - \kappa) \cdot \min\left\{\frac{1}{8}, \frac{1}{6}\ell(\ell-1)\varepsilon\right\}, \end{cases} \qquad 2 \leq \ell \leq 1/\varepsilon \ .$$

Proof. Let ε_i be the advantage of adversary $\mathcal{A}$ against the public key $y = g^i$. By the construction we know that $\varepsilon = \frac{1}{q}(\varepsilon_0 + \cdots + \varepsilon_{q-1})$ and $\kappa_0 = \ldots = \kappa_{q-1} = 1/q = 2^{-k}$. Directly applying Thm. 8 gives

$$\Pr[x \leftarrow \mathbb{Z}_g, y = g^x : \mathcal{B}(g, y) = x] \geq (\varepsilon - 2^{-k}) \cdot \min\left\{\frac{1}{8}, \frac{1}{6}\ell(\ell-1)\varepsilon\right\}$$

where $\mathcal{B}$ runs Rewind-Basic for ℓ probes to restore x. Note that the running-time of $\mathcal{B}$ is ℓt and the claim follows. $\square$

[5] A more careful analysis would allow to obtain a reduction that loses a factor d instead of d^2. However, this would require reproving Thm. 8 with all its auxiliary results.

Optimal time-success ratio. Thm. 9 shows that for $\ell \geq \frac{1}{\sqrt{\varepsilon}}$, the worst case success probability grows slowly and that reductions with $\ell \in [2, \frac{1}{\sqrt{\varepsilon}}]$ are reasonable. Let $\varepsilon > 2\kappa = 2^{-k+1}$. If we have a (t, ε)-attack against the Schnorr identification protocol then we must have $(t\ell, 2 \cdot \ell^2\varepsilon^2)$-secure DL-group $\mathbb{G}$. In order to compare such reductions, we can observe the change in the time-success ratio $\alpha = \frac{t}{\varepsilon}$. The time-success ratio characterises the average number of computational steps an adversary must do to break the primitive, if he is allowed to restart the attack, i.e, y is resampled. Let α_0 be the best time-success ratio for finding the discrete logarithm in $\mathbb{G}$ and let α be the time-success ratio of an attack on the protocol. Then, Thm. 9 gives $\frac{\alpha}{\alpha_0} \leq 0.5 \cdot \ell\varepsilon \leq O(\sqrt{\varepsilon})$ and the number of probes really makes a difference. For example, let the desired success bound be $\varepsilon = 2^{-80}$. Then for the Resetting Lemma where $\ell = 2$ the average guarantee for breaking time decreases by 2^{80}, whereas for the optimal choice $\ell = \sqrt{\varepsilon}$ the average breaking time decreases only 2^{41} times. Nevertheless, observe that the decrease $O(\sqrt{\varepsilon})$ is always quite significant. Hence, such reduction techniques should be avoided if possible.

And Now For Something Completely Different... Fair Coin-Flipping. In order to show the power of Thm. 8, we look at a completely different application. Namely, we analyse the security of Blum's coin flipping protocol [Blu81] for k-bit string commitments. The protocol consists of four steps: (a) a trusted party uses a probabilistic algorithm Gen to fix the public parameters pk of the commitment scheme; (b) Alice chooses a random k-bit string x and sends a commitment $\mathsf{Com}_{\mathsf{pk}}(x)$ to Bob; (c) Bob sends a random b-bit string y to Alice; (d) Alice opens the commitment to x and both compute $z = x \oplus y$.

It is intuitively clear that Bob cannot control the value of z if the commitment is hiding and Alice cannot control z if the commitment is binding. However, there is an important quantitative difference. Reduction to hiding is straightforward, whereas reduction to the binding property is more complex. The precise complexity of the latter reduction has been studied only for the case of 1-bit strings. Recall that a commitment is (t, ε)-binding if any t-time adversary $\mathcal{B}$ can generate a commitment c and then open it to two different values with probability less than ε, where the probability is taken over the random coins of $\mathcal{B}$ and Gen. Hence, we have to somehow extract a double opening from malicious Alice.

Theorem 10. *Let $\mathcal{Z}$ be an s-element subset of $\{0,1\}^k$ with an efficient membership test and let the commitment scheme be (t_1, ε_1)-binding. If Bob is honest, then a τ-time malicious Alice can force outcome $z \in \mathcal{Z}$ with probability $\varepsilon_2 \leq s \cdot 2^{-k} + \max\{8\varepsilon_1, \frac{\sqrt{6\varepsilon_1}}{\ell-1}\}$, where $\ell = \frac{t}{\tau}$. In particular, if $\tau = \Theta(t\sqrt{\varepsilon_1})$, then $\varepsilon_2 \leq s \cdot 2^{-k} + 8\varepsilon_1$.*

Proof. Assume for the sake of contradiction that a τ-time Alice can force $z \in \mathcal{Z}$ with probability $\varepsilon_2 > s \cdot 2^{-k} + \max\{8\varepsilon_1, \sqrt{6\varepsilon_1}/(\ell-1)\}$. Let $i \in [u]$ be the randomness used by the Gen algorithm and let $\mathsf{A}_i(r,c)$ be the status matrix corresponding to pk_i, where r is the randomness of Alice and $c = y$ is the reply of Bob. Let $\mathsf{A}_i(r,c) = 0$ if Alice fails to open the commitment after receiving y so that $x \oplus y \in \mathcal{Z}$, and let $\mathsf{A}(r,c) = 1$ otherwise. Note that $\kappa = s \cdot 2^{-k}$, as for different values of y, $x \oplus y \in \mathcal{Z}$ must be different. Now, applying Thm. 8, we get that $\mathsf{Rewind\text{-}Basic}$ reveals a double opening with probability at least $\left(\varepsilon_2 - s \cdot 2^{-k}\right) \cdot \min\left\{\frac{1}{8}, \frac{1}{6}\ell(\ell-1)\varepsilon_2\right\} > \varepsilon_1$ after the first ℓ probes. The result follows. □

Recent manual authentication schemes [Vau05b,PV06,LN06] that use Blum's coin flipping protocol as a sub-primitive indeed require such a property. A partial control over z such that

$z \in \mathcal{Z}$ allows to bypass security tests so that the protocol becomes insecure. Moreover, as all τ_0-time distinguishers D naturally define a set $\mathcal{Z} = \{z : D(z) = 1\}$, we have obtained that the protocol outcome is (τ_0, ε_2)-pseudorandom—an expected result with precise security guarantees.

Similar reductions are done in [BS04] to prove the security of timestamping schemes. The authors gave only a polynomial reduction for $\ell = 2$. Our results allow to quantify the exact security of a wide range of reductions. In particular, choosing $\ell = \Theta\left(\frac{1}{\sqrt{\varepsilon}}\right)$ significantly decreases the drop in the time-success ratio.

5 Open Questions

We derived bounds on black-box knowledge extraction, assuming that all ε-fractional configurations are achievable. However, it is relatively straightforward to verify that τ-time adversaries cannot generate all possible matrix configurations, as there exist matrix configurations with Kolmogorov complexity mn. Since the adversarial code must be shorter than $\tau \ll mn$, we get that only a negligible fraction of possible configurations are realisable. It might be possible to exploit this in knowledge extraction algorithms. Whether this is possible and what the biggest possible gain in running-time can be, are theoretically very intriguing questions. Another interesting question is whether white-box knowledge extraction, where one can exploit knowledge about the internal structure of an adversarial algorithm, can achieve results that are dramatically more efficient, e.g., possibly bypassing the quadratic computational blow-up in the case of generic signatures.

Acknowledgments. We would like to thank Hendrik Nigul and Serge Vaudenay for useful comments.

References

[BL06] Ahto Buldas and Sven Laur. Do broken hash functions affect the security of time-stamping schemes? In Jianying Zhou, Moti Yung, and Feng Bao, editors, *4th International Conference on Applied Cryptography and Network Security – ACNS'06*, volume 3989 of *Lecture Notes in Computer Science*, pages 50–65, Singapore, 2006. Springer-Verlag.

[Blu81] Manuel Blum. Coin Flipping by Telephone. In Allen Gersho, editor, *Advances in Cryptology: A Report on CRYPTO 81, CRYPTO 81, IEEE Workshop on Communications Security*, pages 11–15, Santa Barbara, CA, USA, August 24–26, 1981. U.C. Santa Barbara, Dept. of Elec. and Computer Eng., ECE Report No 82-04, 1982.

[BN06] Mihir Bellare and Gregory Neven. Multisignatures in the Plain Public-Key Model and a General Forking Lemma. In Rebecca Wright and Sabrina De Capitani di Vimercati, editors, *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pages ?–?, Alexandria, VA, USA, October 30–November 3, 2006.

[BP02] Mihir Bellare and Adriana Palacio. GQ and Schnorr Identification Schemes: Proofs of Security against Impersonation under Active and Concurrent Attacks. In Moti Yung, editor, *Advances in Cryptology — CRYPTO 2002, 22nd Annual International Cryptology Conference*, volume 2442 of *Lecture Notes in Computer Science*, pages 162–177, Santa Barbara, USA, August 18–22, 2002. Springer-Verlag.

[BR93] Mihir Bellare and Phillip Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In Victoria Ashby, editor, *1st ACM Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, 3–5 November 1993. ACM Press.

[BS04] Ahto Buldas and Märt Saarepera. On Provably Secure Time-Stamping Schemes. In Pil Joong Lee, editor, *Advances on Cryptology — ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 500–514, Jeju Island, Korea, December 5-9 2004. Springer-Verlag.

[DF02] Ivan Damgård and Eiichiro Fujisaki. An Integer Commitment Scheme Based on Groups with Hidden Order. In Yuliang Zheng, editor, *Advances on Cryptology — ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 125–142, Queenstown, New Zealand, December 1–5, 2002. Springer-Verlag.

[FS86] Amos Fiat and Adi Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology—CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194, Santa Barbara, California, USA, 11–15 August 1986. Springer-Verlag, 1987.

[HS03] Javier Herranz and Germán Sáez. Forking Lemmas for Ring Signature Schemes. In Thomas Johansson and Subhamoy Maitra, editors, *INDOCRYPT 2003*, volume 2904 of *Lecture Notes in Computer Science*, pages 266–279, New Delhi, India, December 8–10 2003. Springer-Verlag.

[LN06] Sven Laur and Kaisa Nyberg. Efficient Mutual Data Authentication Using Manually Authenticated Strings. In Yi Mu and David Pointcheval, editors, *CANS 2006*, volume 4301 of *Lecture Notes in Computer Science*, pages 99–107, Suzhou, China, December 8-10, 2006. Springer-Verlag. To appear. Full version available as ePrint Report 2005/424.

[OO98] Kazuo Ohta and Tatsuaki Okamoto. On Concrete Security Treatment of Signatures Derived from Identification. In Hugo Krawczyk, editor, *Advances in Cryptology — CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 354–369, Santa Barbara, USA, 23–27 August 1998. Springer-Verlag.

[PS00] David Pointcheval and Jacques Stern. Security Arguments for Digital Signatures and Blind Signatures. *Journal of Cryptology*, 13(3):361–396, 2000.

[PV06] Sylvain Pasini and Serge Vaudenay. SAS-Based Authenticated Key Agreement. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *Public Key Cryptography - PKC 2006, 9th International Conference on Theory and Practice of Public-Key Cryptography*, volume 3958 of *Lecture Notes in Computer Science*, pages 395–409, New York, NY, USA, April 24–26, 2006. Springer-Verlag.

[Sch91] Claus-Peter Schnorr. Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, 4(3):161–174, 1991.

[Vau05a] Serge Vaudenay. *A Classical Introduction to Cryptography: Applications for Communications Security*. Springer, 2005.

[Vau05b] Serge Vaudenay. Secure Communications over Insecure Channels Based on Short Authenticated Strings. In Victor Shoup, editor, *Advances in Cryptology — CRYPTO 2005, 25th Annual International Cryptology Conference*, volume 3621 of *Lecture Notes in Computer Science*, pages 309–326, Santa Barbara, USA, August 14–18, 2005. Springer-Verlag.

A Proof of Optimality (Theorem 2)

Proof. Consider a simpler task: find two nonzero entries in the matrix and let $\mathcal{S}$ be a corresponding deterministic probing strategy. Now, lets compute the average case probe complexity of $\mathcal{S}$ over matrices with εmn nonzero entries. W.l.o.g. we assume that $\mathcal{S}$ probes always new entries. As $\mathcal{S}$ is deterministic and matrix chosen randomly, we can show that any new choice of a new matrix entry is equivalent to random probing of A. In the following, we show that random probing without repetition requires at least $2(mn+1)/(\varepsilon mn+1)$ probes on average. Hence, the average case bound must hold also for all probabilistic algorithms. As the worst case complexity is always as large as the average, then for some matrix $\mathsf{A}_{\mathcal{S}}$ the bound holds.

More precisely, we show that if entries are probed randomly without resampling then the expected number of probes that reveal two nonzero entries from a u-element list that contains v nonzero entries is $\mathbf{E}[\mathsf{probes}] = 2(u+1)/(v+1) = (2-o(1))\cdot u/v$. Really, let probes be the number of probes until two nonzero entries are revealed. Then by a simple combinatorial argument,

$$\begin{aligned}\Pr\left[\mathsf{probes} > \ell\right] &= \frac{(u-v)\cdots(u-v-\ell+1)}{u\cdots(u-\ell+1)} + \ell\cdot\frac{v(u-v)\cdots(u-v-\ell+2)}{u\cdots(u-\ell+1)}\\ &= \binom{u-\ell}{v}\Big/\binom{u}{v} + \ell\cdot\binom{u-\ell}{v-1}\Big/\binom{u}{v}\ ,\end{aligned}$$

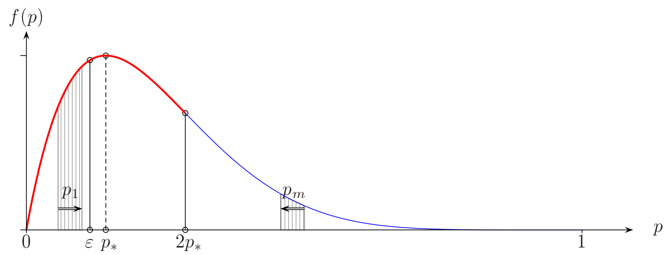

Fig. 1. Suboptimal matrix configuration aligned with $f(p) = p(1-p)^\ell$

as there are two mutually exclusive options: either the first ℓ entries are equal to 0, or there is exactly one nonzero element among them. Hence, the average number of probes can be computed

$$\mathbf{E}[\mathsf{probes}] = \sum_{\ell=1}^{\infty} \Pr[\mathsf{probes} \geq \ell] = \frac{u+1}{v+1} + \frac{u+1}{v+1} = \frac{2(u+1)}{v+1} = (2 - o(1)) \cdot \frac{u}{v} \ ,$$

using standard combinatorial equalities. $\square$

B The Worst Matrix Configuration

Lemma 2. *Let $\varepsilon \in [0,1]$ and $\ell \leq 1/\varepsilon - 1$. Then the relaxed configuration $p_r = \varepsilon$ for all rows $r \in [m]$ maximises the failure probability* $\Pr[\mathsf{probes}_2 > \ell]$ *for* Rewind-Basic.

Proof. For a fixed configuration $\boldsymbol{p} = (p_1, \ldots, p_m)$, express $\Pr[\mathsf{probes}_2 > \ell]$ as

$$F_\ell(\boldsymbol{p}) = \sum_{r=1}^{m} \Pr[R = r] \cdot \Pr[\mathsf{probes}_2 > \ell | R = r] = \frac{1}{\varepsilon m} \cdot \sum_{r=1}^{m} p_r (1 - p_r)^\ell \ .$$

A tight upper bound on $\Pr[\mathsf{probes}_2 > \ell]$ can be found by maximising $F_\ell(\boldsymbol{p})$ with relaxed constraints $p_i \in [0,1]$ and $p_1 + \cdots + p_m = \varepsilon m$. We call such points *feasible*. A feasible point is *maximal* if $F_\ell(\boldsymbol{p})$ is maximal among all feasible points. Now, consider a function $f(p) = p(1-p)^\ell$. From $f'(p) = (1-p)^{\ell-1}(1 - \ell p - p)$ and $f''(p) = \ell(1-p)^{\ell-2}(\ell p + p - 2)$, it is easy to see that f has one local maximum point $p_* = 1/(\ell+1)$ in the interesting region $[0,1]$. Also, f is convex-cap in $[0, 2p_*]$ and convex-cup in $[2p_*, 1]$. The restriction $\ell \leq 1/\varepsilon - 1$ then implies $\varepsilon \leq p_*$. Let $\boldsymbol{p} = (p_1, p_2, \ldots, p_m)$ be a maximal feasible point. W.l.o.g., assume that $p_1 \leq p_2 \leq \cdots \leq p_m$. Assume that p_m is in the convex-cup region, i.e., $p_m \in [2p_*, 1]$. Then, from $p_m > p_* \geq \varepsilon$ and $\sum_{i=1}^{m} p_i = \varepsilon m$, we get that $p_1 < \varepsilon$. Let $\delta = \min\{\varepsilon - p_1, p_m - \varepsilon\} > 0$. Taking a feasible $\boldsymbol{p}_\circ = (p_1 + \delta, p_2, \ldots, p_{m-1}, p_m - \delta)$ gives $F_\ell(\boldsymbol{p}_\circ) > F_\ell(\boldsymbol{p})$, a contradiction with the maximality of $\boldsymbol{p}$ (See Fig. 1 as an illustrative example). Thus, all coordinates $p_1, p_2, \ldots, p_m$ of maximal feasible points lie in $[0, 2p_*]$. Since f is convex-cap in $[0, 2p_*]$ Jensen's inequality implies that there is a unique maximum $p_1 = \cdots = p_m = \varepsilon$. $\square$

C Analytical Optimisation Problem

The proof of Thm. 8 requires on analytical solution to the following optimisation task

$$f(p_1, p_2, \varepsilon_2) = p_1 + c\varepsilon_2(\varepsilon_2 - \kappa) \to \min \tag{F}$$

$$\begin{cases} p_1 + p_2 = 1, \quad p_1 + p_2\varepsilon_2 = \varepsilon, \\ \kappa \leq \varepsilon_2 \leq \varepsilon_\circ, \quad p_1, p_2 \geq 0 \end{cases} \tag{Ω}$$

for $\varepsilon \leq \varepsilon_\circ$ and $c = c(\ell)$ growing as a function of probes ℓ.

Lemma 4. *For any $c > 0$ and $\varepsilon \geq \kappa$, the minimisation task (F) – (Ω) has a solution*

$$\min_{\Omega} f(p_1, p_2, \varepsilon_2) = \begin{cases} \frac{\varepsilon - \kappa}{1 - \kappa} & \text{if } \frac{1}{c} \in (0, \kappa - \kappa^2) \ , \\ 1 - c(1-\varepsilon)(2 - \kappa - 2\sqrt{1 - \kappa - 1/c}) & \text{if } \frac{1}{c} \in [\kappa - \kappa^2, 2\varepsilon - \varepsilon^2 - \kappa] \ , \\ c\varepsilon(\varepsilon - \kappa) & \text{if } \frac{1}{c} \in (2\varepsilon - \varepsilon^2 - \kappa, \infty) \ . \end{cases}$$

In particular, $\min_\Omega f(p_1, p_2, \varepsilon_2) \geq (\varepsilon - \kappa) \min\{c\varepsilon, \frac{1}{2(1-\kappa)}\}$.

Proof. First, note that $\varepsilon_2 = (\varepsilon + p_2 - 1)/p_2$ and thus we have to minimise

$$f(p_2) = 1 - c(1-\varepsilon)(2-\kappa) + \frac{c(1-\varepsilon)2}{p_2} + c\left(1 - \kappa - \frac{1}{c}\right)p_2$$

where $\frac{1-\varepsilon}{1-\kappa} \leq p_2 \leq 1$. Note that $g(x) = a/x + bx$ for $a, b > 0$ has a single minimum $x_* = \sqrt{a/b}$ in the region $x \geq 0$ and $g(x_*) = 2\sqrt{ab}$ and no minimum if a and b have different signs. Hence, if $1 - \kappa - 1/c > 0$, the minimum is achieved in the point $p_2^* = \frac{1-\varepsilon}{\sqrt{1-\kappa-1/c}}$. Note that p_2^* is in the feasible region if $\kappa - \kappa^2 \leq 1/c \leq 2\varepsilon - \varepsilon^2 - \kappa$ and then $f(p_2^*) = 1 - c(1-\varepsilon)(2 - \kappa - 2\sqrt{1 - \kappa - 1/c})$. Otherwise $f(p_2)$ is minimised at the endpoints, i.e, $f(p_2) \geq \min\{(\varepsilon - \kappa)/(1 - \kappa), c\varepsilon(\varepsilon - \kappa)\}$. As $(\varepsilon - \kappa)/(1 - \kappa) \leq c\varepsilon(\varepsilon - \kappa)$ if $\frac{1}{c} \leq \varepsilon - \varepsilon\kappa$ and $\kappa - \kappa^2 \leq \varepsilon - \kappa\varepsilon \leq 2\varepsilon - \varepsilon^2 - \kappa \leq 1 - \kappa$ the first equation follows. Since at the limiting point of linear growth $1/c = 2\varepsilon - \varepsilon^2 - \kappa$ the ratio

$$\frac{\min_\Omega f(p_1, p_2, \varepsilon_2)(1-\kappa)}{\varepsilon - \kappa} = \frac{\varepsilon(1-\kappa)}{2\varepsilon - \varepsilon^2 - \kappa} \geq \frac{1}{2}$$

and $\min_\Omega f(p_1, p_2, \varepsilon_2)$ is obviously growing if c is growing, we get $\min_\Omega f(p_1, p_2, \varepsilon_2) \geq \frac{\varepsilon - \kappa}{2(1-\kappa)}$ if $\frac{1}{c} < 2\varepsilon - \varepsilon^2 - \kappa$. The second claim follows. See Fig. 2 as an illustration. □

D Useful Approximation Formulas

The famous Taylor's Theorem states that $f(x) = f(0) + \frac{f'(0)}{1!} \cdot x + \cdots + \frac{f^{(n)}(0)}{n!} \cdot x^n + \frac{f^{(n+1)}(\xi)}{(n+1)!} \cdot x^{n+1}$ for some $\xi \in [0, x]$ if $f^{(n+1)}(x)$ is continuous in the interval $[0, x]$. We use this fact to derive some inequalities.

Lemma 5 (Third order Taylor expansion). *For any $\varepsilon \in [0, 1]$ and integer $\ell \leq 1/\varepsilon$, the following inequality* $1 - (1-\varepsilon)^\ell - \ell\varepsilon(1-\varepsilon)^{\ell-1} \geq \frac{1}{2}\ell(\ell-1)\varepsilon^2 - \frac{1}{3}\ell(\ell-1)(\ell-2)\varepsilon^3 \geq \frac{1}{6}\ell(\ell-1)\varepsilon^2$ *holds.*

Proof. Let $f(\varepsilon) = 1 - (1-\varepsilon)^\ell - \ell\varepsilon(1-\varepsilon)^{\ell-1}$. By straightforward computation, $f(0) = 0$, $f'(0) = 0$, $f^{(2)}(0) = \ell(\ell-1)$, $f^{(3)}(0) = -2\ell(\ell-1)(\ell-2)$ and $f^{(4)}(\xi) = \ell(\ell-1)(\ell-2)(\ell-3)(1-\xi)^{\ell-5}(3 - \xi\ell + \xi) \geq 0$ for $\xi \in [0, 1/\ell]$. The claim follows from Taylor's Theorem. □

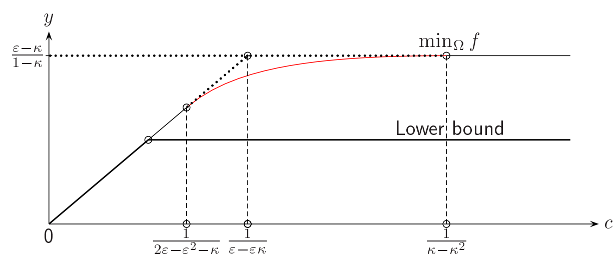

Fig. 2. Behaviour of $\min_{\Omega} f(p_1, p2, \varepsilon_2)$ and the corresponding lower bound