Beyond 128 Bits: The Concrete Security of EKE

Jiawei Bao[1], Tibor Jager[2], Eike Kiltz[3], Aysan Nishaburi[3], Samin Nooripoor[3], and Jiaxin Pan[1]

[1] University of Kassel, Kassel, Germany
[2] Bergische Universität Wuppertal, Wuppertal, Germany
[3] Ruhr-Universität Bochum, Bochum, Germany

Abstract. Can a relevant cryptographic primitive, when instantiated over the NIST P-256 elliptic curve, achieve a bit-security level exceeding 128 bits? Yes. We formally prove that the well-known password-authenticated key exchange protocol EKE, introduced by Bellovin and Merritt (S&P 1992), achieves a generic security level of $128 + \frac{1}{2}\log_2(N)$ bits, where N denotes the size of the password space.

To prove this result, we introduce and develop a new approach for showing that breaking a cryptosystem with a prescribed advantage requires solving *many* instances of an underlying computational assumption. To this end, we formulate the Hidden-Target Diffie–Hellman assumption. In this assumption, the adversary is given a set of N Diffie–Hellman challenge instances. The Diffie–Hellman key of one uniformly random instance is designated as the hidden target. The adversary does not know which instance is the target, but may output an arbitrary subset of candidate solutions and succeeds only if this subset contains the target. We formally prove that breaking the Hidden-Target Diffie–Hellman assumption with probability greater than $(k-1)/N$ requires solving at least k of the N Diffie–Hellman instances.

We then show that the security of EKE in the ideal-cipher model is equivalent to the Hidden-Target Diffie–Hellman assumption. A somewhat surprising consequence of this equivalence is that EKE achieves the claimed generic security level of $128 + \frac{1}{2}\log_2(N)$ bits. Moreover, the equivalence implies that EKE remains secure even in settings where the hardness of DLOG or CDH is weaker than expected: an adversary may still need to solve on the order of hundreds or thousands of discrete logarithm instances in order to succeed, a task that may remain infeasible even for powerful attackers, including those equipped with early quantum computers.

1 Introduction

Basing security on weakest-possible assumptions. In modern cryptography, we aim to establish the security of cryptosystems based on the weakest possible hardness assumptions, making them as realistic and well-founded as possible. This is desirable because it increases the confidence that the assumed difficulty will hold even as cryptanalytic research and computational capabilities evolve. Security based on weak assumptions is also more robust against

unforeseen algorithmic breakthroughs. In cryptography over algebraic groups, the *Discrete Logarithm* (DLOG) and the *Computational Diffie-Hellman* (CDH) assumption are the weakest commonly used hardness assumptions. DLOG asserts that, given a generator g of a cyclic group $\mathbb{G}$ of prime order p and a random element g^x, computing x is computationally infeasible. CDH asserts hardness of computing g^{xy}, given two random group elements g^x and g^y. Generic attacks on DLOG and CDH over prime-order groups have matching upper and lower bounds on their running times $T = \Theta(\sqrt{p})$. Therefore it is widely accepted that a cryptosystem instantiated, for example, over the *NIST P-256 elliptic curve* provides at most $\log p/2 = 128$ *bits of security.*[4] *In this work we challenge this belief.*

Achieving security when hardness assumptions fail. As long as we lack the ability to prove useful lower bounds on the complexity of computational problems used in cryptography, the possibility that a hardness assumption may break down is a persistent threat. Indeed, cryptographic history has repeatedly shown how hardness assumptions can turn out to be wrong. This includes, for example, the broken security of ciphers like RC4 [6,24], the collision vulnerabilities in hash functions such as MD5 [42,43,45], and the collapses of schemes like SIKE [20,33,37] and Rainbow [15].

While DLOG and CDH have remained resistant to attacks for decades, cryptanalytic advances through improved algorithms, new mathematical insights, or the eventual realization of a cryptanalytically relevant quantum computer that is able to run Shor's algorithm [40] for practical algebraic groups could change this. A fundamental question in cryptography is whether and how security can still be achieved even if weak hardness assumptions, such as DLOG or CDH, eventually fail. *Is it possible to define even weaker hardness assumptions than* DLOG *or* CDH*, which are still useful for the security analysis of practical schemes?*

Motivation of hidden-target security. At first glance, achieving security under such conditions may seem impossible, because DLOG and CDH seem to be about the weakest hardness assumptions one can reasonably make over algebraic groups. However, a promising approach, which is formalized by the hidden-target assumptions considered in this paper, is to investigate whether breaking a cryptosystem with good advantage requires solving *many* challenge instances of DLOG or CDH.

It is very conceivable that if DLOG and CDH eventually fail for groups commonly used in practice, then in such a way that breaking them requires very substantial, but not unattainable, computational resources. The currently most concrete example for this is the impact of quantum computing on the hardness of DLOG: in theory, a sufficiently large quantum computer can solve DLOG efficiently. However, in practice, building and operating a cryptanalytically relevant

[4] In April 2026, we asked a group of 50 senior cryptographers from the Dagstuhl seminar "Privacy-Preserving Authentication" about the provable security level of a standard PAKE protocol instantiated over NIST P-256. Roughly 30 respondents guessed less than 128 bits, about 20 guessed exactly 128 bits, and none guessed more than 128 bits.

quantum computer will likely be extraordinarily costly, at least in the foreseeable future. *Do cryptographic protocols provide meaningful security even when* DLOG *and* CDH *become solvable, but only at high costs?*

If an adversary must compute, say, hundreds or thousands of discrete logarithms to succeed, then even a computationally very powerful attacker might be practically limited by resource constraints. If the expected costs of a successful attack exceed the expected gain, then performing an attack becomes economically irrational. For example, if an adversary needs to run their quantum computer hundreds or thousands of times to break a scheme, the total cost of the attack may exceed the practical limits of even well-funded adversaries. This shifts the focus from an absolute notion of hardness to a more fine-grained notion, where an attack is not prevented outright, but still remains impractical due to overwhelming computational requirements.

1.1 Our contributions

This paper makes the following contributions. We introduce the *hidden-target* CDH assumption, which incorporates an information-theoretic component into the winning condition. This assumption formalizes the intuition that achieving a higher security advantage necessitates breaking multiple independent instances of the underlying problem. We show that any adversary that succeeds with probability greater than $(k-1)/N$ must solve at least k (out of N) instances of the CDH problem. We then apply this assumption to the security analysis of the EKE variant originally proposed by Bellare and Merritt [14] and further studied by Bellare, Pointcheval, and Rogaway [10], within a variant of the BPR model [10]. This yields strictly stronger concrete security bounds than those obtained in prior reductions. Figure 1 provides a schematic overview of the relationships among our results. We conclude this section with a technical overview of our contributions.

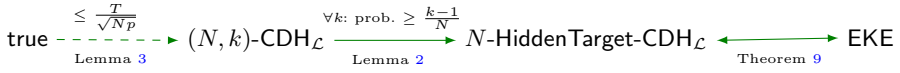


Fig. 1. Relations among the k-out-of-N CDH ((N,k)-CDH$_{\mathcal{L}}$) assumption, the Hidden-Target CDH (N-HiddenTarget-CDH$_{\mathcal{L}}$) assumption, and the security of EKE in the BPR model. Here, N represents the size of the password space and p the group order. Solid arrows indicate tight implications, while the dashed arrow represents a lower bound in the generic group model.

Hidden-Target Diffie-Hellman. Fix a group $\mathbb{G}$ of prime-order p with generator g. We define the N-HiddenTarget-CDH$_{\mathcal{L}}$ problem for an integer N as follows (see Section 3 for details).

1. The security experiment chooses random $x_1, \ldots, x_N, y_1, \ldots, y_N \xleftarrow{\$} \mathbb{Z}_p$ and runs the adversary $\mathcal{A}$ on input $g^{x_1}, \ldots, g^{x_N}, g^{y_1}, \ldots, g^{y_N}$. Intuitively, the input of $\mathcal{A}$ corresponds to N independent instances (g^{x_i}, g^{y_i}) of the CDH problem. Furthermore, it picks a uniformly random hidden index $\tau \xleftarrow{\$} [N]$.
2. The adversary outputs a list $\mathcal{L} = \{Z_1, \ldots, Z_q\}$ of group elements. If this list contains the "hidden target" $g^{x_\tau y_\tau}$, then the adversary wins.

The N-Hidden-Target Diffie–Hellman problem can also be interpreted as a variant of a *Merkle's Puzzle* [35], in which each puzzle corresponds to an independent Diffie–Hellman instance rather than a symmetric ciphertext. We consider hidden-target assumptions to be of independent interest, and we anticipate that they will have applications beyond those investigated in this work. Investigating their connection to Merkle's Puzzles and, more broadly, to fine-grained cryptography [21] appears to be a promising direction for future research.

HARDNESS OF THE N-$\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}}$ PROBLEM. Due to the random choice of index τ, the success probability of the adversary is information-theoretically bounded by k/N, where k is the number of correct CDH-solutions contained in the list, i.e., $k = |\{i \in [N] \mid g^{x_i y_i} \in \mathcal{L}\}|$. In Lemma 2 we formally prove that, in order to achieve a success probability greater than $(k-1)/N$, the adversary has to solve at least k instances of the CDH problem. More formally, for any $k \in [N]$,

$$\mathsf{Adv}^{N\text{-}\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}}}(\mathcal{A}) \leq \frac{k-1}{N} + \mathsf{Adv}^{(N,k)\text{-}\mathsf{CDH}_{\mathcal{L}}}(\mathcal{A}). \tag{1}$$

The best known attack against (N,k)-$\mathsf{CDH}_{\mathcal{L}}$ is to solve k discrete logarithm instances which can be done in time roughly $\sqrt{kp}$ using the algorithm by Kuhn and Struik [31]. The upper bound implied by the attack matches the known lower bounds of the (N,k)-CDH problem [8] in the generic group [41]. In Lemma 3 we prove that the advantage of a generic group adversary $\mathcal{A}$ making at most T group operations is roughly upper bounded as

$$\mathsf{Adv}^{N\text{-}\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}}}(\mathcal{A}) \lesssim \min_{k \in [N]} \left(\frac{k-1}{N} + \left(\frac{T^2}{kp} \right)^k \right) \lesssim \frac{T}{\sqrt{Np}}. \tag{2}$$

A consequence of (2) is that solving the N-$\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}}$ problem with constant success probability requires a running time of $T = \sqrt{Np}$. In other words, the N-$\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}}$ problem offers $\frac{1}{2}\big(\log_2(p) + \log_2(N)\big)$ bits of security. This is in sharp contrast to the known generic bound $T = \sqrt{p}$ for solving the CDH problem [41] which implies $\frac{1}{2}\log_2(p)$ bits of security for CDH. Putting it differently, the N-$\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}}$ problem provides $\frac{1}{2}\log_2(N)$ additional bits of security compared to CDH.

HOW TO USE HIDDEN-TARGET SECURITY: SECURITY OF EKE. The EKE (*encrypted key exchange*) protocol framework was originally introduced by Bellovin and Merritt [14]. Their work considered several protocols, namely $\mathsf{RSA\text{-}EKE}$, $\mathsf{ElGamal\text{-}EKE}$, and $\mathsf{DH\text{-}EKE}$. We will focus on $\mathsf{DH\text{-}EKE}$, which is a rather simple

protocol: Essentially, a user U and a server S perform a standard Diffie-Hellman key exchange, where the user sends $X = g^x$ and the server replies with $Y = g^y$, except that X and Y are encrypted with the password pw. Thus, the user sends $\hat{X} := \mathsf{E}(\mathsf{pw}, X)$, and the server replies with $\hat{Y} := \mathsf{E}(\mathsf{pw}, Y)$. The key is then derived as $\mathsf{SK} := g^{xy}$.

Bellare, Pointcheval, and Rogaway [10] considered a variant of the DH-EKE protocol (called EKE2), where in addition the user identities and the message transcript are used to derive the session key as $\mathsf{SK} := \mathsf{H}(\mathsf{U}, \mathsf{S}, \hat{X}, \hat{Y}, g^{xy})$. We will consider a variant where additionally the password is used for key derivation, i.e., $\mathsf{SK} := \mathsf{H}(\mathsf{U}, \mathsf{S}, \hat{X}, \hat{Y}, g^{xy}, \mathsf{pw})$ (see Section 5 for a detailed description), because this will allow us to get a tighter proof and thus better concrete security. In the sequel, we will consider this variant of DH-EKE and refer to it simply as EKE.

Proving the Security of EKE using N-HiddenTarget-CDH$_{\mathcal{L}}$. We use the standard game-based BPR security model for PAKE protocols [10]. Previous work showed that, in the ideal cipher model, EKE is secure under the CDH assumption [10] and tightly secure under the strong CDH assumption [36].

Following the standard approach of the BPR security model [10], we will first consider a setting where the password space is $\{\mathsf{pw}_1, \ldots, \mathsf{pw}_N\}$ and the password is chosen as pw_τ, for τ chosen uniformly from $\{1, \ldots, N\}$. We will show that the PAKE security experiment with EKE can be simulated by a reduction in a way such that every password pw_i corresponds to one instance (g^{x_i}, g^{y_i}) from the N-HiddenTarget-CDH$_{\mathcal{L}}$ problem. We will show that for a protocol session where the adversary is *passive*, it can only test the correctness of a password guess pw_i by providing a solution $g^{x_i y_i}$ to the corresponding CDH instance. Our reduction outputs all these solutions to the N-HiddenTarget-CDH$_{\mathcal{L}}$ experiment. If the PAKE-adversary outputs the "hidden target" $g^{x_\tau y_\tau}$, then we have solved the N-HiddenTarget-CDH$_{\mathcal{L}}$ problem. Like previous security proofs [10, 36], ours uses the ideal cipher model.

We stress that the above is just a rough sketch of the main approach. In Section 5, we will prove *active* PAKE security from N-HiddenTarget-CDH$_{\mathcal{L}}$, though the proof is significantly more involved than in the passive case.

Why we use the BPR security model instead of UC. We adopt the game-based BPR security model rather than a UC-based framework for two fundamental reasons.

1. The primary reason is that UC-based security models (e.g., [9, 19]) are inherently "double-quantifier" models in nature: they require that *for all* adversaries $\mathcal{A}$ *there exists* a simulator $\mathcal{S}$ such that some advantage function is low. As argued in [11, Page 4], this double-quantifier structure complicates concrete-security analysis; we refer the reader to [11] for a more detailed discussion.
   We remark that [19, Appendix A] proves for PAKE that UC security implies BPR security. Moreover, inspecting the proof, one observes that it does not require the UC simulator to run in polynomial time. It may therefore seem

that a concrete UC-security proof for EKE immediately yields a concrete BPR-security proof for EKE, regardless of the simulator's running time. This reasoning, however, is not sound. The reason is that the concrete running time of the simulator may become relevant (and for typical security proofs in UC does become relevant) when the concrete security bounds of the UC-based security proof are considered. That is, while the concrete running time of the simulator is indeed not relevant in the step from BPR security to UC security that is considered in [19, Appendix A], it is relevant for the step from UC security to some hardness assumption, such as CDH, for example. In particular, this is where the "double-quantifier" appears. Of course, this step must also be considered when estimating concrete security bounds for the protocol over the entire reduction chain, from BPR-security, through UC-security, to the underlying hardness assumption(s).

2. A nice property of UC-based security definitions for (PA)KE [9, 19] is that they capture *forward secrecy*, i.e., the guarantee that session keys remain secure even if the adversary subsequently compromises the password. This seems to be an inherent property of UC definitions, i.e., it is unclear how to define UC-security for such protocols without forward secrecy.

   Our goal is to show, for certain relevant protocols and for a meaningful security definition, that it is possible to achieve better concrete security than one would expect. Such as, for example, beyond 128-bit security on a 256-bit curve. However, in order to be able to achieve this, the security definition that we consider for EKE must *not* include forward secrecy. This is because in standard PAKE protocols, including as EKE but also other PAKEs, such as CPace, solving *one single* Diffie–Hellman instance suffices for an adversary $\mathcal{A}$ to reconstruct the session key. Consequently, forward secrecy in EKE is in fact *equivalent* to solving a *single* CDH problem and security beyond the known generic bounds for CDH is impossible. Therefore, forward secrecy must be excluded from our security model in order to establish security under the weaker N-$\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}}$ assumption. However, note that BPR-security without forward secrecy is still a very meaningful security notion for PAKEs, even if it is weaker than notions that additionally include forward secrecy.

The main objective of this paper is to demonstrate the perhaps counterintuitive insight that, for natural protocols and security definitions, one can achieve stronger concrete security guarantees than might be expected. Importantly, we do not claim that the BPR model is preferable to UC for real-world PAKE protocols, or vice versa. Rather, we only argue that it is the appropriate model for achieving this objective.

Exact Security of EKE. We now give a concrete estimate of the exact security of EKE, instantiated over the NIST P-256 curve (i.e., $\log_2(p) \approx 256$) and assuming a password space of cardinality $N = 2^{32}$.

For adversaries not attacking forward secrecy, our new results apply and show that EKE achieves approximately $\frac{1}{2}\big(\log_2(p) + \log_2(N)\big) \approx 128 + 16 = 144$

bits of security. For adversaries attacking forward secrecy, the existing security guarantees of [10,36] remain applicable, implying that EKE provides 128 bits of security. In summary,

$$\text{bit security of } \mathsf{EKE} = \begin{cases} \frac{1}{2}\big(\log(p) + \log(N)\big) = 144 & : \text{without forward secrecy} \\ \frac{1}{2}\log(p) = 128 & : \text{with forward secrecy.} \end{cases}$$

Beyond uniform passwords. To focus on the use of hidden-target security in a security proof, we will start from the the standard BPR security model [10] that model password as chosen uniformly random (from a possibly relatively small space). However, it has been realized that real-world distributions of human passwords are not uniform, but typically follow a *Zipf distribution* [22,48]. Therefore, we will also explain how our results generalize to such password distributions. We formalize the notion of password distributions, adopt the BPR security model and the N-HiddenTarget-CDH$_{\mathcal{C}}$ assumption to such distributions, and give concrete security bounds for EKE in this setting in Appendix A.

1.2 Discussion and Related work

Quantum Annoyance. Our approach is related to the notion of *quantum annoyance*, formalized by Eaton and Stebila [23], which aims to prove that quantum computer attacks on certain protocols may be prohibitively expensive. Their work analyzed CPace [27] and has been extended in [44] to the KHAPE-HMQV PAKE protocol. Both of these works analyze the security of PAKE schemes within the generic group model (GGM), augmented with access to a discrete logarithm oracle. Consequently, their proofs rely on the idealized assumption that PAKE adversaries interact with the group solely in a black-box fashion and can compute discrete logarithms only in an isolated, one-at-a-time manner.

The key difference in our approach is that we do not rely on the generic group model in the actual security proof of the scheme, in particular not to model the idea of "quantum-annoyance". As already noted in [23], the GGM extended with a discrete logarithm oracle is not an adequate model of a quantum adversary, as it is conceivable that quantum algorithms exist that extend Shor's attack to the multi-instance setting, bypassing the assumptions underlying the generic group model and the one-query-per-DLOG restriction. This is a critical gap, even in the setting of classical computing, because we know that solving N independent DLOG instances in a generic group can be done faster than by solving N single instances separately, specifically by a factor of $\sqrt{N}$ due to batching techniques [31,47]. Another limitation is that quantum adversaries cannot perform group operations on superpositions of elements.

We overcome these limitations by introducing new, concrete hidden-target hardness assumptions that capture multi-instance security and do not rely on the generic group model or any other idealized model. We believe this captures the essence of multi-challenge security more accurately, as it eliminates the incorrect assumption that computing N discrete logarithms must necessarily involve running an individual DLOG solver N times.

EKE MAY ACHIEVE STRONGER SECURITY THAN OTHER PAKE PROTOCOLS. Based on our findings, a natural question arises: do other PAKE protocols also provide stronger security guarantees than previously established? Unfortunately, for a broad class of PAKE protocols, this is not the case. For instance, the security of the OEKE protocol [17]—an optimized variant of EKE —can be shown in the BPR model to be equivalent to the standard CDH assumption. In other words, the ability to compute just a *single* CDH key enables an offline dictionary attack, thereby compromising its security. Consequently, it is not possible to establish its security from the difficulty of solving multiple CDH instances, as was done in EKE. Another example is an asymmetric PAKE: the OPAQUE instantiation with DH-OPRF and HMQV proposed in [30], where the computation of a single discrete logarithm suffices to mount an offline password search.

The important CPace protocol [27], based on Jablon's SPEKE protocol [29], is the IRTF working group's recommended choice for (symmetric) PAKE. Prior work hinting towards CPace's quantum annoyance [23] suggest the possibility of stronger security proofs relying on the difficulty of computing many CDH instances (or related problems). As an initial step, we proved the *passive* security of CPace under the N-Hidden-Target Inversion Diffie-Hellman problem, a straightforward adaptation of our N-HiddenTarget-CDH$_{\mathcal{L}}$ problem to the Inversion Diffie-Hellman setting.[5] However, CPace is only known to be *actively* secure under the Strong Simultaneous Diffie-Hellman (SSCDH) problem [3, 9].[6] Since SSCDH is a non-standard problem, establishing a connection between the active security of CPace and a suitable "SSCDH variant" of the Hidden-Target problem—one that can be analyzed in the generic group model—has remained elusive. We believe that recent generic lower bound techniques [5, 28] may be adaptable to yield a non-trivial upper bound in the generic group model. We encountered similar difficulties with the protocols PROTOSS [25] and the (asymmetric) KHAPE [26, 44]. We therefore leave proving stronger security guarantees of CPace, PROTOSS, and KHAPE as interesting open problems.

FURTHER RELATED WORK. The question of how security scales with the number of instances an adversary must solve for an attack has been considered before, but in different settings and using very different techniques. Bellare, Ristenpart, and Tessaro [13] considered the security of password hashing in a setting where it is computationally feasible to compromise a single instance, but the adversary aims to break N out of N instances. Auerbach, Giacon, and Kiltz [8] adopted this to the public-key setting. They considered how the security of Hashed ElGamal scales with k in a setting where an adversary aims to break k out of N instances. Here, every instance corresponds to one public key, thus, they consider how security scales across different users. In contrast, like [23, 44], we consider the question how security scales individually for a user for certain primitives.

[5] The N-Hidden-Target Inversion Diffie-Hellman problem is, given $(g^{a_1}, \ldots, g^{a_N})$, compute g^{1/a_τ} for a random hidden-target $\tau \in [N]$.

[6] The SSCDH problem is, given (g^{a_1}, g^{a_2}, g^x), compute a tuple of the form $(Y, Y^{x/a_1}, Y^{x/a_2})$ with the help of a (restricted) DDH oracle.

2 Preliminaries

For an integer $n \in \mathbb{N}$, let $[n] = \{1, \ldots, n\}$. We define $\mathbb{R}_{\geq 0} := [0, \infty)$. We use $[\![X]\!]$ to evaluate a Boolean term X to either 1 (true) or 0 (false). For a (finite) set $\mathcal{S}$, we write $s \xleftarrow{\$} \mathcal{S}$ for sampling an element s uniformly at random from $\mathcal{S}$. Similarly, for a probabilistic algorithm $\mathcal{A}$, we write $y \xleftarrow{\$} \mathcal{A}(x_1, \ldots, x_n)$ when $\mathcal{A}$ outputs y on input $(x_1, \ldots, x_n)$ using fresh random coins. For deterministic algorithms, we use $y \leftarrow \mathcal{A}(x_1, \ldots, x_n)$. All sets are initialized to the empty set unless stated otherwise. If an algorithm $\mathcal{A}$ has access to an oracle O, we write $\mathcal{A}^{\mathrm{O}}$.

3 Multi-Instance and Hidden Target CDH problems

Throughout this paper we assume the existence of a group description $\mathcal{G} = (\mathbb{G}, p, g)$, where $\mathbb{G}$ is a group of prime order p and g is a generator. Except for group generators, all group elements will be denoted with uppercase letters, e.g., $X = g^x$.

The Computational Diffie Hellman (CDH) problem is the task of computing g^{xy} from given group elements g^x and g^y, where x, y are chosen uniformly from $\mathbb{Z}_p$. The Strong Computational Diffie Hellman (StrongCDH) problem [16] is solving the CDH problem while having access to a restricted Diffie-Hellman oracle $\mathsf{DDH}_x(\hat{Y}, \hat{Z})$ that returns 1 iff $\hat{Y}^x = \hat{Z}$.

3.1 Multi-Instance Computational Diffie-Hellman

For $k \in [N]$, the CDH problem can be generalized to the k-out-of-N multi-instance setting [8], in which the adversary wins if it can solve k out of N given CDH challenges. Formally, we define the advantage function of the (N, k)-CDH problem as

$$\mathsf{Adv}^{(N,k)\text{-CDH}}(\mathcal{A}) := \Pr[(N, k)\text{-CDH} \Rightarrow 1],$$

where (N, k)-CDH is the game described in Figure 2 (left). In the "strong" version of the problem, denoted (N, k)-StrongCDH [8], the adversary furthermore is additionally given access to a restricted Diffie-Hellman oracle $\mathsf{DDH}_{x_1,\ldots,x_N}(i, \hat{Y}, \hat{Z})$, which outputs 1 iff $\hat{Y}^{x_i} = \hat{Z}$. The advantage function $\mathsf{Adv}^{(N,k)\text{-StrongCDH}}(\mathcal{A})$ is defined accordingly.

For stating our results it is more convenient to defined a "list-version" of the (N, k)-CDH problem, denoted as (N, k)-$\mathsf{CDH}_{\mathcal{L}}$. In (N, k)-$\mathsf{CDH}_{\mathcal{L}}$ the adversary outputs a list $\mathcal{L} = \{Z_1, \ldots Z_q\}$ of group elements which requires to contain the k CDH solutions (at arbitrary positions). We define the advantage function of the (N, k)-$\mathsf{CDH}_{\mathcal{L}}$ problem as

$$\mathsf{Adv}^{(N,k)\text{-CDH}_{\mathcal{L}}}(\mathcal{A}) := \Pr[(N, k)\text{-CDH}_{\mathcal{L}} \Rightarrow 1],$$

where (N, k)-$\mathsf{CDH}_{\mathcal{L}}$ is the game described in Figure 2 (right).

The next lemma states that (N, k)-$\mathsf{CDH}_{\mathcal{L}}$ is at least as hard as (N, k)-StrongCDH.

Lemma 1. *For every adversary $\mathcal{A}$ against the (N,k)-$\mathsf{CDH}_{\mathcal{L}}$ problem, there exists adversary $\mathcal{B}$ against the (N,k)-$\mathsf{StrongCDH}$ problem such that*

$$\mathsf{Adv}^{(N,k)\text{-}\mathsf{CDH}_{\mathcal{L}}}(\mathcal{A}) \leq \mathsf{Adv}^{(N,k)\text{-}\mathsf{StrongCDH}}(\mathcal{B}) \ .$$

The number of $\mathcal{B}$'s calls to the DDH *oracle is $N \cdot |\mathcal{L}|$.*

Proof. We sketch the simple proof. We build adversary $\mathcal{B}$ against the (N,k)-$\mathsf{StrongCDH}$ problem such that it runs adversary $\mathcal{A}$ on the same inputs. Given the output $\mathcal{L} = \{Z_1, \ldots, Z_q\}$ of adversary $\mathcal{A}$, adversary $\mathcal{B}$ can correctly identify the k Diffie-Hellman solutions using $N \cdot |\mathcal{L}|$ calls to its DDH oracle. Clearly, if $\mathcal{A}$ wins the (N,k)-$\mathsf{CDH}_{\mathcal{L}}$ game, then $\mathcal{B}$ wins the (N,k)-$\mathsf{StrongCDH}$ game.

Game (N,k)-CDH	**Game** (N,k)-$\mathsf{CDH}_{\mathcal{L}}$
01 **for** $i \in [N]$	06 **for** $i \in [N]$
02 $\quad x_i, y_i \xleftarrow{\$} \mathbb{Z}_p$	07 $\quad x_i, y_i \xleftarrow{\$} \mathbb{Z}_p$
03 $(\mathcal{F}, Z_1, \ldots, Z_N) \xleftarrow{\$} \mathcal{A}((g^{x_i}, g^{y_i})_{i\in[N]})$	08 $\mathcal{L} := (Z_1, \ldots, Z_q) \xleftarrow{\$} \mathcal{A}((g^{x_i}, g^{y_i})_{i\in[N]})$
04 $\ell := \lvert\mathcal{F}\rvert$	09 $\ell := \lvert\{i \in [N] \mid g^{x_i y_i} \in \{Z_1, \ldots, Z_q\}\rvert$
05 **return** $[\![\ell \geq k]\!] \wedge \bigwedge_{i\in\mathcal{F}} [\![Z_i = g^{x_i y_i}]\!]$	10 **return** $[\![\ell \geq k]\!]$

Fig. 2. The (N,k)-CDH game, where set $\mathcal{F} \subseteq [N]$ contains the indices of the correct Diffie-Hellman keys (left) and its list version, the (N,k)-$\mathsf{CDH}_{\mathcal{L}}$ game (right), where $\mathcal{L}$ contains a list of possible Diffie-Hellman keys.

3.2 Hidden-Target Computational Diffie Hellman

We extend the multi-instance CDH problem with an information-theoretic perspective: the adversary must produce a CDH solution for one of the given Diffie–Hellman instances, where the specific target instance is selected uniformly at random from the N instances and remains information-theoretically hidden from the adversary.

Formally, we define the advantage function of the N-$\mathsf{HiddenTarget}$-$\mathsf{CDH}_{\mathcal{L}}$ problem as

$$\mathsf{Adv}^{N\text{-}\mathsf{HiddenTarget}\text{-}\mathsf{CDH}_{\mathcal{L}}}(\mathcal{A}) := \Pr[N\text{-}\mathsf{HiddenTarget}\text{-}\mathsf{CDH}_{\mathcal{L}} \Rightarrow 1],$$

where N-$\mathsf{HiddenTarget}$-$\mathsf{CDH}_{\mathcal{L}}$ is the game described in Figure 3.

The following lemma shows that achieving success probability greater than $\frac{k-1}{N}$ for the N-$\mathsf{HiddenTarget}$-$\mathsf{CDH}_{\mathcal{L}}$ problem necessarily requires solving the (N,k)-$\mathsf{CDH}_{\mathcal{L}}$ problem.

Lemma 2. *Let N be an integer and let $\mathcal{A}$ be an adversary against the N-*$\mathsf{HiddenTarget}$-$\mathsf{CDH}_{\mathcal{L}}$ *problem. Then, for any $k \in [N]$,*

$$\mathsf{Adv}^{N\text{-}\mathsf{HiddenTarget}\text{-}\mathsf{CDH}_{\mathcal{L}}}(\mathcal{A}) \leq \frac{k-1}{N} + \mathsf{Adv}^{(N,k)\text{-}\mathsf{CDH}_{\mathcal{L}}}(\mathcal{A}).$$

Game N-HiddenTarget-CDH$_{\mathcal{L}}$

01 **for** $i \in [N]$
02 $\quad x_i, y_i \xleftarrow{\$} \mathbb{Z}_p$
03 $(Z_1, \ldots, Z_q) \xleftarrow{\$} \mathcal{A}((g^{x_i}, g^{y_i})_{i \in [N]})$
04 $\tau \xleftarrow{\$} [N]$
05 **return** $[\![g^{x_\tau y_\tau} \in \{Z_1, \ldots, Z_q\}]\!]$

Fig. 3. The N-HiddenTarget-CDH$_{\mathcal{L}}$ game.

Proof. Let $\mathcal{A}$ be an adversary against the N-HiddenTarget-CDH$_{\mathcal{L}}$ problem and let $k \in [N]$. We analyze the same adversary $\mathcal{A}$ in the (N,k)-CDH$_{\mathcal{L}}$ game which shares the same input/output interface as the N-HiddenTarget-CDH$_{\mathcal{L}}$ game but differs in its winning condition. Let $\mathcal{L} = \{Z_1, \ldots, Z_q\}$ be the the group elements output by the adversary and let

$$\ell := |\{i \in [N] \mid g^{x_i y_i} \in \mathcal{L}\}|$$

be the number of CDH solutions output by the adversary. Conditioning on ℓ, we obtain

$$\begin{aligned}\Pr[N\text{-HiddenTarget-CDH}_{\mathcal{L}} \Rightarrow 1] &= \Pr[g^{x_\tau y_\tau} \in \mathcal{L}] \\ &\leq \Pr[\ell \geq k] + \Pr[g^{x_\tau y_\tau} \in \mathcal{L} \mid \ell \leq k-1] \\ &= \mathsf{Adv}^{(N,k)\text{-CDH}_{\mathcal{L}}}(\mathcal{A}) + \Pr[g^{x_\tau y_\tau} \in \mathcal{L} \mid \ell \leq k-1].\end{aligned}$$

Finally, since $\tau \xleftarrow{\$} [N]$ is completely hidden from $\mathcal{A}$'s view,

$$\Pr[g^{x_\tau y_\tau} \in \mathcal{L} \mid \ell \leq k-1] = \frac{\ell}{N} \leq \frac{k-1}{N}.$$

Note that the latter bound even holds in case of a collision in the Diffie-Hellman keys, i.e., if $g^{x_i y_i} = g^{x_j y_j}$ for $i \neq j$ holds.

3.3 Lower Bounds in the Generic Group Model

The generic group model [34,41] is an idealized framework for analyzing cryptographic assumptions in which group elements are represented as random encodings, and algorithms can only manipulate them through oracle-provided group operations (addition, inversion, equality testing). This abstraction prevents algorithms from exploiting any special structure of the encoding, ensuring that only the algebraic properties of the group are accessible.

In order to simplify our bounds, in this section we will use "$\approx$" and "$\lesssim$" notation which ignores small constants and lower-order terms.

We state our main result of this subsection, a lower bound on the success probability for any attacker against the N-HiddenTarget-CDH$_{\mathcal{L}}$ problem.

Lemma 3. *For any generic group adversary $\mathcal{A}$ making at most T queries to the group oracles we have:*

$$\mathsf{Adv}^{N\text{-}\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}}}(\mathcal{A}) \lesssim \min_{k\in[N]}\left(\frac{k-1}{N} + \left(\frac{T^2}{kp}\right)^k\right) \lesssim \frac{N^{\frac{1}{2N}}}{\sqrt{Np}}\cdot T \approx \frac{T}{\sqrt{Np}} \ .$$

To prove Lemma 3, we will require two more helper lemmas.

First we recall the generic lower bounds from [8,46] on attacking the (N,k)-$\mathsf{StrongCDH}$ problem.

Lemma 4. *For any generic group adversary $\mathcal{A}$ making at most T queries to the group oracles and q_{DDH} queries to the DDH oracle we have*

$$\mathsf{Adv}^{(N,k)\text{-}\mathsf{StrongCDH}}(\mathcal{A}) \lesssim \left(\frac{T^2+q_{\mathsf{DDH}}}{kp}\right)^k \ .$$

By Lemma 1, the generic lower bound also applies to the list version (N,k)-$\mathsf{CDH}_{\mathcal{L}}$. Additionally, we note that by Lemma 1, one requires $T \geq \Omega(\sqrt{Np})$ queries to the generic group oracle to break (N,k)-$\mathsf{StrongCDH}$ with overwhelming advantage, implying $T \geq N$.

Finally, we require the following analytic lemma that we will prove in Section B.

Lemma 5. *For integers $N \leq p$, define $T_{max} := \frac{\sqrt{Np}}{N^{\frac{1}{2N}}}$. Then, for any integer $T \in [T_{max}]$,*

$$\min_{k\in[N]}\left(\frac{k-1}{N} + \left(\frac{T^2}{kp}\right)^k\right) \leq \frac{1}{T_{max}}\cdot T.$$

For intuition as to why the lemma holds, see Figure 4. The lemma states that the blue minimum curve is upper bounded by the dashed orange line.

We are now ready to prove Lemma 3.

Proof (of Lemma 3). By first applying Lemma 2 and then combining Lemma 1 with Lemma 4 (with $q_{\mathsf{DDH}} \leq N|\mathcal{L}|$), for any $k \in [N]$ we obtain the generic lower bound

$$\begin{aligned}\mathsf{Adv}^{N\text{-}\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}}}(\mathcal{A}) &\leq \frac{k-1}{N} + \mathsf{Adv}^{(N,k)\text{-}\mathsf{CDH}_{\mathcal{L}}}(\mathcal{A}) \\ &\lesssim \frac{k-1}{N} + \left(\frac{T^2+N|\mathcal{L}|}{kp}\right)^k \\ &\lesssim \frac{k-1}{N} + \left(\frac{T^2}{kp}\right)^k \ .\end{aligned}$$

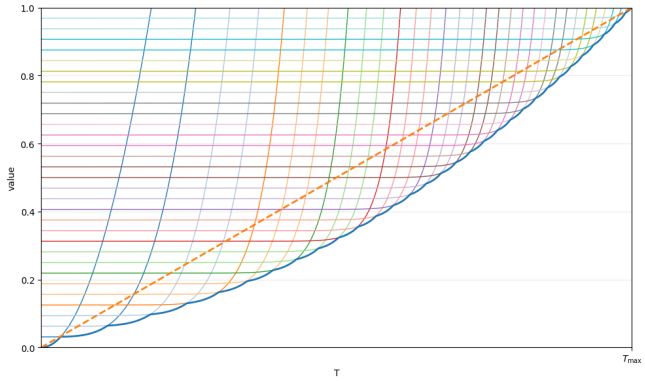


Fig. 4. This plot displays the probabilities $f_k(T) = \frac{k-1}{N} + \left(\frac{T^2}{kp}\right)^k$ for $k \in [N]$, where different colors correspond to different values of k, their point-wise minimum over k (bold blue), and the linear function $g(T) = T/T_{max}$ (dashed orange) on the interval $T \in [0, T_{max}]$. Note that the dashed orange line is always above the blue line (as proven in Lemma 5).

For the last inequality we used that, without loss of generality, $N \leq T$. This implies $N|\mathcal{L}| \leq T^2$, where $\mathcal{L}$ is the list output by the (N,k)-$\mathsf{CDH}_{\mathcal{L}}$ adversary. As the generic group bound holds for all $k \in [N]$, we can upper bound it by the minimum over all k:

$$\mathsf{Adv}^{N\text{-HiddenTarget-CDH}_{\mathcal{L}}}(\mathcal{A}) \lesssim \min_{k\in[N]} \left(\frac{k-1}{N} + \left(\frac{T^2}{kp}\right)^k \right) ,$$

This proves the first inequality of Lemma 3. The second inequality follows by applying Lemma 5 and observing that $N^{1/(2N)}$ is upper bounded by $3^{1/6} \approx 1.2$.

4 Password Authenticated Key Exchange

We consider *two-message* PAKE protocols, consisting of algorithms $\mathsf{PAKE} := (\mathsf{Init}, \mathsf{Resp}, \mathsf{TerInit})$.

Let U and S be two users of PAKE sharing a password pw. In the client-server setting, a session begins with U running $(\mathsf{M_U}, \mathsf{st}) \xleftarrow{\$} \mathsf{Init}(\mathsf{U}, \mathsf{S}, \mathsf{pw})$ to compute a message $\mathsf{M_U}$ and an internal session state st on input of the password and the identities of the client and the server. The message $\mathsf{M_U}$ is then sent to S. Upon receiving $\mathsf{M_U}$, S runs $(\mathsf{M_S}, \mathsf{SK_S}) \xleftarrow{\$} \mathsf{Resp}(\mathsf{U}, \mathsf{S}, \mathsf{M_U}, \mathsf{pw})$ to generate a server message $\mathsf{M_S}$ and a session key $\mathsf{SK_S}$. The server then sends $\mathsf{M_S}$ to U. Finally, U executes $\mathsf{SK_U} \xleftarrow{\$} \mathsf{TerInit}(\mathsf{st}, \mathsf{M_S})$ to compute the session key $\mathsf{SK_U}$. See Figure 5 for an illustration.

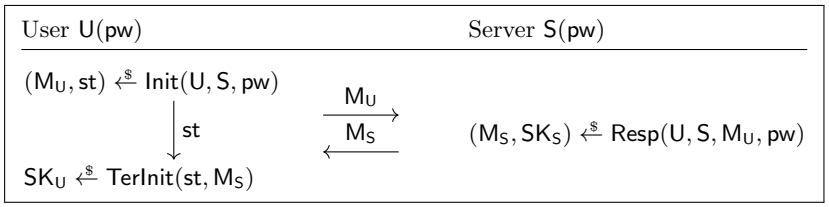


Fig. 5. A two-message PAKE protocol between a user U and a server S.

For correctness, we require that an honest protocol execution will produce the same session key $\mathsf{SK_U} = \mathsf{SK_S}$ for the user and the server, i.e.,

$$\Pr\left[\mathsf{SK_U} = \mathsf{SK_S} \;\middle|\; \begin{array}{c}(\mathsf{M_U}, \mathsf{st}) \xleftarrow{\$} \mathsf{Init}(\mathsf{U}, \mathsf{S}, \mathsf{pw}) \\ (\mathsf{M_S}, \mathsf{SK_S}) \xleftarrow{\$} \mathsf{Resp}(\mathsf{U}, \mathsf{S}, \mathsf{M_U}, \mathsf{pw}) \\ \mathsf{SK_U} \xleftarrow{\$} \mathsf{TerInit}(\mathsf{st}, \mathsf{M_S})\end{array}\right] = 1\ .$$

Security Model of PAKE. We define the two-party variant of the Bellare-Pointcheval-Rogaway (BPR) model [1, 2, 10], where security is defined with respect to a single user–server pair. Our model does not allow password corruption. Using a hybrid argument over the number pairs of user and servers, we can show that our model implies security even in a setting with corruptions, provided that neither the user nor the server in the Test session is corrupted. The reduction loses a factor of the number of users times the number of servers. Forward secrecy is deliberately not captured by our model; for a discussion, see the introduction. Similar, no-forward-secrecy variants have been used in [4,17,18], but their models only allowed at most one Test query and ours allows multiple Test queries.

In the BPR model, we consider a set of users $\mathcal{U}$ and a set of servers $\mathcal{S}$, which are disjoint. Oracles in this model reject when queries are inconsistent with names.

We denote the session key space by $\mathcal{SK}$. The password space is defined as $\mathcal{PW} = \{\mathsf{pw}_1, \ldots, \mathsf{pw}_N\}$. A user and server pair $(\mathsf{U}, \mathsf{S}) \in \mathcal{U} \times \mathcal{S}$ holds a shared secret password $\mathsf{pw} \xleftarrow{\$} \mathcal{PW}$.

Let P be a party (either a user or a server). Each party in $\mathcal{U} \cup \mathcal{S}$ has multiple instances π^i_{P} (i is some index) and each instance has its internal state. The state of an instance π^i_{P} is a tuple $(\mathtt{e}, \mathtt{tr}, \mathtt{key}, \mathtt{acc}, \mathtt{test})$ where

- $\mathtt{e}$ is the ephemeral secret of P.
- $\mathtt{tr}$ is the trace of the instance, that is, the user and the server names in the instance and the messages sent and received by P in the instance.
- $\mathtt{key}$ is the session key of π^i_{P}.
- $\mathtt{acc}$ is a Boolean value indicating whether the instance has accepted the session key. If the instance did not receive the last message, then $\mathtt{acc}$ is undefined, denoted by $\mathtt{acc} = \perp$.
- $\mathtt{test}$ is a Boolean value indicating whether an adversary has asked a Test query on the instance.

To access individual components of the state, we write $\pi_{\mathsf{P}}^{i}.\mathsf{x}$, where $\mathsf{x} \in \{\mathtt{e}, \mathtt{tr}, \mathtt{key}, \mathtt{acc}, \mathtt{test}\}$. We define partnership via matching instance trace.

Definition 6 (Partnering). *A user instance $\pi_{\mathsf{U}}^{t_0}$ and a server instance $\pi_{\mathsf{S}}^{t_1}$ are partnered if and only if*

$$\pi_{\mathsf{U}}^{t_0}.\mathtt{acc} = \mathbf{true} = \pi_{\mathsf{S}}^{t_1}.\mathtt{acc} \quad \textbf{and} \quad \pi_{\mathsf{U}}^{t_0}.\mathtt{tr} = \pi_{\mathsf{S}}^{t_1}.\mathtt{tr}$$

Two user instances are never partnered, neither are two server instances. We define a partnership predicate $\mathsf{Partner}(\pi_{\mathsf{U}}^{t_0}, \pi_{\mathsf{S}}^{t_1})$ which outputs **true** *if and only if $\pi_{\mathsf{U}}^{t_0}$ and $\pi_{\mathsf{S}}^{t_1}$ are partnered.*

Security Game. The security game is defined between an adversary $\mathcal{A}$ and a fixed user-server pair $(\mathsf{U}^*, \mathsf{S}^*) \in \mathcal{U} \times \mathcal{S}$. The experiment first samples a random challenge bit $\beta \xleftarrow{\$} \{0,1\}$ and a password pw uniformly from the password space $\mathcal{PW}$, and then runs $\mathcal{A}$ on $(\mathsf{U}^*, \mathsf{S}^*)$. $\mathcal{A}$ is allowed to query the following oracles:

- Execute$(\mathsf{U}^*, t_1, \mathsf{S}^*, t_2)$: This oracle outputs the protocol messages of an honest protocol execution between instances $\pi_{\mathsf{U}^*}^{t_1}$ and $\pi_{\mathsf{S}^*}^{t_2}$. This models the passive attacks on a PAKE protocol.
- SendInit, SendResp, SendTerInit: These oracles model active attacks. By querying these oracles, the adversary sends protocol messages to protocol instances. For sake of simplicity, we assume that the adversary does not use these oracles to launch passive attacks (which are already captured by the Execute oracle).
- Reveal(P, t): By this oracle, the adversary reveals the session key of π_{P}^{t}.
- Test(P, t): If π_{P}^{t} is fresh (cf. Definition 7), then, depending on the challenge bit β, the oracle outputs either the session key of π_{P}^{t} or a uniformly random key. Otherwise, the oracle outputs $\perp$. After this query, the flag $\pi_{\mathsf{P}}^{t}.\mathtt{test}$ will be set as **true**.

We denote the game by $\mathsf{BPR}_{\mathsf{PAKE}}$. Note that since we are considering the two-party version of the $\mathsf{BPR}_{\mathsf{PAKE}}$, the oracles only answer queries involving the chosen user U^* and server S^*; all others return $\perp$. The pseudocode is given in $\mathbf{G}_0$ in Figure 7, instantiated with our PAKE protocol. Before defining PAKE security, we define freshness to avoid trivial attacks in this model.

Definition 7 (Freshness). *An instance π_{P}^{t} is fresh if*

1. *π_{P}^{t} was not queried* Test *or* Reveal*, and*
2. *at least one of the following conditions holds:*
   (a) *A unique partner instance exists, which was not queried to* Test *or* Reveal*.*
   (b) *There is no partner instance.*

Furthermore, π_{P}^{t} is also fresh if it has more than one partner instance.[7] *If none of the above hold, then π_{P}^{t} is not fresh.*

[7] In this way, we can define the requirement that a secure PAKE protocol should not have more than one partnered instance.

By these definitions, we are ready to define the security of PAKE protocols.

Definition 8 (Security of PAKE). *Let* PAKE *be a PAKE protocol and* $\mathcal{A}$ *be an adversary. The advantage of* $\mathcal{A}$ *against* PAKE *is defined as*

$$\mathsf{Adv}^{\mathsf{BPR}}_{\mathsf{PAKE}}(\mathcal{A}) := \left| \Pr\left[\mathsf{BPR}^{\mathcal{A}}_{\mathsf{PAKE}} \Rightarrow 1\right] - \frac{1}{2} \right| - \frac{q_{\mathsf{Act}}}{|\mathcal{PW}|} \ .$$

Here q_{Act} *is the number of active queries, namely, the total number of* SendResp *and* SendTerInit *queries made by* $\mathcal{A}$*.*

That is, a PAKE protocol is considered secure if the best an adversary can do is to perform an online dictionary attack.

5 The PAKE protocol EKE

5.1 The EKE protocol

Let $\mathcal{SK}$ be the session key space, $\mathsf{H}\colon \{0,1\}^* \to \mathcal{SK}$ be a hash function, and $\mathsf{IC} := (\mathsf{E}, \mathsf{D})$ be a block cipher whose key space includes $\mathcal{PW}$, with plaintext space and ciphertext space $\mathbb{G}$.[8] The two-message PAKE protocol $\mathsf{EKE} := (\mathsf{Init}, \mathsf{Resp}, \mathsf{TerInit})$ is defined in Figure 6. We assume that both the user and the server always generate fresh messages, i.e., messages that have not been sent or received before.

Alg $\mathsf{Init}(\mathsf{U},\mathsf{S},\mathsf{pw})$	**Alg** $\mathsf{TerInit}(\mathsf{U},\mathsf{S},\mathsf{st},\hat{Y},\mathsf{pw})$	**Alg** $\mathsf{Resp}(\mathsf{U},\mathsf{S},\hat{X},\mathsf{pw})$
01 $x \xleftarrow{\$} \mathbb{Z}_p$	06 **let** $(x,\hat{X}) := \mathsf{st}$	13 $y \xleftarrow{\$} \mathbb{Z}_p$; $Y := g^y$
02 $X := g^x$	07 **if** $\hat{X} = \hat{Y}$	14 $\hat{Y} := \mathsf{E}(\mathsf{pw}, Y)$
03 $\hat{X} := \mathsf{E}(\mathsf{pw}, X)$	08 **return** $\perp$	15 **if** $\hat{Y} = \hat{X}$
04 $\mathsf{st} := (x, \hat{X})$	09 $Y := \mathsf{D}(\mathsf{pw}, \hat{Y})$	16 **return** $\perp$
05 **return** $(\hat{X}, \mathsf{st})$	10 $K := Y^x$	17 $X := \mathsf{D}(\mathsf{pw}, \hat{X})$
	11 $\mathsf{SK} := \mathsf{H}(\mathsf{U},\mathsf{S},\hat{X},\hat{Y},K,\mathsf{pw})$	18 $K := X^y$
	12 **return** SK	19 $\mathsf{SK} := \mathsf{H}(\mathsf{U},\mathsf{S},\hat{X},\hat{Y},K,\mathsf{pw})$
		20 **return** $(\hat{Y}, \mathsf{SK})$

Fig. 6. The PAKE protocol EKE

The following theorem proves EKE secure modeling IC as an ideal cipher and H as a random oracle.

Theorem 9 (N-$\mathsf{HiddenTarget}$-$\mathsf{CDH}_{\mathcal{C}} \Rightarrow \mathsf{EKE}$). *Let* $N = |\mathcal{PW}|$ *be the cardinality of the password space. For every adversary* $\mathcal{A}$ *against* EKE *in the* BPR *model, starting at most* q_{Act} *active and* q_{Pas} *passive sessions and making at most* q_{IC}

[8] We refer to [39] for details on implementing such a block cipher.

queries to $\mathsf{IC} := (\mathsf{E}, \mathsf{D})$ *and* q_{H} *queries to* H*, there exists an adversary* $\mathcal{B}$ *against* N-$\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}}$ *with roughly the same running time as* $\mathcal{A}$ *such that*

$$\mathsf{Adv}^{\mathsf{BPR}}_{\mathsf{PAKE}}(\mathcal{A}) \leq \mathsf{Adv}^{N\text{-}\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}}}(\mathcal{B}) + (q_{\mathsf{IC}} + 2 \cdot q_{\mathsf{Pas}} + q_{\mathsf{Act}}) \cdot \left(\frac{2 \cdot q_{\mathsf{Pas}}}{p} + \frac{q_{\mathsf{Act}}}{p} \right),$$

where $|\mathcal{L}| \leq q_{\mathsf{H}}$.

We note that the original EKE protocol of [10] does not incorporate the password into the session key derivation. As a result, without hashing the password, we can only establish the weaker bound $|\mathcal{L}| \leq N \cdot q_{\mathsf{H}}$ and $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + O(N \cdot q_{\mathsf{H}})$. This is because the reduction cannot determine which password corresponds to a given hash query, which requires storing a larger number of elements in the list $\mathcal{L}$.

We further note that Theorem 9 is optimal, in the sense that the converse implication $\mathsf{EKE} \Rightarrow N$-$\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}}$ also holds, and hence the two notions are equivalent. In particular, given any adversary $\mathcal{B}$ against N-$\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}}$, one can construct an adversary $\mathcal{A}$ against EKE such that $\mathsf{Adv}^{N\text{-}\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}}}(\mathcal{B}) \leq \mathsf{Adv}^{\mathsf{BPR}}_{\mathsf{PAKE}}(\mathcal{A})$. Intuitively, $\mathcal{A}$ initiates a passive session to obtain $(\hat{X}, \hat{Y})$, and then runs $\mathcal{B}$ on the N Diffie–Hellman instances defined as $(X_i, Y_i) = (\mathsf{D}_{\mathsf{pw}_i}(\hat{X}), \mathsf{D}_{\mathsf{pw}_i}(\hat{Y}))$, for $i \in [N]$. Upon receiving $\mathcal{B}$'s output $g^{x_\tau y_\tau}$, the adversary $\mathcal{A}$ can finally distinguish the real session key of the test session from a random key. Thus, the theorem describes the *exact* security of EKE in this model.

5.2 Bit Security of EKE

Following the definition of Bellare and Ristenpart [12], A PAKE protocol has λ bits of security in the BPR model, if for all adversaries $\mathcal{A}$ the success ratio of $\mathcal{A}$, defined as $\mathsf{SR}(\mathcal{A}) := \mathsf{Adv}^{\mathsf{BPR}}_{\mathsf{PAKE}}(\mathcal{A})/\mathsf{Time}(\mathcal{A})$, is upper bounded by $2^{-\lambda}$.

Corollary 10. *In the generic group model,* EKE *achieves roughly* $\lambda = \frac{1}{2}(\log_2(p) + \log_2(N))$ *bits of security.*

Proof. Let $\mathcal{A}$ be a generic adversary against EKE. Combining Theorem 9 with Lemma 3 (and ignoring lower-order terms and constants), we obtain

$$\mathsf{Adv}^{\mathsf{BPR}}_{\mathsf{PAKE}}(\mathcal{A}) \lesssim \mathsf{Adv}^{N\text{-}\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}}}(\mathcal{B}) \lesssim \frac{T}{\sqrt{Np}} .$$

We can therefore bound the success ratio of $\mathcal{A}$ as

$$\mathsf{SR}(\mathcal{A}) \lesssim \frac{1}{\sqrt{Np}} = 2^{-\lambda} .$$

In contrast, earlier proofs of EKE achieved active security solely from the SCDH assumption, implying only $\frac{1}{2}\log_2(p)$ bits of security. Our result improves this by an additional $\frac{1}{2}\log_2(N)$ bits.

Concretely, instantiating EKE over the NIST P-256 curve (i.e., $\log_2(p) \approx 256$) and assuming a password space of cardinality $N = 2^{32}$, we obtain $\lambda = 128 + 16 = 144$ bits of security.

5.3 Proof of Theorem 9

We define a sequence of games in Figures 7 and 8 to show

$$N\text{-}\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}} \Rightarrow \mathsf{EKE},$$

where $N\text{-}\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}}$ is defined as in Figure 2.

Game 0. This is the original game. Lines 02 and 03 of Figure 7 are preparing for the following game changes and will be used in $\mathbf{G}_2$.

$$\Pr\left[\mathsf{BPR}^{\mathcal{A}}_{\mathsf{PAKE}} \Rightarrow 1\right] = \Pr[\mathbf{G}^{\mathcal{A}}_0 \Rightarrow 1] \ .$$

Game 1. Instead of using the $\mathsf{Freshness}$ procedure in TEST, we assign an additional variable $\mathtt{fr}$ to explicitly indicate the freshness of instance π. Whenever $\mathcal{A}$ issues an oracle query related to π, we will update $\pi.\mathtt{fr}$ in real time according to the freshness definition (cf. Definition 7). This change is conceptual, and we have

$$\Pr[\mathbf{G}^{\mathcal{A}}_0 \Rightarrow 1] = \Pr[\mathbf{G}^{\mathcal{A}}_1 \Rightarrow 1] \ .$$

Game 2. From this game on, our changes are depicted in Figure 8. We first change the decryption oracle $\mathsf{D}(\mathsf{pw}_j, \hat{A})$, namely, instead of directly sampling the random pre-image, we randomize g^{y_i} and return the rerandomization $(g^{y_i})^{r_{j,\hat{A}}}$. Similarly, we define a procedure $\mathsf{D}_x(\mathsf{pw}_j, \hat{A})$ that reprograms the decryption of $\hat{A}$ to $(g^{x_i})^{r_{j,\hat{A}}}$. With the help of this reprogramming technique, we program the decryption of $\hat{X}, \hat{Y}$.

The major change in this game is to change the $(X, \hat{X})$ and $(Y, \hat{Y})$ generated by the simulator. More precisely, we change EXECUTE by picking random $\hat{X}$ and $\hat{Y}$ and decrypting them to X and Y. Similar changes are also made to $(X, \hat{X})$ in SENDINIT and $(Y, \hat{Y})$ in SENDRESP.

Moreover, for $\hat{X}, \hat{Y}$ generated by EXECUTE queries, we decrypt them under all passwords via the procedure $\mathsf{D}_x(\hat{X})$ and oracle $\mathsf{D}(\hat{Y})$. For consistency, the modification to SENDRESP differs slightly depending on whether a $\hat{X}$ has been queried to the decryption oracle. This modification is to ensure in the last reduction we can obtain the CDH solution. After this modification, we change the simulation for all active $\hat{X}, \hat{Y}$ chosen by the adversary. For each $\textsc{SendResp}(\mathsf{S}^*, t_2, \mathsf{U}^*, \hat{X})$ query, we additionally define $\mathsf{D}_x(\mathsf{pw}_j, \hat{X}) := (g^{x_j})^{r_{j,\hat{X}}}$ for all $j \in [N]$, where $r_{j,\hat{X}} \xleftarrow{\$} \mathbb{Z}_p$, unless $\mathsf{D}_x(\mathsf{pw}_j, \hat{X})$ has been defined. This is done by programming $\mathcal{L}_{\mathsf{IC}}$. We generate $\hat{Y}$ in $\textsc{SendTerInit}(\mathsf{U}^*, t_1, \mathsf{S}^*, \hat{Y})$ queries in a similar way.

Now $\mathcal{A}$ can see the change only if before either $\hat{X}$ or $\hat{Y}$ is generated $\mathcal{A}$ has queried it as decryption or it is the output of a previous encryption query. We denote it as $\mathtt{BAD}_1$ and abort the game when it is true (cf. lines 06, 52, and 64). Since $\hat{X}$ and $\hat{Y}$ are random elements from the group $\mathbb{G}$ (with order p), we have

$$\Pr[\mathtt{BAD}_1] \leq (q_{\mathsf{IC}} + 2 \cdot q_{\mathsf{Pas}} + q_{\mathsf{Act}}) \cdot \left(\frac{2 \cdot q_{\mathsf{Pas}}}{p} + \frac{q_{\mathsf{Act}}}{p}\right).$$

$\mathbf{G}_1$ and $\mathbf{G}_2$ are identical unless $\texttt{BAD}_1$ happens, and therefore

$$\left|\Pr[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1]\right| \leq \Pr[\texttt{BAD}_1]$$
$$\leq (q_{\mathsf{IC}} + 2 \cdot q_{\mathsf{Pas}} + q_{\mathsf{Act}}) \cdot \left(\frac{2 \cdot q_{\mathsf{Pas}}}{p} + \frac{q_{\mathsf{Act}}}{p}\right).$$

Game 3. In this game, we replace the session keys with random strings.

$\mathcal{A}$ can only distinguish $\mathbf{G}_3$ from $\mathbf{G}_2$ if $\mathcal{A}$ guesses the correct password pw_τ and performs on-line attacks or computes the correct Diffie Hellman value and issues the corresponding hash query to H. We introduce the events $\texttt{GUESS}$ and $\texttt{BAD}_2$ to denote these two cases, respectively. The event $\texttt{GUESS}$ is formally defined in lines 61 and 93, and the event $\texttt{BAD}_2$ is formally defined in lines 15, 82, 100, and 25.

We define the event of $\texttt{BAD}_2$ being set to **true** before $\texttt{GUESS}$ being set to **true** as $\texttt{FIRST}_{\texttt{BAD}_2}$. It is clear that

$$\left|\Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1]\right| \leq \Pr[\texttt{GUESS} \vee \texttt{BAD}_2]$$
$$\leq \Pr[\texttt{GUESS}] + \Pr[\texttt{FIRST}_{\texttt{BAD}_2}] \ .$$

We construct a reduction $\mathcal{B}$ in Figure 9 to bound the probability of $\texttt{FIRST}_{\texttt{BAD}_2}$ by the probability of breaking N-$\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}}$. We note that if $\texttt{BAD}_2$ is set to **true** in game $\mathbf{G}_3$, then $\mathcal{B}$ breaks N-$\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}}$ successfully by outputting the list $\mathcal{L}_{\mathsf{out}}$ that contains the solution $g^{x_\tau y_\tau}$, as shown in lines 18, 58, 72, and 26. All elements in $\mathcal{L}_{\mathsf{out}}$ are computed in passive or active sessions and in the hash queries. We note that elements are only added to list $\mathcal{L}_{\mathsf{out}}$, if they have been queried to the random oracle. Hence, the size of $\mathcal{L}_{\mathsf{out}}$ is bounded by q_{H}. Also we note that by definition of $\texttt{FIRST}_{\texttt{BAD}_2}$, $\mathcal{B}$ never needs to simulate Line 60.

Additionally in game $\mathbf{G}_3$, τ is uniformly random and completely hidden from $\mathcal{A}$'s view. Also, since $\mathcal{A}$ issues at most q_{Act} active queries, the probability of $\texttt{GUESS}$ can be bounded by $\frac{q_{\mathsf{Act}}}{N}$. Therefore

$$\left|\Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1]\right| \leq \Pr[\texttt{GUESS}] + \Pr[\texttt{FIRST}_{\texttt{BAD}_2}]$$
$$\leq \frac{q_{\mathsf{Act}}}{N} + \mathsf{Adv}^{N\text{-}\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}}}(\mathcal{B}) \ .$$

Finally, since in $\mathbf{G}_3$, all the session keys are chosen uniformly at random, we have

$$\Pr[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1] = \frac{1}{2}.$$

Collecting the probabilities yields the bound of Theorem 9.

Game $\mathbf{G}_0$ - $\mathbf{G}_1$

01 **let** $\mathcal{PW} = \{\mathsf{pw}_1, ..., \mathsf{pw}_N\}$
02 **for** $j \in [N]$
03 $\quad x_j \xleftarrow{\$} \mathbb{Z}_p, y_j \xleftarrow{\$} \mathbb{Z}_p$
04 $\tau \xleftarrow{\$} [N]$
05 $\beta \xleftarrow{\$} \{0,1\}$
06 $\beta' \xleftarrow{\$} \mathcal{A}^{\mathcal{O},\mathsf{H},\mathsf{IC}}(\mathsf{U}^*, \mathsf{S}^*)$
07 **return** $[\![\beta = \beta']\!]$

Oracle $\mathsf{E}(\mathsf{pw}, A)$

08 **if** $\exists(\mathsf{pw}, A, \hat{A}, *) \in \mathcal{L}_{\mathsf{IC}}$: **return** $\hat{A}$
09 **repeat**
10 $\quad \hat{A} \xleftarrow{\$} \mathbb{G}$
11 **until** $(\mathsf{pw}, *, \hat{A}, *) \notin \mathcal{L}_{\mathsf{IC}}$
12 $\mathcal{L}_{\mathsf{IC}} \coloneqq \mathcal{L}_{\mathsf{IC}} \cup \{(\mathsf{pw}, A, \hat{A}, \mathrm{enc})\}$
13 **return** $\hat{A}$

Oracle $\mathsf{D}(\mathsf{pw}, \hat{A})$

14 **if** $\exists(\mathsf{pw}, A, \hat{A}, *) \in \mathcal{L}_{\mathsf{IC}}$: **return** A
15 **repeat**
16 $\quad A \xleftarrow{\$} \mathbb{G}$
17 **until** $(\mathsf{pw}, A, *, *) \notin \mathcal{L}_{\mathsf{IC}}$
18 $\mathcal{L}_{\mathsf{IC}} \coloneqq \mathcal{L}_{\mathsf{IC}} \cup \{(\mathsf{pw}, A, \hat{A}, \mathrm{dec})\}$
19 **return** A

Oracle $\textsc{Execute}(\mathsf{U}^*, t_1, \mathsf{S}^*, t_2)$

20 **if** $\pi_{\mathsf{U}^*}^{t_1} \neq \bot$ **or** $\pi_{\mathsf{S}^*}^{t_2} \neq \bot$
21 $\quad$ **return** $\bot$
22 $x \xleftarrow{\$} \mathbb{Z}_p$, $y \xleftarrow{\$} \mathbb{Z}_p$
23 $X \coloneqq g^x$, $Y \coloneqq g^y$
24 $\hat{X} \coloneqq \mathsf{E}(\mathsf{pw}_\tau, X)$, $\hat{Y} \coloneqq \mathsf{E}(\mathsf{pw}_\tau, Y)$
25 **if** $\hat{X} = \hat{Y}$: **return** $\bot$
26 $K \coloneqq X^y$
27 $\mathsf{SK} \coloneqq \mathsf{H}(\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, K, \mathsf{pw}_\tau)$
28 $\pi_{\mathsf{U}^*}^{t_1} \coloneqq ((x, \bot), (\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y})), \mathsf{SK}, \mathbf{true})$
29 $\pi_{\mathsf{S}^*}^{t_2} \coloneqq ((\bot, y), (\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y})), \mathsf{SK}, \mathbf{true})$
30 $(\pi_{\mathsf{U}^*}^{t_1}.\mathtt{fr}, \pi_{\mathsf{S}^*}^{t_2}.\mathtt{fr}) \coloneqq (\mathbf{true}, \mathbf{true})$ $\quad \backslash\backslash\, \mathbf{G}_1$
31 **return** $(\mathsf{U}^*, \hat{X}, \mathsf{S}^*, \hat{Y})$

Oracle $\textsc{Reveal}(\mathsf{P}, t)$

32 **if** $\mathsf{P} \neq \mathsf{U}^*$ **and** $\mathsf{P} \neq \mathsf{S}^*$
33 $\quad$ **return** $\bot$
34 **if** $\pi_{\mathsf{P}}^t.\mathtt{acc} \neq \mathbf{true}$ **or** $\pi_{\mathsf{P}}^t.\mathtt{test} = \mathbf{true}$
35 $\quad$ **return** $\bot$
36 **if** $\exists \mathsf{P}' \in \mathcal{U} \cup \mathcal{S}, t'$ s.t.
37 $\quad \mathsf{Partner}(\pi_{\mathsf{P}}^t, \pi_{\mathsf{P}'}^{t'}) = \mathbf{true}$
38 $\quad$ **and** $\pi_{\mathsf{P}'}^{t'}.\mathtt{test} = \mathbf{true}$
39 $\quad$ **return** $\bot$
40 **for** $\forall(\mathsf{P}', t')$ s.t. $\pi_{\mathsf{P}'}^{t'}.\mathtt{tr} = \pi_{\mathsf{P}}^t.\mathtt{tr}$ $\quad \backslash\backslash\, \mathbf{G}_1$
41 $\quad \pi_{\mathsf{P}'}^{t'}.\mathtt{fr} \coloneqq \mathbf{false}$ $\quad \backslash\backslash\, \mathbf{G}_1$
42 **return** $\pi_{\mathsf{P}}^t.\mathtt{key}$

Oracle $\textsc{Test}(\mathsf{P}, t)$

43 **if** $\mathsf{P} \neq \mathsf{U}^*$ **and** $\mathsf{P} \neq \mathsf{S}^*$
44 $\quad$ **return** $\bot$
45 **if** $\mathsf{Freshness}(\pi_{\mathsf{P}}^t) = \mathbf{false}$: **return** $\bot$ $\quad \backslash\backslash\, \mathbf{G}_0$
46 **if** $\pi_{\mathsf{P}}^t.\mathtt{fr} \coloneqq \mathbf{false}$: **return** $\bot$ $\quad \backslash\backslash\, \mathbf{G}_1$
47 $\mathsf{SK}_0^* \coloneqq \textsc{Reveal}(\mathsf{P}, t)$; $\mathsf{SK}_1^* \xleftarrow{\$} \mathcal{SK}$
48 $\pi_{\mathsf{P}}^t.\mathtt{test} \coloneqq \mathbf{true}$
49 **return** SK_β^*

Oracle $\textsc{SendInit}(\mathsf{U}^*, t_1, \mathsf{S}^*)$

50 **if** $\pi_{\mathsf{U}^*}^{t_1} \neq \bot$: **return** $\bot$
51 $x \xleftarrow{\$} \mathbb{Z}_p$
52 $X \coloneqq g^x$
53 $\hat{X} \coloneqq \mathsf{E}(\mathsf{pw}_\tau, X)$
54 $\pi_{\mathsf{U}^*}^{t_1} \coloneqq ((x, \bot), (\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \bot), \bot, \bot)$
55 $\pi_{\mathsf{U}^*}^{t_1}.\mathtt{fr} \coloneqq \mathbf{false}$ $\quad \backslash\backslash\, \mathbf{G}_1$
56 **return** $(\mathsf{U}^*, \hat{X})$

Oracle $\textsc{SendResp}(\mathsf{S}^*, t_2, \mathsf{U}^*, \hat{X})$

57 **if** $\pi_{\mathsf{S}^*}^{t_2} \neq \bot$: **return** $\bot$
58 $y \xleftarrow{\$} \mathbb{Z}_p$
59 $Y \coloneqq g^y$
60 $\hat{Y} \coloneqq \mathsf{E}(\mathsf{pw}_\tau, Y)$
61 **if** $\hat{X} = \hat{Y}$: **return** $\bot$
62 $X \coloneqq \mathsf{D}(\mathsf{pw}_\tau, \hat{X})$
63 $K \coloneqq X^y$
64 $\mathsf{SK} \coloneqq \mathsf{H}(\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, K, \mathsf{pw}_\tau)$
65 $\pi_{\mathsf{S}^*}^{t_2} \coloneqq ((\bot, y), (\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}), \mathsf{SK}, \mathbf{true})$
66 $\pi_{\mathsf{S}^*}^{t_2}.\mathtt{fr} \coloneqq \mathbf{true}$ $\quad \backslash\backslash\, \mathbf{G}_1$
67 **return** $(\mathsf{S}, \hat{Y})$

Oracle $\textsc{SendTerInit}(\mathsf{U}^*, t_1, \mathsf{S}^*, \hat{Y})$

68 **if** $\pi_{\mathsf{U}^*}^{t_1} = \bot$ **or** $\pi_{\mathsf{U}^*}^{t_1}.\mathtt{tr} \neq (\mathsf{U}^*, \mathsf{S}^*, *, *)$
69 $\quad$ **return** $\bot$
70 **let** $((x, \bot), (\mathsf{U}^*, \mathsf{S}^*, \hat{X}, *), *, *) \coloneqq \pi_{\mathsf{U}^*}^{t_1}$
71 **if** $\hat{X} = \hat{Y}$: **return** $\bot$
72 $Y \coloneqq \mathsf{D}(\mathsf{pw}_\tau, \hat{Y})$
73 $K \coloneqq Y^x$
74 $\mathsf{SK} \coloneqq \mathsf{H}(\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, K, \mathsf{pw}_\tau)$
75 $\pi_{\mathsf{U}^*}^{t_1} \coloneqq ((x, \bot), (\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}), \mathsf{SK}, \mathbf{true})$
76 $\pi_{\mathsf{U}^*}^{t_1}.\mathtt{fr} \coloneqq \mathbf{true}$ $\quad \backslash\backslash\, \mathbf{G}_1$
77 **return true**

Oracle $\mathsf{H}(\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, K, \mathsf{pw}_j)$

78 **if** $\mathcal{L}_{\mathsf{H}}[\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, K, \mathsf{pw}_j] = \bot$
79 $\quad \mathsf{SK} \xleftarrow{\$} \mathcal{SK}$
80 $\quad \mathcal{L}_{\mathsf{H}}[(\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, K, \mathsf{pw}_j] \coloneqq \mathsf{SK}$
81 **return** $\mathcal{L}_{\mathsf{H}}[\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, K, \mathsf{pw}_j]$

Fig. 7. Games $\mathbf{G}_0$-$\mathbf{G}_1$ for the proof of Theorem 9. $\mathcal{A}$ has oracle access to $\mathcal{O} := \{\textsc{Execute}, \textsc{Reveal}, \textsc{Test}, \textsc{SendInit}, \textsc{SendResp}, \textsc{SendTerInit}\}$, random oracle H, and ideal cipher $\mathsf{IC} := (\mathsf{E}, \mathsf{D})$. Further changes in $\mathbf{G}_2$-$\mathbf{G}_4$ are in Figure 8. Without loss of generality we assume that passwords input to H are always from the dictionary, and the user and server pairs input to the session oracles are always $(\mathsf{U}^*, \mathsf{S}^*)$, otherwise an adversary cannot get any advantage.

Oracle $\textsc{Execute}(\mathsf{U}^*, t_1, \mathsf{S}^*, t_2)$

$\textbf{if } \pi^{t_1}_{\mathsf{U}^*} \neq \bot \textbf{ or } \pi^{t_2}_{\mathsf{S}^*} \neq \bot$
  $\textbf{return } \bot$
$\hat{X}, \hat{Y} \xleftarrow{\$} \mathbb{G}$
$\textbf{if } \hat{X} = \hat{Y}\textbf{: return } \bot$
$\textbf{if } (*, *, \hat{X}, *) \in \mathcal{L}_{\mathsf{IC}} \textbf{ or } (*, *, \hat{Y}, *) \in \mathcal{L}_{\mathsf{IC}}$
  $\mathtt{BAD}_1 \coloneqq \textbf{true}\text{; } \textbf{abort}$
$\textbf{for } j \in [N]$
  $X_j \xleftarrow{\$} \mathsf{D}_x(\mathsf{pw}_j, \hat{X})$
  $Y_j \xleftarrow{\$} \mathsf{D}(\mathsf{pw}_j, \hat{Y})$
  $\mathcal{L}_{\mathrm{r}} \coloneqq \mathcal{L}_{\mathrm{r}} \cup \{((\hat{X}, \hat{Y}), j, (r_{j,\hat{X}}, r_{j,\hat{Y}}))\}$
$x \coloneqq x_\tau r_{\tau,\hat{X}}$
$y \coloneqq y_\tau r_{\tau,\hat{Y}}$
$K \coloneqq g^{xy}$
$\textbf{if } \mathcal{L}_{\mathsf{H}}[\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, K, \mathsf{pw}_\tau] \neq \bot$ $\quad \setminus\setminus \mathbf{G}_3$
  $\mathtt{BAD}_2 \coloneqq \textbf{true}\text{; } \textbf{abort}$ $\quad \setminus\setminus \mathbf{G}_3$
$\mathsf{SK} \coloneqq \mathsf{H}(\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, K, \mathsf{pw}_\tau)$ $\quad \setminus\setminus \mathbf{G}_2$
$\mathsf{SK} \xleftarrow{\$} \mathcal{SK}$ $\quad \setminus\setminus \mathbf{G}_3$
$\pi^{t_1}_{\mathsf{U}^*} \coloneqq ((x, \bot), (\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y})), \mathsf{SK}, \textbf{true})$
$\pi^{t_2}_{\mathsf{S}^*} \coloneqq ((\bot, y), (\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y})), \mathsf{SK}, \textbf{true})$
$(\pi^{t_1}_{\mathsf{U}^*}.\mathtt{fr}, \pi^{t_2}_{\mathsf{S}^*}.\mathtt{fr}) \coloneqq (\textbf{true}, \textbf{true})$
$\textbf{return } (\mathsf{U}^*, \hat{X}, \mathsf{S}^*, \hat{Y})$

Oracle $\mathsf{H}(\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, K, \mathsf{pw}_j)$

$\textbf{if } \exists (r_{j,\hat{X}}, r_{j,\hat{Y}}) \text{ s.t. } ((\hat{X}, \hat{Y}), j, (r_{j,\hat{X}}, r_{j,\hat{Y}})) \in \mathcal{L}_{\mathrm{r}}$ $\quad \setminus\setminus \mathbf{G}_3$
  $K^* \coloneqq K^{(r_{j,\hat{X}} r_{j,\hat{Y}})^{-1}}$ $\quad \setminus\setminus \mathbf{G}_3$
  $\textbf{if } K^* = g^{x_j y_j} \wedge j = \tau$ $\quad \setminus\setminus \mathbf{G}_3$
    $\mathtt{BAD}_2 \coloneqq \textbf{true}\text{; } \textbf{abort}$ $\quad \setminus\setminus \mathbf{G}_3$
$\textbf{if } \mathcal{L}_{\mathsf{H}}[\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, K, \mathsf{pw}_j] = \bot$
  $\mathsf{SK} \xleftarrow{\$} \mathcal{SK}$
  $\mathcal{L}_{\mathsf{H}}[(\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, K, \mathsf{pw}_j] \coloneqq \mathsf{SK}$
$\textbf{return } \mathcal{L}_{\mathsf{H}}[\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, K, \mathsf{pw}_j]$

Oracle $\mathsf{E}(\mathsf{pw}_i, A)$

$\textbf{if } \exists (\mathsf{pw}_i, A, \hat{A}, *) \in \mathcal{L}_{\mathsf{IC}}\textbf{: return } \hat{A}$
$\textbf{repeat}$
  $\hat{A} \xleftarrow{\$} \mathbb{G}$
$\textbf{until } (\mathsf{pw}_i, *, \hat{A}, *) \notin \mathcal{L}_{\mathsf{IC}}$
$\mathcal{L}_{\mathsf{IC}} \coloneqq \mathcal{L}_{\mathsf{IC}} \cup \{(\mathsf{pw}_i, A, \hat{A}, \mathrm{enc})\}$
$\mathcal{L}_{\hat{A}} \coloneqq \mathcal{L}_{\hat{A}} \cup \{i\}$
$\textbf{return } \hat{A}$

Oracle $\mathsf{D}(\mathsf{pw}_i, \hat{A})$

$\textbf{if } \exists (\mathsf{pw}_i, A, \hat{A}, *) \in \mathcal{L}_{\mathsf{IC}}\textbf{: return } A$
$\textbf{repeat}$
  $r_{i,\hat{A}} \xleftarrow{\$} \mathbb{Z}_p$
$\textbf{until } (\mathsf{pw}_i, (g^{y_i})^{r_{i,\hat{A}}}, *, *) \notin \mathcal{L}_{\mathsf{IC}}$
$\mathcal{L}_{\mathsf{IC}} \coloneqq \mathcal{L}_{\mathsf{IC}} \cup \{(\mathsf{pw}_i, (g^{y_i})^{r_{i,\hat{A}}}, \hat{A}, \mathrm{dec})\}$
$\textbf{return } (g^{y_i})^{r_{i,\hat{A}}}$

Procedure $\mathsf{D}_x(\mathsf{pw}_i, \hat{A})$

$\textbf{if } \exists (\mathsf{pw}_i, A, \hat{A}, *) \in \mathcal{L}_{\mathsf{IC}}\textbf{: return } A$
$\textbf{repeat}$
  $r_{i,\hat{A}} \xleftarrow{\$} \mathbb{Z}_p$
$\textbf{until } (\mathsf{pw}_i, (g^{x_i})^{r_{i,\hat{A}}}, *, *) \notin \mathcal{L}_{\mathsf{IC}}$
$\mathcal{L}_{\mathsf{IC}} \coloneqq \mathcal{L}_{\mathsf{IC}} \cup \{(\mathsf{pw}_i, (g^{x_i})^{r_{i,\hat{A}}}, \hat{A}, \mathrm{x})\}$
$\textbf{return } (g^{x_i})^{r_{i,\hat{A}}}$

Oracle $\textsc{SendInit}(\mathsf{U}^*, t_1, \mathsf{S}^*)$

$\textbf{if } \pi^{t_1}_{\mathsf{U}^*} \neq \bot\textbf{: return } \bot$
$\hat{X} \xleftarrow{\$} \mathbb{G}$
$\textbf{if } (*, *, \hat{X}, *) \in \mathcal{L}_{\mathsf{IC}}$
  $\mathtt{BAD}_1 \coloneqq \textbf{true}\text{; } \textbf{abort}$
$\textbf{for } j \in [N]$
  $X_j \xleftarrow{\$} \mathsf{D}_x(\mathsf{pw}_j, \hat{X})$
$x \coloneqq x_\tau \cdot r_{\tau,\hat{X}}$
$\pi^{t_1}_{\mathsf{U}^*} \coloneqq ((x, \bot), (\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \bot), \bot, \bot)$
$\pi^{t_1}_{\mathsf{U}^*}.\mathtt{fr} \coloneqq \textbf{false}$
$\textbf{return } (\mathsf{U}^*, \hat{X})$

Oracle $\textsc{SendResp}(\mathsf{S}^*, t_2, \mathsf{U}^*, \hat{X})$

$\textbf{if } \pi^{t_2}_{\mathsf{S}^*} \neq \bot\textbf{: return } \bot$
$\textbf{if } (\mathsf{pw}_\tau, *, \hat{X}, \mathrm{enc}) \in \mathcal{L}_{\mathsf{IC}}$ $\quad \setminus\setminus \mathbf{G}_3$
  $\mathtt{GUESS} \coloneqq \textbf{true}\text{; } \textbf{abort}$ $\quad \setminus\setminus \mathbf{G}_3$
$\hat{Y} \xleftarrow{\$} \mathbb{G}$
$\textbf{if } (*, *, \hat{Y}, *) \in \mathcal{L}_{\mathsf{IC}}$
  $\mathtt{BAD}_1 \coloneqq \textbf{true}\text{; } \textbf{abort}$
$\textbf{if } (*, *, \hat{X}, \mathrm{dec}) \in \mathcal{L}_{\mathsf{IC}}$
  $\textbf{for } j \in [N]$
    $Y_j \xleftarrow{\$} \mathsf{D}_x(\mathsf{pw}_j, \hat{Y})$
  $\textbf{for } j \in [N] \setminus \mathcal{L}_{\hat{X}}$
    $X_j \xleftarrow{\$} \mathsf{D}(\mathsf{pw}_j, \hat{X})$
  $\mathcal{L}_{\mathrm{r}} \coloneqq \mathcal{L}_{\mathrm{r}} \cup \{((\hat{X}, \hat{Y}), j, (r_{j,\hat{X}}, r_{j,\hat{Y}}))\}$
  $X \coloneqq \mathsf{D}(\mathsf{pw}_\tau, \hat{Y})$
$\textbf{else}$
  $\textbf{for } j \in [N]$
    $Y_j \xleftarrow{\$} \mathsf{D}(\mathsf{pw}_j, \hat{Y})$
  $\textbf{for } j \in [N] \setminus \mathcal{L}_{\hat{X}}$
    $X_j \xleftarrow{\$} \mathsf{D}_x(\mathsf{pw}_j, \hat{X})$
    $\mathcal{L}_{\mathrm{r}} \coloneqq \mathcal{L}_{\mathrm{r}} \cup \{((\hat{X}, \hat{Y}), j, (r_{j,\hat{X}}, r_{j,\hat{Y}}))\}$
  $X \coloneqq \mathsf{D}(\mathsf{pw}_\tau, \hat{X})$
$y \coloneqq y_\tau r_{\tau,\hat{Y}}$
$K \coloneqq X^y$
$\textbf{if } \mathcal{L}_{\mathsf{H}}[\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, K, \mathsf{pw}_\tau] \neq \bot$ $\quad \setminus\setminus \mathbf{G}_3$
  $\mathtt{BAD}_2 \coloneqq \textbf{true}\text{; } \textbf{abort}$ $\quad \setminus\setminus \mathbf{G}_3$
$\mathsf{SK} \coloneqq \mathsf{H}(\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, K, \mathsf{pw}_\tau)$ $\quad \setminus\setminus \mathbf{G}_2$
$\mathsf{SK} \xleftarrow{\$} \mathcal{SK}$ $\quad \setminus\setminus \mathbf{G}_3$
$\pi^{t_2}_{\mathsf{S}^*} \coloneqq ((\bot, y), (\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}), \mathsf{SK}, \textbf{true})$
$\pi^{t_2}_{\mathsf{S}^*}.\mathtt{fr} \coloneqq \textbf{true}$
$\textbf{return } (\mathsf{S}^*, \hat{Y})$

Oracle $\textsc{SendTerInit}(\mathsf{U}^*, t_1, \mathsf{S}^*, \hat{Y})$

$\textbf{if } \pi^{t_1}_{\mathsf{U}^*} = \bot \textbf{ or } \pi^{t_1}_{\mathsf{U}^*}.\mathtt{tr} \neq (\mathsf{U}^*, \mathsf{S}^*, *, *)$
  $\textbf{return } \bot$
$\textbf{let } ((x, \bot), (\mathsf{U}^*, \mathsf{S}^*, \hat{X}, *), *, *) \coloneqq \pi^{t_1}_{\mathsf{U}^*}$
$\textbf{if } \hat{X} = \hat{Y}\textbf{: return } \bot$
$\textbf{if } (\mathsf{pw}_\tau, *, \hat{Y}, \mathrm{enc}) \in \mathcal{L}_{\mathsf{IC}}$ $\quad \setminus\setminus \mathbf{G}_3$
  $\mathtt{GUESS} \coloneqq \textbf{true}\text{; } \textbf{abort}$ $\quad \setminus\setminus \mathbf{G}_3$
$\textbf{for } j \in [N] \setminus \mathcal{L}_{\hat{Y}}$
  $Y_j \xleftarrow{\$} \mathsf{D}(\mathsf{pw}_j, \hat{Y})$
  $\mathcal{L}_{\mathrm{r}} \coloneqq \mathcal{L}_{\mathrm{r}} \cup \{((\hat{X}, \hat{Y}), j, (r_{j,\hat{X}}, r_{j,\hat{Y}}))\}$
$Y \coloneqq \mathsf{D}(\mathsf{pw}_\tau, \hat{Y})$
$K \coloneqq Y^x$
$\textbf{if } \mathcal{L}_{\mathsf{H}}[\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, K, \mathsf{pw}_\tau] \neq \bot$ $\quad \setminus\setminus \mathbf{G}_3$
  $\mathtt{BAD}_2 \coloneqq \textbf{true}\text{; } \textbf{abort}$ $\quad \setminus\setminus \mathbf{G}_3$
$\mathsf{SK} \coloneqq \mathsf{H}(\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, K, \mathsf{pw}_\tau)$ $\quad \setminus\setminus \mathbf{G}_2$
$\mathsf{SK} \xleftarrow{\$} \mathcal{SK}$ $\quad \setminus\setminus \mathbf{G}_3$
$\pi^{t_1}_{\mathsf{U}^*}.(\mathtt{tr}, \mathtt{key}, \mathtt{acc}) \coloneqq ((\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}), \mathsf{SK}, \textbf{true})$
$\pi^{t_1}_{\mathsf{U}^*}.\mathtt{fr} \coloneqq \textbf{true}$
$\textbf{return true}$

Fig. 8. Changes to EXECUTE, SENDINIT, SENDRESP, SENDTERINIT in games $\mathbf{G}_2$-$\mathbf{G}_3$ for the proof of Theorem 9. Other oracles are in Figure 7.

Reduction $\mathcal{B}((g^{x_j}, g^{y_j})_{j\in[N]})$

$\mathcal{PW} = \{\mathsf{pw}_1, ..., \mathsf{pw}_N\}$
$\beta \xleftarrow{\$} \{0,1\}$
$\mathcal{L}_{\mathsf{out}} \coloneqq \emptyset$
$\beta' \xleftarrow{\$} \mathcal{A}^{\mathcal{O},\mathsf{H},\mathsf{IC}}(\mathsf{U}^*, \mathsf{S}^*)$
return $\mathcal{L}_{\mathsf{out}}$

Oracle EXECUTE$(\mathsf{U}^*, t_1, \mathsf{S}^*, t_2)$

if $\pi^{t_1}_{\mathsf{U}^*} \neq \bot$ **or** $\pi^{t_2}_{\mathsf{S}^*} \neq \bot$
  return $\bot$
$\hat{X}, \hat{Y} \xleftarrow{\$} \mathbb{G}$
if $\hat{X} = \hat{Y}$: **return** $\bot$
if $(*, *, \hat{X}, *) \in \mathcal{L}_{\mathsf{IC}}$ **or** $(*, *, \hat{Y}, *) \in \mathcal{L}_{\mathsf{IC}}$
  $\mathtt{BAD}_1 \coloneqq$ **true**; **abort**
for $j \in [N]$
  $X_j \xleftarrow{\$} \mathsf{D}_x(\mathsf{pw}_j, \hat{X})$
  $Y_j \xleftarrow{\$} \mathsf{D}(\mathsf{pw}_j, \hat{Y})$
  $\mathcal{L}_{\mathrm{r}} \coloneqq \mathcal{L}_{\mathrm{r}} \cup \{((\hat{X}, \hat{Y}), j, (r_{j,\hat{X}}, r_{j,\hat{Y}}))\}$
for $(\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, \hat{K}, \mathsf{pw}_{j^*}) \in \mathcal{L}_{\mathsf{H}}$
  $K^* \coloneqq \hat{K}^{(r_{j^*,\hat{X}} r_{j^*,\hat{Y}})^{-1}}$
  if $K^* \notin \mathcal{L}_{\mathsf{out}}$: $\mathcal{L}_{\mathsf{out}} \coloneqq \mathcal{L}_{\mathsf{out}} \cup \{K^*\}$
$\mathsf{SK} \xleftarrow{\$} \mathcal{SK}$
$\pi^{t_1}_{\mathsf{U}^*} \coloneqq ((\bot, \bot), (\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y})), \mathsf{SK}, \mathbf{true})$
$\pi^{t_2}_{\mathsf{S}^*} \coloneqq ((\bot, \bot), (\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y})), \mathsf{SK}, \mathbf{true})$
$(\pi^{t_1}_{\mathsf{U}^*}.\mathtt{fr}, \pi^{t_2}_{\mathsf{S}^*}.\mathtt{fr}) \coloneqq (\mathbf{true}, \mathbf{true})$
return $(\mathsf{U}^*, \hat{X}, \mathsf{S}^*, \hat{Y})$

Oracle $\mathsf{H}(\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, K, \mathsf{pw}_j)$

if $\exists (r_{j,\hat{X}}, r_{j,\hat{Y}})$ s.t. $((\hat{X}, \hat{Y}), j, (r_{j,\hat{X}}, r_{j,\hat{Y}})) \in \mathcal{L}_{\mathrm{r}}$
  $K^* \coloneqq K^{(r_{j,\hat{X}} r_{j,\hat{Y}})^{-1}}$
  if $K^* \notin \mathcal{L}_{\mathsf{out}}$: $\mathcal{L}_{\mathsf{out}} \coloneqq \mathcal{L}_{\mathsf{out}} \cup \{K^*\}$
if $\mathcal{L}_{\mathsf{H}}[\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, K, \mathsf{pw}_j] = \bot$
  $\mathsf{SK} \xleftarrow{\$} \mathcal{SK}$
  $\mathcal{L}_{\mathsf{H}}[\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, K, \mathsf{pw}_j] \coloneqq \mathsf{SK}$
return $\mathcal{L}_{\mathsf{H}}[\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, K, \mathsf{pw}_j]$

Oracle SENDINIT$(\mathsf{U}^*, t_1, \mathsf{S}^*)$

if $\pi^{t_1}_{\mathsf{U}^*} \neq \bot$: **return** $\bot$
$\hat{X} \xleftarrow{\$} \mathbb{G}$
if $(*, *, \hat{X}, *) \in \mathcal{L}_{\mathsf{IC}}$
  $\mathtt{BAD}_1 \coloneqq$ **true**; **abort**
for $j \in [N]$
  $X_j \xleftarrow{\$} \mathsf{D}_x(\mathsf{pw}_j, \hat{X})$
$\pi^{t_1}_{\mathsf{U}^*} \coloneqq ((\bot, \bot), (\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \bot), \bot, \bot)$
$\pi^{t_1}_{\mathsf{U}^*}.\mathtt{fr} \coloneqq$ **false**
return $(\mathsf{U}^*, \hat{X})$

Oracle SENDRESP$(\mathsf{S}^*, t_2, \mathsf{U}^*, \hat{X})$

if $\pi^{t_2}_{\mathsf{S}^*} \neq \bot$: **return** $\bot$
$\hat{Y} \xleftarrow{\$} \mathbb{G}$
if $(*, *, \hat{Y}, *) \in \mathcal{L}_{\mathsf{IC}}$
  $\mathtt{BAD}_1 \coloneqq$ **true**; **abort**
if $(*, *, \hat{X}, \mathrm{dec}) \in \mathcal{L}_{\mathsf{IC}}$
  for $j \in [N]$
    $Y_j \xleftarrow{\$} \mathsf{D}_x(\mathsf{pw}_j, \hat{Y})$
  for $j \in [N] \setminus \mathcal{L}_{\hat{X}}$
    $X_j \xleftarrow{\$} \mathsf{D}(\mathsf{pw}_j, \hat{X})$
    $\mathcal{L}_{\mathrm{r}} \coloneqq \mathcal{L}_{\mathrm{r}} \cup \{((\hat{X}, \hat{Y}), j, (r_{j,\hat{X}}, r_{j,\hat{Y}}))\}$
else
  for $j \in [N]$
    $Y_j \xleftarrow{\$} \mathsf{D}(\mathsf{pw}_j, \hat{Y})$
  for $j \in [N] \setminus \mathcal{L}_{\hat{Y}}$
    $X_j \xleftarrow{\$} \mathsf{D}_x(\mathsf{pw}_j, \hat{X})$
    $\mathcal{L}_{\mathrm{r}} \coloneqq \mathcal{L}_{\mathrm{r}} \cup \{((\hat{X}, \hat{Y}), j, (r_{j,\hat{X}}, r_{j,\hat{Y}}))\}$
for $(\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, \hat{K}, \mathsf{pw}_{j^*}) \in \mathcal{L}_{\mathsf{H}}$
  $K^* \coloneqq \hat{K}^{(r_{j^*,\hat{X}} r_{j^*,\hat{Y}})^{-1}}$
  if $K^* \notin \mathcal{L}_{\mathsf{out}}$: $\mathcal{L}_{\mathsf{out}} \coloneqq \mathcal{L}_{\mathsf{out}} \cup \{K^*\}$
$\mathsf{SK} \xleftarrow{\$} \mathcal{SK}$
$\pi^{t_2}_{\mathsf{S}^*} \coloneqq ((\bot, \bot), (\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}), \mathsf{SK}, \mathbf{true})$
$\pi^{t_2}_{\mathsf{S}}.\mathtt{fr} \coloneqq$ **true**
return $(\mathsf{S}, \hat{Y})$

Oracle SENDTERINIT$(\mathsf{U}^*, t_1, \mathsf{S}^*, \hat{Y})$

if $\pi^{t_1}_{\mathsf{U}^*} = \bot$ **or** $\pi^{t_1}_{\mathsf{U}^*}.\mathtt{tr} \neq (\mathsf{U}^*, \mathsf{S}^*, *, *)$
  return $\bot$
let $((\bot, \bot), (\mathsf{U}^*, \mathsf{S}^*, \hat{X}, *), *, *) \coloneqq \pi^{t_1}_{\mathsf{U}^*}$
if $\hat{X} = \hat{Y}$: **return** $\bot$
for $j \in [N] \setminus \mathcal{L}_{\hat{Y}}$
  $Y_j \xleftarrow{\$} \mathsf{D}(\mathsf{pw}_j, \hat{Y})$
  $\mathcal{L}_{\mathrm{r}} \coloneqq \mathcal{L}_{\mathrm{r}} \cup \{((\hat{X}, \hat{Y}), j, (r_{j,\hat{X}}, r_{j,\hat{Y}}))\}$
for $(\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}, \hat{K}, \mathsf{pw}_{j^*}) \in \mathcal{L}_{\mathsf{H}}$
  $K^* \coloneqq \hat{K}^{(r_{j^*,\hat{X}} r_{j^*,\hat{Y}})^{-1}}$
  if $K^* \notin \mathcal{L}_{\mathsf{out}}$: $\mathcal{L}_{\mathsf{out}} \coloneqq \mathcal{L}_{\mathsf{out}} \cup \{K^*\}$
$\mathsf{SK} \xleftarrow{\$} \mathcal{SK}$
$\pi^{t_1}_{\mathsf{U}^*}.(\mathtt{tr}, \mathsf{key}, \mathtt{acc}) \coloneqq ((\mathsf{U}^*, \mathsf{S}^*, \hat{X}, \hat{Y}), \mathsf{SK}, \mathbf{true})$
$\pi^{t_1}_{\mathsf{U}^*}.\mathtt{fr} \coloneqq$ **true**
return true

Fig. 9. Adversary $\mathcal{B}$ against N-HiddenTarget-CDH$_{\mathcal{L}}$ that wins if $\mathtt{BAD}_2$ is set to **true** before GUESS in game $\mathbf{G}_3$ (event $\mathtt{FIRST}_{\mathtt{BAD}_2}$). We highlight the changes from $\mathbf{G}_3$. All other oracles are the same as in $\mathbf{G}_3$.

Acknowledgements

Aysan Nishaburi and Samin Nooripoor were supported by the European Union (ERC AdG REWORC – 101054911). Eike Kiltz was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2092 CASA - 390781972, and by the European Union (ERC AdG REWORC – 101054911). Jiawei Bao and Jiaxin Pan were funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 572525990. We would like to thank Dennis Hofheinz for insightful discussions on Universal Composability.

References

1. Abdalla, M., Barbosa, M.: Perfect forward security of SPAKE2. Cryptology ePrint Archive, Report 2019/1194 (2019), https://eprint.iacr.org/2019/1194
2. Abdalla, M., Fouque, P.A., Pointcheval, D.: Password-based authenticated key exchange in the three-party setting. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 65–84. Springer, Berlin, Heidelberg (Jan 2005)
3. Abdalla, M., Haase, B., Hesse, J.: Security analysis of CPace. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part IV. LNCS, vol. 13093, pp. 711–741. Springer, Cham (Dec 2021)
4. Abdalla, M., Pointcheval, D.: Simple password-based encrypted key exchange protocols. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 191–208. Springer, Berlin, Heidelberg (Feb 2005)
5. Akshima, Kiltz, E., Nishaburi, A., Nooripoor, S., Wiedijk, E.: Generic bounds for multi-instance problems: The strange case of inverse diffie-hellman. preprint (2026)
6. AlFardan, N.J., Bernstein, D.J., Paterson, K.G., Poettering, B., Schuldt, J.C.N.: On the security of RC4 in TLS. In: King, S.T. (ed.) USENIX Security 2013. pp. 305–320. USENIX Association (Aug 2013), https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/alFardan
7. Apostol, T.M.: Mathematical Analysis. Addison-Wesley, 2 edn. (1974)
8. Auerbach, B., Giacon, F., Kiltz, E.: Everybody's a target: Scalability in public-key encryption. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part III. LNCS, vol. 12107, pp. 475–506. Springer, Cham (May 2020)
9. Barbosa, M., Gellert, K., Hesse, J., Jarecki, S.: Bare PAKE: Universally composable key exchange from just passwords. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part II. LNCS, vol. 14921, pp. 183–217. Springer, Cham (Aug 2024)
10. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated key exchange secure against dictionary attacks. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 139–155. Springer, Berlin, Heidelberg (May 2000)
11. Bellare, M., Ranjan, R., Riepel, D., Aldakheel, A.: The concrete security of two-party computation: Simple definitions, and tight proofs for PSI and OPRFs. In: Chung, K.M., Sasaki, Y. (eds.) ASIACRYPT 2024, Part VI. LNCS, vol. 15489, pp. 328–362. Springer, Singapore (Dec 2024)
12. Bellare, M., Ristenpart, T.: Simulation without the artificial abort: Simplified proof and improved concrete security for Waters' IBE scheme. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 407–424. Springer, Berlin, Heidelberg (Apr 2009)

13. Bellare, M., Ristenpart, T., Tessaro, S.: Multi-instance security and its application to password-based cryptography. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 312–329. Springer, Berlin, Heidelberg (Aug 2012)
14. Bellovin, S.M., Merritt, M.: Encrypted key exchange: Password-based protocols secure against dictionary attacks. In: 1992 IEEE Symposium on Security and Privacy. pp. 72–84. IEEE Computer Society Press (May 1992)
15. Beullens, W.: Breaking rainbow takes a weekend on a laptop. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part II. LNCS, vol. 13508, pp. 464–479. Springer, Cham (Aug 2022)
16. Boneh, D., Boyen, X.: Short signatures without random oracles and the SDH assumption in bilinear groups. Journal of Cryptology 21(2), 149–177 (Apr 2008)
17. Bresson, E., Chevassut, O., Pointcheval, D.: Security proofs for an efficient password-based key exchange. In: Jajodia, S., Atluri, V., Jaeger, T. (eds.) ACM CCS 2003. pp. 241–250. ACM Press (Oct 2003)
18. Bresson, E., Chevassut, O., Pointcheval, D.: New security results on encrypted key exchange. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 145–158. Springer, Berlin, Heidelberg (Mar 2004)
19. Canetti, R., Halevi, S., Katz, J., Lindell, Y., MacKenzie, P.D.: Universally composable password-based key exchange. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 404–421. Springer, Berlin, Heidelberg (May 2005)
20. Castryck, W., Decru, T.: An efficient key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 423–447. Springer, Cham (Apr 2023)
21. Degwekar, A., Vaikuntanathan, V., Vasudevan, P.N.: Fine-grained cryptography. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part III. LNCS, vol. 9816, pp. 533–562. Springer, Berlin, Heidelberg (Aug 2016)
22. Dürmuth, M., Golla, M., Markert, P., May, A., Schlieper, L.: Towards quantum large-scale password guessing on real-world distributions. In: Conti, M., Stevens, M., Krenn, S. (eds.) CANS 21. LNCS, vol. 13099, pp. 412–431. Springer, Cham (Dec 2021)
23. Eaton, E., Stebila, D.: The "quantum annoying" property of password-authenticated key exchange protocols. In: Cheon, J.H., Tillich, J.P. (eds.) Post-Quantum Cryptography - 12th International Workshop, PQCrypto 2021. pp. 154–173. Springer, Cham (2021)
24. Fluhrer, S.R., Mantin, I., Shamir, A.: Weaknesses in the key scheduling algorithm of RC4. In: Vaudenay, S., Youssef, A.M. (eds.) SAC 2001. LNCS, vol. 2259, pp. 1–24. Springer, Berlin, Heidelberg (Aug 2001)
25. Giandomenico, E.D., Li, Y., Schäge, S.: Protoss: Protocol for tight optimal symmetric security. In: Luo, B., Liao, X., Xu, J., Kirda, E., Lie, D. (eds.) ACM CCS 2024. pp. 4718–4731. ACM Press (Oct 2024)
26. Gu, Y., Jarecki, S., Krawczyk, H.: KHAPE: Asymmetric PAKE from key-hiding key exchange. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021, Part IV. LNCS, vol. 12828, pp. 701–730. Springer, Cham, Virtual Event (Aug 2021)
27. Haase, B., Labrique, B.: AuCPace: Efficient verifier-based PAKE protocol tailored for the IIoT. Cryptology ePrint Archive, Report 2018/286 (2018), https://eprint.iacr.org/2018/286
28. Hhan, M.: A new approach to generic lower bounds - classical/quantum MDL, quantum factoring, and more. pp. 345–374. LNCS, Springer, Cham (Jun 2025)
29. Jablon, D.P.: Strong password-only authenticated key exchange. Comput. Commun. Rev. 26(5), 5–26 (1996), https://doi.org/10.1145/242896.242897

30. Jarecki, S., Krawczyk, H., Xu, J.: OPAQUE: An asymmetric PAKE protocol secure against pre-computation attacks. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part III. LNCS, vol. 10822, pp. 456–486. Springer, Cham (Apr / May 2018)
31. Kuhn, F., Struik, R.: Random walks revisited: Extensions of Pollard's rho algorithm for computing multiple discrete logarithms. In: Vaudenay, S., Youssef, A.M. (eds.) SAC 2001. LNCS, vol. 2259, pp. 212–229. Springer, Berlin, Heidelberg (Aug 2001)
32. Lee, J.M.: Introduction to Smooth Manifolds, Graduate Texts in Mathematics, vol. 218. Springer, 2 edn. (2013)
33. Maino, L., Martindale, C.: An attack on SIDH with arbitrary starting curve. Cryptology ePrint Archive, Report 2022/1026 (2022), https://eprint.iacr.org/2022/1026
34. Maurer, U.M.: Abstract models of computation in cryptography (invited paper). In: Smart, N.P. (ed.) 10th IMA International Conference on Cryptography and Coding. LNCS, vol. 3796, pp. 1–12. Springer, Berlin, Heidelberg (Dec 2005)
35. Merkle, R.C.: Secure communications over insecure channels. Commun. ACM 21(4), 294–299 (1978), https://doi.org/10.1145/359460.359473
36. Pan, J., Zeng, R.: A generic construction of tightly secure password-based authenticated key exchange. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023, Part VIII. LNCS, vol. 14445, pp. 143–175. Springer, Singapore (Dec 2023)
37. Robert, D.: Breaking SIDH in polynomial time. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 472–503. Springer, Cham (Apr 2023)
38. Rudin, W.: Principles of Mathematical Analysis. McGraw-Hill, New York, 3 edn. (1976)
39. Santos, B.F.D., Gu, Y., Jarecki, S.: Randomized half-ideal cipher on groups with applications to UC (a)PAKE. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 128–156. Springer, Cham (Apr 2023)
40. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: 35th FOCS. pp. 124–134. IEEE Computer Society Press (Nov 1994)
41. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT'97. LNCS, vol. 1233, pp. 256–266. Springer, Berlin, Heidelberg (May 1997)
42. Stevens, M., Lenstra, A.K., de Weger, B.: Chosen-prefix collisions for MD5 and colliding X.509 certificates for different identities. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 1–22. Springer, Berlin, Heidelberg (May 2007)
43. Stevens, M., Sotirov, A., Appelbaum, J., Lenstra, A.K., Molnar, D., Osvik, D.A., de Weger, B.: Short chosen-prefix collisions for MD5 and the creation of a rogue CA certificate. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 55–69. Springer, Berlin, Heidelberg (Aug 2009)
44. Tiepelt, M., Eaton, E., Stebila, D.: Making an asymmetric PAKE quantum-annoying by hiding group elements. In: Tsudik, G., Conti, M., Liang, K., Smaragdakis, G. (eds.) ESORICS 2023, Part I. LNCS, vol. 14344, pp. 168–188. Springer, Cham (Sep 2023)
45. Wang, X., Yu, H.: How to break MD5 and other hash functions. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 19–35. Springer, Berlin, Heidelberg (May 2005)
46. Ying, J.H.M., Kunihiro, N.: Bounds in various generalized settings of the discrete logarithm problem. In: Gollmann, D., Miyaji, A., Kikuchi, H. (eds.) ACNS 17Inter-

national Conference on Applied Cryptography and Network Security. LNCS, vol. 10355, pp. 498–517. Springer, Cham (Jul 2017)
47. Yun, A.: Generic hardness of the multiple discrete logarithm problem. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 817–836. Springer, Berlin, Heidelberg (Apr 2015)
48. Zipf, G.: Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology. Addison-Wesley Press, Cambridge, Massachusetts, USA (1949)

A PAKE with Non-Uniform Password Distributions

In this section, we outline how to extend our results to a PAKE security model that supports arbitrary password distributions D. All our definitions and results for PAKE with uniformly distributed passwords are recovered as a special case of this generalized model by choosing the uniform password distribution.

A.1 Password Distributions

The BPR security model introduced in Section 4 assumes that passwords are drawn uniformly at random. However, this assumption is unrealistic when passwords are chosen by humans. To address this, we first formalize the notion of password distributions and then discuss the $\textsc{Zipf}_s$ distribution, a practically relevant model for real-world password choices.

Let $\mathcal{PW} = \{\mathsf{pw}_1, \ldots, \mathsf{pw}_N\}$ be an ordered set of N passwords. Let D be an arbitrary distribution over the password space $\mathcal{PW}$, where for each $i \in [N]$,

$$p_i := \Pr_{\mathsf{pw} \xleftarrow{\mathrm{D}} \mathcal{PW}}[\mathsf{pw} = \mathsf{pw}_i].$$

We write $\tau \xleftarrow{\mathrm{D}} [N]$ for the induced distribution on $[N]$ such that, for every $i \in [N]$, $\Pr_{\tau \xleftarrow{\mathrm{D}} [N]}[\tau = i] := p_i$. We assume that the passwords are ordered in non-increasing probability, i.e., $p_1 \geq p_2 \geq \ldots \geq p_N$, so that pw_1 is the most likely password.

Zipf Distribution. It has been realized that real-world distributions of human passwords typically follow a $\textsc{Zipf}_s$ distribution [22, 48]. To define $\textsc{Zipf}_s$, we introduce the generalized harmonic number as $H_s(N) := \sum_{i=1}^{N} \frac{1}{i^s}$. The $\textsc{Zipf}_s$ distribution of a random variable pw with real steepness parameter $0 \leq s < 1$ is defined as

$$p_i := \Pr_{\mathsf{pw} \xleftarrow{\mathrm{D}} \mathcal{PW}}[\mathsf{pw} = \mathsf{pw}_i] := \frac{1}{H_s(N)} \frac{1}{i^s},$$

where $H_s(N)$ normalizes the distribution over $\mathcal{PW}$. For $s = 0$, we have $H_0(N) = N$ and $p_i = \frac{1}{N}$ corresponds to a uniform distribution. As the steepness s approaches 1, the probability mass becomes increasingly concentrated on the most likely passwords. Leakage of real-world password data (e.g., LinkedIn, RockYou, and 000Webhost) shows that human passwords commonly fellow the $\textsc{Zipf}_s$ distribution with $N \approx 2^{30}$ and steepness parameter s between 0.68 and 0.79 [22].

It is well known [22] that the partial sums of $\textsc{Zipf}_s$ probabilities admit the representation

$$\sum_{i=1}^{k} p_i = \frac{H_s(k)}{H_s(N)} = \Theta\left(\left(\frac{k}{N}\right)^{1-s}\right).$$

A.2 PAKE Security Model

The BPR_D model for an arbitrary password distribution D is identical to the BPR model defined in Section 4, except that the experiment samples the password from D, i.e., $\mathsf{pw} \xleftarrow{\mathrm{D}} \mathcal{PW}$. Since, in the q_{Act} active sessions, the adversary may guess the q_{Act} most probable passwords $\mathsf{pw}_1, \ldots, \mathsf{pw}_{q_{\mathsf{Act}}}$ under distribution D, the advantage defined in Definition 8 must be normalized by $\sum_{i=1}^{q_{\mathsf{Act}}} p_i$.

Definition 11 (Security of PAKE with password distribution D). *Let* PAKE *be a PAKE protocol and* $\mathcal{A}$ *be an adversary. The advantage of* $\mathcal{A}$ *against* PAKE *is defined as*

$$\mathsf{Adv}^{\mathsf{BPR}_{\mathrm{D}}}_{\mathsf{PAKE}}(\mathcal{A}) := \left|\Pr\left[\mathsf{BPR}^{\mathcal{A}}_{\mathsf{PAKE}} \Rightarrow 1\right] - \frac{1}{2}\right| - \sum_{i=1}^{q_{\mathsf{Act}}} p_i \ .$$

Here q_{Act} *is the number of active queries, namely, the total number of* SendInit, SendResp, *and* SendTerInit *queries made by* $\mathcal{A}$.

A.3 Hidden-Target Computational Diffie Hellman

We adapt the N-$\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}}$ problem of Section 3.2 to an arbitrary password distribution D by picking the hidden-target τ from distributions D. Formally, we define the advantage function as

$$\mathsf{Adv}^{N\text{-}\mathsf{HiddenTarget\text{-}CDH}_{\mathrm{D},\mathcal{L}}}(\mathcal{A}) := \Pr[N\text{-}\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}} \Rightarrow 1],$$

where N-$\mathsf{HiddenTarget\text{-}CDH}_{\mathrm{D},\mathcal{L}}$ is the game described in Figure 10.

Game N-$\mathsf{HiddenTarget\text{-}CDH}_{\mathrm{D},\mathcal{L}}$

01 **for** $i \in [N]$
02 $\quad x_i, y_i \xleftarrow{\$} \mathbb{Z}_p$
03 $(Z_1, \ldots, Z_q) \xleftarrow{\$} \mathcal{A}((g^{x_i}, g^{y_i})_{i\in[N]})$
04 $\tau \xleftarrow{\mathrm{D}} [N]$
05 **return** $[\![g^{x_\tau y_\tau} \in \{Z_1, \ldots, Z_q\}]\!]$

Fig. 10. The N-$\mathsf{HiddenTarget\text{-}CDH}_{\mathrm{D},\mathcal{L}}$ game for password distribution D.

The following lemma is a simple generalization of Lemma 2 to arbitrary password distributions D over $\mathcal{PW}$.

Lemma 12. *Let N be an integer and let $\mathcal{A}$ be an adversary against the N-*$\mathsf{HiddenTarget\text{-}CDH}_{\mathrm{D},\mathcal{L}}$ *problem. Then, for any $k \in [N]$,*

$$\mathsf{Adv}^{N\text{-}\mathsf{HiddenTarget\text{-}CDH}_{\mathrm{D},\mathcal{L}}}(\mathcal{A}) \leq \sum_{i=1}^{k-1} p_i + \mathsf{Adv}^{(N,k)\text{-}\mathsf{CDH}_{\mathcal{L}}}(\mathcal{A}).$$

A.4 Security of EKE

It is not hard to see that the same concrete security statement for EKE of Theorem 9 holds for an arbitrary password distribution such as D by replacing $\mathsf{Adv}^{N\text{-}\mathsf{HiddenTarget\text{-}CDH}_{\mathcal{L}}}(\mathcal{B})$ with $\mathsf{Adv}^{N\text{-}\mathsf{HiddenTarget\text{-}CDH}_{\mathrm{D},\mathcal{L}}}(\mathcal{B})$.

Concrete Security with Zipf Distribution. We now compute the concrete bit security of EKE in the $\mathsf{BPR}_{\mathrm{D}}$ model, where $\mathrm{D} = \textsc{Zipf}_s$ is the Zipf distribution with steepness s.

We formulate the following conjecture that generalizes Lemma 5.[9]

Conjecture 13. For integers $N \leq p$, define $T_{max} = \sqrt{N^{1-s}p}$. Then, for any integer $T \in [T_{max}]$,

$$\min_{k\in[N]} \left(\frac{k-1}{N} + \left(\frac{T^2}{kp}\right)^k \right) \lesssim \frac{1}{T_{max}} \cdot T.$$

Using this conjecture we can generalize Lemma 3 as

$$\begin{aligned}
\mathsf{Adv}^{N\text{-}\mathsf{HiddenTarget\text{-}CDH}_{\mathrm{D},\mathcal{L}}}(\mathcal{A}) &\lesssim \min_{k\in[N]} \left(\frac{H_s(k-1)}{H_s(N)} + \left(\frac{T^2}{kp}\right)^k \right) \\
&\approx \min_{k\in[N]} \left(\left(\frac{k-1}{N}\right)^{1-s} + \left(\frac{T^2}{kp}\right)^k \right) \\
&\lesssim \frac{1}{\sqrt{N^{1-s}p}} \cdot T \ .
\end{aligned}$$

This implies a generalization of Corollary 10 on the bit security of EKE in the $\mathsf{BPR}_{\textsc{Zipf}_s}$ model.

Corollary 14. *In the generic group model,* EKE *achieves roughly $\lambda = \frac{1}{2}(\log_2(p) + (1-s)\log_2(N))$ bits of security.*

B Proof of Lemma 5

For $k \in [N]$, we define $f_k : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ and $g : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ as follows:

$$f_k(T) = \frac{k-1}{N} + \left(\frac{T^2}{kp}\right)^k, \quad g(T) = \frac{1}{T_{max}} \cdot T. \tag{3}$$

[9] Currently, the conjecture is supported only by empirical evidence; a formal proof is left for future work.

For $k \in [N]$, set

$$\hat{f}_0 := \frac{k-1}{N}, \quad \hat{f}_1 := (kp)^{-k}, \quad \hat{g}_1 := \frac{1}{T_{max}}.$$

Then, we can represent f_k and g as follows:

$$f_k(T) = \hat{f}_1 \, T^{2k} + \hat{f}_0, \quad g(T) = \hat{g}_1 \, T.$$

For $k \in [N]$, we define $h_k : \mathbb{R}_{\geq 0} \to \mathbb{R}$ as follows:

$$\begin{aligned} h_k(T) &= f_k(T) - g(T) \\ &= \hat{f}_1 \, T^{2k} - \hat{g}_1 \, T + \hat{f}_0. \end{aligned}$$

Since $\hat{f}_1 \geq 0$ and

$$h_k^{''}(T) = 2k(2k-1) \, \hat{f}_1 \, T^{2k-2} \;\geq\; 0,$$

the function h_k is convex on $\mathbb{R}_{\geq 0}$.

Consider the roots $t_1(k)$ and $t_2(k)$ of function h_k, which are the intersections of $f_k(T)$ and $g(T)$ as seen in Figure 4.

Claim 1 *For $k \in [N]$, h_k has exactly two positive distinct roots such that*

$$t_1(k) < t_2(k). \tag{4}$$

The proof of Claim 1 is deferred to Appendix B.1. Since h_k is convex, for the interval $T \in [t_1(k), t_2(k)]$, we have:

$$f_k(T) \leq g(T). \tag{5}$$

Claim 2 *For all $k \in [N-1]$, we have:*

$$t_2(k+1) > t_2(k) \; . \tag{6}$$

The proof of Claim 2 is postponed to Appendix B.2. Note that $t_2(N) = T_{max}$ and therefore

$$g(t_2(N)) = f_N(t_2(N)) \; .$$

Given Claim 2, the interval $[0, \, t_2(N)]$ can be partitioned using consecutive intersection points as follows:

$$[0, \, t_2(N)] = [t_1(1), \, t_2(1)) \; \cup \; [t_2(1), \, t_2(2)) \; \cup \; \cdots \; \cup \; [t_2(N-1), \, t_2(N)].$$

With this in hand, we define $\hat{f} : [0, t_2(N)] \to [0, 1]$ as follows:

$$\hat{f}(T) = \begin{cases} f_1(T), & T \in [0, \, t_2(1)), \\ f_2(T), & T \in [t_2(1), \, t_2(2)), \\ \quad \vdots & \quad \vdots \\ f_N(T), & T \in [t_2(N-1), \, t_2(N)]. \end{cases}$$

Claim 3 *For all $k \in [N]$, we have:*

$$t_2(k-1) \in [t_1(k), t_2(k)] \,. \tag{7}$$

The proof of Claim 3 is postponed to Appendix B.4.
By Equation (5) and Claim 3, on each interval $[t_2(k-1),\, t_2(k))$, we have:

$$h_k(T) \leq 0, \tag{8}$$

and hence

$$f_k(T) = \hat{f}(T) \leq g(T).$$

Therefore, for all $T \in [0, t_2(N)]$, one can write:

$$\min_{k\in[N]} f_k(T) \;\leq\; \hat{f}(T) \;\leq\; g(T). \tag{9}$$

Combining (3) with inequality (9) completes the proof of the lemma.

B.1 Proof of Claim 1

First, we define $h : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ as follows:

$$h(k, T) = \frac{k-1}{N} + \Big(\frac{T^2}{kp}\Big)^k - \frac{T}{T_{\max}} \tag{10}$$

For each real $k \in [1, N]$, we extend the function h_k to $h_k : \mathbb{R}_{\geq 0} \to \mathbb{R}$ by

$$h_k(T) := h(k, T). \tag{11}$$

Here, we state and prove a stronger version of Claim 1 as follows by allowing k to be a real number in $[1, N]$.

Lemma 15. *Fix $k \in [1, N]$ and let h_k be defined as in (11). Then the equation $h_k(T) = 0$ has exactly two distinct roots in $[0, T_{\max}]$, denoted by $t_1(k) < t_2(k)$.*

Proof (of Lemma 15). First, we show that for every $k \in [1, N]$, the equation $h_k(T) = 0$ has at most two distinct roots in $\mathbb{R}_{\geq 0}$ (and hence in $[0, T_{\max}]$). For $T > 0$, we can compute:

$$h_k^{''}(T) = 2k(2k-1)\,(kp)^{-k} T^{2k-2}.$$

As a result, h_k is strictly convex on $(0, \infty)$. Thus, $h_k^{'}$ is strictly increasing on $(0, \infty)$. If h_k had three distinct zeros in $\mathbb{R}_{\geq 0}$, then by Rolle's theorem from [7, page 110] applied on the intervals between them, $h_k^{'}$ would have at least two distinct zeros in $(0, \infty)$, contradicting strict monotonicity of $h_k^{'}$. We now show that h_k has at least two roots on $\mathbb{R}_{\geq 0}$ for $k \in [1, N]$, by a sign-change argument. Fix $k \in [1, N]$ and define

$$T_k^* := \frac{k}{N}\, T_{\max}. \tag{12}$$

Claim 4 *For $k \in (1, N)$, we have:*

$$h_k(T_k^*) < 0. \tag{13}$$

Proof (of Claim 4). A direct computation gives

$$h_k(T_k^*) = -\frac{1}{N} + \frac{k^k}{N^{\,k+k/N}}.$$

Thus $h_k(T_k^*) < 0$ is equivalent to

$$k^k < N^{\,k-1+k/N} \quad \Longleftrightarrow \quad k \ln k < (k - 1 + k/N) \ln N. \tag{14}$$

Define $\phi(x) := (x - 1 + x/N) \ln N - x \ln x$ on $[1, N]$. Then

$$\phi''(x) = -\frac{1}{x} < 0,$$

so ϕ is strictly concave. It is easy to verify that:

$$\phi(1) > 0, \tag{15}$$
$$\phi(N) = 0. \tag{16}$$

A strictly concave function lying above 0 at $x = 1$ and equal to 0 at $x = N$ satisfies $\phi(x) > 0$ for all $x \in (1, N)$. This implies that for $x \in (1, N)$, we have:

$$\phi(x) := (x - 1 + x/N) \ln N - x \ln x > 0 \tag{17}$$

Given equivalences (14), for $k \in (1, N)$ we obtain:

$$h_k(T_k^*) < 0.$$

which completes the proof of the claim.

Claim 5 *For every $k \in [1, N]$, there exist roots $t_1(k), t_2(k) \in [0, T_{\max}]$ of h_k such that*

$$t_1(k) < T_k^* \leq t_2(k).$$

Proof (of Claim 5). First, assume $k \in (1, N)$. Given Claim 4, we have:

$$h_k(T_k^*) < 0. \tag{18}$$

By a direct calculation, for $k \in (1, N)$ we can see:

$$h_k(0) > 0, \tag{19}$$
$$h_k(T_{\max}) > 0. \tag{20}$$

By the Intermediate Value Theorem in [38, page 93] and given Equation (18) and Equation (19), there exists $t_1(k) \in (0, T_k^*)$ such that $h_k(t_1(k)) = 0$. Similarly,

using Equation (18) and Equation (20), there exists $t_2(k) \in (T_k^*, T_{\max})$ such that $h_k(t_2(k)) = 0$. Since h_k is strictly convex on $(0, \infty)$, these two roots are distinct.

For $k = 1$, we can explicitly find the two distinct roots of h_1 and easily verify that Claim 5 holds.

For $k = N$, we can show that:

$$\begin{aligned} h_N(0) &> 0, \\ h_N(T_{\max}) &= 0, \\ h_N(\frac{2N-1}{2N} T_{\max}) &< 0. \end{aligned}$$

Applying the same Intermediate Value Theorem argument, one can show that Claim 5 holds for $k = N$. This completes the proof of the claim.

Using Claim 5, we showed that for each $k \in [1, N]$, it follows that $h_k(T) = 0$ has exactly two distinct roots in $[0, T_{\max}]$, denoted by $t_1(k) < t_2(k)$, which proves Lemma 15.

B.2 Proof of Claim 2

In order to show that Claim 2 holds, we first recall the notion of a C^1 function:

Definition 16 (C^1 function). *Let $U \subset \mathbb{R}^2$ be an open set [10] and let $h : U \to \mathbb{R}$ be a function. We say that h is of class C^1 on U if the partial derivatives $\frac{\partial h}{\partial k}$ and $\frac{\partial h}{\partial T}$ exist at every point of U and are continuous on U.*

Then, we introduce the global implicit function lemma.

Lemma 17 (Larger Root Global Implicit Function Lemma). *Let $I \subset \mathbb{R}$ be an interval and let $h : I \times \mathbb{R} \to \mathbb{R}$ be a C^1 function. Assume that for every $k \in I$, the equation $h(k, T) = 0$ has exactly two real roots in T, denoted $t_1(k)$ and $t_2(k)$ with $t_1(k) < t_2(k)$, and for all $k \in I$, $\frac{\partial h}{\partial T}(k, t_1(k)) \neq 0$ and $\frac{\partial h}{\partial T}(k, t_2(k)) \neq 0$ holds. Then the map $t_2 : I \to \mathbb{R}$ which assigns to each $k \in I$ the larger root $t_2(k)$ of the equation $h(k, T) = 0$ is of class C^1 on I. Moreover, for every $k \in I$ the derivative of t_2 is given by*

$$t_2'(k) = -\frac{\dfrac{\partial h}{\partial k}(k, t_2(k))}{\dfrac{\partial h}{\partial T}(k, t_2(k))}.$$

The proof of the lemma is provided in Section B.3. We then apply Lemma 17 to function $h(k, T)$ defined in (10), which yields the following result:

[10] This definition can also be extended to close sets by considering the interior of U, where the partial derivatives exist and are continuous and requiring that the one-sided partial derivatives exist at the boundary points.

Lemma 18. *Let $h_k(T)$ be as in (11) and define $h(k,T)$ as in (10) for $k \in [1,N]$ and $T \geq 0$. Then the larger root $t_2(k)$ is of class C^1 on $[1,N]$, and, for all $k \in [1,N]$,*

$$t_2'(k) = -\frac{\frac{\partial h}{\partial k}\big(k, t_2(k)\big)}{\frac{\partial h}{\partial T}\big(k, t_2(k)\big)} > 0.$$

In particular, t_2 is strictly increasing in k.

Proof (of Lemma 18). Lemma 15 shows that for each $k \in [1,N]$ the equation $h_k(T) = 0$ has exactly two distinct roots $t_1(k) < t_2(k)$. Define:

$$\frac{\partial h}{\partial T}(k,T) = h_k'(T). \tag{21}$$

For $i \in \{1,2\}$, we claim:

$$h_k'(t_i(k)) \neq 0. \tag{22}$$

The statement holds because if without loss of generality for $i = 1$, $h_k'(t_1(k)) = 0$, since h_k is strictly convex on $(0,\infty)$ and h_k' is strictly increasing, then $h_k(t_2(k)) > 0$ which is a contradiction. As a result, for $i \in \{1,2\}$, we have:

$$\frac{\partial h}{\partial T}(k, t_i(k)) \neq 0. \tag{23}$$

Therefore, h satisfies the assumptions of the Larger Root Global Implicit Function Lemma 17 on the interval $I = [1,N]$. It follows that the larger root $t_2 : [1,N] \to \mathbb{R}$ is of class C^1, and that for every real $k \in [1,N]$,

$$t_2'(k) = -\frac{\frac{\partial h}{\partial k}\big(k, t_2(k)\big)}{\frac{\partial h}{\partial T}\big(k, t_2(k)\big)}. \tag{24}$$

Claim 6 *For all $k \in [1,N]$, we have:*

$$\frac{\partial h}{\partial T}\big(k, t_2(k)\big) > 0. \tag{25}$$

Proof (of Claim 6). By strict convexity, the derivative in T is strictly increasing. Applying the mean value theorem from [38, page 107] between $t_1(k)$ and $t_2(k)$, there exists some $\xi_k \in (t_1(k), t_2(k))$ such that

$$\frac{\partial h}{\partial T}(k, \xi_k) = \frac{h\big(k, t_2(k)\big) - h\big(k, t_1(k)\big)}{t_2(k) - t_1(k)} = 0.$$

Since h_k' is strictly increasing in T and $t_1(k) < \xi_k < t_2(k)$, we must have

$$\frac{\partial h}{\partial T}(k, t_1(k)) < 0 < \frac{\partial h}{\partial T}(k, t_2(k)).$$

This proves Claim 6.

Claim 7 *For $k \in [1, N]$, we have:*

$$\frac{\partial h}{\partial k}(k, t_2(k)) < 0. \tag{26}$$

Proof (of Claim 7). A direct computation gives:

$$\frac{\partial h}{\partial k}(k, T) = \frac{1}{N} + \Big(\frac{T^2}{kp}\Big)^k \Big(\log\Big(\frac{T^2}{kp}\Big) - 1\Big). \tag{27}$$

For $k \in [1, N]$, define:

$$z_k = \frac{t_2(k)^2}{kp}. \tag{28}$$

Now, Equation (27) evaluated at $t_2(k)$ and given definition (28) can be rewritten as:

$$\frac{\partial h}{\partial k}(k, t_2(k)) = \frac{1}{N} + z_k^k \Big(\log(z_k) - 1\Big). \tag{29}$$

Since for $k \in [1, N]$ equation $h(k, t_2(k)) = 0$ holds, using definition (28) we have:

$$z_k^k - \frac{t_2(k)}{T_{\max}} + \frac{k-1}{N} = 0. \tag{30}$$

Given that $t_2(k) \leq T_{\max}$ holds for $k \in (1, N]$, $t_2(1) < T_{max}$, and by Equation (30), we can conclude:

$$z_k^k = \frac{t_2(k)}{T_{\max}} - \frac{k-1}{N} < 1. \tag{31}$$

Using Equation (12) and Equation (30) and Claim 5, we get:

$$z_k^k = \frac{t_2(k)}{T_{\max}} - \frac{k-1}{N} \geq \frac{k}{N} - \frac{k-1}{N} \geq \frac{1}{N}. \tag{32}$$

Equation (31) implies:

$$\log\Big(z_k\Big) < 0. \tag{33}$$

From Equation (33), we get:

$$\frac{\partial h}{\partial k}(k, t_2(k)) < \frac{1}{N} - z_k^k. \tag{34}$$

Combining Equation (32), Equation (34), proves Claim 7.

Given Claim 7, Claim 6, and Equation (24), we have:

$$t_2'(k) > 0. \tag{35}$$

which completes the proof of the lemma.

Now, we apply Equation (35) and get that $t_2(k)$ is a strictly increasing function on $[1, N]$. Therefore, for any integers $k \in [N-1]$, we have

$$t_2(k+1) > t_2(k). \tag{36}$$

B.3 Proof of Lemma 17 from Section B.2

In order to prove Lemma 17, we first state the local implicit function lemma from [38, page 223], which we will use:

Lemma 19 (Local Implicit Function Lemma). *Let $h : \mathbb{R}^2 \to \mathbb{R}$ be a C^1 function. Let $(k_0, T_0) \in \mathbb{R}^2$ be a point such that $h(k_0, T_0) = 0$ and $\frac{\partial h}{\partial T}(k_0, T_0) \neq 0$. Then there exist an open interval I centered at k_0, an open interval J centered at T_0, and a unique function $\varphi : I \to J$ such that φ is of class C^1 on I, $\varphi(k_0) = T_0$, and*

$$h(k, \varphi(k)) = 0 \quad \text{for all } k \in I.$$

Moreover, for every $k \in I$ the derivative of φ is given by

$$\varphi'(k) = -\frac{\dfrac{\partial h}{\partial k}(k, \varphi(k))}{\dfrac{\partial h}{\partial T}(k, \varphi(k))}.$$

We then go ahead and prove the larger root local implicit function lemma using Lemma 19 as follows:

Lemma 20 (Larger Root Local Implicit Function Lemma). *Let $I \subset \mathbb{R}$ be an interval and let $h : I \times \mathbb{R} \to \mathbb{R}$ be a C^1 function. Assume that for every $k \in I$, the equation $h(k, T) = 0$ has exactly two real roots in T, denoted $t_1(k)$ and $t_2(k)$ with $t_1(k) < t_2(k)$, and for all $k \in I$, $\frac{\partial h}{\partial T}(k, t_1(k)) \neq 0$ and $\frac{\partial h}{\partial T}(k, t_2(k)) \neq 0$ holds.*

Then, for every $k^* \in I$, there exists an open interval J centered at k^*, together with a unique function $\varphi : J \cap I \to \mathbb{R}$, such that φ is of class C^1 on $J \cap I$. In addition, for all $k \in J \cap I$, $h(k, \varphi(k)) = 0$ holds, and $\varphi(k) = t_2(k)$ and

$$\varphi'(k) = -\frac{\dfrac{\partial h}{\partial k}(k, \varphi(k))}{\dfrac{\partial h}{\partial T}(k, \varphi(k))}.$$

Proof (of Lemma 20). For each $k^* \in I$, by the assumption, the equation $h(k^*, T) = 0$ has two roots, namely $(k^*, t_1(k^*))$ and $(k^*, t_2(k^*))$. We apply Lemma 19 on each of the points. At $(k^*, t_1(k^*))$, we conclude that there exists an open interval V_1 centered at k^* and a C^1 function $\varphi_1 : V_1 \cap I \to \mathbb{R}$ exists such that

$$\varphi_1(k^*) = t_1(k^*), \tag{37}$$

and for $k \in V_1 \cap I$, we have:

$$h(k, \varphi_1(k)) = 0. \tag{38}$$

Following the same logic, at $(k^*, t_2(k^*))$, there exists an open interval V_2 centered at k^* and a C^1 function $\varphi_2 : V_2 \cap I \to \mathbb{R}$ exists such that

$$\varphi_2(k^*) = t_2(k^*), \tag{39}$$

and for $k \in V_2 \cap I$, we have:

$$h(k, \varphi_2(k)) = 0. \tag{40}$$

Define interval U centered at k^* as follows:

$$U = V_1 \cap V_2.$$

Note that since $k^* \in U \cap I$, $U \cap I \neq \emptyset$. Thus, we can define function $d : U \cap I \to \mathbb{R}$

$$d(k) = \varphi_2(k) - \varphi_1(k).$$

From Equation (37), Equation (39), and $t_2(k^*) > t_1(k^*)$, we have:

$$\varphi_2(k^*) > \varphi_1(k^*) \implies d(k^*) > 0. \tag{41}$$

As d is a continuous function, and $d(k^*) > 0$, continuity of d implies that there exists an interval J centered at k^* such that $J \subseteq U$ and for all $k \in J \cap I$, we have:

$$\varphi_2(k) > \varphi_1(k) \tag{42}$$

For all $k \in J \cap I$, from Equation (38) and Equation (40), we have:

$$h(k, \varphi_1(k)) = 0, \qquad h(k, \varphi_2(k)) = 0.$$

By the assumption, the equation $h(k, T) = 0$ has exactly two real roots, namely $(k, t_2(k))$ and $(k, t_1(k))$ such that $t_2(k) > t_1(k)$. Therefore, the two roots $(k, \varphi_1(k))$ and $(k, \varphi_2(k))$ must coincide with these, in the correct order. Thus, from Equation (42), we can conclude that for all $k \in J \cap I$:

$$\varphi_2(k) = t_2(k), \qquad \varphi_1(k) = t_1(k) \tag{43}$$

Now define $\varphi : J \cap I \to \mathbb{R}$ such that:

$$\varphi(k) = \varphi_2(k)$$

In addition, since we applied Lemma 19 on point $(k^*, t_2(k^*))$, we can conclude that φ is C^1 on $J \cap I$ and the following arguments for $k \in J \cap I$ hold:

$$h(k, \varphi(k)) = 0, \qquad \varphi'(k) = -\frac{\dfrac{\partial h}{\partial k}(k, \varphi(k))}{\dfrac{\partial h}{\partial T}(k, \varphi(k))}.$$

Moreover, from Equation (43), we have:

$$\varphi(k) = t_2(k).$$

Note that the uniqueness of φ in $J \cap I$ follows from the uniqueness part of Lemma 19 applied at $(k^*, t_2(k^*))$ with the condition $\varphi(k) = t_2(k)$. This completes the proof of Lemma 20.

We now prove the larger root global implicit function lemma (Lemma 17) stated in Section B.2 using Lemma 20 as follows:

By assumption, for each $k \in I$, the equation $h(k,T) = 0$ has exactly two real roots, namely $(k, t_1(k))$ and $(k, t_2(k))$ such that $t_2(k) > t_1(k)$. Thus, the map $t_2 : I \to \mathbb{R}$, which assigns to each k the larger root $t_2(k)$, is well-defined on all of I.

We now apply Lemma 20 on every $k_0 \in I$. This implies that, for every $k_0 \in I$, there exists an open interval J_{k_0} centered at k_0 and a unique function $\varphi_{k_0} : J_{k_0} \cap I \to \mathbb{R}$ of class C^1 on $J_{k_0} \cap I$ such that

$$h\big(k, \varphi_{k_0}(k)\big) = 0 \qquad \text{for all } k \in J_{k_0} \cap I.$$

Moreover, for every $k \in J_{k_0} \cap I$ we have $\varphi_{k_0}(k) = t_2(k)$. Finally, for all $k \in J_{k_0} \cap I$, the derivative is given by

$$\varphi'_{k_0}(k) = -\,\frac{\dfrac{\partial h}{\partial k}\big(k, \varphi_{k_0}(k)\big)}{\dfrac{\partial h}{\partial T}\big(k, \varphi_{k_0}(k)\big)}.$$

In particular, for each $k_0 \in I$ we have constructed an open interval $J_{k_0} \subset I$ centered at k_0 and a C^1 function φ_{k_0} on J_{k_0} such that

$$\varphi_{k_0}(k) = t_2(k) \quad \text{for all } k \in J_{k_0} \cap I.$$

Since each J_{k_0} contains k_0, the family $\{J_{k_0}\}_{k_0 \in I}$ forms an open cover of I as follows:

$$I = \bigcup_{k_0 \in I} J_{k_0} \cap I.$$

We now check compatibility on overlaps. Let $k_0, k_1 \in I$ and consider a point $k \in J_{k_0} \cap J_{k_1} \cap I$. By construction,

$$\varphi_{k_0}(k) = t_2(k) = \varphi_{k_1}(k).$$

Thus, for every k in the intersection $J_{k_0} \cap J_{k_1}$ the two local functions φ_{k_0} and φ_{k_1} agree:

$$\varphi_{k_0}(k) = \varphi_{k_1}(k).$$

For the remainder of the proof, we will use the gluing lemma from [32, page 35].

Lemma 21 (Gluing Lemma). *Let $I \subset \mathbb{R}$ be an interval and let $\{U_\alpha\}_{\alpha \in A}$ be a family of open subsets of I such that*

$$I = \bigcup_{\alpha \in A} U_\alpha.$$

For each $\alpha \in A$, let $f_\alpha : U_\alpha \to \mathbb{R}$ be a function. Assume that for all $\alpha, \beta \in A$ and all $x \in U_\alpha \cap U_\beta$ one has

$$f_\alpha(x) = f_\beta(x).$$

Then there exists a unique function $f : I \to \mathbb{R}$ such that $f(x) = f_\alpha(x)$ for all $x \in U_\alpha$ and all $\alpha \in A$. If each f_α is of class C^1 on U_α, then f is of class C^1 on I.

We now apply Lemma 21 to the open cover $\{U_{k_0}\}_{k_0 \in I}$ of I, where $U_{k_0} := J_{k_0} \cap I$, and to the family of functions $f_{k_0} := \varphi_{k_0}$. Since $\varphi_{k_0} = \varphi_{k_1}$ on $U_{k_0} \cap U_{k_1}$, the lemma yields a unique function $f : I \to \mathbb{R}$ such that $f = \varphi_{k_0}$ on U_{k_0} for every $k_0 \in I$. Moreover, because each φ_{k_0} is of class C^1 on U_{k_0}, f is of class C^1 on I. Finally, for each $k \in U_{k_0}$ we also have $\varphi_{k_0}(k) = t_2(k)$, hence $f(k) = t_2(k)$ for each $k \in I$, and therefore t_2 is of class C^1 on I.

Now fix $k \in I$ and choose $k_0 \in I$ such that $k \in U_{k_0}$. On U_{k_0} we have $t_2 = \varphi_{k_0}$, and thus

$$t_2'(k) = \varphi_{k_0}'(k) = -\frac{\frac{\partial h}{\partial k}\big(k, \varphi_{k_0}(k)\big)}{\frac{\partial h}{\partial T}\big(k, \varphi_{k_0}(k)\big)} = -\frac{\frac{\partial h}{\partial k}\big(k, t_2(k)\big)}{\frac{\partial h}{\partial T}\big(k, t_2(k)\big)}.$$

Since k was arbitrary, the formula holds for all $k \in I$, which completes the proof.

B.4 Proof of Claim 3

Let $h_k(T)$ be as in (11) and define $h(k, T)$ as in (10) for $k \in [1, N]$ and $T \geq 0$. A direct computation yields:

$$\frac{\partial h}{\partial k}(k, T) = \frac{1}{N} + \Big(\frac{T^2}{kp}\Big)^k \Big(\log\Big(\frac{T^2}{kp}\Big) - 1\Big). \tag{44}$$

Moreover,

$$\frac{\partial^2 h}{\partial k^2}(k, T) = \Big(\frac{T^2}{kp}\Big)^k \left[\Big(\log\Big(\frac{T^2}{kp}\Big) - 1\Big)^2 - \frac{1}{k}\right]. \tag{45}$$

Fix an integer $k \in [N]$. By the mean value theorem applied on the interval $[k-1, k]$, there exists some $\hat{k} \in (k-1, k)$ such that

$$\frac{\partial h}{\partial k}(\hat{k}, T) = h(k, T) - h(k-1, T) \tag{46}$$

Applying the mean value theorem again on the interval $[k-1, \hat{k}]$, there exists some $k^* \in (k-1, \hat{k})$ such that

$$\frac{\partial^2 h}{\partial k^2}(k^*, T) = \frac{\frac{\partial h}{\partial k}(\hat{k}, T) - \frac{\partial h}{\partial k}(k-1, T)}{\hat{k} - k + 1} \tag{47}$$

Note that using the fact that $k^* > k - 1$ and Equation (35), we have:

$$t_2(k^*) > t_2(k-1) \tag{48}$$

Using Equation (45) ,Equation (33), and Equation (48), we can easily show that

$$\frac{\partial^2 h}{\partial k^2}(k^*, t_2(k-1)) < 0. \tag{49}$$

Evaluating Equation (47) at $T = t_2(k-1)$ and using Equation (49), we obtain:

$$\frac{\partial h}{\partial k}(\hat{k}, t_2(k-1)) < \frac{\partial h}{\partial k}(k, t_2(k-1)). \tag{50}$$

Combining Claim 7 with Equation (50) yields

$$\frac{\partial h}{\partial k}(\hat{k}, t_2(k-1)) < 0. \tag{51}$$

Evaluating Equation (46) at $T = t_2(k-1)$ and using Equation (51), we conclude that

$$h(k, t_2(k-1)) < h(k-1, t_2(k-1)). \tag{52}$$

By definition of $t_2(k)$,

$$h\big(k-1, t_2(k-1)\big) = 0. \tag{53}$$

Therefore, Equation (52) and Equation (53) imply

$$h(k, t_2(k-1)) < 0. \tag{54}$$

By Claim 1, $t_1(k) < t_2(k)$ are the only two roots of h_k. Since a convex function with exactly two roots is negative between the roots and non-negative outside them, together with the convexity of h_k and Equation (54), we have

$$t_2(k-1) \in [t_1(k), t_2(k)]. \tag{55}$$