Limits of Constructive Security Proofs

Michael Backes[1,2] and Dominique Unruh[1]

[1] Saarland University, Saarbrücken, Germany, {backes,unruh}@cs.uni-sb.de
[2] Max-Planck-Institute for Software Systems, Saarbrücken, Germany, backes@mpi-sws.mpg.de

Abstract. The collision-resistance of hash functions is an important foundation of many cryptographic protocols. Formally, collision-resistance can only be expected if the hash function in fact constitutes a parametrized family of functions, since for a single function, the adversary could simply know a single hard-coded collision. In practical applications, however, unkeyed hash functions are a common choice, creating a gap between the practical application and the formal proof, and, even more importantly, the concise mathematical definitions.
A pragmatic way out of this dilemma was recently formalized by Rogaway: instead of requiring that no adversary exists that breaks the protocol (existential security), one requires that given an adversary that breaks the protocol, we can efficiently construct a collision of the hash function *using an explicitly given reduction* (constructive security).
In this paper, we show the limits of this approach: We give a protocol that is existentially secure, but that provably cannot be proven secure using a constructive security proof.
Consequently, constructive security—albeit constituting a useful improvement over the state of the art—is not comprehensive enough to encompass all protocols that can be dealt with using existential security proofs.

1 Introduction

The collision-resistance of hash functions is an important ingredient of many cryptographic protocols. Formally, collision-resistance can only be expected if the hash function in fact constitutes a parametrized family of functions, since for a single function, the adversary could simply have a collision hard-coded into its program. In practical applications, however, such unkeyed hash functions are often used (e.g., SHA-1), creating a gap between the practical application and the formal proof, and, even more importantly, the concise mathematical definitions.

A pragmatic way out of this dilemma was discussed by Stinson [10] and recently formalized by Rogaway [9]: instead of requiring that no adversary exists that breaks the protocol (existential security), one requires that given an adversary that breaks the protocol, one can efficiently construct a collision of the hash function *using an explicitly given reduction* (constructive security).

Slightly more formally, the dilemma can be described as follows: An existential security proof for a protocol π shows the following: If there exists a polynomial-time adversary A that has a non-negligible advantage in breaking the protocol, then there exists a polynomial-time adversary B that has a non-negligible advantage in breaking at least one of the assumptions of the protocol. Here, the exact meaning of the word *advantage* depends on the security notion under consideration; in a proof system for example, the advantage would be the probability to convince the verifier of a wrong fact. For collision-resistant hash functions, it would be the probability of finding a collision. Considering a protocol π whose security is based on the collision-resistance of an *unkeyed* hash function H, an existential security proof would show the following: If an adversary A has non-negligible advantage in breaking π, there is an adversary B that outputs a collision of H with non-negligible probability. However, this is vacuously true: There always exists an adversary that has a collision of H hard-coded into its program and outputs this collision with probability one. We, that is the totality of all human beings, might not know this adversary, but it exists nonetheless. To circumvent this problem, mathematical definitions and proofs usually make use of keyed hash functions. In this case, for every key K the collision might be different so that the assumption that no polynomial-time adversary can compute collisions for more than a small fraction of the keys is sensible.

But what if we are forced to use unkeyed hash functions, e.g., because of efficiency considerations or simply because industrial applications often rely on unkeyed hash functions? Do we lose all possibility to prove security, since we cannot expect an existential security proof in this case? Fortunately, this is not necessarily the case: we may ground security on the observation that although there always exists an adversary finding a collision of an unkeyed hash function, this adversary might not be explicitly known. This leads to the following approach that was recently formalized by Rogaway [9]: A constructive security proof for a protocol π that uses a hash function H is an efficient transformation C (that must be explicitly given) that, upon input an adversary A and the hash function H, outputs a collision of H. If someone finds a successful adversary A, he hence also knows a collision, thereby breaking the collision-resistance of the hash function.

Rogaway [9] stresses that most existential security proofs already constitute constructive security proofs and that all that must be done for concisely handling unkeyed hash functions is to rephrase those proofs in a constructive setting. Indeed, folklore has always believed that protocols with existential security proofs can be transformed into constructive ones. In some cases it may be as easy as rephrasing the theorem statement, in other cases it may be as hard as finding a different proof. E.g., [9] writes: "In general, it is well understood that one can rephrase provable-security results as assertions about explicitly given reductions". Although this folklore statement may hold true in many cases of practical interest, we show that it does not hold true in general. We construct a protocol (more exactly, a zero-knowledge argument of knowledge) that we show secure with an existential security proof, but for which we further show that there provably does not exist any constructive security proof.

Hence although constructive security proofs may constitute a useful improvement over the state of the art, there are applications where the use of unkeyed hash functions cannot be justified even with this technique.

1.1 Our Contribution

We show how hash functions can be used to construct protocols that can be shown secure using an existential security proof, but that cannot be proven secure using a constructive security proof.

The main idea underlying this separating example is to construct a protocol whose security is based on a *non-uniform* security reduction. Then, this reduction will only lead to a *non-uniform* collision-finding algorithm. Since an unkeyed hash function can only be secure against *uniform* adversaries, such a reduction does not lead to a contradiction when basing the protocol on an unkeyed hash function. Thus, in particular, a *non-uniform* reduction does not give rise to a constructive security proof. The main technical difficulty lies in actually proving that the security of the protocol can *only* be shown using a non-uniform reduction.

More specifically, we investigate argument systems (computationally sound proof systems) as our security notion of interest. The approach can be adapted to other notions as well, e.g., by constructing a protocol for another task that uses and depends on the argument system presented in this paper.

In more detail, we construct, depending on a hash function H, a proof system (P^H, V^H) of which we can show the following properties:

- Under two nonstandard but reasonable assumptions (discussed below in the paragraph on complexity assumptions and formalized in Assumption 1 in the body of the paper) and the assumption that H is a non-uniform collision-resistant hash function, we can give an *existential* security proof for (P^H, V^H).
- Using Assumption 1, we can prove that one cannot give a *constructive* security proof that reduces the security of (P^H, V^H) to the collision-resistance of H. This even holds independent of any additional assumptions we might use for the constructive security proof (as long as these assumptions are not false).

At a first glance, this separation may seem confusing because of the different layers of assumptions (in the proofs themselves and in the proofs about proofs). Thus the following view might help to improve the intuition underlying our result: In a world where Assumption 1 has been *proven* to hold, it will be possible to show *existentially* that (P^H, V^H) is secure if H is collision-resistant, but a *constructive* security proof for (P^H, V^H) reducing to the collision resistance of H will be impossible.

At this point, we consider it important to stress that our assumptions and in particular our proofs strongly rely on the careful distinction of non-uniform and uniform complexity. In particular, we use non-uniform techniques to prove results about uniform algorithms.

Basic Idea of the Construction. In order to construct a zero-knowledge argument of knowledge that has an existential proof of security but no constructive security proof, we use the following general approach. We take an existing zero-knowledge proof of knowledge $(P^\dagger, V^\dagger)$ and modify it as follows: Instead of directly showing that a given statement σ holds, the prover P^H shows (using $P^\dagger$) that one of the following two statements holds:

– he knows a witness for the statement σ, or
– he knows a ciphertext c that is the encryption of a collision of H.

The basic idea is that given an adversary that knows such a ciphertext c, one can break the argument. However, given an adversary with a hard-coded ciphertext, a constructive security proof should not be able to extract the collision contained in the ciphertext. We have to achieve the following two goals:

– If H is a collision-resistant keyed hash function, it is hard to find a ciphertext c that is the encryption of a collision of H. Otherwise the argument can be easily broken even if the hash function is secure, thus even defying the existential security proof.
– Given c, it is hard to extract a collision from c; in particular, the decryption key should be secret. Otherwise a constructive security proof can use a knowledge extractor to extract c from a successful prover and then extract a collision from c. Further, the decryption of c should not be part of the witness used for the proof system $(P^\dagger, V^\dagger)$ since this witness could then be extracted from the adversary.

We achieve the first goal as follows: To ensure that it is hard to find a ciphertext given a collision-resistant keyed hash function, we use an encryption scheme that can be broken by non-uniform adversaries, but that is secure against uniform adversaries. An adversary that breaks (P^H, V^H) entails an adversary that finds a ciphertext c that is the encryption of a collision of H. This again entails the existence of a *non-uniform* adversary decrypting these ciphertexts and thus finding collisions. Consequently, if we require H to be a keyed hash function that is collision-resistant against *non-uniform* adversaries, we obtain a contradiction. On the other hand, a constructive security proof cannot obtain the collisions in this way, since in such a proof the reduction would have to be explicitly given and thus in particular be a uniform algorithm.

The second goal is achieved as follows: We do not directly show (using $P^\dagger$) that c is the encryption of a collision of H, since this would necessitate to use the plaintext, i.e., the collision, as a witness, which in turn would allow to extract this witness. Instead, we introduce another proof system (P^*, V^*). This proof system is non-interactive (in the strong sense that it does not even use a common reference string), statistically sound (otherwise the overall scheme could be broken by non-uniform adversaries that know a single wrong proof) and it should hide the plaintext of the encryption c. The last condition roughly means that if some adversary can extract the plaintext of c given a proof N, then it could also extract the plaintext without knowledge of N with non-negligible probability. We call such a proof system a *content-hiding proof of content.* Given a content-hiding proof of content, we do not directly prove that c is an encryption of a collision, but that we know a non-interactive proof N that c is an encryption of a collision. Then in the constructive security proof, c and N might be extractable from an adversary, but this would not be of help: If one could extract a collision from c and N, one could extract one from c alone as well (since (P^*, V^*) is content-hiding). If the encryption scheme is IND-CPA secure, the encryption c alone is indistinguishable from a random encryption. Thus one could also find the collision without using c at all. A constructive security proof would hence imply the existence of an algorithm to find collisions.

Summary of the Construction. We now summarize our construction in a more detailed and a more concise manner. Let f be a one-way permutation that is secure against uniform adversaries, but can be inverted by non-uniform adversaries (Definition 2). From f we construct an encryption scheme $\mathcal{E}_f$ such that for each security parameter, there is a fixed public key, and such that the corresponding secret key can be found by a non-uniform adversary (Definition 3). The scheme $\mathcal{E}_f$ is shown to be IND-CPA secure (Lemma 2).

Let then (P^*, V^*) be a content-hiding proof of content for the encryption scheme $\mathcal{E}_f$ (Definitions 4 and 5). That is, using P^* we can show non-interactively that a given ciphertext c is the encryption of a cleartext m that fulfills a given property π. Since P^* is content-hiding, we know that if we can extract the plaintext from c given the non-interactive proof, we can also do so without access to the proof. Let $(P^\dagger, V^\dagger)$ be a computational zero-knowledge proof of knowledge. Let H be a hash function (keyed or unkeyed). Then we construct the argument system (P^H, V^H) as follows (Definition 6):

- The prover P^H takes as input a SAT-instance σ and a corresponding witness w. The verifier V^H expects a SAT-instance σ.
- To show his knowledge of w, the prover P^H invokes the prover $P^\dagger$ to show that either
  - he knows a witness w for σ, or
  - he knows a ciphertext c and a non-interactive proof N such that the proof N convinces the verifier V^* that the ciphertext c is an encryption of a collision of H.

  The prover can easily perform this proof since he knows the witness w.
- The verifier V^H uses $V^\dagger$ to verify the above proof.

Note that the prover P^* is never used in the above construction. The existence of P^* will however be used in the proofs.

On our Complexity Assumptions. Our proof is based on the existence of content-hiding proofs of content as well as on the existence of one-way permutations with non-uniform trapdoors, which constitute nonstandard complexity assumptions. To motivate these assumptions, we prove that relative to a random oracle these assumptions follow from standard ones.

At a first glance, it may seem that a result that needs such strong assumptions and involved constructions will not be of relevance for the provability of natural protocol constructions, i.e., construction which do not have the creation of a counterexample in mind. We would like to point out the following counter-arguments: First, one reason why we need such strong assumptions is that we do not only want a protocol that cannot be proven secure using constructive proofs, but that *provably* cannot be proven secure using constructive proofs. The reason for the complexity of our example may hence not follow from the fact that all natural protocols have constructive proofs, but rather from the fact that proving unprovability is in general a difficult task. Secondly, somewhat similar techniques have already been used in the literature: Barak [3] presents an argument system in which the prover proves that the statement under consideration is true or that he knows a short circuit describing (the data sent by) the verifier. This seemingly contrived construction then was shown to allow for argument systems that enjoy properties that where shown to be impossible for zero-knowledge argument systems that do not use the circuit of the adversary (i.e., black-box zero-knowledge argument systems). In that light it may well be possible that some useful protocol will have to use constructions similar to the ones presented in this work and therefore will have no constructive security proof.

1.2 Related Work

Hash functions where first formalised in [4]. In [9] the notion of a constructive security proof was made explicit, although the concept was already discussed or implicitly used in many other papers.

The idea of considering problems relative to oracles to analyze complexity assumptions was introduced by [2]. See also [6] for a survey and a discussion of such relativisation techniques.

An example of a non-constructive security proof can be found in [5, Section 8]. They give a resettable zero-knowledge proof in the timing-model, and the proof of soundness uses a non-constructive reduction. However, it is not shown that their protocol does not have a constructive proof. In contrast, the complexity of our constructions result from the necessity of creating a scheme where we can *prove* that no constructive security proof exists. We believe that the result of [5] and our result complement each other: [5] show that there are *natural* protocols where we *do not know* constructive security proofs, while we show that there are *contrived* protocols where constructive security proofs *do not exist* (under certain complexity assumptions).

2 Preliminaries and Notation

By $x \leftarrow A$ we mean assigning the output of the probabilistic algorithm A to x, and by $x \overset{\$}{\leftarrow} M$ assigning a uniformly randomly chosen element of M to x. By $\langle A, B\rangle$ we mean the output of B after an interaction of the interactive machines A and B. The variable k will always denote the security parameter.

An *unkeyed hash function* H is a function from $\{0,1\}^*$ to $\{0,1\}^n$ for some n that can be computed in deterministic polynomial time. A *keyed (family of) hash functions* consists of a family $\{H_K\}$ of functions together with an efficient key generation algorithm G_H such that the following holds: Given K and x, the image $H_K(x)$ can be computed in deterministic polynomial time. Further, for $K \leftarrow G_H(1^k)$, the function H_K maps $\{0,1\}^*$ to $\{0,1\}^{\ell(k)}$ for some polynomially bounded function ℓ.

Of central interest to this paper is the notion of a constructive security proof. In principle, a constructive security proof consists of two parts: an explicitly given reduction C from adversaries to collisions, and a proof that C is indeed such a reduction. Since we are only interested in negative results in this paper, it will be sufficient to show that no such reduction C exists. We therefore slightly abuse notation and define a constructive security proof to solely consist of this reduction C. That is, we do not even require that the reduction is proven to be correct.

Furthermore, we will confine ourselves to constructive security proofs that a given protocol is an argument system. This results in a less abstract definition, which is sufficient for our application. Examples of constructive security proofs for other properties are given in [9].

Let (P^H, V^H) be a proof system parametrized by an unkeyed hash function H that is assumed to be given as a circuit. For an adversary A (given as a circuit) and an unsatisfiable SAT-formula σ, we define

$$\mathrm{Adv}^{\mathrm{arg}}_{V^H,k}(A,\sigma) := \Pr[\langle A, V^H(1^k,\sigma)\rangle = 1].$$

Further, for an algorithm C, let

$$\mathrm{Adv}^{\mathrm{col}}_{H,k}(C,A,\sigma) := \Pr[(x,x') \leftarrow C(1^k,H,A,\sigma) : x \neq x' \text{ and } H(x) = H(x')].$$

Definition 1 (Constructive Security Proof). *Let (P^H, V^H) be a proof system parametrised by an unkeyed hash function H. We call an algorithm C a* constructive security proof that (P^H, V^H) is an argument *if C runs in uniform probabilistic polynomial-time and there exist some $c > 0$ and some negligible function μ such that for all circuits A, all unsatisfiable boolean formulas σ and all $k \in \mathbb{N}$ we have*

$$\mathrm{Adv}^{\mathrm{col}}_{H,k}(C,A,\sigma) \geq \left(\frac{\mathrm{Adv}^{\mathrm{arg}}_{V^H,k}(A,\sigma)}{k + |A| + |H| + |\sigma|}\right)^c - \mu(k).$$

Our notion of a constructive security proof slightly deviates from the notion put forward in [9]. The most obvious difference is that [9] does not contain any asymptotic definition of a constructive security proof. Instead, all results are given in terms of concrete security, i.e., the relation between the advantage to break the protocol and the advantage to find collisions is given explicitly. A negative statement, i.e., a claim that a given protocol has no constructive security proof, cannot rely on concrete security since one does not aim to show that a given relation between the two advantages does not hold, but that no (useful) lower bound for $\mathrm{Adv}^{\mathrm{col}}$ in terms of $\mathrm{Adv}^{\mathrm{arg}}$ exists. To characterize such useful lower bounds we have introduced the above asymptotic formulation. Since we are interested in a negative result, we have made the lower bound as weak as possible.

A notion of black-box constructive proofs has also been formalized in [9]. Since black-box is the stricter kind of reduction, our negative result encompasses this notion as well.

3 Assumptions Underlying our Negative Result

In this section, we will present two cryptographic assumptions that are needed in our proof.

3.1 One-Way Permutations with Non-Uniform Trapdoors

The first assumption roughly states that there are one-way permutations that are secure against uniform adversaries but that can be inverted by non-uniform ones.

Definition 2 (One-Way Permutations with Non-Uniform Trapdoors). *A function* $f : \{0,1\}^* \to \{0,1\}^*$ *is a* one-way permutation with non-uniform trapdoors*, if*

- *The function* f *is a length-preserving permutation that is computable in deterministic polynomial time.*
- *The function* f *is one-way against uniform adversaries.*
- *There exists a sequence* t_k *of polynomial-sized circuits, such that* $t_k(f(x)) = x$ *for all* $k \in \mathbb{N}$ *and all* $x \in \{0,1\}^k$*.*

The existence of one-way permutations with non-uniform trapdoors constitutes a nonstandard complexity assumption in cryptography. Although we did not succeed in reducing the existence of one-way permutations with non-uniform trapdoors to more common assumptions in general, we show that there is an oracle relative to which this is possible.

Lemma 1. *Assume that trapdoor one-way permutations with dense public keys*[3] *exist that are one-way against uniform probabilistic polynomial-time adversaries. Then there exists an oracle* $\mathcal{O}$ *relative to which one-way permutations with non-uniform trapdoors exist.*

The proof of this lemma is given in the full version [1].

The proof of Lemma 1 in fact shows a stronger statement: choosing a random oracle entails one-way permutations with non-uniform trapdoors with probability one. If we accept the random oracle heuristic, the following conjecture is thus made realistic by the proof of Lemma 1:

[3] We say a family of trapdoor permutations has dense public keys if the distribution of the public keys is near the uniform distribution on the set of strings of a given length. Intuitively, this means that we can choose the public key using only public coins.

Conjecture 1. Let R be a sufficiently unstructured, efficiently computable function. Then using R in the construction of the proof of Lemma 1 yields one-way permutations with non-uniform trapdoors.

Using one-way permutations with non-uniform trapdoors, we can use the standard construction for creating IND-CPA secure encryption schemes from one-way permutations. The result is an encryption scheme where for each security parameter there is a single public key, and where the corresponding secret keys can be recovered by non-uniform adversaries (but not by uniform ones).

Definition 3 (Singleton Encryption). *Let f be a one-way permutation with non-uniform trapdoors. We define the* singleton encryption scheme $\mathcal{E}_f, \mathcal{D}_f$ for *f as follows: Let $pk_k := 1^k$ and $sk_k := t_k$, where t_k denotes the trapdoors of the function f. For $x \in \{0,1\}$, we have $\mathcal{E}_f(pk, x) := (f(r_1), r_2, (r_1 \cdot r_2) \oplus x)$ where r_1, r_2 are uniformly random from $\{0,1\}^{|pk|}$. For $x \in \{0,1\}^*$, we have $\mathcal{E}_f(pk, x) := (\mathcal{E}_f(pk, x_1), \ldots, \mathcal{E}_f(pk, x_{|x|}))$.*

The corresponding (deterministic) decryption algorithm $\mathcal{D}_f$ proceeds as follows: Upon input $(pk, sk, (c_1, r_2, c_2))$ where sk is a circuit and (c_1, r_2, c_2) the encryption of a single bit, the decryption algorithm first verifies that $f(sk(c_1)) = c_1$ and that $|c_1| = |pk|$. If so, it outputs $(sk(c_1) \cdot r_2) \oplus c_2$. Otherwise, it outputs $\perp$. The encryption of multiple bits is handled by decrypting each bit individually (with output $\perp$ if one of the decryptions fails).

The set of valid public keys of $\mathcal{E}_f$ for security parameter k is hence $\{pk_k\}$; consequentely the public key generation algorithm is trivial. The corresponding secret-keys sk_k, i.e., the trapdoors of f, are guaranteed to exist, but they are not efficiently computable by a uniform adversary. We have $\mathcal{D}_f(pk_k, sk_k, \mathcal{E}_f(pk_k, m)) = m$ for all m by construction; moreover, $\mathcal{D}_f(pk_k, sk, c) = m \neq \perp$ for some (possibly invalid) secret key sk implies $\mathcal{D}_f(pk_k, sk_k, c) = m$ since the checks performed by $\mathcal{D}_f$ guarantee $sk(c_1) = sk_k(c_1)$.

The following lemma states that the construction given above indeed results in an IND-CPA secure encryption scheme, at least against uniform adversaries:

Lemma 2. *Let f be a one-way permutation with non-uniform trapdoors and let $\mathcal{E}_f$ be the singleton encryption scheme for f. Then $\mathcal{E}_f$ is IND-CPA secure against uniform adversaries in the following sense: For all uniform probabilistic polynomial-time algorithms A_1, A_2, we have that*

$$\Pr\Big[(m_0, m_1, z) \leftarrow A_1(1^k), b \xleftarrow{\$} \{0,1\}, c \leftarrow \mathcal{E}_f(pk_k, m_b) : \\ A_2(1^k, c, z) = b \ \wedge \ |m_0| = |m_1|\Big]$$

is negligible in k.

A proof of Lemma 2 can be found in [8, Section 5.3.4.1]. Although this proof applies to a slightly different definition of public-key encryption where the public and secret keys are chosen by an explicit key generation algorithm, the proof carries over, mainly because the secret keys are not used in the definition of IND-CPA security.

3.2 Proofs of Content

We now introduce the novel notion of a non-interactive proof of content. Intuitively, a proof of content is a non-interactive proof system that proves that a given ciphertext c is the encryption of some plaintext m that fulfills some predicate π.

We first introduce some additional notation: Given an encryption scheme $(\mathcal{E}, \mathcal{D})$ with deterministic decryption, a Boolean circuit π, a ciphertext c, a public key pk and a private key sk, let $\pi^{pk,sk}[c] := \mathit{true}$ if and only if $m := \mathcal{D}(pk, sk, c) \neq \bot$ and $\pi(m) = 1$, and let $\pi^{pk}[c] = \mathit{true}$ if there exists a secret key sk such that $\pi^{pk,sk}[c] = \mathit{true}$.

Definition 4 (Non-Interactive Proofs of Content). *A* non-interactive proof of content for an encryption scheme $(\mathcal{E}, \mathcal{D})$ *(where $\mathcal{D}$ is deterministic) consists of a polynomial-time prover P and a polynomial-time verifier V such that the following holds:*

- Polynomial length. *There exists a polynomial p such that for every π, c, pk, sk, and k, we have $|P(1^k, \pi, c, pk, sk)| \leq p(|(1^k, \pi, c, pk, sk)|)$.*
- Completeness. *There is a negligible function μ such that for every π, c, pk and sk satisfying $\pi^{pk,sk}[c] = \mathit{true}$ and for every k, we have*

$$\Pr\bigl[V(1^k, pk, \pi, c, P(1^k, \pi, c, pk, sk)) = 0\bigr] \leq \mu(k).$$

- Soundness. *There is a negligible function μ such that for every π, c, and pk satisfying $\pi^{pk}[c] = \mathit{false}$ and for every k and every string N, we have*

$$\Pr\bigl[V(1^k, pk, \pi, c, N) = 1\bigr] \leq \mu(k).$$

So far, a proof of content can be quite easily realized by revealing the secret key of the encryption scheme. This of course is not satisfying; hence we need an additional secrecy property. We cannot expect the proof system to be zero-knowledge (since it is non-interactive without a common reference string), but we can require that a proof will not help us to extract the plaintext from the ciphertext m (which would be clearly violated if we learned the secret key). We will call this property *content-hiding*.

We now define content-hiding proofs of content. This notion will crucially depend on the notion of a valid public key of a given encryption scheme, and of the notion of the corresponding secret key. The notion of a valid public key and corresponding secret key has a natural meaning for most public-key cryptosystems, but it may not be well-defined in general. However, in the remainder of the paper we will only consider the encryption scheme from Definition 3 where a public key is valid if and only if it has the form 1^k, and where the secret key corresponding to a given public key is uniquely determined as t_k. So for the sake of readability we abstain from formally specifying what a valid public key and the corresponding secret key are.

Definition 5 (Content-Hiding Proofs of Content). *A non-interactive proof of content (P, V) for an encryption scheme $(\mathcal{E}, \mathcal{D})$ is called* content-hiding *if the following holds:*

Let G be any polynomial-time algorithm that upon input 1^k outputs a valid public key pk for $\mathcal{E}$, a message $m \in \{0,1\}^*$, a circuit π and some auxiliary information $z \in \{0,1\}^*$. Let A be any polynomial-time algorithm such that

$$\Pr\Big[(pk, m, \pi, z) \leftarrow G(1^k),\ c \leftarrow \mathcal{E}(pk, m),\ N \leftarrow P(1^k, \pi, c, pk, sk),\\ m' \leftarrow A(1^k, pk, c, \pi, z, N)\ :\ m = m'\Big]$$

is not *negligible in k, where sk denotes the secret key corresponding to pk.*

Then there exists a polynomial-time algorithm S outputting a list of strings, such that

$$\Pr\Big[(pk, m, \pi, z) \leftarrow G(1^k),\ c \leftarrow \mathcal{E}(pk, m),\ M' \leftarrow S(1^k, pk, c, \pi, z) : m \in M'\Big].$$

is not *negligible in k.*

While the definition of content-hiding proof is similar to that of witness-hiding proofs, there is an important difference: Witness-hiding proofs guarantee that the witness cannot be guessed if the statement is chosen according to some fixed distribution, while we require that the content-hiding property holds for *any* efficiently sampleable distribution on the messages m. Furthermore, a witness-hiding proof only guarantees that the witness is not disclosed as a whole, while we only require that the *message* m is not disclosed as a whole; the latter requirement is weaker since a witness would consist of m and the randomness used for encryption.

The existence of content-hiding proofs of content constitutes a novel cryptographic assumption. We did not succeed in reducing it to existing assumptions, but we show that at least there is an oracle relative to which this is possible.

Lemma 3. *Assume that trapdoor one-way permutations with dense public keys exist that are secure against non-uniform probabilistic polynomial-time adversaries. Then there exists an oracle $\mathcal{O}$ relative to which content-hiding proofs of content with deterministic verifiers exist for any encryption scheme $(\mathcal{E}, \mathcal{D})$.*

The proof of Lemma 3 (which is given in the full version [1]) establishes the following slightly stronger statement: choosing a random oracle entails content-hiding proofs of content with probability one. Hence the following conjecture is again justified by the random oracle heuristic:

Conjecture 2. Let R be a sufficiently unstructured efficiently computable function. Then using R in the construction of the proof of Lemma 3 yields content-hiding proofs of content with deterministic verifiers.

In the next section we will need both the existence of one-way permutations with non-uniform trapdoors as well as of content-hiding proofs of content. We additionally use some standard complexity assumptions. All assumptions used are summarized in the following statement:

Assumption 1 *There exist a one-way function with non-uniform trapdoors f (Definition 2) and a content-hiding proof of content with a deterministic verifier*[4] *for the singleton encryption scheme $\mathcal{E}_f$ for f (Definition 3).*

Further, we assume the existence of one-way functions secure against non-uniform adversaries and the existence of a keyed family of hash functions that is collision-resistant against non-uniform adversaries.

4 Limits of Constructive Security Proofs

Based on the definitions and assumptions from the preceding sections, we are now ready to show the existence of an existentially secure argument system that does not have a constructive security proof.

In the following, let f be a length-regular one-way function with non-uniform trapdoors, let $\mathcal{E}_f$ be the singleton encryption scheme for f, and let (P^*, V^*) denote a content-hiding proof of content for $\mathcal{E}_f$. Let $(P^\dagger, V^\dagger)$ be a computational zero-knowledge proof of knowledge, which can be constructed from one-way functions secure against non-uniform polynomial-time adversaries (see e.g., [7, Section 4.7.3]). When passing an algorithm A as argument to a function or algorithm, we assume that A is encoded as a circuit in some canonical way. Let H be the description of a function from $\{0,1\}^*$ to $\{0,1\}^*$. When considering H as a circuit, we will always mean the circuit describing the function H restricted to the domain $\{0,1\}^k$.

Stating the construction in a concise manner necessitates a few auxiliary definitions:

- Let $\pi_H(x_1, x_2) := \mathit{true}$ if and only if $x_1, x_2 \in \{0,1\}^k$, $x_1 \neq x_2$ and $H(x_1) = H(x_2)$.
- Let $\gamma(H, c, N) := \mathit{true}$ if and only if $V^*(1^k, pk_k, \pi_H, c, N) = 1$.
- Let $\eta(H, \sigma, c, N, w) := \mathit{true}$ if and only if $\sigma(w) = 1$ or $\gamma(H, c, N) = \mathit{true}$.
- Let $l_c(k) := |\mathcal{E}_f(1^k, 1^{2k})|$ denote the length of an encryption of a $2k$-bit plaintext.
- Let l_P be a polynomial such that for all $k \in \mathbb{N}$ and $c \in \{0,1\}^{l_c(k)}$, the value $l_P(k+|H|)$ is an upper bound on $|P^*(1^k, \pi_H, c, t_k)|$ where $|H|$ denotes the size of the circuit H and t_k is the non-uniform trapdoor for f (cf. Definition 2). Such a polynomial l_P exists, since there are polynomial upper bounds on all arguments of P^*, and P^* satisfies the polynomial length property from Definition 4.
- Let L_η be the language consisting of all (H, σ) such that there exist a triple (c, N, w) with $|c| \leq l_c(k)$ and $|N| \leq l_P(k + |H|)$ that satisfies $\eta(H, \sigma, c, N, w) = \mathit{true}$. Obviously, $L_\eta \in \mathrm{NP}$. Note that if $\sigma(w) = 1$, then w is a witness for $(H, \sigma) \in L_\eta$.

[4] We could also weaken the assumption slightly by allowing a probabilistic verifier. While our results hold as well for probabilistic verifiers, we have chosen to use this slightly stronger formulation since it makes the separating example and the proof easier.

Using this notation, we can now describe the protocol that will have an existential security proof, but that will provably not have a constructive proof:

Definition 6 (The Separating Argument System). *The proof system (P^H, V^H) where H may be a keyed or unkeyed hash function, is defined as follows:*

- *The prover P^H is invoked with input $(1^k, \sigma, w)$ where σ is a Boolean circuit and w is an assignment such that $\sigma(w) = 1$. The verifier is invoked with input $(1^k, \sigma)$.*
- *The prover P^H invokes $P^\dagger$ on security parameter 1^k, L_η-instance (H, σ) and witness w; here H is treated as a circuit mapping $\{0,1\}^k$ to $\{0,1\}^*$.*
- *The verifier V^H invokes $V^\dagger(1^k, \sigma)$ to verify the proof given by the prover P^H.*

The notation introduced in front of Definitions 4 and 6 (e.g., $\pi^{pk}[c]$, γ, $P^\dagger$, etc.) will be used in the following proofs without explicit reference.

We have assumed in Assumption 1 that V^* is deterministic. If V^* was probabilistic, we would have to change the above proof system as follows: First, the prover commits to a witness (c, N, w). The prover and the verifier then perform a coin-toss to choose a random tape R for V^*. Finally the prover proves that $\sigma(w) = 1$ or that the verifier V^* accepts with random tape R. We have opted to consider the case of a deterministic verifier V^* to make the presentation more readable.

Theorem 1. *Under Assumption 1, if H_K is a keyed hash-function that is secure against non-uniform adversaries then the proof system (P^H, V^H) is a (non-uniformly secure) computational zero-knowledge argument of knowledge for SAT. (We assume the key K to be chosen by some key generation algorithm $\mathcal{K}(1^k)$.)*

Proof. Since $(P^\dagger, V^\dagger)$ is a computational zero-knowledge proof, the computational zero-knowledge property and the completeness of (P^H, V^H) follow from the construction.

We show that (P^H, V^H) is an argument of knowledge, i.e., we construct a knowledge extractor E such that there exists a polynomial q such that for any non-uniform polynomial-time prover $\tilde{P}$ and any sequence σ of SAT-instances of polynomial length, there is a negligible function μ such that the following holds for each $k \in \mathbb{N}$:

$$\begin{aligned} &\Pr[K \leftarrow \mathcal{K}(1^k) : \ E^{\tilde{P}(1^k, K)}(1^k, H_K, \sigma_k) \text{ is a SAT-witness for } \sigma_k] \\ \geq \ &\frac{1}{q(k)} \Pr[K \leftarrow \mathcal{K}(1^k) : \ \langle \tilde{P}(1^k, K), V^{H_K}(1^k, \sigma_k) \rangle = 1] - \mu(k). \end{aligned} \tag{1}$$

Here $E^{\tilde{P}(1^k, K)}(1^k, H_K, \sigma_k)$ denotes the extractor E with black-box access to $\tilde{P}(1^k, K)$ and that is given a description of H_K.

Let $E_\dagger$ be the knowledge-extractor of $(P^\dagger, V^\dagger)$. Then there is a polynomial q such that for every non-uniform polynomial-time prover $\hat{P}$ and every sequence of polynomial-sized L_η-instances (H_k, σ_k) there exists a negligible function ν such that for all k the following holds:

$$\begin{aligned}&\Pr[E_\dagger^{\hat{P}(1^k)}(1^k, H_k, \sigma_k) \text{ is an } L_\eta\text{-witness for } (H_k, \sigma_k)]\\ &\geq \frac{1}{q(k)} \Pr[\langle \hat{P}(1^k), V^\dagger(1^k, H_k, \sigma_k)\rangle = 1] - \nu(k). \qquad (2)\end{aligned}$$

Here $E^{\hat{P}(1^k)}$ denotes the extractor $E_\dagger$ with black-box access to H_K and $\hat{P}(1^k, K)$.

We construct the knowledge-extractor E as follows: When invoked with black-box access to $\tilde{P}$ and with input $(1^k, H, \sigma)$, it invokes $(c, N, w) \leftarrow E_\dagger^{\tilde{P}}(1^k, H, \sigma)$ and then returns w.

It is left to show that E satisfies (1). Let $\tilde{P}$ be a non-uniform polynomial-time prover as in (1) and σ a sequence of SAT-instances of polynomial length. Let K be a sequence of keys for the hash-function H. By (2) and by definition of L_η, there exists a negligible function ν such that

$$\begin{aligned}&\Pr[(c, N, w) \leftarrow E_\dagger^{\tilde{P}(1^k, K_k)}(1^k, H_k, \sigma_k) : \eta(H_{K_k}, \sigma_k, c, N, w) = \mathit{true}]\\ &\geq \frac{1}{q(k)} \Pr[\langle \tilde{P}(1^k, K_k), V^\dagger(1^k, H_{K_k}, \sigma_k)\rangle = 1] - \nu(k) \qquad (3)\end{aligned}$$

Since this holds for every sequence K of keys, we have for some negligible ν and all $k \in \mathbb{N}$:

$$\begin{aligned}&\Pr[K \leftarrow \mathcal{K}(1^k),\ (c, N, w) \leftarrow E_\dagger^{\tilde{P}(1^k, K)}(1^k, H_K, \sigma_k) :\\ &\qquad\qquad \eta(H_K, \sigma_k, c, N, w) = \mathit{true}]\\ &\geq \frac{1}{q(k)} \Pr[K \leftarrow \mathcal{K}(1^k) : \langle \tilde{P}(1^k, K), V^\dagger(1^k, H_K, \sigma_k)\rangle = 1] - \nu(k). \qquad (4)\end{aligned}$$

(Otherwise we could simply use the worst-case sequence of keys to contradict (3).)

Let μ_1 be defined as follows:

$$\mu_1(k) := \Pr[K \leftarrow \mathcal{K}(1^k), (c, N, w) \leftarrow E_\dagger^{\tilde{P}(1^k, K)}(1^k, H_K, \sigma_k) : \gamma(H_K, c, N) = \mathit{true}].$$

By definition, $\gamma(H_K, c, N) = \mathit{true}$ is equivalent to $V^*(1^k, pk_k, \pi_{H_K}, c, N) = 1$ which in turn implies $\pi_{H_K}^{pk_k}[c] = \mathit{true}$. Hence there exists a secret key sk such that $\mathcal{D}_f(pk_k, sk, c) =: m \neq \bot$ and $\pi_{H_K}(m) = \mathit{true}$. Since $\mathcal{D}_f(pk_k, sk, c) = m \neq \bot$ implies $\mathcal{D}_f(pk_k, sk_k, c) = m$ by construction, it follows that $\pi_{H_K}(\mathcal{D}_f(pk_k, sk_k, c)) = \mathit{true}$. We therefore have

$$\begin{aligned}\mu_1(k) \leq \Pr[&K \leftarrow \mathcal{K}(1^k),\ (c, N, w) \leftarrow E_\dagger^{\tilde{P}(1^k, K)}(1^k, H_K, \sigma_k),\\ &\qquad m \leftarrow \mathcal{D}_f(pk_k, sk_k, c) :\ \pi_{H_K}(m) = \mathit{true}].\end{aligned}$$

Since $(c, N, w) \leftarrow E_\dagger^{\tilde{P}(1^k, K)}(1^k, H_K, \sigma_k)$, $m \leftarrow \mathcal{D}_f(pk_k, sk_k, c)$ can be computed by a non-uniform polynomial-time algorithm (given 1^k and K), and since

$\pi_{H_K}(m) = \mathit{true}$ implies that m encodes a collision of H_K, we have constructed a non-uniform polynomial-time algorithm that finds collisions of H_K with probability at least μ_1. Since by assumption, H_K is collision-resistant against non-uniform polynomial-time adversaries, this implies that μ_1 is negligible.

By definition, we have $\eta(H_K, \sigma_k, c, N, w) = \mathit{true}$ if and only if $\sigma_k(w) = 1$ or $\gamma(H_K, c, N) = \mathit{true}$. So using the definition of E and V^H we get

$$\begin{aligned}
&\Pr[K \leftarrow \mathcal{K}(1^k),\ w \leftarrow E^{\tilde{P}(1^k,K)}(1^k, H_K, \sigma_k) : \sigma_k(w) = 1] \\
=\ &\Pr[K \leftarrow \mathcal{K}(1^k),\ (c, N, w) \leftarrow E_\dagger^{\tilde{P}(1^k,K)}(1^k, H_K, \sigma_k) : \sigma_k(w) = 1] \\
\geq\ &\Pr[K \leftarrow \mathcal{K}(1^k),\ (c, N, w) \leftarrow E_\dagger^{\tilde{P}(1^k,K)}(1^k, H_K, \sigma_k) : \\
&\qquad\qquad\qquad\qquad \eta(H_K, \sigma_k, c, N, w) = \mathit{true}] - \mu_1(k) \\
\overset{(4)}{\geq}\ &\frac{1}{q(k)} \Pr[K \leftarrow \mathcal{K}(1^k) : \langle \tilde{P}(1^k, K), V^\dagger(1^k, H_K, \sigma_k)\rangle = 1] - \nu(k) - \mu_1(k). \\
=\ &\frac{1}{q(k)} \Pr[K \leftarrow \mathcal{K}(1^k) : \langle \tilde{P}(1^k, K), V^{H_K}(1^k, \sigma_k)\rangle = 1] - \nu(k) - \mu_1(k). \qquad (5)
\end{aligned}$$

Setting $\mu := \nu + \mu_1$, this gives us (1) and thus shows that (P^H, V^H) is a (non-uniformly secure) computational zero-knowledge argument of knowledge. □

Theorem 2. *Under Assumption 1, there exists no constructive security proof C that (P^H, V^H) is an argument.*

In particular, the theorem implies that no constructive security proof exists that (P^H, V^H) is a computational zero-knowledge argument of knowledge.

Proof. Assume for contradiction that a constructive security proof C exists that (P^H, V^H) is an argument.

Let f be a one-way permutation with non-uniform trapdoors and let $\{\tilde{H}_K\}_{K\in\mathcal{K}}$ be a keyed family of hash functions that is one-way against non-uniform adversaries. Let $G_{\tilde{H}}$ be the key generation algorithm for $\tilde{H}_K$, and assume w.l.o.g. that for $K \leftarrow G_{\tilde{H}}(1^k)$ the function $\tilde{H}_K$ maps from $\{0,1\}^*$ to $\{0,1\}^k$.

We first construct a keyed family $\{H_{a,b,K}\}_{(a,b,K)\in Y\times\mathcal{K}}$ of hash functions $H_{a,b,K} : \{0,1\}^* \to \{0,1\}^{k+1}$ with $Y := \bigcup Y_k$ and $Y_k := \{(a,b) : a, b \in \{0,1\}^k, a \neq b\}$ as follows:

$$H_{a,b,K}(x) := \begin{cases} 0\|\tilde{H}_K(x), & |x| \neq k, \\ 1\|f(x), & |x| = k,\ f(x) \neq a, \\ 1\|b, & |x| = k,\ f(x) = a. \end{cases} \qquad \text{for } a, b, x \in \{0,1\}^k.$$

It is easy to see that the only collision (x, x') of $H_{a,b,K}$ that satisfies $|x| = |x'| = k$ is $(f^{-1}(a), f^{-1}(b))$. Hence finding such a collision of $H_{a,b,K}$ for random (a, b) implies inverting f at a. Finding collisions (x, x') with $|x| \neq k$ or $|x'| \neq k$ breaks the collision-resistance of $\tilde{H}_K$. So $H_{a,b,K}$ is collision-resistant against uniform polynomial-time adversaries.

In the following, we write *k-collision* to denote a collision (x, x') with $|x| = |x'| = k$. Then there exists only a single k-collision (x, x') of $H_{a,b,K}$ (where $k = |a| = |b|$).

Let σ_{false} denote some fixed unsatisfiable circuit. Let $\tilde{P}$ be a prover that upon input $(1^k, H, c, N)$ invokes $P^\dagger$ on security parameter 1^k, L_η-instance (H, σ_{false}) and witness (c, N, w).

By construction of (P^H, V^H) and since $(P^\dagger, V^\dagger)$ is complete, there exists a negligible function μ_1 such that for all c, N with $|c| \le l_c(k)$ and $|N| \le l_P(k + |H|)$ such that N is a valid proof for $\pi_H^{pk_k}[c] = \mathit{true}$ (i.e., such that $V^*(1^k, pk_k, \pi_H, c, N) = 1$), we have

$$\Pr\Big[\langle \tilde{P}(1^k, H, c, N), V^H(1^k, \sigma_{false})\rangle = 1\Big] \ge 1 - \mu_1(k). \tag{6}$$

Consider the following game G_0:

$$(\tilde{a}, \tilde{b}) \xleftarrow{\$} Y_k,\ a := f(\tilde{a}),\ b := f(\tilde{b}),\ K \leftarrow G_{\tilde{H}}(1^k),\ H := H_{a,b,K}, \tag{7}$$
$$c \leftarrow \mathcal{E}_f(pk_k, (\tilde{a}, \tilde{b})),\ N \leftarrow P^*(1^k, \pi_H, c, pk_k, sk_k), \tag{8}$$
$$(\hat{a}, \hat{b}) \leftarrow C(1^k, H, \tilde{P}(1^k, H, c, N), \sigma_{false}). \tag{9}$$

That is, first, in (7) we construct a hash-function H such that we know the (only) k-collision $(\tilde{a}, \tilde{b})$. Then in (8) we construct an encryption c of that k-collision and a proof that c indeed contains a k-collision (i.e., that $\pi_H^{pk_k}[c] = \mathit{true}$). Finally, in (9) we invoke the generic security proof C with a description of the hash-function H, with a description of $\tilde{P}$ (instantiated with input $(1^k, H, c, N)$) and with the SAT-instance σ_{false}.

By the completeness of (P^*, V^*), there is a negligible function μ_2 such that in G_0 the following holds: $\Pr[V^*(1^k, pk_k, \pi_H, c, N) = 1] \ge 1 - \mu_2(k)$. Further, by definition of l_c and l_P it is $|c| \le l_c(k)$ and $|N| \le l_P(k + |H|)$. Then using (6) we get

$$\mathrm{Adv}_k^{\mathrm{arg}} := \Pr\Big[\langle \tilde{P}(1^k, H, c, N), V^H(1^k, \sigma_{false})\rangle = 1\Big] \ge 1 - \mu_1(k) - \mu_2(k)$$

when H, c and N are chosen as in game G_0.

Since σ_{false} is not satisfiable, this violates the soundness of the argument system (P^H, V^H). So by the definition of constructive security proofs, C should be able to extract a collision given 1^k, H, $\tilde{P}(1^k, H, c, N)$ and σ_{false}. More exactly, let p be a polynomial such that $p(k)$ bounds the length of $(1^k, H, \tilde{P}(1^k, H, c, N), \sigma_{false})$. Such a polynomial exists, since H is constructed by a polynomial-time algorithm and $\tilde{P}$ runs in polynomial time. Then there is a $c > 0$ and a negligible function μ_5 such that

$$\begin{aligned}\Pr\big[(\hat{a}, \hat{b}) \text{ is a collision of } H\big] &\ge \left(\frac{\mathrm{Adv}_k^{\mathrm{arg}}}{p(k)}\right)^c - \mu_5(k)\\ &\ge \left(\frac{1 - \mu_1(k) - \mu_2(k)}{p(k)}\right)^c - \mu_5(k) =: \nu(k).\end{aligned}$$

Then ν is not negligible. On the other hand, since $\tilde{H}_K$ is collision-resistant against non-uniform adversaries, and $(\hat{a},\hat{b})$ is computed by non-uniform polynomial-time algorithms in (7–9),[5] there is a negligible function μ_4 bounding the probability that $(\hat{a},\hat{b})$ is a collision of $\tilde{H}_K$. Since by construction of $H := H_{a,b,K}$, the only collision of H that is not a collision of $\tilde{H}_K$ is the k-collision $(f^{-1}(a), f^{-1}(b)) = (\hat{a},\hat{b})$, it follows that

$$\Pr\bigl[(\hat{a},\hat{b}) = (\tilde{a},\tilde{b})\bigr] \geq \nu(k) - \mu_4(k). \tag{10}$$

Let now $A(1^k, pk, c, \pi, H, N) := C(1^k, H, \tilde{P}(1^k, H, c, N), \sigma_{false})$. Since C and $\tilde{P}$ are polynomial-time algorithms, so is A. Further let $G(1^k)$ be an algorithm that chooses $m := (\tilde{a},\tilde{b})$ and H as in game G_0 and then outputs (pk_k, m, π_H, H). Then G runs in polynomial-time, too. Then the following game G_1 is just a rewriting of game G_0:

$$\begin{aligned}&(pk, m, \pi, H) \leftarrow G(1^k),\ c \leftarrow \mathcal{E}_f(pk, m),\\ &\qquad\qquad N \leftarrow P^*(1^k, \pi, c, pk, sk),\ m' \leftarrow A(1^k, pk, c, \pi, H, N)\end{aligned}$$

with $(\hat{a},\hat{b}) := m'$ and with sk being the secret key corresponding to pk. So by (10) it follows that $\Pr[m = m'] \geq \nu(k) - \mu_4(k)$ in game G_0. This is not negligible. Since (P^*, V^*) is content-hiding, it follows that there is a polynomial-time simulator S such that

$$\begin{aligned}\nu_2(k) := \Pr\bigl[&(pk, m, \pi, H) \leftarrow G(1^k),\ c \leftarrow \mathcal{E}_f(pk, m),\\ &M' \leftarrow S(1^k, pk, c, \pi, H)\ :\ m \in M'\bigr]\end{aligned} \tag{11}$$

is *not* negligible. Since $\mathcal{E}_f$ is IND-CPA by Lemma 2, and the algorithms in (11) are all uniform polynomial-time algorithms, we can replace $\mathcal{E}_f(pk, m)$ by $\mathcal{E}_f(pk, 0^{2k})$ (since $|m| = 2k$). (For this, note that G chooses $pk := pk_k$.) Then, for some negligible function μ_3, we have

$$\begin{aligned}\Pr[&(pk, m, \pi) \leftarrow G(1^k),\ c \leftarrow \mathcal{E}_f(pk, 0^{2k}),\\ &M' \leftarrow S(1^k, pk, c, \pi, H)\ :\ m \in M'] \geq \nu_2(k) - \mu_3(k)\end{aligned}$$

Since given a description of $H_{a,b,K}$ with $a = f(\tilde{a})$ and $b = f(\tilde{b})$, we can efficiently verify whether for some m' we have $m' = (\tilde{a},\tilde{b})$, we can modify S so that it directly outputs $m = (\tilde{a},\tilde{b})$ if that m is in M'. Call the resulting algorithm S'. By substituting the definition of G we get

$$\begin{aligned}\Pr[&(\tilde{a},\tilde{b}) \xleftarrow{\$} Y_k,\ a := f(\tilde{a}),\ b := f(\tilde{b}),\ K \leftarrow G_{\tilde{H}}(1^k),\\ &(\hat{a},\hat{b}) \leftarrow S'(1^k, pk_k, \mathcal{E}_f(pk_k, 0^{2k}), \pi_{H_{a,b,K}}, H_{a,b,K})\ :\\ &(\hat{a},\hat{b}) = (\tilde{a},\tilde{b})] \geq \nu_2(k) - \mu_3(k).\end{aligned}$$

[5] The non-uniformity stems from the appearance of sk_k in game G_0.

Let the algorithm $T(1^k, a)$ perform as follows: First, it chooses b uniformly from $\{0,1\}^k \setminus \{a\}$ and K using $G_{\tilde{H}}(1^k)$. Then it executes $(\hat{a}, \hat{b}) \leftarrow S'(1^k, pk_k, \mathcal{E}_f(pk_k, 0^{2k}), \pi_{H_{a,b,K}}, H_{a,b,K})$ and outputs $\hat{a}$. Then the previous probability can be rewritten as

$$\Pr[\tilde{a} \xleftarrow{\$} \{0,1\}^k, \hat{a} := T(1^k, f(\tilde{a})) : \tilde{a} = \hat{a}] \geq \nu_2(k) - \mu_3(k).$$

Since $\nu_2 - \mu_3$ is not negligible and T is a uniform polynomial-time algorithm, this is a contradiction to f being one-way against uniform polynomial-time adversaries. Hence our assumption that C is a constructive security proof was wrong. $\square$

References

1. Michael Backes and Dominique Unruh. Limits of constructive security proofs. `http://www.infsec.cs.uni-sb.de/~unruh/publications/backes08limits.html`, 2008. Full version of this paper.
2. T. Baker, J. Gill, and R. Solovay. Relativizations of the p $\overset{?}{=}$ NP question. *SIAM Journal on Computing*, 4:431–442, 1975.
3. Boaz Barak. How to go beyond the black-box simulation barrier. In *42th Annual Symposium on Foundations of Computer Science, Proceedings of FOCS 2001*, pages 106–115. IEEE Computer Society, 2001. Extended abstract, full version online available at `http://www.wisdom.weizmann.ac.il/~boaz/Papers/nonbb.ps`.
4. Ivan Damgård. Collision free hash functions and public key signature schemes. In *Advances in Cryptology, Proceedings of EUROCRYPT '87*, volume 304 of *Lecture Notes in Computer Science*, pages 203–216. Springer-Verlag, 1987.
5. Cynthia Dwork and Moni Naor. Zaps and their applications. ECCC TR02-001, 2002. Online available at `http://eccc.hpi-web.de/eccc-reports/2002/TR02-001/index.html`.
6. Lance Fortnow. The role of relativization in complexity theory. In *Bulletin of the EATCS 52*, February 1994. Online available at `http://people.cs.uchicago.edu/~fortnow/papers/relative.ps`.
7. Oded Goldreich. *Foundations of Cryptography – Volume 1 (Basic Tools)*. Cambridge University Press, August 2001. Previous version online available at `http://www.wisdom.weizmann.ac.il/~oded/frag.html`.
8. Oded Goldreich. *Foundations of Cryptography – Volume 2 (Basic Applications)*. Cambridge University Press, May 2004. Previous version online available at `http://www.wisdom.weizmann.ac.il/~oded/frag.html`.
9. Phillip Rogaway. Formalizing human ignorance: Collision-resistant hashing without the keys. In *Vietcrypt 2006*, volume 4341 of *Lecture Notes in Computer Science*, pages 221–228. Springer-Verlag, 2006. Online available at `http://eprint.iacr.org/2006/281`.
10. Douglas R. Stinson. Some observations on the theory of cryptographic hash functions. IACR ePrint Archive, March 2001. Online available at `http://eprint.iacr.org/2001/020`.