On The Practical Advantage of Committing Challenges in Zero-Knowledge Protocols

David Naccache and Ofer Yifrach-Stav

DIÉNS, ÉNS, CNRS, PSL University, Paris, France
45 rue d'Ulm, 75230, Paris CEDEX 05, France
ofer.friedman@ens.fr, david.naccache@ens.fr

Abstract. The Fiat-Shamir transform is a classical technique for turning any zero-knowledge Σ-protocol into a signature scheme.
In essence, the idea underlying this transform is that deriving the challenge from the digest of the commitment suppresses simulatability and hence provides non-interactive proofs of interaction.
It follows from that observation that if one wishes to preserve deniability the challenge size (per round) must be kept low. For instance in the original Fiat-Shamir protocol the authors recommend 18 bits but suggest that the challenge size can be made larger to reduce communication overhead, e.g. the value of 20 is proposed in [12].
We show that even with relatively small challenge sizes *practical* deniability can be destroyed by having the verifier artificially impose upon himself the use of slowed-down hash function or by resorting to a trusted agency proposing an on-line deniability enforcement service against the provers community's will.

1 Introduction

Authentication is a cornerstone of information security, and much effort has been put in trying to design efficient authentication primitives.

The Fiat-Shamir transform is a classical technique for turning any zero-knowledge Σ-protocol into a signature scheme.

In essence, the idea underlying this transform is that deriving the challenge from the digest of the commitment suppresses simulatability and hence provides non-interactive proofs of interaction.

It follows from that observation that if one wishes to preserve deniability the challenge size (per round) must be kept low. For instance in the original Fiat-Shamir protocol the authors recommend 18 bits but suggest that the challenge size can be made larger to reduce communication overhead, e.g. the value of 20 is proposed in [12].

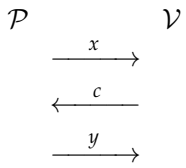
We show that even with relatively small challenge sizes *practical* deniability can be destroyed by having the verifier artificially impose upon himself the use of slowed-down hash function or by resorting to a trusted agency proposing an on-line deniability enforcement service against the provers community's will.

To that end, we will start by presenting generic notions and notations and later discuss our observation.

1.1 Σ-protocols

A Σ-protocol [1, 8, 9] is a generic 3-step interactive protocol, whereby a prover $\mathcal{P}$ communicates with a verifier $\mathcal{V}$. The goal of this interaction is for $\mathcal{P}$ to convince $\mathcal{V}$ that $\mathcal{P}$ knows some value – without revealing anything beyond this assertion. The absence of information leakage is formalized by the existence of a simulator $\mathcal{S}$, whose output is indistinguishable from the recording (trace) of the interaction between $\mathcal{P}$ and $\mathcal{V}$.

The three phases of a Σ protocol can be summarized by the following exchanges:



Namely,

- The prover sends a *commitment* x to the verifier;
- The verifier replies with a *challenge* c;
- The prover gives a *response* y.

Upon completion, $\mathcal{V}$ may accept or reject $\mathcal{P}$, depending on whether $\mathcal{P}$'s answer is satisfactory. Such a description encompasses well-known identification protocols such as Feige-Fiat-Shamir [3] and Girault-Poupard-Stern [5].

Formally, let R be some (polynomial-time) recognizable relation, then the set $L = \{v \text{ s.t. } \exists w, (v, w) \in R\}$ defines a *language*. Proving that $v \in L$ therefore amounts to proving knowledge of a witness w such that $(v, w) \in R$. A Σ-protocol satisfies the following three properties:

- *Completeness*: given an input v and a witness w such that $(v, w) \in R$, $\mathcal{P}$ is always able to convince $\mathcal{V}$.
- *Special honest-verifier zero-knowledge*[1]: there exists a probabilistic polynomial-time simulator $\mathcal{S}$ which, given v and a c, outputs triples (x, c, y) that have the same distribution as in a valid conversation between $\mathcal{P}$ and $\mathcal{V}$.
- *Special soundness*: given two accepting conversations for the same input v, with different challenges but an identical commitment x, there exists a probabilistic polynomial-time extractor procedure $\mathcal{E}$ that computes a witness w such that $(v, w) \in R$.

Many generalizations of zero-knowledge protocols have been discussed in the literature. One critical question for instance is to compose such protocols in parallel [7, 11], or to use weaker indistinguishably notions (e.g., computational indistinguishability).

[1] Note that *special* honest-verifier zero-knowledge implies honest-verifier zero-knowledge.

1.2 The Fiat-Shamir Transform

Hashing commitments is not a new idea: hashing x with a message m and using the result as c was used by Fiat and Shamir to purposely destroy deniability[2]. The Fiat-Shamir transform is a technique used to convert a zero-knowledge proof of knowledge (in particular Σ-protocols) into a digital signature scheme. The basic idea behind the transform is to replace the interaction between the prover and the verifier in the Σ-protocol with the use of a publicly computable function.

The function is constructed such that it takes as input the statement being proven, along with the challenge, and produces a response. The response, along with the statement, can then be used as a signature. The verifier can check the validity of the signature by re-computing the function using the statement and the response, and comparing the result to the original challenge.

In this way, the Fiat-Shamir transform allows one to convert a Σ-protocol, which only gives evidence of knowledge of a certain value, into a signature, which provides evidence of authenticity. The key benefit of this conversion is that the resulting signature scheme can be more efficient than a traditional zero-knowledge proof, since it eliminates the need for interaction between the prover and the verifier.

It is important to note that the Fiat-Shamir transform can only be applied to Σ-protocol where the proof is sound, that is if the verifier can efficiently check the correctness of the prover's proof.

2 Manufacturing Proofs of Interaction

The question around which revolves this paper is that of *deniability*. A key feature of a ZKP is the fact that the verifier cannot bring a proof of interaction with the prover. This corollary of simulatability is useful in many practical applications. Consider for instance a doctor's card reader $\mathcal{V}$ interacting with a patient's contactless card allowing to get an anonymous methadone prescription. The card holder is entitled to get such services, however, for medical privacy reasons he does not want any proof to exist of such an interaction with the doctor $\mathcal{V}$ and plausibly deny such interactions in case of need. A standard implementation of a ZKP perfectly answers this need: the doctor obtains the assurance that the patient is entitled for care whereas the patient leaves no traces after the consultation. Note that the card provided to the patient does not need to be anonymous, it can identify the patient (e.g. with a photo or a fingerprint) but leaves no traces.

Here we observe that for several parameter settings a malicious doctor $\mathcal{V}$ may extract a proof of interaction from a card implementing a Σ-protocol.

The idea is that of purposely slowing down hashing. Consider a slowed-down version of a hash function H, denoted H^u. H^u consists in simply iterating u times the operation $H(x)$ to increase hashing time by a factor of u.

[2] In a way the roots of this technique seem to even strech back to ElGamal's celebrated signature scheme [2].

We note that this does not contradict the theoretical *asymptotic* definition of zero-knowledge security, as it slows-down operations by a constant factor[3]. Yet it suffices to *practically* suppress deniability in several real-life parameter settings.

What happens if the prover $\mathcal{P}$ submits as a challenge $c = H^u(x)$ instead of a random c or $c = H(x)$?

A $\mathcal{V}$ wishing to deprive $\mathcal{P}$ from his deniability must exhibit a session trace x, c, y such that $c = H^u(x)$. Because c is of size k and each evaluation of H^u costs u it follows that the probability $P_k(w)$ that a $\mathcal{V}$ investing a workload $w \times u$ can falsely pretend that $\mathcal{P}$ participated in the session (while $\mathcal{P}$ did not) is:

$$P_k(w) = 1 - (1 - \frac{1}{2^k})^w$$

As a numerical application, let $u = 2^{40}$ and $k = 40$. A $\mathcal{V}$ wishing to falsely pretend, with a success probability of 0.5, that $\mathcal{P}$ participated in a session is expected to perform $w = 7.6 \times 10^{11} \times u \simeq 2^{39.5} \times u = 2^{79.5}$ hashing operations. This clearly puts the blame on $\mathcal{P}$.

With $u = 2^{45}$ and $k = 20$. A $\mathcal{V}$ wishing to falsely pretend, with a success probability of 0.5, that $\mathcal{P}$ participated in a session is expected to perform $w = 726817u \simeq 2^{19.5} \times u = 2^{64.5}$ hashing operations which, again, clearly puts the blame on $\mathcal{P}$.

Remark 1. The reader may object that there is an easy fix consisting in checking by $\mathcal{P}$ that $H^u(x) \neq c$. This is unfortunately insufficient because $\mathcal{V}$ can enrich hashing with a long secret random number r, unbeknownst to $\mathcal{P}$ (in other words hash $c = H^u(\{x, r\})$) and later exhibit r as part of the proof.

Remark 2. Even the most succinct authentication protocols require collision-resistant commitments. Interestingly, while Girault and Stern [6] proved that breaking beyond the collision-resistance size barrier is impossible, a previous research [4] showed that if we add the assumption that the verifier can measure the prover's response time, then commitment collision-resistance becomes unnecessary. The present work shows that measuring $\mathcal{V}$'s response time can also be beneficial but for another goal: preserving the prover's deniability.

2.1 Concurrent sessions

In a parallel ZKP the same $\mathcal{P}$ sends ℓ commitments $x_0, \ldots, x_{\ell-1}$ to one or several $\mathcal{V}$s and then gets ℓ challenges $c_0, \ldots, c_{\ell-1}$ to which he answers with $y_0, \ldots, y_{\ell-1}$.

It is noted that if we derive the challenges $c_0, \ldots, c_{\ell-1}$ by hashing $x_0, \ldots, x_{\ell-1}$ then, again, deniability is broken if the entropy of $c_0, \ldots, c_{\ell-1}$ is large enough (e.g. 80 bits). In this scenario we do not require a slowed-down H although slowing-down H can again serve to compensate for a smaller entropy in the $c_0, \ldots, c_{\ell-1}$. This works even if each c_i is a single bit.

[3] e.g. as is the case in [10].

A large number of papers was published on concurrent zero-knowledge, we recommend to the reader the excellent state of the art reference [13].

2.2 Using a trusted deniability enforcement agency

In this section we assume that a national agency $\mathcal{A}$ opposed to undeniability proposes an online service allowing verifiers to obtain and keep proofs of interaction with provers. We note that such a service can also be proposed by an association opposed to undeniability for ideological reasons. Here $\mathcal{A}$ is not opposed to the fact that provers are able to prove their identities, but proposes a service allowing verifiers to generate proofs of their interactions with the provers even without the provers' consent or knowledge. It is important to underline that $\mathcal{A}$ does not "cheat" or "collude" in any way but just honestly performs the service it advertises to the verifiers' community.

When $\mathcal{V}$ gets a commitment x from a prover $\mathcal{P}$, $\mathcal{V}$ forwards x and $\mathcal{P}$'s public-key (denoted $\mathsf{pk}_{\mathcal{P}}$) to $\mathcal{A}$.

$\mathcal{A}$ keeps a table counting the number of requests performed by verifiers for each $\mathsf{pk}_{\mathcal{P}}$, we denote this counter by $\omega_{\mathcal{P}}$. When an $\omega_{\mathcal{P}}$ exceeds a limit η, $\mathcal{A}$ will stop answering queries concerning $\mathsf{pk}_{\mathcal{P}}$. The goal of the bound η is to prevent verifiers from forging proofs of interaction without actually interacting with targeted provers.

If $\omega_{\mathcal{V}} < \eta$, $\mathcal{A}$ will increase $\omega_{\mathcal{P}}$ and answer the query with a signature σ on the data $x, \mathsf{pk}_{\mathcal{P}}$.

$\mathcal{V}$ will keep σ as a proof and derive the challenge c from σ by hashing.

At a later stage, $\mathcal{V}$ can exhibit σ and prove the interaction under the hypothesis that $\mathcal{A}$ played by the rules. Indeed $\mathcal{A}$ will not sign more than η commitments per verifier and this rules-out the possibility of exhaustive search by $\mathcal{V}$.

Evidently, if a central $\mathcal{A}$ is insufficient in terms of public trust, $\mathcal{A}$ can be replaced by any group signature involving a multiparty protocol that ascertains that several agencies or entities collaborated to produce σ. The odds that *all* such agencies cheat diminishes the probative value of $\mathcal{P}$'s future deniability claims.

For $k = 40$ and $\eta = 2^{20}$ (i.e. the possibility to use $\mathcal{A}$'s services one million times per prover), the odds that a $\mathcal{V}$ exhibiting a proof of interaction is falsely accusing $\mathcal{P}$ drops to 2^{-20}. Thereby, again, destroying $\mathcal{P}$'s undeniability without even $\mathcal{P}$ knowing about this.

Remark 3. To avoid resistance movements from flooding $\mathcal{A}$ with signature requests associated to a given $\mathsf{pk}_{\mathcal{P}}$ and hence protect $\mathcal{P}$[4], $\mathcal{A}$ may charge a fee for each signature and/or request $\mathcal{V}$ to identify himself and blacklist dishonest $\mathcal{V}$s as soon as those provides too many incorrect y_is corresponding to the x_is on which they requested $\mathcal{A}$'s signatures. This opens yet another resistance strategy on $\mathcal{P}$'s behalf consisting in purposely failing authentication attempts. A fix can be implemented by having verifiers refuse to identify any $\mathcal{P}$ whose $\omega_{\mathcal{P}}$ reached

[4] by pushing artificially the counter $\omega_{\mathcal{P}}$ to the limit η.

η (i.e. a $\mathcal{P}$ for which $\mathcal{A}$ denies signatures). Hence we see that this scenario is hybrid adversarial scenario involving both information security measures and cryptographic strategies. Yet in real-world settings it can be problematic and of practical significance.

3 Mitigation

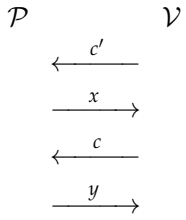
Protections against those deniability deprival scenarios revolve around three mitigation measures:

- Reduce k (preferably to say less than 8 bits) so as to preserve simulatability while preserving $\mathcal{P}$'s efficiency.
- Add to the protocol the extra specification that a time-out between the sending of x and the receiving of c must be respected. If such a time-out is not respected $\mathcal{P}$ is instructed to abort and not send y.
- Concurrent sessions should be avoided, i.e. $\mathcal{P}$ should agree to open a new session only when the previous is over (or consider a current session as interrupted[5] as soon as a new challenge y_{i+1} arrives.

In particular, countering the trusted deniability enforcement agency scenario requires reducing k and repeating the protocol to achieve the desired security level.

4 A generic mitigation

A further, less trivial and more generic mitigation, consists in banning cuncurrent interactions and modifying the protocol as follows:

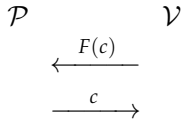


Here c' is a commitment on c. This prevents $\mathcal{V}$ from feeding $\mathcal{P}$ with a doctored c and allows using any k because $\mathcal{P}$ will abort[6] if c does not correspond to the c' received at the protocol's start. If the entropy of c is low $\mathcal{V}$ can generate a sufficiently long random r and define $c' = H(r, c)$ with r being revealed at the commitment opening stage (this requires r to be added along with c and the third exchange.

This idea can also be used to derive a deniable zero-knowledge mode of operation for any public-key cryptosystem (denoted F and F^{-1}). Consider first the following protocol:

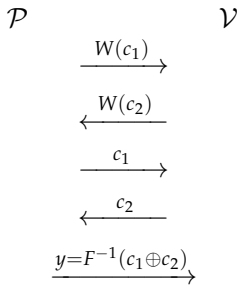
[5] i.e. $\mathcal{P}$ will not agree to send the y_i anymore.

[6] i.e. not send y.



This protocol is obviously not zero-knowledge because $\mathcal{V}$ may select as $F(c)$ a number presenting a redundancy which destroys deniability. A first solution consists in selecting c featuring a redundancy and have $\mathcal{P}$ check that c was indeed chosen honestly before returning it. This setting is trivial to simulate but it requires a random oracle or specific assumptions on the padding function used to introduce redundancy into c.

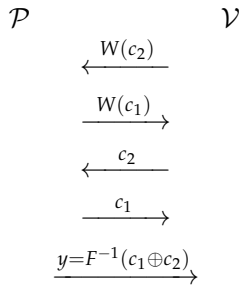
A more elegant approach not resorting to random oracles consists in jointly agreeing about a common challenge $c_1 \oplus c_2$. There are two ways of doing so. We will now use two one-way permutations U, W that can be of the same family as F to avoid requiring additional complexity assumptions[7]. The first (insecure) protocol is:



It is easy to see that this does not guarantee deniability, indeed, having received $\mathcal{P}$'s commitment first $\mathcal{V}$ may manufacture $c_2 = U(W(c_1))$, which would result in a $y = F^{-1}(c_1 \oplus U(W(c_1)))$. Because $\mathcal{V}$ is unable to invert F his only solution is to pick a random pre-image α and solve the equation $\alpha = x \oplus U(W(x))$ which is impossible. Hence exhibiting a solution demonstrates that using $\mathcal{P}$ is the only way in which the proof of interaction was obtained, which in turn destroys $\mathcal{P}$'s deniability.

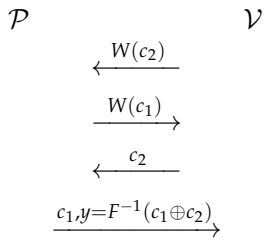
Consider now the same protocol in which $\mathcal{V}$ speaks first:

[7] There remains the technical question of generating the public parameters for U, W that we skip here as there are several algorithmic ways to do so. For instance in the case of RSA very long moduli extracted from a public constant such as π can be used thereby ascertaining that with high probability roots cannot be computed by anybody.



The situation is now radically different. We see that, having committed himself on c_2[8] $\mathcal{V}$ must work with c_2 into which he cannot inject any information coming from $\mathcal{P}$ in subsequent deniability destruction attempts.

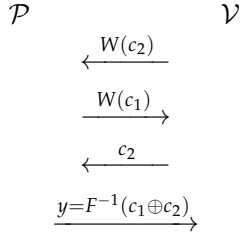
We can now note that the last steps of the above protocol are communications from $\mathcal{P}$ to $\mathcal{V}$ and hence simplify the protocol into a 4-move one:



To simulate the protocol, $\mathcal{V}$ can pick in advance a random $\bar{y}$, compute $F(\bar{y})$, pick a random $\bar{c}_1$ and compute $c_2 = \bar{y} \oplus \bar{c}_1$. $\mathcal{V}$ is now able to complete the simulation by computing the commitments $W(\bar{c}_1), W(c_2)$ to output:

$$\{W(\bar{y} \oplus \bar{c}_1), W(\bar{c}_1), \bar{y} \oplus \bar{c}_1, \bar{y}\}$$

We now note, as a last simplification step, that the transmission of c_1 at the last step is superfluous. Indeed, $\mathcal{V}$ can easily derive from y the quantity $c_1 \oplus c_2$ and, knowing c_2, derive c_1. He can hence check $W(c_1)$ and complete the protocol. This results in the simplified version:



[8] and given that the protocol will fail if this commitment is subsequently found to be false by $\mathcal{P}$

5 Conclusion

In this paper we underlined the practical risk that stems from the use of too long challenges in zero-knowledge protocols. We show that for practical purposes, even 20 or 40 bit challenges can result in situations where the prover's deniability is compromised. A generic solution, ascertaining that the challenge was chosen randomly seems to cleanly settle the issue and, given its simplicity, we recommend to implement it in practical settings where deniability is of importance.

References

1. I. Damgård. On Σ Protocols, 2010. http://www.cs.au.dk/~ivan/Sigma.pdf.
2. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *Advances in Cryptology—CRYPTO'85*, page 10–18, 1985.
3. U. Feige, A. Fiat, and A. Shamir. Zero-knowledge proofs of identity. *J. Cryptology*, 1(2):77–94, 1988.
4. H. Ferradi, R. Géraud, and D. Naccache. Slow motion zero knowledge identifying with colliding commitments. Cryptology ePrint Archive, Paper 2016/399, 2016. https://eprint.iacr.org/2016/399.
5. M. Girault. An identity-based identification scheme based on discrete logarithms modulo a composite number. In I. Damgård, editor, *Advances in Cryptology - EUROCRYPT '90, Workshop on the Theory and Application of of Cryptographic Techniques, Aarhus, Denmark, May 21-24, 1990, Proceedings*, volume 473 of *Lecture Notes in Computer Science*, pages 481–486. Springer, 1990.
6. M. Girault and J. Stern. On the length of cryptographic hash-values used in identification schemes. In Y. Desmedt, editor, *Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings*, volume 839 of *Lecture Notes in Computer Science*, pages 202–215. Springer, 1994.
7. O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991.
8. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In R. Sedgewick, editor, *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*, pages 291–304. ACM, 1985.
9. C. Hazay and Y. Lindell. *Efficient secure two-party protocols: Techniques and constructions*. Springer Science & Business Media, 2010.
10. R. C. Merkle. Protocols for public key cryptosystems. *Proceedings of the IEEE*, 75(1):56–62, 1987.
11. S. Micali and R. Pass. Local zero knowledge. In J. M. Kleinberg, editor, *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 306–315. ACM, 2006.
12. S. Micali and A. Shamir. An improvement of the Fiat-Shamir identification and signature scheme. In S. Goldwasser, editor, *Advances in Cryptology — CRYPTO' 88*, pages 244–247, New York, NY, 1990. Springer New York.
13. R. Pass. A tutorial on concurrent zero-knowledge. *IACR Cryptology ePrint Archive*, 2015:543, 2015.