Updatable proofs for hard languages
Draft, September 30, 2019

Michał Zając

Clearmatics, UK

Abstract. During a lecture at the Simons Institute workshop, Riad Wahby proposed a list of four problems that may be crucial for wide adoption of proof systems, one of them was the problem of proof updatability. Proof updatability is strictly connected updatable sound NIZK proof systems (usNIZKs), which allow users to update the SRS assuring that no party knows the SRS secret trapdoor and is able to provide acceptable proofs for incorrect statements.
Unfortunately, currently known usNIZKs do not allow to update proofs along with SRS-s. That is, a proof π procured under an SRS srs cannot be easily updated to proof π' valid in an updated SRS srs'. Thus, to assure that π could be verified in the future, the parties has to store a part of the SRS srs allowing for the proof verification, what is a great inconvenience and rises a lot of questions regarding storing the SRS-s. Proof updatability could change that – nobody would need to store the SRS-s as the only the most recent one would be enough to verify an old proof. In this short note we show that proof updatability is impossible for usNIZKs for hard languages. More precisely, given proof updatable usNIZK for a language $\mathcal{L}$ one can determine $\mathcal{L}$.

Definition 1 (Decisionally hard language). *We call language $\mathcal{L}$* decisionally hard *if for all λ and PPT algorithm $\mathcal{D}$*

$$\left|\Pr[\mathcal{D}(x) = 1 \mid x \in \mathcal{L}\,] - \Pr[\mathcal{D}(x) = 1 \mid x \notin \mathcal{L}\,]\right| \leq \mathsf{negl}(\lambda)\ .$$

Given NP decisional hard language $\mathcal{L}$ we define $\mathbf{R}$ its corresponding relation if $\mathsf{x} \in \mathcal{L}$ iff there exists w such that $(\mathsf{x}, \mathsf{w}) \in \mathbf{R}$.

We skip in this draft full formal definition of zero knowledge system, see e.g. [Gro06], and recall only on the updatable knowledge soundness property as defined in [MBKM19]. First of all, an updatable knowledge sound proof system contains three algorithms related to the SRS, instead of just one for the SRS generation:

$\mathsf{SRS.Gen}$ that on input $\mathbf{R}$ outputs an SRS srs, proof of its correctness ρ and the trapdoor td.

$\mathsf{SRS.Upd}$ that on input $\mathsf{srs}_n, Q = \{\rho_i\}_{i=0}^{n}$ outputs a new SRS srs', corresponding proof of update correctness ρ' and adds the latter to set Q.

$\mathsf{SRS.Vf}$ that takes an SRS srs_n, sequence of proofs of correct updates $\{\rho_i\}_{i=0}^{n}$ and outputs 1 iff srs_n is correctly generated.

```
$\mathsf{U\text{-}KSND}^{\mathcal{A},\mathsf{Ext}_{\mathcal{A}}}(\mathbf{R})$

$\mathsf{srs} \leftarrow \bot$
$(\mathsf{x}, \pi) \leftarrow \mathcal{A}^{\mathsf{U\text{-}}\mathcal{O}_{\mathsf{s}}}(\mathbf{R}; r)$
$\mathsf{w} \leftarrow \mathsf{Ext}_{\mathcal{A}}(\mathsf{srs}, r)$
return $\mathsf{V}(\mathsf{srs}, \mathsf{x}, \pi) \wedge (\mathsf{x}, \mathsf{w}) \notin \mathbf{R}$
```

```
$\mathsf{U\text{-}}\mathcal{O}_{\mathsf{s}}(\mathbf{R}, \mathsf{intent}, \mathsf{srs}_n, \{\rho_i\}_{i.=0}^{n})$

if $\mathsf{srs} \neq \bot$ return $\bot$
if $\mathsf{intent} = \mathrm{setup}$
    $(\mathsf{srs}', \rho') \leftarrow \mathsf{SRS.Gen}(\mathbf{R})$
    $Q \leftarrow Q \cup \{\rho'\}$
    return $(\mathsf{srs}', \rho')$
if $\mathsf{intent} = \mathrm{update}$
    $b \leftarrow \mathsf{SRS.Vf}(\mathbf{R}, \mathsf{srs}_n, \{\rho_i\}_{i=0}^{n})$
if $b = 0$ return $\bot$
    $(\mathsf{srs}', \rho') \leftarrow \mathsf{SRS.Upd}(\mathbf{R}, \mathsf{srs}_n, \{\rho_i\}_{i=0}^{n})$
    $Q \leftarrow Q \cup \{\rho'\}$
    return $(\mathsf{srs}', \rho')$
if $\mathsf{intent} = \mathrm{final}$
    $b \leftarrow \mathsf{SRS.Vf}(\mathbf{R}, \mathsf{srs}_n, \{\rho_i\}_{i=0}^{n})$
    if $(b = 0) \vee (Q \cap \{\rho_i\}_{i=0}^{n}) = \emptyset$ return $\bot$
    $\mathsf{srs} \leftarrow \mathsf{srs}_n$
    return $\mathsf{srs}$
else return $\bot$
```

Fig. 1. Updatable knowledge soundness game.

We require that the SRS verification is complete, i.e. $\mathsf{SRS.Vf}$ accepts all SRS-s generated by $\mathsf{SRS.Gen}$ and $\mathsf{SRS.Upd}$.

Definition 2 (Updatable knowledge soundness). *Zero knowledge proof system* NIZK *for relation* $\mathbf{R}$ *is updatable knowledge sound if for all PPT algorithms* $\mathcal{A}$ *there exists an extractor* $\mathsf{Ext}_{\mathcal{A}}$ *such that the probability* $\Pr\left[\mathsf{U\text{-}KSND}^{\mathcal{A},\mathsf{Ext}_{\mathcal{A}}}(1^{\lambda})\right]$*, cf. Fig. 1, is negligible in* λ*.*

Definition 3 (Fully updatable proof system). *We call a zero-knowledge proof system* NIZK fully updatable *if it is updatable-sound and equipped with an efficiently computable function* $\mathsf{Proof.Upd}$ *that given an instance* π*, proof* π_0 *valid for an SRS* srs_0*, updated SRS* srs_n*, and proofs of correct updates* $\{\rho_i\}_{i=0}^{n}$ *returns proof* π_n*, such that* $\mathsf{V}(\mathbf{R}, \mathsf{srs}_n, \{\rho_i\}_{i=0}^{n}, \mathsf{x}, \pi_n)$ *accepts.*

Theorem 1. *Let* $\mathcal{L}$ *be a decisionally hard* NP *language and* $\mathbf{R}$ *its corresponding instance-witness relation. Then there exists no fully updatable zero-knowledge proof system for* $\mathbf{R}$*.*

Proof. We proceed by contradiction. Let NIZK be such a fully updatable proof system for $\mathbf{R}$, we build an algorithm $\mathcal{D}$ that on input x decides whether x belongs to $\mathcal{L}$ with non-negligible advantage.

First, $\mathcal{D}$ sets up an SRS update oracle $\mathsf{U}\text{-}\mathcal{O}_\mathsf{s}$ and queries it on the setup intent getting SRS srs_0 and proof ρ_0 of its correctness. Since $\mathcal{D}$ runs $\mathsf{U}\text{-}\mathcal{O}_\mathsf{s}$ internally, she learns a trapdoor td_0 corresponding to srs_0 and is able to produce a simulated proof π_0 for x. Second, $\mathcal{D}$ runs a PPT algorithm $\mathcal{A}$ that is given the initial SRS srs_0, $\{\rho_0\}$, and access to $\mathsf{U}\text{-}\mathcal{O}_\mathsf{s}$. In the end, the SRS is srs_n, for some natural n, and the sequence of update correctness proofs is $\{\rho_i\}_{i=0}^n$. Since NIZK is fully updatable, a verifier $\mathsf{V}(\mathbf{R}, \mathsf{srs}_n, \{\rho_i\}_{i=0}^n, \mathsf{x}, \mathsf{Proof.Upd}(\mathsf{srs}_n, \{\rho\}_{i=0}^n, \pi_0))$ accepts $\mathsf{x} \notin \mathcal{L}$ with negligible probability only. Furhtermore, since the proof system is complete, the verifier accepts an honestly generated proof for $\mathsf{x} \in \mathcal{L}$. Finally, $\mathcal{D}$ outputs 1 iff the verifier accepts an updated proof.

References

Gro06. Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In Xuejia Lai and Kefei Chen, editors, *ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 444–459. Springer, Heidelberg, December 2006. `doi:10.1007/11935230_29`. 1

MBKM19. Mary Maller, Sean Bowe, Markulf Kohlweiss, and Sarah Meiklejohn. Sonic: Zero-knowledge snarks from linear-size universal and updateable structured reference strings. *IACR Cryptology ePrint Archive*, 2019:99, 2019. URL: `https://eprint.iacr.org/2019/099`. 1