Slow Motion Zero Knowledge Identifying With Colliding Commitments

No Author Given

No Institute Given

Abstract. Discrete-logarithm authentication protocols are known to present two interesting features: The first is that the prover's commitment, $x = g^r$, claims most of the prover's computational effort. The second is that x does not depend on the challenge and can hence be computed in advance. Provers exploit this feature by pre-loading (or pre-computing) ready to use commitment pairs r_i, x_i. The r_i can be derived from a common seed but storing each x_i still requires 160 to 256 bits when implementing DSA or Schnorr.
This paper proposes a new concept called *slow motion zero-knowledge* (SM-ZK). SM-ZK allows the prover to slash commitment size (by a factor of 4 to 6) by combining classical zero-knowledge and a timing side-channel. We pay the conceptual price of requiring the ability to measure time but, in exchange, obtain communication-efficient protocols.

1 Introduction

Authentication is a cornerstone of information security, and much effort has been put in trying to design efficient authentication primitives. However, even the most succinct authentication protocols require collision-resistant commitments. As proved by Girault and Stern [12], breaking beyond the collision-resistance size barrier is impossible. This paper shows that if we add the assumption that the verifier can measure the prover's response time, then commitment collision-resistance becomes unnecessary. We call this new construction *slow-motion zero knowledge* (SM-ZK).

As we will show, the parameter determining commitment size in SM-ZK protocols is the attacker's online computational power rather than the attacker's overall computational power. As a result, SM-ZK allows a significant reduction (typically by a factor of 4 to 6) of the prover's commitment size.

The prover's on-line computational effort remains unchanged (enabling instant replies in schemes such as GPS [11]). The prover's offline work is only slightly increased. The main price is paid by the verifier who has to solve a time-puzzle per session. The time taken to solve this time-puzzle determines the commitment's shortness.

The major contribution of this work is thus a technique forcing a cheating prover to either attack the underlying zero-knowledge protocol or exhaust the space of possible replies in the presence of a time-lock function that slows down his operations. When this time-lock function is properly tuned, a simple time-out on the verifier's side rules out cheating provers. It is interesting to contrast this approach to the notion of *knowledge tightness* introduced by Goldreich, Micali and Widgerson [13], and generalizations such as *precise/local ZK* introduced by Micali and Pass [17], which uses similar time-constraint arguments but to prove reduced knowledge leakage bounds.

2 Building Blocks

SM-ZK combines two existing building blocks that we now recall: three-pass zero-knowledge protocols and time-lock functions.

2.1 Three-Pass Zero-Knowledge Protocols

A Σ-protocol [5, 14, 15] 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. 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 $(v, w) \in R$.

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

$$\begin{array}{ccc} & \xrightarrow{x} & \\ \mathcal{P} & \xleftarrow{c} & \mathcal{V} \\ & \xrightarrow{y} & \end{array}$$

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.

For our purposes, the conversation is satisfactory if, for some public function f we have $f(x, c, y) = x$. In particular this encompasses well-known identification protocols such as Feige-Fiat-Shamir [9] and Girault-Poupard-Stern [10].

A Σ-protocol must furthermore satisfy 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 $\mathcal{V}$.

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

– *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 $(v, w) \in R$.

When a function f is such that these properties are satisfied, we shall say that f *defines* a Σ protocol.

2.2 Commitment Pre-Processing

Because x does not depend on c, authors quickly noted that x can be prepared in advance. This is of little use in protocols where the creation of x is easy (e.g. Fiat-Shamir [9]). Discrete-logarithm commitment pre-processing is a well-known optimization technique (e.g. [18,21]) that exploits two properties of DLP.

The first is the fact that the commitment generation $x = g^r$ claims most of the prover's efforts.

The second is that x does not depend on the challenge c and can hence be computed in advance. A pre-computed commitment is hence defined a r, x computed in advance for $\mathcal{P}$ (or by $\mathcal{P}$). Because several pre-computed commitments usually need to be saved by $\mathcal{P}$ for later use, it is possible to derive all the r_i components by hashing a common seed.

2.3 Time-Lock Puzzles

Time-lock puzzles [16,20] are problems designed to guarantee that they will take (approximately) τ units of time to solve. Like proof-of-work protocols [7], time-locks have found applications in settings where delaying requests is desirable, such as fighting spam or denial-of-service attacks, as well as in electronic cash [1,6,8].

Time-lock puzzles may be based on computationally demanding problems, but not all such problems make good time-locks. For instance, inverting a weak one-way function would in general not provide a good time-lock candidate [20]. The intuition is that the time it takes to solve a time-lock should not be significantly reduced by using more computers (*i.e.* parallel brute-force) or more expensive machines.

Definition 1 (Time-lock puzzle). *A time-lock puzzle is a problem such that there is a super-polynomial gap between the work required to generate the puzzle, and the parallel time required to solve it (for a polynomial number of parallel processors).*

Rivest, Shamir and Wagner [20], and independly Boneh and Naor [4] proposed a time-lock puzzle construction relying on the assumption that factorization is hard. This is the construction we retain for this work, and to the best of our knowledge the only known one to achieve interesting security levels.

Indeed, Mahmoody, Moran and Vadhan [16] showed that time-lock puzzles with large difficulty gap are impossible in the Random Oracle Model : for every time-lock puzzle there exists a parallel adversary that can solve the puzzle in no more time than it takes to generate and makes only polynomially more queries to the random oracle than the best honest (serial) solver. The same authors show however how to construct a linear-gap puzzle in the ROM.

Definition 2 (Rivest-Shamir-Wagner Time-Lock). *The original Rivest-Shamir-Wagner (RSW) time-lock [20] is based on the "intrinsically sequential" problem of computing:*

$$2^{2^{\tau}} \bmod n$$

for specified values of τ and an RSA modulus n. The parameter τ controls the puzzle's difficulty. The puzzle can be solved by performing τ successive squares modulo n.

As pointed out in [20], there is no known shortcut allowing to compute the puzzle without knowing the factors of n.

3 Slow Motion Zero-Knowledge Protocols

3.1 Definition

We can now introduce the following notion:

Definition 3 (SM-ZK). *A* Slow Motion Zero-Knowledge *(SM-ZK) protocol $(f, T_\tau, \Delta_{max})$, where f defines a Σ protocol, T_τ is a time-lock puzzle and $\Delta_{max} \in \mathbb{R}$, is defined by the three following steps:*

1. *Commitment: $\mathcal{P}$ sends a commitment x to $\mathcal{V}$*
2. *Timed challenge: $\mathcal{V}$ sends a challenge c to $\mathcal{P}$, and starts a timer.*
3. *Response: $\mathcal{P}$ provides a response y to $\mathcal{V}$, which stops the timer.*

$\mathcal{V}$ accepts iif $T_\tau\left(f(x, c, y)\right) = x$ and *if the time elapsed, as measured by the timer, is smaller than Δ_{max}.*

Such a protocol must satisfy a new notion of soundness:

Definition 4 (Time-constrained soundness). *Let $\epsilon' > 0$, then there exists $\tau > 0$ such that, if a prover $\mathcal{A}$ is accepted with non-negligible probability $\epsilon = 1/B^N + \epsilon'$ by honest verifiers, then with overwhelming probability, $\mathcal{A}$ knows s.*

3.2 Commitment shortening

Commitments in a Σ protocol are under the control of $\mathcal{P}$, which may be malicious. If commitments are not collision-resistant, the protocol's security is weakened. Hence commitments need to be long, and in classical Σ protocols breaking below the collision-resistance size barrier is impossible as proved by [12].

However, as we now show, commitment collision-resistance becomes unnecessary in the case of SM-ZK protocols.

4 An example SM-ZK

While SM-ZK can be instantiated with any three-pass ZK protocol, we will illustrate the construction using the Girault-Poupard-Stern (GPS) protocol [10,11,19], and the time-lock construction due to Rivest, Shamir and Wagner [20].

4.1 Girault-Poupard-Stern Protocol

$\mathcal{P}$		$\mathcal{V}$
$r \in_R [0, A-1]$ $x \leftarrow g^r \bmod n$		
	$\xrightarrow{x}$	
		$c \in_R [0, B-1]$
	$\xleftarrow{c}$	

if $c \notin [0, B-1]$ **then**
abort
else
$y \leftarrow r + c \times s$		

	$\xrightarrow{y}$	
		if $y \in [0, A + (B-1)(S-1) - 1]$ **and** $g^y v^c = x \bmod n$ **then** valid **else** invalid

Fig. 1. Girault-Poupard-Stern identification protocol.

GPS key generation consists in generating a composite modulus n, choosing a public generator $g \in [0, n-1]$ and integers A, B, S such that $A \gg BS$. The secret key is an integer $s \in [0, S-1]$, and the corresponding public key is $v = g^{-s} \bmod n$. Authentication is performed as in Figure 1.

Implicitly, parameters A, B, S are functions of the security parameter k.

We recall in Figure 2 the way in which pre-computed commitments are generated and used. Note that the authority's role (separated here for the sake of clarity) can be played by $\mathcal{P}$.

4.2 GPS-RSW SM-ZK

We can now combine the previous building-blocks to construct a pre-processing scheme that requires little commitment storage. The starting point is a slightly

Authority		$\mathcal{P}$
		$J \in_R [1, 2^{512}]$
	$\xleftarrow{J}$	

for $i = 1$ to k **do**
$\quad r_i \leftarrow H(J, i, s)$
$\quad x_i \leftarrow g^{r_i} \bmod n$
end for		

	$\xrightarrow{x_1, \ldots, x_k}$	
		Store $J, x_1, \ldots, x_k$

$\mathcal{P}$		$\mathcal{V}$
$i \leftarrow i + 1$		
	$\xrightarrow{x_{i-1}}$	
$r \leftarrow H(J, i-1, s)$		$c \in_R [0, B-1]$
	$\xleftarrow{c}$	
$y \leftarrow r + s \times c$		
	$\xrightarrow{y}$	
		$\mathsf{Verify}(x_{i-1}, y, c)$

Fig. 2. Commitment pre-processing as applied to GPS. The first stage describes the preliminary interaction with a trusted authority, where pre-computed commitments are generated and stored. The second stage describes the interaction with a verifier. For the sake of clarity the range-tests on c and y were omitted. The trusted authority can be easily replaced by $\mathcal{P}$ himself.

modified version of the RSW time-lock function:

$$f_{\tau,\ell}(x) = \left(\mu(x)^{2^\tau} \bmod \overline{n}\right) \bmod 2^\ell$$

Here, $\overline{n}$ is an RSA modulus (different from the n used for the GPS), μ is a deterministic RSA signature padding function (*e.g.* the Full Domain Hash [2]), τ is a parameter controlling computation time (for parties who do not know the factors of $\overline{n}$) and ℓ is a parameter controlling f's output size.

The function $f_{\tau,\ell}$ only differs from the RSW time-lock in two respects: the value x is masked by μ, and $f_{\tau,\ell}$ is reduced modulo 2^ℓ. These modifications do not affect the arguments backing the RSW time-lock, and under the same assumptions (hardness of factorization) $f_{\ell,\tau}$ is also a time-lock function.

Then, we adapt a construction of M'Raïhi and Naccache [18] to GPS [10]. This is done by defining a common secret J and having the trusted authority generate the quantities:

$$x'_i = g^{H(J,i,s)} \bmod n$$

However, the authority compresses these x'_i by computing $x_i = f_{\tau,\ell}(x'_i)$. Note that because the authority knows the factors of $\overline{n}$, computing the x_i is fast. $\mathcal{P}$ is loaded with k pre-computed commitments $x_1, \ldots, x_k$ as shown in Figure 3.

Authority		$\mathcal{P}$
		$J \in_R [1, 2^{512}]$
	$\xleftarrow{J}$	

for $i = 1$ to k **do**
$r_i \leftarrow H(J, i, s)$
$x'_i \leftarrow g^{r_i} \bmod n$
$x_i \leftarrow f_{\ell,\tau}(x'_i)$
end for		

	$\xrightarrow{x_1,\ldots,x_k}$	
		Store $J, x_1, \ldots, x_k$

Fig. 3. Slow motion commitment pre-processing for GPS.

When $\mathcal{V}$ wishes to authenticate $\mathcal{P}$ the parties execute the protocol shown in Figure 4.

With a proper choice of ℓ, τ we can have a reasonable verification time (assuming that $\mathcal{V}$ is more powerful than $\mathcal{P}$), extremely short commitments (*e.g.* 40-bit ones) and very little on-line computations required from $\mathcal{P}$.

4.3 Choice of Parameters

What drives the choice of parameters is the ratio between:

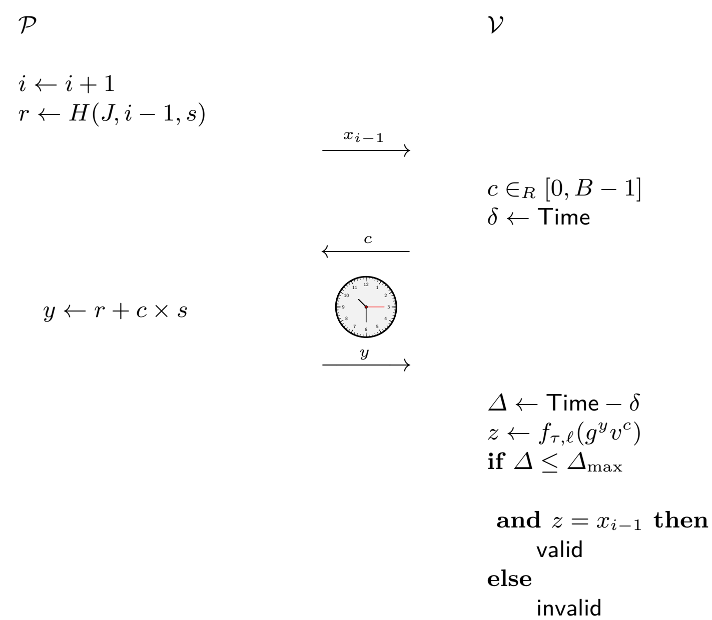

Fig. 4. Slow Motion GPS. Range tests on c and y omitted for the sake of clarity.

- The time t it takes to a legitimate prover to compute y and transmits it. In GPS this is simply one multiplication of operands of sizes $\log_2 B$ and $\log_2 S$ (additions neglected), this takes time $\lambda \log(B) \log(S)$ for some constant λ (not assuming optimizations such as [3] based on the fact that operand s is constant).
- The time T it takes for the fastest adversary to evaluate once the time-lock function $f_{\tau,\ell}$. T does not really depend on ℓ, and is linear in τ. We hence let $T = \nu\tau$. Note that there is no need to take into account the size of $\overline{n}$, all we require from $\overline{n}$ is to be hard to factor. That way, the slowing effect will solely depend on τ.

In a brute-force attack, there are 2^ℓ possibilities to exhaust. The most powerful adversary may run $\kappa \leq 2^\ell$ parallel evaluations of the time-lock function, and succeed to solve the puzzle in t time units with probability

$$\epsilon = \frac{\kappa t}{2^\ell T} = \frac{\kappa \log(B) \log(S) \lambda}{\nu 2^\ell \tau}$$

A typical instance resulting in 40-bit commitments is $\{\kappa = 2^{24}, T = 1, t = 2^{-4}, \epsilon = 2^{-20}\} \Rightarrow \ell = 40$. Here we assume that the attacker has 16.7 million (2^{24}) computers capable of solving one time-lock challenge per second ($T = 1$) posing as a prover responding in one sixteenth of a second ($t = 2^{-4}$). Assuming the least secure DSA parameters (160-bit q) this divides commitment size by 4. For 256-bit DSA the gain ratio becomes 6.4.

The time-out constant $\Delta_{\max}$ in Figure 4 is tuned to be as small as possible, but not so short that it prevents legitimate provers from authenticating. Therefore the only constraint is that $\Delta_{\max}$ is greater or equal to the time t it takes to the *slowest legitimate prover* to respond. Henceforth we assume $\Delta_{\max} = t$.

5 Security Proof

The security of this protocol is related to that of the standard GPS protocol analysed in [11, 19]. We recall here the main results and hypotheses.

5.1 Preliminaries

The following scenario is considered. A randomized polynomial-time algorithm Setup generates the public parameters $(\mathcal{G}, g, S)$ on input the security parameter k. Then a second probabilistic algorithm GenKey generates pairs of public and private keys, sends the secret key to $\mathcal{P}$ while the related public key is made available to anybody, including of course $\mathcal{P}$ and $\mathcal{V}$. Finally, the identification procedure is a protocol between $\mathcal{P}$ and $\mathcal{V}$, at the end of which $\mathcal{V}$ accepts or not.

An adversary who doesn't corrupt public parameters and key generation has only two ways to obtain information: either passively, by eavesdropping on a regular communication, or actively, by impersonating (in a possibly non protocol-compliant way) $\mathcal{P}$ and $\mathcal{V}$.

The standard GPS protocol is proven complete, sound and zero-knowledge by reduction to the *discrete logarithm with short exponent problem* [11]:

Definition 5 (Discrete logarithm with short exponent problem). *Given a group $\mathcal{G}$, $g \in \mathcal{G}$, and integer S and a group element g^x such that $x \in [0, S-1]$, find x.*

5.2 Compressed Commitments For Time-Locked GPS

We now consider the impact of shortening the commitments to ℓ bits on security, while taking into account the time constraint under which $\mathcal{P}$ operates. The shortening of commitments will indeed weaken the protocol [12] but this is compensated by the time constraint, as explained below.

Lemma 1 (Completeness). *Execution of the protocol of Figure 4 between a prover $\mathcal{P}$ who knows the secret key corresponding to his public key, and replies in bounded time $\Delta_{\max}$, and a verifier $\mathcal{V}$ is always successful.*

Proof. This is a direct consequence of the completeness of the standard GPS protocol [11, Theorem 1]. By assumption, $\mathcal{P}$ computes y and sends it within the time allotted for the operation. This computation is easy knowing the secret s and we have

$$g^y v^c = g^{r_i + cs} v^c = x'_i g^{cs} v^c = x'_i v^{c-c} = x'_i$$

Consequently, $f_{\tau,\ell}(g^y v^c) = f_{\tau,\ell}(x'_i) = x_i$. Finally,

$$y = r + cs \leq (A-1) + (B-1)(S-1) < y_{\max}.$$

Therefore all conditions are met and the identification succeeds. □

Lemma 2 (Zero-Knowledge). *The protocol of Figure 4 is statistically zero-knowledge if it is run a polynomial number of times N, B is polynomial, and NSB/A is negligible.*

Proof. The proof follows [11] and can be found in Appendix A.

The last important property to prove is that if $\mathcal{V}$ accepts, then with overwhelming probability $\mathcal{P}$ must know the discrete logarithm of v in base g.

Lemma 3 (Time-constrained soundness). *Under the assumption that the discrete logarithm with short exponent problem is hard, and the time-lock hardness assumption, this protocol achieves time-constrained soundness.*

Proof. After a commitment x has been sent, if $\mathcal{A}$ can correctly answer with probability $> 1/B$ then he must be able to answer to two different challenges, c and c', with y and y' such that they are both accepted, *i.e.* $f_{\ell,\tau}(g^y v^c) = x = f_{\ell,\tau}(g^{y'} v^{c'})$. When that happens, we have

$$\mu\left(g^y v^c\right)^{2^\tau} = \mu\left(g^{y'} v^{c'}\right)^{2^\tau} \bmod \overline{n} \bmod 2^\ell$$

Here is the algorithm that extracts these values from the adversary $\mathcal{A}$:

Step 1. Pick a random tape ω_2 and a tuple c of N integers $c_1, \dots, c_N$ in $[0, B-1]$. If $\text{Success}(\omega_2, \text{c}) = \mathsf{false}$, then abort.

Step 2. Probe random N-tuples c' that are different from each other and from c, until $\text{Success}(\omega_2, \text{c}') = \mathsf{true}$. If after $B^N - 1$ probes a successful c' has not been found, abort.

Step 3. Let j be the first index such that $c_j \neq c'_j$, write y_j and y'_j the corresponding answers of $\mathcal{A}$. Output c_j, c'_j, y_j, y'_j.

This algorithm succeeds with probability $\geq \epsilon - 1/B^N = \epsilon'$, and takes at most $4\Delta_{\max}$ units of time [11]. This means that there is an algorithm finding collisions in $f_{\ell,\tau}$ with probability $\geq \epsilon'$ and time $\leq 4\Delta_{\max}$.

Assuming the hardness of the discrete logarithm with short exponents problem, the adversary responds in time by solving a hard problem, where as pointed out earlier the probability of success is given by

$$\zeta = \frac{\kappa \log(B) \log(S) \lambda}{\nu 2^\ell \tau}$$

where κ is the number of concurrent evaluations of $f_{\ell,\tau}$ performed by $\mathcal{A}$. There is a value of τ such that $\zeta \ll \epsilon$. For this choice of τ, $\mathcal{A}$ is able to compute $f_{\ell,\tau}$ much faster than brute-force, which contradicts the time-lock hardness assumption. □

6 Conclusion and Further Research

This paper introduced a new class of protocols, called Slow Motion Zero Knowledge (SM-ZK) showing that if we pay the conceptual price of allowing time measurements during a three-pass ZK protocol then commitments do not need to be collision-resistant.

Because of its interactive nature, SM-ZK does not yield signatures but seems to open new research directions. For instance, SM-ZK permits the following interesting construction, that we call a *fading signature*: Alice wishes to send a signed email m to Bob without allowing Bob to keep a long-term proof of her involvement. By deriving $c \leftarrow H(x, m, \rho)$ where ρ is a random challenge chosen by Bob, Bob can can convince himself[2] that m comes from Alice. This conviction is however not transferable if Alice prudently uses a short commitment as described in this paper.

References

1. Abadi, M., Burrows, M., Manasse, M.S., Wobber, T.: Moderately hard, memory-bound functions. ACM Trans. Internet Techn. 5(2), 299–327 (2005)
2. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 3-5, 1993. pp. 62–73 (1993)
3. Bernstein, R.L.: Multiplication by integer constants. Softw., Pract. Exper. 16(7), 641–652 (1986)
4. Boneh, D., Naor, M.: Timed commitments. In: Bellare, M. (ed.) Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings. Lecture Notes in Computer Science, vol. 1880, pp. 236–254. Springer (2000)
5. Damgård, I.: On Σ Protocols (2010), http://www.cs.au.dk/~ivan/Sigma.pdf
6. Dwork, C., Goldberg, A., Naor, M.: On memory-bound functions for fighting spam. In: Boneh, D. (ed.) Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2729, pp. 426–444. Springer (2003)
7. Dwork, C., Naor, M.: Pricing via processing or combatting junk mail. In: Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings. pp. 139–147 (1992)
8. Dwork, C., Naor, M., Wee, H.: Pebbling and proofs of work. In: Shoup, V. (ed.) Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3621, pp. 37–54. Springer (2005)
9. Feige, U., Fiat, A., Shamir, A.: Zero-knowledge proofs of identity. J. Cryptology 1(2), 77–94 (1988)

[2] If y was received before $\Delta_{\max}$.

10. Girault, M.: An identity-based identification scheme based on discrete logarithms modulo a composite number. In: Advances in Cryptology - EUROCRYPT '90, Workshop on the Theory and Application of of Cryptographic Techniques, Aarhus, Denmark, May 21-24, 1990, Proceedings. pp. 481–486 (1990)
11. Girault, M., Poupard, G., Stern, J.: On the fly authentication and signature schemes based on groups of unknown order. J. Cryptology 19(4), 463–487 (2006)
12. Girault, M., Stern, J.: On the length of cryptographic hash-values used in identification schemes. In: Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings. pp. 202–215 (1994)
13. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. J. ACM 38(3), 691–729 (1991)
14. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems (extended abstract). In: Sedgewick, R. (ed.) Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA. pp. 291–304. ACM (1985)
15. Hazay, C., Lindell, Y.: Efficient secure two-party protocols: Techniques and constructions. Springer Science & Business Media (2010)
16. Mahmoody, M., Moran, T., Vadhan, S.P.: Time-lock puzzles in the random oracle model. In: Rogaway, P. (ed.) Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6841, pp. 39–50. Springer (2011), http://dx.doi.org/10.1007/978-3-642-22792-9
17. Micali, S., Pass, R.: Local zero knowledge. In: Kleinberg, J.M. (ed.) Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006. pp. 306–315. ACM (2006)
18. M'Raïhi, D., Naccache, D.: Couponing scheme reduces computational power requirements for dss signatures. In: Proceedings of CardTech/SecurTech. pp. 99—-104 (1994)
19. Poupard, G., Stern, J.: Security analysis of a practical "on the fly" authentication and signature generation. In: Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding. pp. 422–436 (1998)
20. Rivest, R., Shamir, A., Wagner, D.: Time-lock puzzles and timed-release crypto (1996), technical report, MIT/LCS/TR-684
21. de Rooij, P.: On schnorr's preprocessing for digital signature schemes. Journal of Cryptology 10(1), 1–16 (1997)

A Proof of Lemma 2

Proof. The zero-knowledge property of the standard GPS protocol is proven by constructing a polynomial-time simulation of the communication between a prover and a verifier [11, Theorem 2]. We adapt this proof to the context of the proposed protocol. The function δ is defined by $\delta(\mathsf{true}) = 1$ and $\delta(\mathsf{false}) = 0$, and $\wedge$ denotes the logical operator "and". For clarity, the function $f_{\tau,\ell}$ is henceforth written f.

The scenario is that of a prover $\mathcal{P}$ and a dishonest verifier $\mathcal{A}$ who can use an adaptive strategy to bias the choice of the challenges to try to obtain information about s. In this case the challenges are no longer chosen at random, and this must be taken into account in the security proof. Assume the protocol is run N times and focus on the i-th round.

$\mathcal{A}$ has already obtained a certain amount of information η from past interactions with $\mathcal{P}$. $\mathcal{P}$ sends a pre-computed commitment x_i. Then $\mathcal{A}$ chooses a commitment using all information available to her, and a random tape ω: $c_i(x_i, \eta, \omega)$.

The following is an algorithm (using its own random tape ω_M) that simulates this round:

Step 1. Choose $\overline{c_i} \in_R [0, B-1]$ and $\overline{y_i} \in_R [(B-1)(S-1), A-1]$ using ω_M.
Step 2. Compute $\overline{x_i} = f_{\ell,\tau}\left(g^{\overline{y_i}} v^{\overline{c_i}}\right)$.
Step 3. If $c_i(\overline{x_i}, \eta, \omega) = \overline{c_i}$ then return to step 1 and try again with another pair $(\overline{c_i}, \overline{y_i})$, else return $(\overline{x_i}, \overline{c_i}, \overline{y_i})$.[3]

The rest of the proof shows that, provided $\Phi = (B-1)(S-1)$ is much smaller than A, this simulation algorithm outputs triples that are indistinguishable from real ones, for any fixed random tape ω.

Formally, we want to prove that

$$\Sigma_1 = \sum_{\alpha,\beta,\gamma} \left| \Pr_{\omega_P}\left[(x, c, y) = (\alpha, \beta, \gamma)\right] - \Pr_{\omega_M}\left[(\overline{x}, \overline{c}, \overline{y}) = (\alpha, \beta, \gamma)\right] \right|$$

is negligible, *i.e.* that the two distributions cannot be distinguished by accessing a polynomial number of triples (even using an infinite computational power). Let (α, β, γ) be a fixed triple, and assuming a honest prover, we have the following probability:

$$\begin{aligned}
p &= \Pr_{\omega_P}\left[(x, c, y) = (\alpha, \beta, \gamma)\right] \\
&= \Pr_{0 \le r < A}\left[\alpha = f(g^r) \wedge \beta = c(\alpha, \eta, \omega) \wedge \gamma = r + \beta s\right] \\
&= \sum_{r=0}^{A-1} \frac{1}{A} \delta\left(\alpha = f(g^\gamma v^\beta) \wedge \beta = c(\alpha, \eta, \omega) \wedge r = \gamma - \beta s\right) \\
&= \frac{1}{A} \delta\left(\alpha = f(g^\gamma v^\beta) \wedge \beta = c(\alpha, \eta, \omega) \wedge \gamma - \beta s \in [0, A-1]\right) \\
&= \frac{1}{A} \delta\left(\alpha = f(g^\gamma v^\beta)\right) \delta\left(\beta = c(\alpha, \eta, \omega)\right) \delta\left(\gamma - \beta s \in [0, A-1]\right).
\end{aligned}$$

where $f = f_{\ell,\tau}$.

We now consider the probability $\overline{p} = \Pr_{\omega_M}\left[(\overline{x}, \overline{c}, \overline{y}) = (\alpha, \beta, \gamma)\right]$ to obtain the triple (α, β, γ) during the simulation described above. This is a conditional probability given by

[3] The probability of success at step 3 is essentially $1/B$, and the expected number of executions of the loop is B, so that the simulation of N rounds runs in $O(NB)$: the machine runs in expected polynomial time.

$$\overline{p} = \Pr_{\substack{\overline{y} \in [\Phi, A-1] \\ \overline{c} \in [0, B-1]}} \left[\alpha = f\left(g^{\overline{y}} v^{\overline{c}}\right) \wedge \beta = \overline{c} \wedge \gamma = \overline{y} \;\middle|\; \overline{c} = c\left(f\left(g^{\overline{y}} v^{\overline{c}}\right), \eta, \omega\right)\right]$$

Using the definition of conditional probabilities, this equals

$$\overline{p} = \frac{\Pr_{\substack{\overline{y} \in [\Phi, A-1] \\ \overline{c} \in [0, B-1]}} \left[\alpha = f\left(g^{\overline{y}} v^{\overline{c}}\right) \wedge \beta = \overline{c} \wedge \gamma = \overline{y}\right]}{\Pr_{\substack{\overline{y} \in [\Phi, A-1] \\ \overline{c} \in [0, B-1]}} \left[\overline{c} = c\left(f\left(g^{\overline{y}} v^{\overline{c}}\right), \eta, \omega\right)\right]}$$

Let us introduce

$$Q = \sum_{\substack{\overline{y} \in [\Phi, A-1] \\ \overline{c} \in [0, B-1]}} \delta\left(\overline{c} = c\left(f\left(g^{\overline{y}} v^{\overline{c}}\right), \eta, \omega\right)\right)$$

then the denominator in $\overline{p}$ is simply $Q/B(A-\Phi)$. Therefore:

$$\begin{aligned}
\overline{p} &= \sum_{\overline{c} \in [0, B-1]} \frac{1}{B} \Pr_{\overline{y} \in [\Phi, A-1]} \left[\alpha = f\left(g^{\overline{y}} v^{\overline{c}}\right) \wedge \gamma = \overline{y} \wedge \beta = \overline{c} = c(\alpha, \eta, \omega)\right] \frac{B(A-\Phi)}{Q} \\
&= \Pr_{\overline{y} \in [\Phi, A-1]} \left[\alpha = f\left(g^{\gamma} v^{\beta}\right) \wedge \gamma = \overline{y} \wedge \beta = c(\alpha, \eta, \omega)\right] \frac{A-\Phi}{Q} \\
&= \sum_{\overline{y} \in [\Phi, A-1]} \frac{1}{A-\Phi} \delta\left(\alpha = f\left(g^{\gamma} v^{\beta}\right) \wedge \gamma = \overline{y} \wedge \beta = c(\alpha, \eta, \omega)\right) \frac{A-\Phi}{Q} \\
&= \frac{1}{Q} \delta\left(\alpha = f\left(g^{\gamma} v^{\beta}\right)\right) \delta\left(\beta = c(\alpha, \eta, \omega)\right) \delta\left(\gamma \in [\Phi, A-1]\right)
\end{aligned}$$

We will now use the following combinatorial lemma:

Lemma 4. *If $h : \mathcal{G} \to [0, B-1]$ and $v \in \{g^{-s}, s \in [0, S-1]\}$ then the total number M of solutions $(c, y) \in [0, B-1] \times [\Phi, A-1]$ to the equation $c = h(g^y v^c)$ satisfies $A - 2\Phi \le M \le A$.*

Proof (Proof of Lemma 4). [11, Appendix A]

Specialising Lemma 4 to the function that computes $c(f(g^{\overline{y}} v^{\overline{c}}), \eta, \omega)$ from $(\overline{c}, \overline{y})$ gives $A - 2\Phi \le Q \le A$. This enables us to bound Σ_1:

$$\begin{aligned}
\Sigma_1 &= \sum_{\alpha,\beta,\gamma} \left| \Pr_{\omega_P}[(x,c,y)=(\alpha,\beta,\gamma)] - \Pr_{\omega_M}[(\overline{x},\overline{c},\overline{y})=(\alpha,\beta,\gamma)] \right| \\
&= \sum_{\alpha,\beta,\gamma\in[\Phi,A-1]} \left| \Pr_{\omega_P}[(x,c,y)=(\alpha,\beta,\gamma)] - \Pr_{\omega_M}[(\overline{x},\overline{c},\overline{y})=(\alpha,\beta,\gamma)] \right| \\
&\quad + \sum_{\alpha,\beta,\gamma\notin[\Phi,A-1]} \Pr_{\omega_P}[(x,c,y)=(\alpha,\beta,\gamma)] \\
&= \sum_{\substack{\gamma\in[\Phi,A-1]\\ \beta\in[0,B-1]\\ \alpha=f(g^\gamma v^\beta)}} \left| \frac{1}{A}\delta\left(\beta=c(\alpha,\eta,\omega)\right) - \frac{1}{Q}\delta(\beta=c(\alpha,\eta,\omega)) \right| \\
&\quad + \left(1 - \sum_{\alpha,\beta,\gamma\in[\Phi,A-1]} \Pr_{\omega_P}[(x,c,y)=(\alpha,\beta,\gamma)] \right) \\
&= \left| \frac{1}{A} - \frac{1}{Q} \right| Q + 1 - \sum_{\substack{\gamma\in[\Phi,A-1]\\ \beta\in[0,B-1]\\ \alpha=f(g^\gamma v^\beta)}} \frac{1}{A}\delta\left(\beta=c(\alpha,\eta,\omega)\right) \\
&= \frac{|Q-A|}{A} + 1 - \frac{Q}{A}
\end{aligned}$$

Therefore $\Sigma_1 \leq 2|Q-A|/A \leq 4\Phi/A < 4SB/A$, which proves that the real and simulated distributions are statistically indistinguishable if SB/A is negligible.

□