Proof of Latency Using a Verifiable Delay Function

Type of thesis
University of Turku
Department of Future Technologies
Department of Future Technologies
2020
Jani Anttonen

TURUN YLIOPISTO
Informaatioteknologian laitos

JANI ANTTONEN: Proof of Latency Using a Verifiable Delay Function

M.Sc., 8 s., 0 liites.

Kuukausi 2020

Tarkempia ohjeita tiivistelmäsivun laadintaan läytyy opiskelijan yleisoppaasta, josta alla lyhyt katkelma.
Bibliografisten tietojen jälkeen kirjoitetaan varsinainen tiivistelmä. Sen on oletettava, että lukijalla on yleiset tiedot aiheesta. Tiivistelmän tulee olla ymmärrettävissä ilman tarvetta perehtyä koko tutkielmaan. Se on kirjoitettava täydellisinä virkkeinä, väliotsakeluettelona. On käytettävä vakiintuneita termejä. Viittauksia ja lainauksia tiivistelmään ei saa sisällyttää, eikä myäskään tietoja tai väitteitä, jotka eivät sisälly itse tutkimukseen. Tiivistelmän on oltava mahdollisimman ytimekäs n. 120 – 250 sanan pituinen itsenäinen kokonaisuus, joka mahtuu ykkäsvälillä kirjoitettuna vaivatta tiivistelmäsivulle. Tiivistelmässä tulisi ilmetä mm. tutkielman aihe tutkimuksen kohde, populaatio, alue ja tarkoitus käytetyt tutkimusmenetelmät (mikäli tutkimus on luonteeltaan teoreettinen ja tiettyyn kirjalliseen materiaaliin, on mainittava tärkeimmät lähdeteokset; mikäli on luonteeltaan empiirinen, on mainittava käytetyt metodit) keskeiset tutkimustulokset tulosten perusteella tehdyt päätelmät ja toimenpidesuositukset asiasanat

Asiasanat: tähän, lista, avainsanoista

Contents

Chapter 1

Introduction

1.1 Background and Motivation

Distributed systems are vulnerable to problems related to data integrity, because in many cases there is no single source of truth. Work pioneered by the likes of Leslie Lamport has tried to mitigate these problems with clock synchronization and consensus algorithms, which have fixed some but not all of the underlying issues related to the field.

Proof of Work is the most well-known consensus algorithm, as it is used in most public blockchains today, including Bitcoin, of in which whitepaper it was first described in 2008. New algorithms have been introduced since to battle it's resource intensiveness, including Proof of Stake, which requires network nodes participating in the voting of new blocks to stake a part of their assets as a pawn. If the voters get labeled as malicious by a certain majority, they can get slashed, losing all or a part of the staked asset in the process. This serves as an incentive for honest co-operation, with sufficient computation resources.

Problem is that the block generation votes are not done globally, but by a selected group of peers called the validators, which vote for the contents of proposed blocks by one peer selected as the block generator. The validators are selected randomly. This has generated an increasing demand for verifiable public randomness, that is pre-image resistant, meaning the output of the algorithm generating the randomness cannot be influenced beforehand. This created a motivation for an algorithm that would prevent multiple malicious actors from being selected to vote at once, effectively creating a more secure public blockchain.

In 2018, two research papers were released independently with similar formalizations of a VDF.[1][2] By definition, a VDF is an algorithm that requires a specified number of sequential steps to evaluate, but produces a unique output that can be efficiently and publicly verified.[3] To achieve pre-image resistance, a VDF is sequential in nature, and cannot be sped up by parallel processing. There are multiple formulations of a VDF, and not all even have a generated proof, instead using parallel processing with graphics processors to check that the calculation is sequential.[4] This bars less powerful devices, like embedded devices, from verifying the result efficiently. Thus, generating an efficient proof that requires little time to verify is more ideal.[3]

Using verifiable delay functions, I propose a novel algorithm for producing a publicly verifiable proof of network latency and computation resource difference between two participants in a peer-to-peer network. This proof can be used for dynamic routing in peer-to-peer networks to reduce latency between peers, and for making sybil attacks harder to achieve.

Chapter 2

Cryptography

2.1 RSA

2.2 Asymmetric Cryptography

2.2.1 Diffie-Hellman Key Exchange

2.3 Elliptic Curve Cryptography

Chapter 3

Verifiable Delay Functions

3.1 Variations

3.2 Similar Constructs

A VDF can only be calculated sequentially, but even without a proof there's a possibility to make the verification faster through parallellism. [4]

Chapter 4

Proof of Latency

Proof of Latency is a algorithm that uses an asynchronically stoppable verifiable delay function as its basis. The calculated amount of squarings until the VDF is stopped serves as a publicly verifiable measurement metric for latency between two peers.

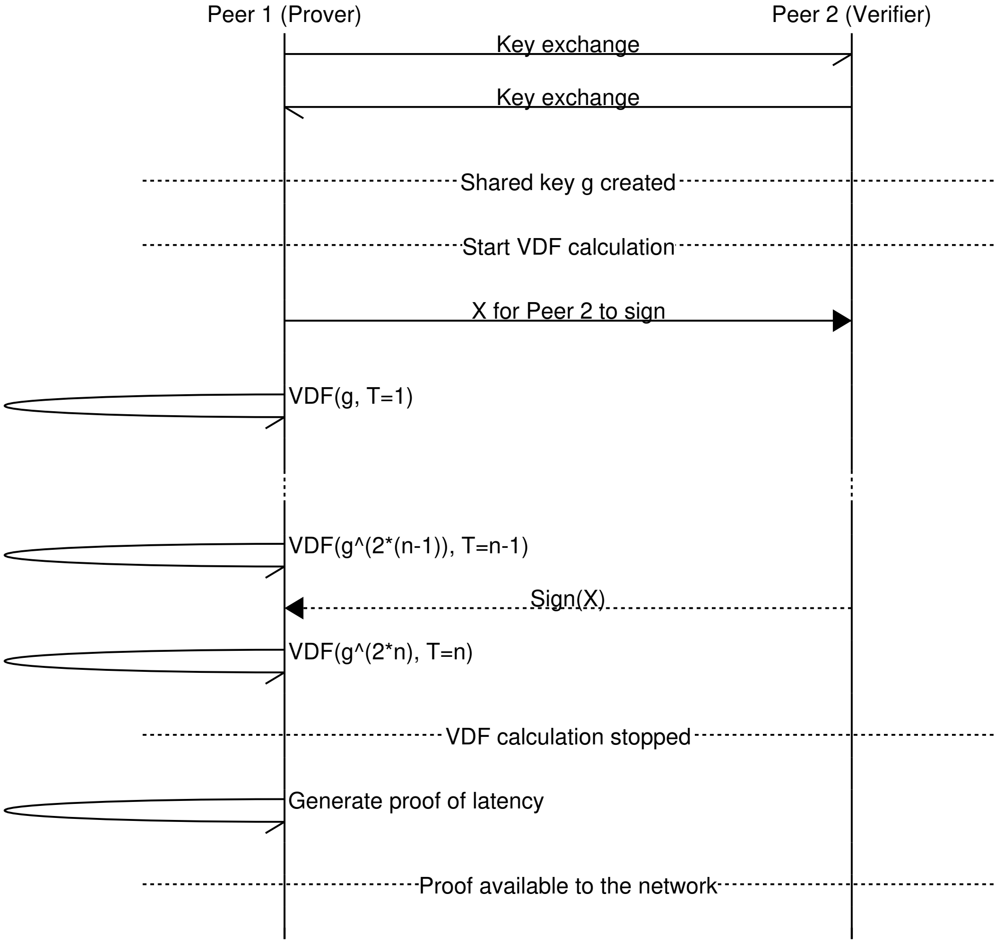

Figure 4.1: Protocol diagram of Proof of Latency

Chapter 5

Results

References

[1] Benjamin Wesolowski. Efficient verifiable delay functions. Technical Report 623, 2018.

[2] Krzysztof Pietrzak. Simple Verifiable Delay Functions. Technical Report 627, 2018.

[3] Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. Verifiable Delay Functions. Technical Report 601, 2018.

[4] Anatoly Yakovenko. Solana: A new architecture for a high performance blockchain. page 32.