

# MPC for Group Reconstruction Circuits

Lúcás Críostóir Meier

June 8, 2022

## Abstract

In this paper, we present a thing.

## 1 Introduction

Write the introduction

## 2 Background

Throughout this paper, we let  $\mathbb{G}$  denote a group of prime order  $q$ , with generators  $G$  and  $H$ . Let  $\mathbb{F}_q$  denote the scalar field associated with this group, and let  $\mathbb{Z}/(q)$  denote the additive group of elements in this field.

We make heavy use of group homomorphisms throughout this paper. We let

$$\varphi(P_1, \dots, P_m) : \mathbb{A} \rightarrow \mathbb{B}$$

denote a homomorphism from  $\mathbb{A}$  to  $\mathbb{B}$ , parameterized by some public values  $P_1, \dots, P_m$ . Commonly  $\mathbb{A}$  will be a product of several groups  $\mathbb{G}_1, \dots, \mathbb{G}_n$ , in which case we'd write:

$$\varphi(P_1, \dots, P_m)(x_1, \dots, x_n)$$

to denote the application of  $\varphi$  to an element  $(x_1, \dots, x_n)$  of the product group. Such an element is sometimes treated as a vector  $\mathbf{x}$ , in which case we write  $\varphi(P_1, \dots, P_m)(\mathbf{x})$ . We also often leave the public values  $P_i$  implicit.

### 2.1 Pedersen Commitments

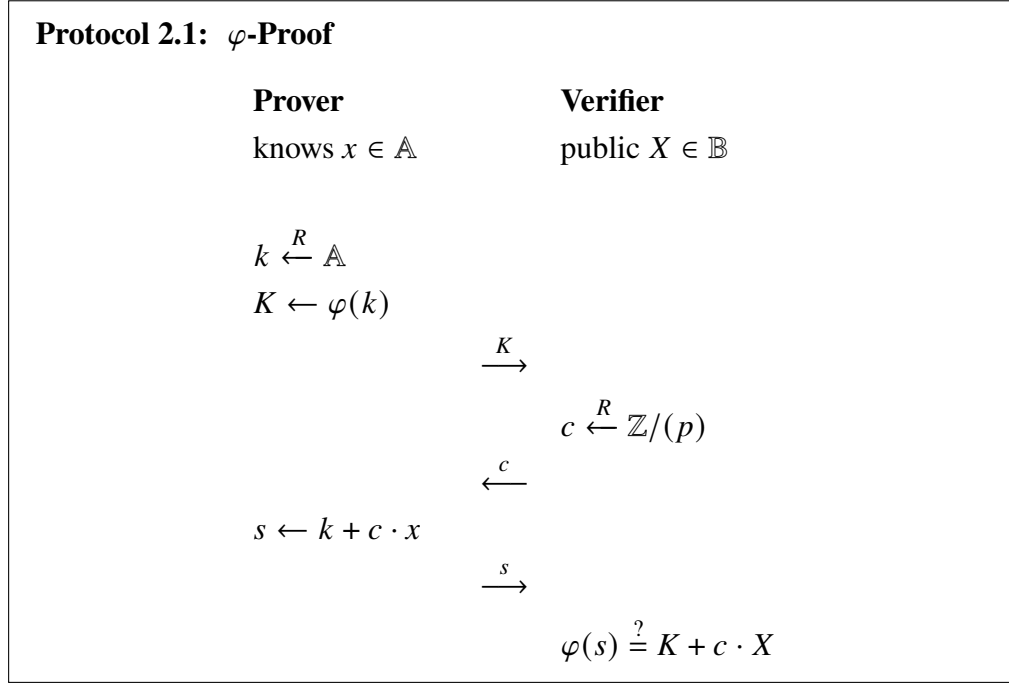
### 2.2 Sigma Protocols

### 2.3 Maurer's $\varphi$ -Proof

In [Mau09], Maurer generalized Schnorr's sigma protocol for knowledge of the discrete logarithm [Sch90] to a much larger class of relations. In particular, Maurer provided a sigma protocol for proving knowledge of the pre-image of a group

homomorphism  $\varphi$ . We denote this protocol as a “ $\varphi$ -proof”, and recapitulate the scheme here.

Given a homomorphism  $\varphi : \mathbb{A} \rightarrow \mathbb{B}$ , and a public value  $X \in \mathbb{B}$ , the prover wants to demonstrate knowledge of a private value  $x \in \mathbb{A}$  such that  $\varphi(x) = X$ . The prover does this by means of Protocol 2.1:



Here,  $p$  is chosen such that  $\forall B \in \mathbb{B}. p \cdot B = 0$ . In practice, we’ll set  $p = q$ , which will work perfectly for the groups we use, which are all products of  $\mathbb{G}$  or  $\mathbb{Z}/(q)$ .

**Claim 2.1.** Protocol 2.1 is a valid sigma protocol.

Completeness follows directly from the fact that  $\varphi$  is a homomorphism.

For the HVZK property, the simulator  $\mathcal{S}(X, c)$  works by generating a random  $s \xleftarrow{R} \mathbb{A}$ , and then setting  $K := \varphi(s) - c \cdot X$ .

Finally, we prove 2-extractability. Given two verifying transcripts  $(K, c, s)$  and  $(K, c', s')$  sharing the first message, we extract a value  $\hat{x}$  satisfying  $\varphi(\hat{x}) = X$  as follows:

$$\begin{aligned}
\varphi(s) - c \cdot X &= K = \varphi(s') - c' \cdot X \\
\varphi(s) - \varphi(s') &= c \cdot X - c' \cdot X \\
\frac{1}{c - c'} \cdot \varphi(s - s') &= X \\
\varphi\left(\frac{s - s'}{c - c'}\right) &= X
\end{aligned}$$

Thus, defining  $\hat{x} := (s - s') / (c - c')$ , we successfully extract a valid pre-image.

We conclude that the protocol is a valid sigma protocol.

■

Maurer's protocol can also work even in the case where the order of the groups are not known, but this makes the challenge generation a bit more complicated, and we don't need this functionality in this work.

## 2.4 UC Security and the Hybrid Model

## 2.5 Ideal Functionalities for Sigma Protocols

### Functionality 2.1: Zero-Knowledge Functionality $\mathcal{F}(\text{ZK}, \varphi)$

A functionality  $\mathcal{F}$  for parties  $P_1, \dots, P_n$ .

On input (prove, sid,  $x$ ) from  $P_i$ :

$\mathcal{F}$  checks that sid has not been used by  $P_i$  before.

$\mathcal{F}$  generates a new token  $\pi$ , and sets  $x_\pi \leftarrow x$ .

$\mathcal{F}$  replies with (proof,  $\pi$ ).

On input (verify,  $X, \pi$ ):

$\mathcal{F}$  replies with (verify-result,  $\varphi(x_\pi) \stackrel{?}{=} X$ ).

## 2.6 Broadcast Functionalities

### Functionality 2.2: Authenticated Broadcast Functionality $C$

A functionality  $C$  for parties  $P_1, \dots, P_n$ .

On receiving (broadcast-in,  $\text{sid}, m$ ) from  $P_i$ :  
 $C$  checks that  $\text{sid}$  has not been used by  $P_i$  before.  
 $C$  sends (broadcast-out,  $\text{pid}_i, \text{sid}, m$ ) to every party  $P_j$ .

## 3 Group Reconstruction Circuits

### 3.1 Formal Definition

### 3.2 Normalized Form

## 4 MPC Protocol for GRCs

### 4.1 Ideal Functionality

#### Functionality 4.1: GRC functionality $\mathcal{F}(\text{GRC}, \Phi, \mathbf{X}^j, \mathbf{Y}^j)$

A functionality  $\mathcal{F}$  for parties  $P_1, \dots, P_n$ .

After receiving (input,  $\text{sid}, \mathbf{x}^j, \mathbf{y}^j, \alpha^j, \mathbf{k}^j, \beta^j$ ) from every party  $P_j$ :  
 $\mathcal{F}$  checks, for every  $j \in [n]$ , that:

$$\mathbf{X}^j \stackrel{?}{=} \mathbf{x}^j \cdot G$$

$$\mathbf{Y}^j \stackrel{?}{=} \mathbf{y}^j \cdot G + \alpha^j \cdot H$$

$\mathcal{F}$  computes, for each round  $r \in [d]$ :

$$\mathbf{V}_r^j := \varphi_r(\mathbf{V}_1, \dots, \mathbf{V}_{r-1})(\mathbf{x}^j, \mathbf{y}^j, \mathbf{k}^j)$$

$$\mathbf{V}_r := \sum_j \mathbf{V}_r^j$$

$\mathcal{F}$  sends (output,  $\text{sid}, \mathbf{V}_1^1, \dots, \mathbf{V}_d^n$ ) to every party  $P_j$ .

## **4.2 Protocol**

## **4.3 Security Analysis**

## **4.4 Practical Considerations**

# **5 Applications**

# **6 Limitations and Further Work**

# **7 Conclusion**

## **References**

- [Mau09] Ueli Maurer. Unifying Zero-Knowledge Proofs of Knowledge. In *AFRICACRYPT 2009*, volume 5580 of *LNCS*, pages 272–286. Springer, Berlin, Heidelberg, 2009.
- [Sch90] C. P. Schnorr. Efficient Identification and Signatures for Smart Cards. In *CRYPTO 1989*, volume 435 of *LNCS*, pages 239–252, New York, NY, 1990. Springer.