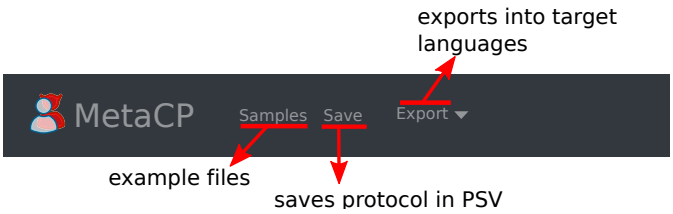
Instructions For MetaCP

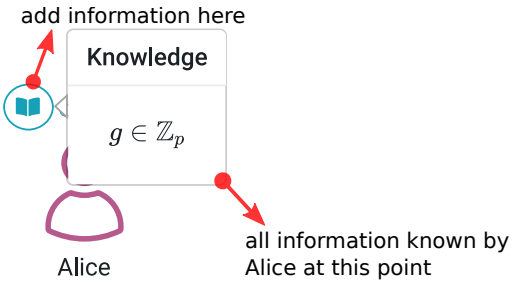
1 Getting Started

Welcome to MetaCP! As you access the tool you will be greated by a interface that should be quite familiar to most protocol designers. It is composed of:

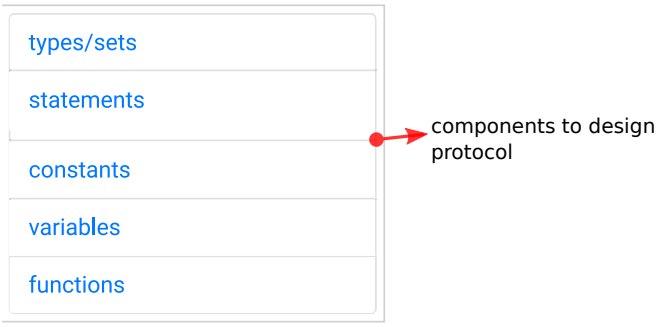
a top menu to help the navigation through the tool



two parties, Alice and Bob each with their knowledge



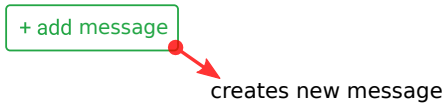
a draggable toolbox, with all the different elements needed to draw the protocol



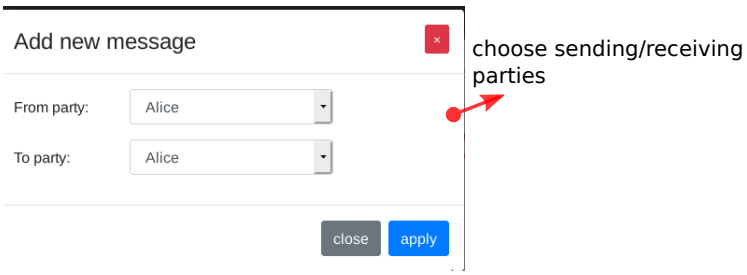
the ability to begin writing the first message of the protocol.

2 Adding New Messages

This feature is achieved by selecting the *add first message* button (and consequently the *+ add message* button.



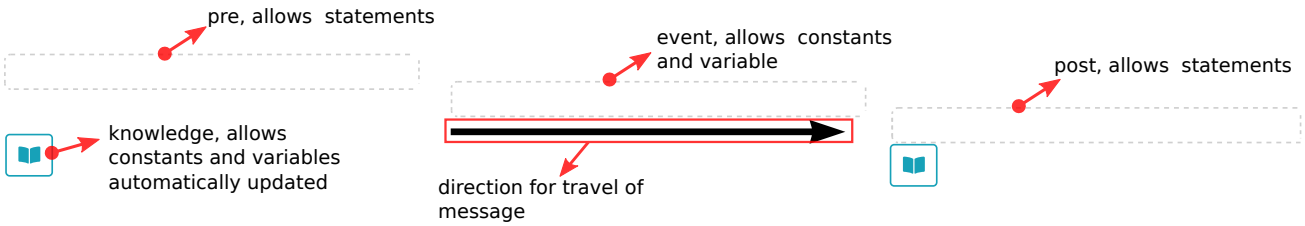
upon clicking the buttons a menu will appear



- you will need to select a source and destination for the message[1]
- once the message is applied a message will appear.

2.1 Message Structure

A message will have the form of a $[\text{pre}]\xrightarrow{[event]}$ / $\xleftarrow{[event]}[\text{post}]$.



- Each of these components is represented by a droppable area.
- The user will now be able to drag components from the toolbox into the droppable area.
- Each area has restrictions as to what toolbox components can be dropped within it.
- This ensures correctness of the protocol design.

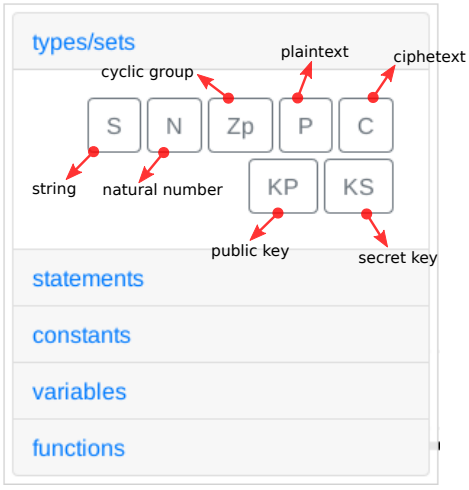
[1] messages with same source and destintion are currently still possible

3 ToolBox

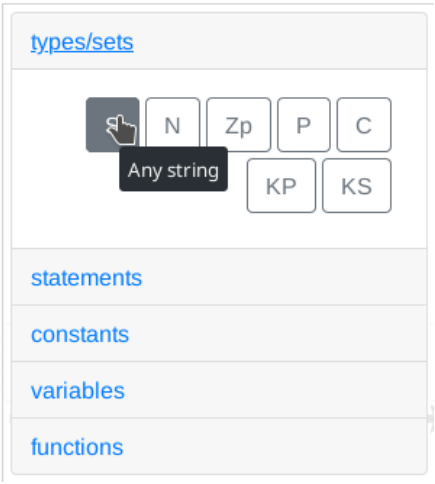
The toolbox contains all the elements needed to design your protocol. Each elements is a draggable component that can be positioned within the messages, or knowledge.

3.1 Types/Sets

The currently supported types are, string, natural numbers, cyclic group, plaintext, ciphertext, public key and secret key.



We show the available types graphically above, but if you hover over each button a hint bubble will appear with the definition.

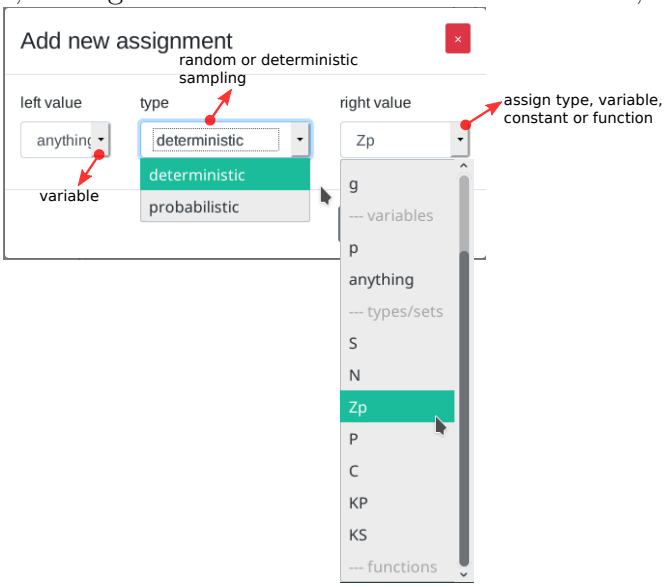


The types are used in the statements to ensure correctness of input, the type of the variable/constant must match the expected input of the statement.

3.2 Statements

Statements are used to construct components of the protocol. To add one simply click the *+ assignment* button.

They are composed of a left side, of type variable; a middle, to choose the type of assignment; and, the right side which can be another variable, a typeset, a constant or a function.



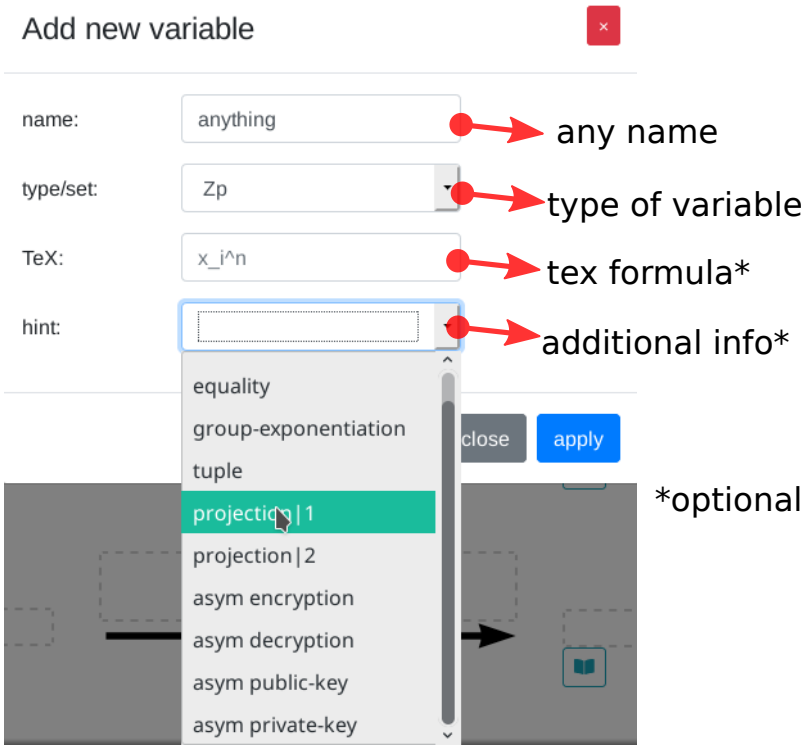
4 Constants

As of now we only provide constants, Alice, Bob and g. More can be used in the PSV.

5 Variables

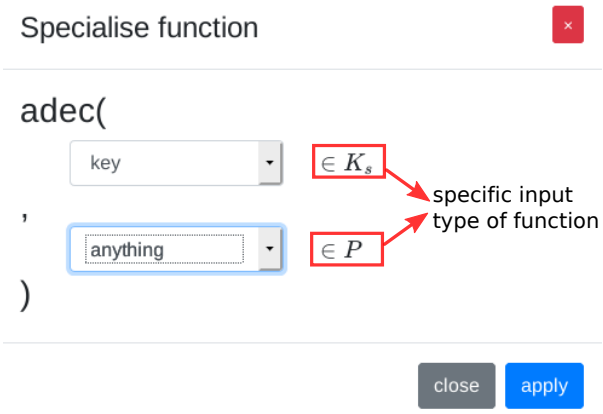
A variable is the main component of the protocol. Upon specifying a variable you need to select it's type. Two further optional components are the **TeX**, to influence how it will display on screen; and the **hint**, these are tips useful for the plugins to interpret these variables in a specific way, they are used to provide further information about the variable[2]. If they are not recognized by a plugin they will simply be ignored and can be helpful to cunstruct further plugins.

[2]whilst not currently available, anything can be specified within it



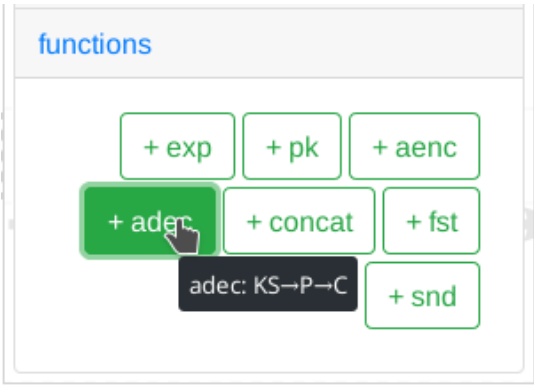
6 Functions

Currently the GDE suppoers all the required functions to replicate the sample protocols[3]. A function will take the form of $X \to Y \to Z$ where X,Y are of the types of the inputs and Z is the type of the output.



[3]Further functions will be added shortly

this is also shown if the function is hovered.

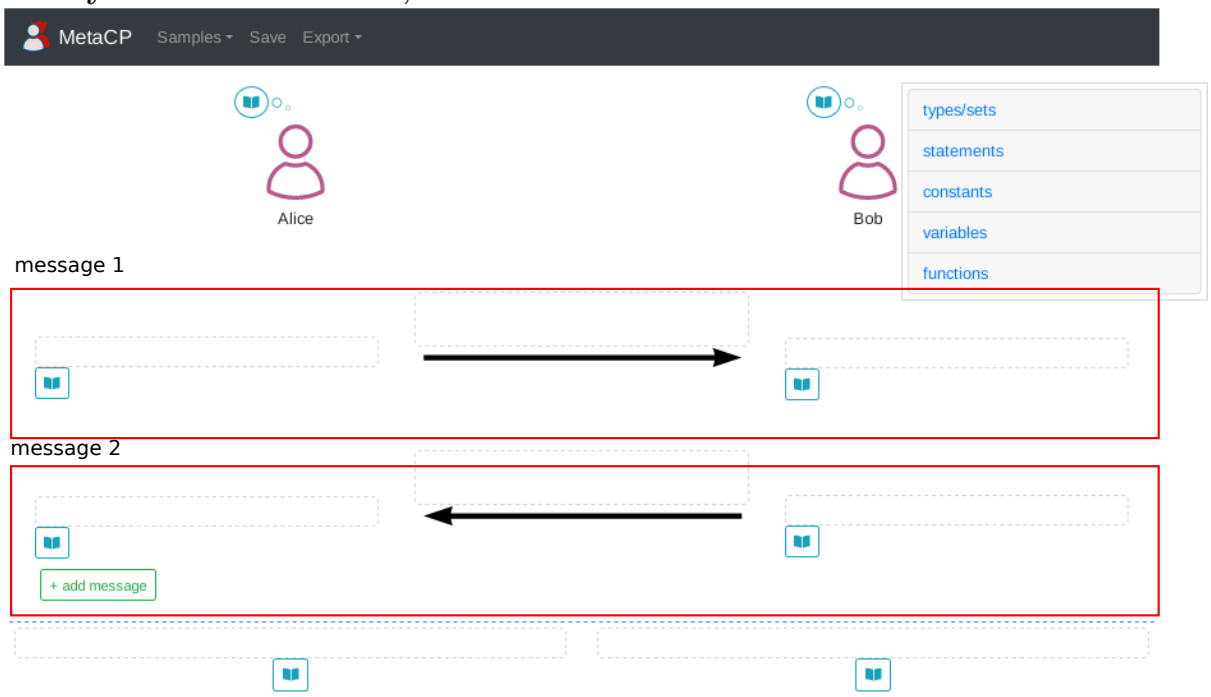


7 Example Protocol: Diffie-Hellman

We will now show how a simple protocol can be specified using the GDE. The steps can be performed in a different order if preferred.

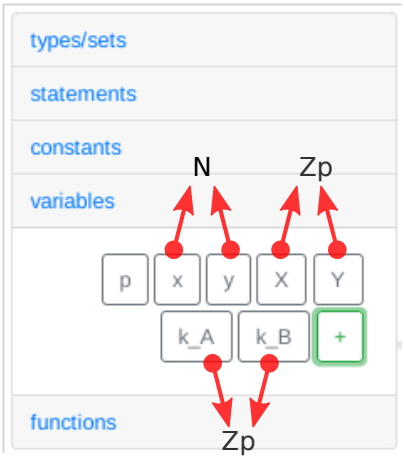
7.1 Step 1: Add messages

Firstly from Alice to Bob, and then the reverse.



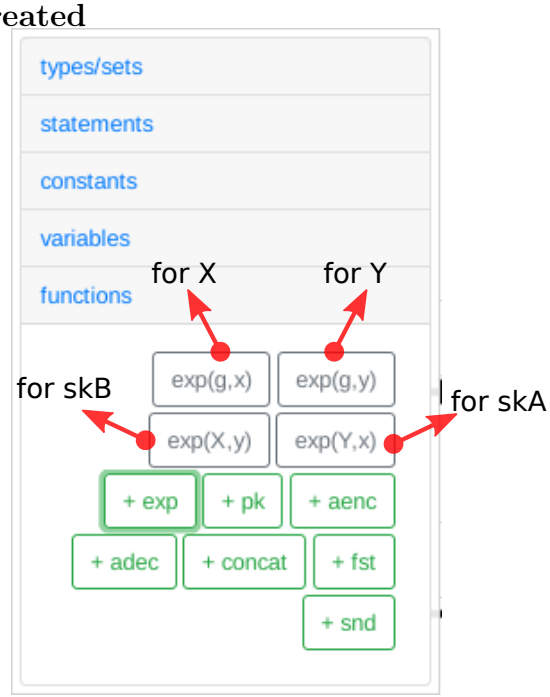
7.2 Step 2: Declare Variables

The keys and the shared values in the Diffie-Hellman key exchange are outcomes of modulus exponentiation so they are of type $\mathbb{Z}_p$.



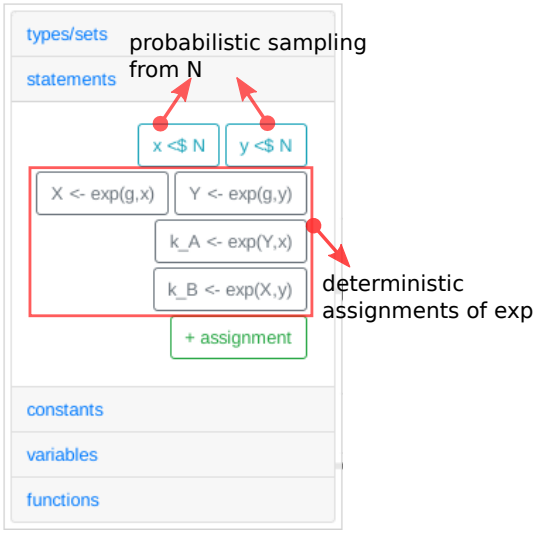
7.3 Step 3: Declare Functions

Before the protocol statements are constructed the exponentiation functions must be created

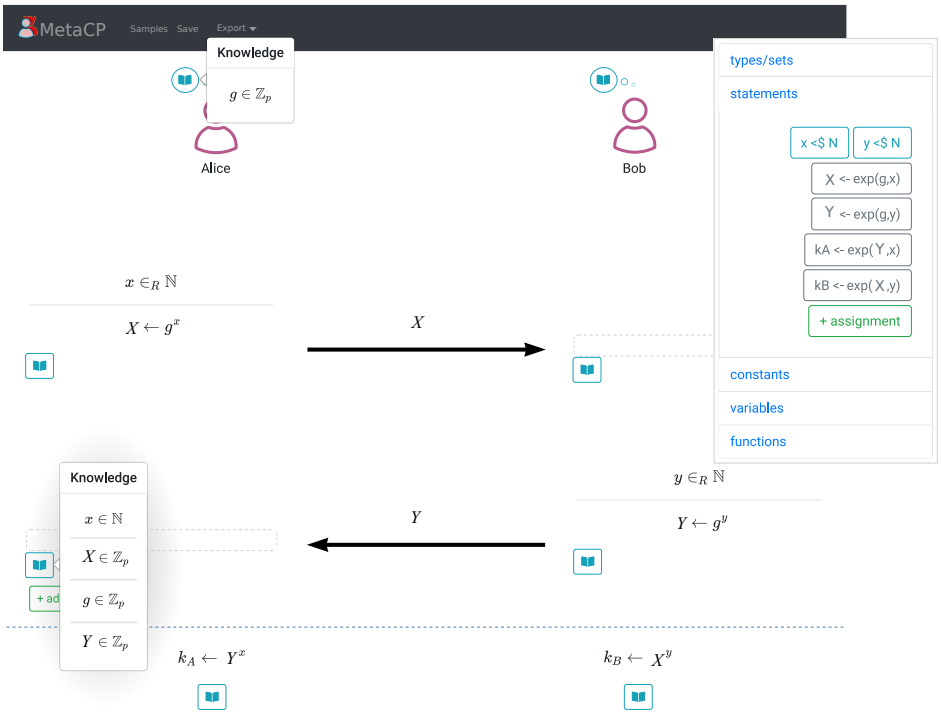


7.4 Step 4: Declare Statements

The last tools needed are the statements and assignment that constitute the messages



7.5 Step 5: Drag them in!



You may now save or export your protocol!