1 Introduction

There are only 2 sentences in this whole paragraph. This makes it very difficult to read. Try to avoid long sentences by splitting them into 2 or more.

In this project, I aim to introduce the technique of obfuscation as a tool to secure the privacy of users of the biological modelling languages SBML and Bio-PEPA to keep intellectual ownership of their models by obfuscating them in a way which does not change the meaning of the model to a computer but makes it much more difficult for a human to interpret it. These biological modelling languages represent a "biological model" which may be a mathematical model of a biological system or, "it refers to a specific organism which may be studied extensively with the goal of generating data which can be applied to other organisms" [1].

The long quotation here is unnecessary as it would be very easy to re-word the concept. This is especially bad as it is taken from a website!

Generally, a newly discovered biological mechanism could be the basis of a highly desirable scientific publication. Hence these biological models and biological parameter data are closely guarded by their owners which make it very difficult for modellers to trust networked services to analyse their models and data. Though this explains the reason for privacy, the questions arise are "why obfuscation", "why not encryption", etc. General bad formatting here.

In case of encryption we encrypt the data and send it to the server for results. Here data is secured until it reaches the server. In the server, this data and the model should be decrypted and then gets processed to obtain results. Here data needs to be decrypted i.e. the original representation of data is exposed. Encryption does not eliminate the need to protect some 'secret data' (like keys).

When explaining a term, don't use the same word to explain it! What if the reader is not familiar with the word 'obfuscate'?

'Obfuscate' used 3 times in same sentence. Avoid repetition like this.

In case of obfuscation we obfuscate data at the client side in such a way that the obfuscated data has the minimum possible information that is necessary for server to get results. Though the minimum possible information contains biological parameter data in the form of biological models, but makes no sense without a de-obfuscated model thereby the results from the server which are deduced from this obfuscated data also makes no sense or meaning to anyone.

More repetition!

2 Background

A short introduction to what will be presented in the 'Background' section should be given here. Instead the student immediately starts to describe SBML

2.1 SBML (Systems Biology Markup Language) [2] is an XML-based language.

Normally better to do this the other way around i.e. 'Systems Biology Markup Language (SBML)'

Computational modelling of data has been a requirement for biological systems. Since modelling is practiced by all the scientists, the next step is to formulate it into computable form that can be analysed using simulation. This requirement led to the emergence of SBML.

SBML is capable of representing systems consisting of biological entities in machine-readable format. It makes models encoded using XML. Many software tools use SBML as format for communicating and storing models.

Since this is a modelling language it has an abstract datatype called SBase which is capable of handling information of annotation.

We can skip this annotation part in parsing SBML because of the following reasons:

This whole section is unclear without an example.

- We want to reduce models to their minimum possible content.
- It is just used for representation.
- In SBML, id and name are not defined on SBase because of restrictions in definition of abstract data type SBase(absence of default scoping rule) and it is unnecessary to define an Id or name.

An SBML model has the following attributes: [2]

Function definition: A named mathematical function that may be used throughout the rest of a model.
Unit definition: A name for a unit used in the expression of quantities in a model.

Compartment: A container of finite size for substances.

Species: A substance or entity that takes part in a reaction.

Parameter: A quantity with a symbolic name.

Rule: A mathematical expression

Reaction: A statement describing some transformation, transport or binding process that can change the amount of one or more species.

Event: A statement describing an instantaneous, discontinuous change in a set of variables of any type (species concentration, compartment size or parameter value) when a triggering condition is satisfied.

2.2 How the attributes are interpreted:

• The identifier includes FunctionDefinition, CompartmentType, SpeciesType, Compartment, Species, Parameter, Reaction, SpeciesReference, ModifierSpeciesReference, Event,and Model.

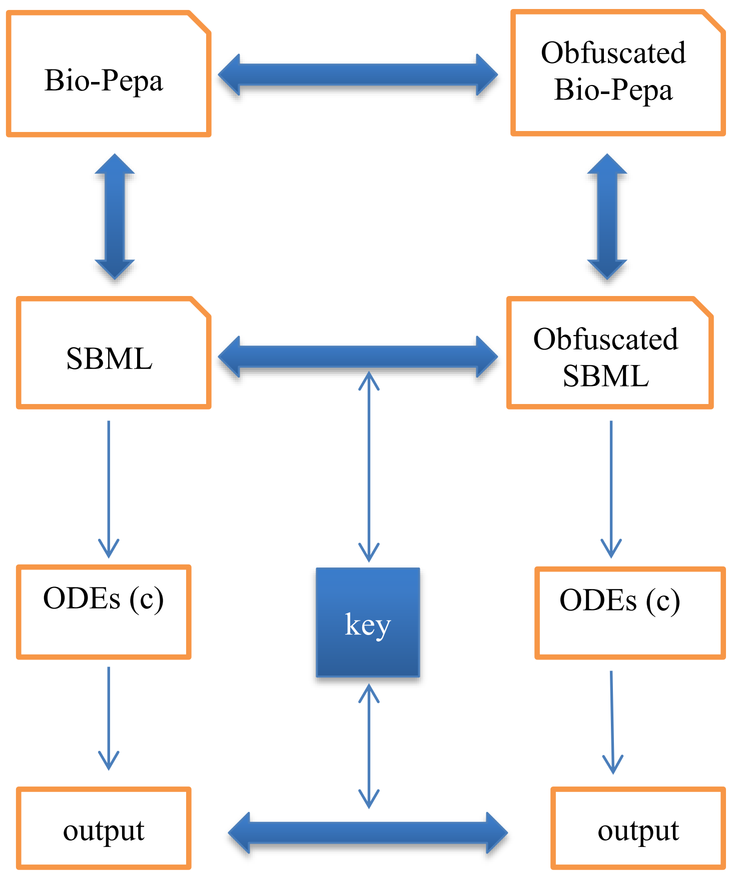
• The identifier of every UnitDefinition

• Reaction: Each reaction instance has its own local Parameter and identifiers.
The KineticLaw, StoichiometryMath, EventAssignment and Rule structures

This page is just lists without explanations. The student has simply copied this from somewhere and has not shown any understanding of what it all means.
Lists/bullet-points can be a great tools for presenting information but you must take care to explain the items.

3 Method:

3.1 Sample architecture of the proposed model:



This diagram is not explained at all.

If you do not explain diagrams it will look like you are only trying to fill space.

Figure 1: Sample Architecture...

Always label figures and add descriptions of what they are.

Explanation of the proposed model in detail:

1 sentence is not 'in detail'!

Obfuscated SBML: After the applying obfuscation we get the output as obfuscated SBML. In this obfuscated SBML, terms like names of the species, element names in the reactions are obfuscated according to the requirements.

3.2 Requirements of the obfuscation function:

The level of obfuscation in obfuscation function should be efficient by satisfying the basic requirement like:

Prevent clashes with reserved words:

Generally, SBML models are often compiled to C/C++ source code. If there is any obfuscated word that matches the reserved word in that language, it may lead to the compilation error of logical error which is very undesirable.

It should be capable of producing outputs (obfuscated terms) that should have the qualities like

- Non-obvious

The output cannot be easily predicted or one output should not resemble with other outputs.

- Randomness

  One output pattern should not resemble with other outputs.
- Short

  Obfuscated term cannot be too large.
- Usable as identifiers in many programming languages.

  Since SBML can be used in various domains, it should be able to designate as an identifier in most of programming languages.

Consider a reaction: [3]

$$E + S \underset{k_{\text{off}}}{\overset{k_{\text{on}}}{\rightleftharpoons}} ES \xrightarrow{k_{\text{cat}}} E + P$$

Avoid bad copy and paste like this. It is better to rewrite the function using Latex or an equivalent function of your word processor.
All equations should be numbered.

This is not a good explanation and presentation is careless. The student has not even used the correct symbols or subscripting. Very lazy!

If we look at the reaction: Reactant elements are E, S and ES along with addition of Kon, Koff, Kcat produces products as E, P.

The following table shows an abstract view of what an actual SBML code look like on the left-hand side and the expected Obfuscated SBML code on right-hand side:

SBML code	Obfuscated SBML code
<kineticLaw>…	<kineticLaw>…
<times />	<times />
<ci> kon </ci>	<ci> Z42HB </ci>
<ci> E </ci>	<ci> X0ZFP </ci>
<ci> S </ci>	<ci> Y3GHQ </ci>
</apply>	</apply>
<apply>	<apply>
<times />	<times />
<ci> koff </ci>	<ci>P5NJE </ci>
<ci> ES </ci>	<ci>A3KJH </ci>
</apply>	</apply>
……..	……..
</kineticLaw>	</kineticLaw>
<kineticLaw> …	<kineticLaw> …
<apply>	<apply>
<times />	<times />
<ci> kcat </ci>	<ci> J7ITU </ci>
<ci> ES </ci>	<ci> A3KJH </ci>
</apply>	</apply>
…	…
</kineticLaw>	</kineticLaw>

Figure 2: ....

In the obfuscated SBML code the element names are obfuscated which will be done according to the rules in obfuscation function. The above example can be classified in to two reactions, where in code it has designated as two kinetic laws. In the first kinetic law we have ES as one of the attribute and it is obfuscated. In the second kinetic law we again have ES as reactant (or) attribute but the obfuscated name at this part is same the first part.

Key: After this obfuscation the hash table containing the original with corresponding obfuscated words is maintained at the owner (user) acts as a key which is used to de-obfuscate.

Here in this example hash table (key), which maps actual term to obfuscated term would look like as follows

Actual term	Obfuscated term
Kon	Z42HB
E	X0ZFP
S	Y3GHQ
Koff	P5NJE
ES	A3KJH
Kcat	J7ITU

You normally do not need to explain well established mathematical functions and equations such as ODEs. This gives the impression that the student was trying to 'fill space'. If you are not sure what content to include then ask me or your supervisor.

ODEs (Ordinary Differential Equations): An **ordinary differential equation** is an equation which expresses a relationship between the derivatives of an unknown function, the independent variable, and the unknown function itself [5].

An example **first order** differential equation is (*P(t)* is an unknown function)

$$\frac{d}{dt}P(t) = kP(t)$$

Where k=constant.

In this project, the intermediate component ODEs (c) represents that Ordinary Differential Equations in C/C++ language (to which SBML models are compiled).

3.3 Where this obfuscation function is implemented

We might assume that JSBML is a Java implementation of SBML, but the student should explicitly say so. It is always better to explain acronyms and initialisms.

In JSBML library, the SBML code is interpreted as abstract syntax tree nodes. The JSBML ASTNode provides methods that are needed to transform these abstract syntax trees to other formats.

The ASTNodeCompiler **class [6]** This section has bad formatting and very disjointed prose.

This increases the range of implementation by providing the capability to customize the interpreters that encode mathematical equations. Here ASTNode class returns ASTNodeValue object by calling in a recursive way. JSBML provides many implementations like one of the ASTNode object can be translated to LATEX or Strings etc.

4 Hardware and Software Requirements

Software: Java, eclipse IDE.

Always use terminology in a consistent way i.e. use capital letters or not.

Jsbml Library: The JSBML provide an SBML parser and programming library that maps all SBML elements to a flexible and extended type hierarchy.

LibSBML Library: LibSBML is an open-source programming library to help you read, write, manipulate, translate, and validate SBML files and data streams [4].

Hardware: Standard PC.

5 Evaluation

This section is very important. You should be able to say something about how you will test if your project is successful or not. How will you test if it works? How will you measure performance? You should also consider a section of 'Potential Setbacks/Issues' to preempt what might go wrong. There is not enough here and it is too vague.

The goal of this project is to implement such a system in Java for both SBML and Bio-PEPA. Further, translations exist between SBML and Bio-PEPA which allow models in one language to be converted into equivalent models in the other language. These translations should continue to function in the presence of obfuscation so that, for example, an SBML model could be obfuscated, then translated into Bio-PEPA, then analysed, then have the results de-obfuscated and applied to the SBML model.

6 Work Plan

The final section of this proposal identifies the anticipated workflow of my project along with expected completion dates. Each phase will have some overlap from other phases. For example, implement and improvement of retrieval efficiency are closely dependent because changes in latter involves changes in previous phase. This can be seen in the summary table directly below or in the more comprehensive Gantt chart listed below that.

Fully justified text looks bad inside tables.

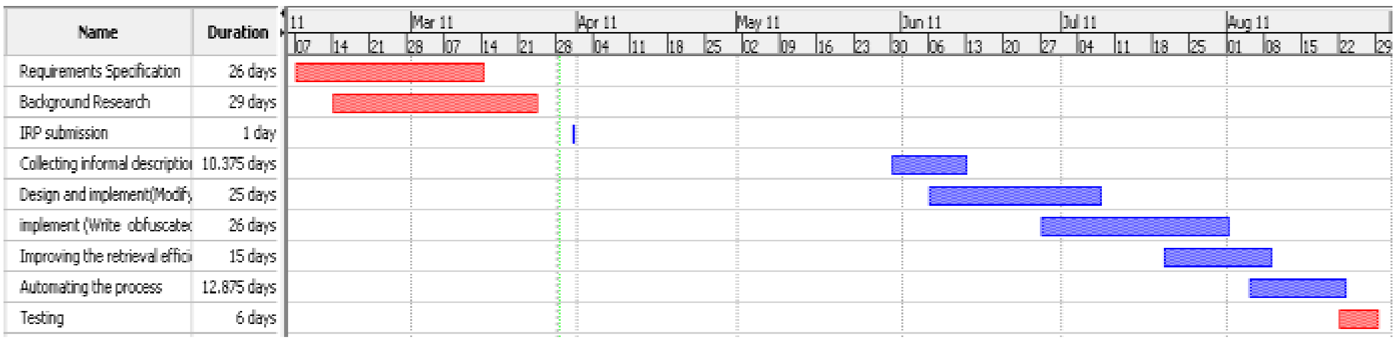
Most of the items in this list have not been described well (or even at all!).

Milestone	Estimated Time of Completion
Collecting informal descriptions(Study SBML,Read SBML models)	Week 2
Design and implement(Modify SBML, write obfuscation SBML)	Week 3
implement (Write obfuscated types, read obfuscated data)	Week 4
Improving the retrieval efficiency(De-obfuscation of data, read plain text data)	Week 6
Automating the process	Week 8
Testing	Week 10

6.1 Gantt Chart

This does not need to be a new section.

The large gap in the middle of the Gantt chart looks bad. Either start from June or explain it.



Name	Duration
Requirements Specification	26 days
Background Research	29 days
IRP submission	1 day
Collecting informal descriptio	10.375 days
Design and implement(Modify	25 days
implement (Write obfuscate	26 days
Improving the retrieval effici	15 days
Automating the process	12.875 days
Testing	6 days

Avoid diagrams and images that extend into the margin like this, it looks untidy and may not print correctly if required.

References:

[1] http://www.wisegeek.com/what-is-a-biological-model.htm

This is a terrible idea. The student should be able to find a formal definition of a biological model in a peer-reviewed paper or a published book.

[2] Hucka M, Finney A, Sauro HM et al. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 19 (4): 524–31, 2003.

[3] Dräger A, Hassis N, Supper J, Schröder A, Zell A. SBMLsqueezer: a CellDesigner plug-in to generate kinetic rate equations for biochemical networks, BMC Systems Biology 2008, 2:39, 2008.

[4] Bornstein BJ, Keating SM, Jouraku A, Hucka M. LibSBML: An API Library for SBML. Bioinformatics, 24(6):880–881, 2008.

[5] http://www.ems.bbk.ac.uk/for_students/msc_finance/Rita22sept08.pdf

A link is not enough. What is this? paper? website?
Always give details of author, title, date, authors institution and where the paper was published.

[6] Andreas Dr¨ager, Nicolas Rodriguez†, Alexander D¨orr, Marine Dumousseau†, Clemens Wrzodek. Documentation: A short description of the main differences between JSBML and libSBML.

In general there are not enough references and only some of them are 'Primary Sources' (see wiki for last years IRR notes on types of sources). The bibliography formatting is inconsistent.

GENERAL COMMENTS

Again the language (spelling, grammar and semantics) throughout is very bad. (see IRP Bad Example 1 for more details on this)

The amount of content is very low and is mostly just background. Almost none of it explains what the student actually 'proposes' to do. This is perhaps the most important aspect of IRP; you must clearly identify your research topic and illustrate how you intend to approach it. If you are finding it difficult to do this then you should speak with either myself or your supervisor.

Formatting and presentation are sub-standard throughout. This shows the student has been careless and has made little effort.

This work does not meet the standards of IRP.