

Студ. А.Д. Талатынник  
Науч. рук. доц. Н.И. Гурин  
(Кафедра информационных систем и технологий, БГТУ)

### **3D-СИМУЛЯТОР ПРОМЫШЛЕННОЙ УСТАНОВКИ ДЛЯ СИНТЕЗА АММИАКА**

Создан 3D симулятор промышленной установки для синтеза аммиака, которая установлена и применяется на заводе Гродно-Азот. Данный симулятор разработан для обучения студентов и позволяет ознакомиться с элементами установки, ввести данные для расчета (температура, давление, доли исходных газов), запустить симуляцию и посмотреть, как протекает реакция в установке. С помощью анимации газа и выполнения расчетов, студенты могут увидеть результаты реакции и понять, как изменение параметров влияет на ход процесса синтеза аммиака. Данный симулятор может быть использован как в учебных целях, так и в качестве средства для повышения квалификации лекторов и инженеров.

Процесс синтеза аммиака – это сложная химическая реакция, в которой азот и водород превращаются в аммиак в присутствии катализатора [1]. Реакция происходит при высоком давлении и высокой температуре, обычно в диапазоне от 150 до 250 градусов Цельсия и от 150 до 350 атмосфер. Реакционная смесь подается в реактор, где она проходит через слой катализатора. Катализатор, обычно состоящий из железа или металлических соединений, помогает ускорить химическую реакцию, снизив температуру, при которой она происходит [2].

В процессе реакции азот и водород соединяются, образуя аммиак. Также в процессе реакции выделяется тепло, которое нужно отводить, чтобы сохранить оптимальные условия реакции. После окончания реакции продукты охлаждают, чтобы избавиться от избыточного водорода, не прореагировавшего в реакции, и аммиак отделяют от оставшихся газов [3]. Для разработки модели установки симулятора было использовано приложение 3ds Max, с помощью которого была создана детальная 3D-модель установки для синтеза аммиака, включая все ее элементы и детали. Для создания самого симулятора была использована игровая платформа Unity, которая обеспечивает возможность быстрого прототипирования и разработки игровых приложений. Для реализации логики и функциональности симулятора был использован язык программирования C#.

#### **ЛИТЕРАТУРА**

1. Процесс Габера [Электронный ресурс]. Режим доступа: <https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0>

%B5%D1%81%D1%81\_%D0%93%D0%B0%D0%B1%D0%B5%D1%80  
%D0%B0. Дата доступа: 17.04.2023 г.

2. Катализатор. [Электронный ресурс]. Режим доступа:  
<https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%82%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%82%D0%BE%D1%80>. Дата  
доступа: 18.04.2023 г.

3. Получение аммиака. [Электронный ресурс]. Режим доступа:  
<https://obrazovaka.ru/himiya/poluchenie-ammiaka.html>. Дата доступа:  
19.04.2023 г.

УДК [004.056+003.26](075.8)

Студ. Д.С. Шкабров  
Науч. рук. проф. П.П. Урбанович  
(Кафедра информационных систем и технологий, БГТУ)

## АНАЛИЗ АЛГОРИТМА ХЭШИРОВАНИЯ КЕССАК

Безопасность является важнейшей характеристикой современных ИТ. Многие решения проблемы основаны на использовании хэш-преобразований [1, 2]. В докладе анализируется алгоритм Кессак, который генерирует хэши различных длин: 224, 256, 384 и 512. Следует отметить, что Кессак является криптографически стойкой хэш-функцией, что означает, что для любой длины хэша, которую вы выберете, вероятность обнаружения двух сообщений с одинаковым хэшем крайне мала.

Следует учитывать, что более длинные хэши обеспечивают более высокий уровень безопасности, но требуют больше ресурсов для вычисления. Кроме того, более длинные хэши обычно занимают больше места в памяти и на диске, что может быть проблемой в некоторых приложениях. Так же, по логике, должно быть, что вычисление хэша с большей длиной будет дольше по времени, чем с меньшей длиной и мы это проверим.

```
-----224-----
6ca6d17d3fee853494b6f2799d99e267326025eca5c453080b0d6dfe
1: 0.0010008811950683594
-----256-----
b943c70842371200ca1fb6df8979fdb258173617924210b60fe0c01d017d1794
1: 0.0010023117065429688
-----384-----
391fd29a076dda4ead53c531f42bb17f03a555ea76b97188fbff661fe1d562a2f2f9902b91dd918bd862d425f0bd1721
1: 0.0009987354278564453
-----512-----
21e58dd347b6b73fbbc281fe173c7c1b3f10898b6880c797852c54060a2fe17de501ae38422e6282b7a01a5c45972b0a176d403923ec60278383e23e7099f117
1: 0.0010013580322265625
-----224-----
3d75596695192c558d50081fd5fddc947fe9fd1fa9249bde657f90
2: 0.0
-----256-----
b943c70842371200ca1fb6df8979fdb258173617924210b60fe0c01d017d1794
2: 0.0
-----384-----
391fd29a076dda4ead53c531f42bb17f03a555ea76b97188fbff661fe1d562a2f2f9902b91dd918bd862d425f0bd1721
2: 0.000999275207519531
```

Рисунок 1 – Вывод результата хэширования и времени вычисления