Q&A Interface design for symmetric block cipher encryption primitives through handmade and ML decision trees

Alvin Crighton
Rohin Dasari
Brandon Vo

For our project, we decided to opt for project 2 type 2 where we would design a Q&A system such that it would inquire and guide a fresh user to choose the ideal cryptographic scheme for their project. Our decision tree provides a series of yes/no questions designed to allow the user to obtain the best block cipher for their particular use case.

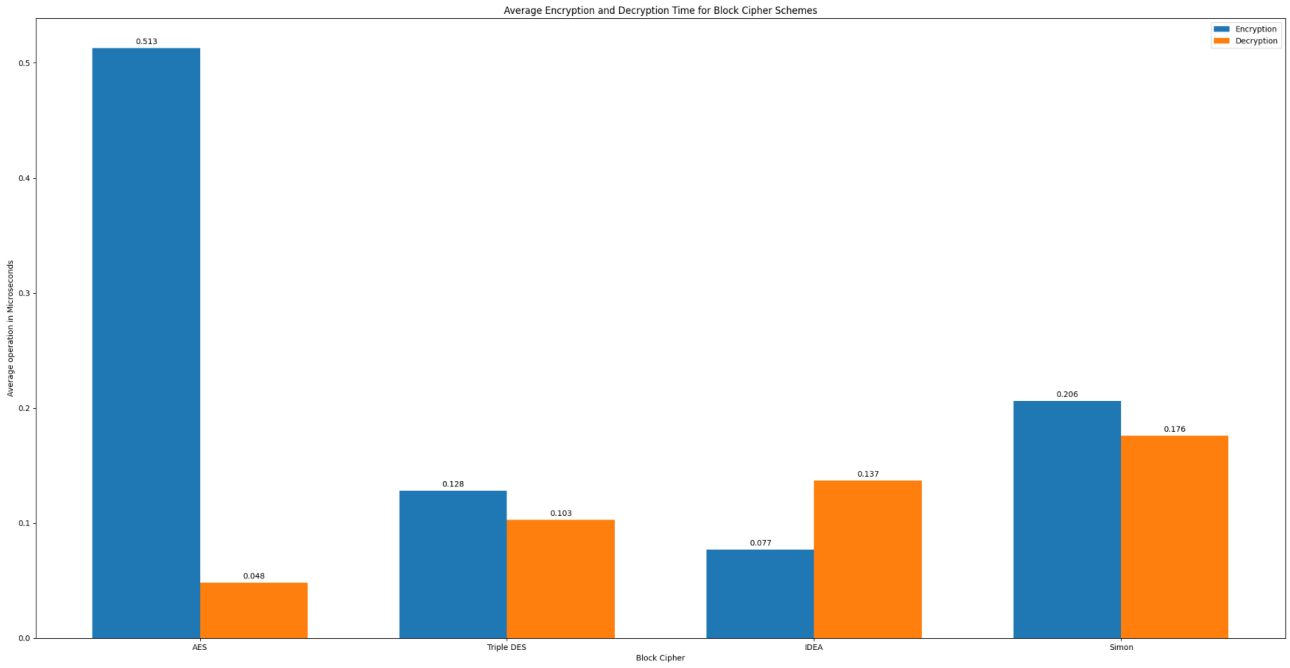
For our implementation, we decided on using a classification tree to decide what general method of encryption the user might want. The tree was implemented using Scikit's decision tree machine learning package. The tree was serialized into a format that is permissible for step-by-step user interaction and implemented as a web app using React, a common javascript frontend framework.

Because this decision tree would be designed for the symmetric block cipher cryptographic primitive, extensive research was done on 5 unique block cipher schemes that had different design goals and intentions to achieve secrecy and correctness. The set was also chosen due to being recognized as a secure scheme and proven to have not been declared broken within reasonable probability.

Runtime Analysis

While doing research on each of the encryption schemes, we searched for working implementations whether from imported cryptographic libraries or from experimental examples. We ran a benchmark test on four of the five cryptographic schemes using a 32-byte input.

3DES and AES used a 32-byte randomly generated string as input. Both schemes shared the same input for the purposes of this benchmark test. SIMON and IDEA, however, accepted only integers and had to use a separate set of inputs. SIMON accepted only 32-byte integers, and IDEA accepted only 8-byte integers. Furthermore, Threefish is not included in the benchmark because Threefish-1024 is not widely supported and struggles to find public use. We could not find a stable, non-deprecated implementation of Threefish-1024, so we could not gather quantitative data regarding the block cipher scheme. However, we have cited benchmarks pulled from other researchers and open source resources for three fishes runtime.



AES-128

AES is widely considered the standard choice for encryption schemes being efficient and effective in its performance and security. Its non-linearity and high diffusion have been achieved through its 10-round operation despite using simple mathematical operations and single encryption and decryption key, which would cost approximately 2^{190} operations just to break AES's encryption (assuming AES uses atleast 7 rounds of operations).

As such, it has been widely supported and implemented in a variety of cryptographic security libraries and has been proven to be infeasible to break for the past two decades.

Triple DES

Triple DES is an encryption scheme that will have a considerably weaker presence for our Q&A interface compared to the rest of the block ciphers. This is because in addition to being considerably weaker and slower than the rest of the more recent version of the symmetric encryption schemes, Triple DES is now considered broken due to the discovery of the Sweet32 Attack.

The Sweet32 Attack discovered a vulnerability in block ciphers that exploited the small block sizes to allow for a brute force crack of the encryption key. As such, NIST has considered Triple DES broken and declared that the scheme will be deprecated as of 2023. As a result, Triple DES's presence will be given lowest priority and will be selected only if the user is completely certain in his choice of encryption scheme.

IDEA

While AES was designed for simple but secure operations, IDEA can be described as the more complex counterpart to AES being conceptualized in 1991 but officially designed and patented sometime between 2000 and 2004. IDEA is now more accessible to use since its patent expired in 2012, allowing for official implementation aside from potential licensing issues. IDEA uses 8.5 operations where the last round finishes halfway to undo a previous operation's swap mechanic.

Despite being designed for more complex algorithms, IDEA costs less memory than AES, operating for fewer rounds. However, IDEA's speed and efficiency has been outpaced by more modern schemes and may not match modern standards for performance. Regardless, IDEA's security has been mostly proven, requiring 2^{128} operations to break the scheme.

However, one major weakness with IDEA is that the scheme is extremely vulnerable when poorly implemented. Attacks have been discovered against versions of IDEA that run for 6 rounds or less, meaning that all versions of IDEA must run for atleast 7 rounds in order to remain secure. IDEA is also exploitable if a weak private key was provided for encryption, meaning that a user must take careful precautions and be more involved with the design and implementation of this block cipher scheme to prevent showing any vulnerabilities when sending encrypted ciphertexts.

SIMON

SIMON is the most recently designed algorithm, created in 2013 with the creation being recognized as controversial due to the reputation of its creation by the NSA. Regardless, SIMON was cited to be as lightweight as possible while allowing for an acceptable amount of security being an encryption scheme that operates for 44 rounds using a 128 key length. Because of this design, SIMON is able to run effectively on low-end hardware that struggles to run even AES.

SIMON was considered to be widely supported as it was officially integrated and supported by Linux Kernels on February 2018.

Due to this reasoning, we decided that the attributes for SIMON would be an encryption scheme that prioritizes security over speed although depending on the hardware being used, SIMON is capable of maintaining security and speed.

Threefish-1024

Threefish is a recently designed algorithm relative to the rest of the block cipher schemes presented, being created in 2008 based on the Skein hash function. Threefish was created as a successor to Twofish, an encryption scheme that competed against AES and lost. Unlike the other encryption schemes, Threefish was designed with a much larger focus on security with number of brute force operations to crack Threefish being 2^{222} operations needed, an additional $* 2^{100}$ operations needed than the rest of the encryption schemes, giving Threefish-1024 the most potential to have the strongest security compared to the other schemes.

Threefish, however, sacrifices more resources than normal in order to achieve this level of security. Threefish costs more memory because it uses a keysize of up to 1024 bits as opposed to the 128, 256, and 512 standard key lengths, and Threefish operates for 72 or 80 rounds in its permutation and diffusion cycles, much more compared to the contemporary encryption schemes that run for no more than 10 rounds on average.

In addition to the large number of cycles, the encryption scheme alternates between rotating XOR execution and additions with every set. This adds more complexity to the encryption method, but this achieves nonlinearity, enabling Threefish to serve as a pseudo-random permutation.

Using the information regarding Threefish, we have deduced that Threefish is an encryption algorithm that prioritizes security and would be willing to sacrifice the most amount of resources necessary to achieve its high level of security compared to the remaining four encryption schemes.

However, one main issue with Threefish is that, possibly due to Twofish losing to AES, the encryption method is rarely implemented or documented. Many implementations found were done more as a proof of concept rather than designed for practical use, and many libraries that had included Threefish often had an obsolete and unusable library. We had great difficulty finding a working implementation of Threefish,

so we deemed that Threefish has difficulty with widespread support, documentation, and implementation.

Number of operations to crack	**Runtime (clock cycles per byte)**	**Block Size (Bits)**	**Old and proven or new and unknown?**	**Widely supported?**	**Complex or Simple**	**Output**
2^{113}	108	64	Old (1975)	Until 2023	Simple	Triple DES
infeasible	18	18	Old (1998)	Yes	Simple	AES-128
$2^{125.7}$	7.5	Flexible	New (2013)	Yes	Simple	SIMON
$2^{126.1}$	132	64	New (~2000)	Yes	Complex	IDEA
2^{222}	881	1024	New (2008)	No	Complex	Threefish-1024

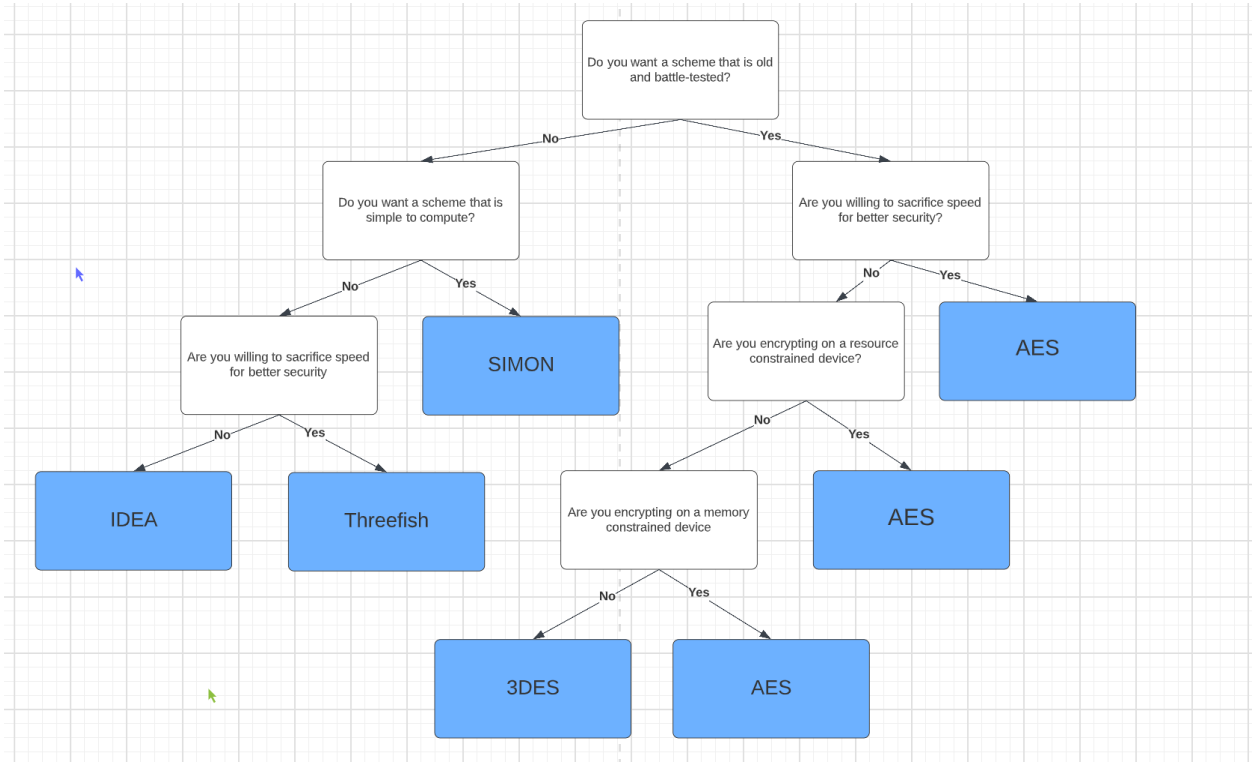
Based on data gathered from the table created and research on each of the five block ciphers, a series of 6 attributes were applied to each encryption scheme. A combination of these 6 labels were used to best match the potential output scheme. In order to allow for a certain degree of tolerance and deviation, multiple entries and different combinations were also written to describe the same type of block scheme. For example, AES will include an extra variation of input to identify that AES can be chosen for secure block cipher schemes or fast block cipher schemes. While SIMON was designed for resource-constrained devices, additional data was inputted to allow for leniency in offering SIMON for non-lightweight devices. The resulting data set is provided below:

Question 1	Question 2	Question 3	Question 4	Question 5	Question 6	Scheme
Old and proven ove modern and experimental? modern and experimental?	Do you want a widely Supported Scheme?	Are you encrypting data on a resource-co nstrained device?	Security Over Speed?	Simple Or complex?	Are you fine with high memory usage?	
Yes	No	No	Security	Simple	Yes	Triple DES

Yes	No	Yes	Security	Simple	Yes	Triple DES
Yes	Yes	Yes	Speed	Simple	No	AES
Yes	Yes	No	Speed	Simple	Yes	AES
Yes	No	No	Speed	Simple	No	AES
Yes	Yes	Yes	Security	Simple	Yes	AES
Yes	Yes	No	Security	Simple	Yes	AES
Yes	No	No	Security	Simple	No	AES
No	Yes	Yes	Security	Complex	No	IDEA
No	No	Yes	Security	Complex	Yes	IDEA
No	Yes	Yes	Security	Complex	Yes	IDEA
No	No	Yes	Security	Complex	Yes	IDEA
No	No	Yes	Security	Complex	No	IDEA
No	Yes	Yes	Security	Simple	No	SIMON
No	Yes	Yes	Security	Simple	No	SIMON
No	Yes	Yes	Speed	Simple	Yes	SIMON
No	No	No	Speed	Simple	Yes	SIMON
No	No	No	Security	Simple	No	SIMON
No	No	No	Security	Complex	Yes	Threefish
No	No	Yes	Security	Complex	Yes	Threefish
No	No	No	Speed	Complex	Yes	Threefish
No	No	Yes	Speed	Complex	Yes	Threefish

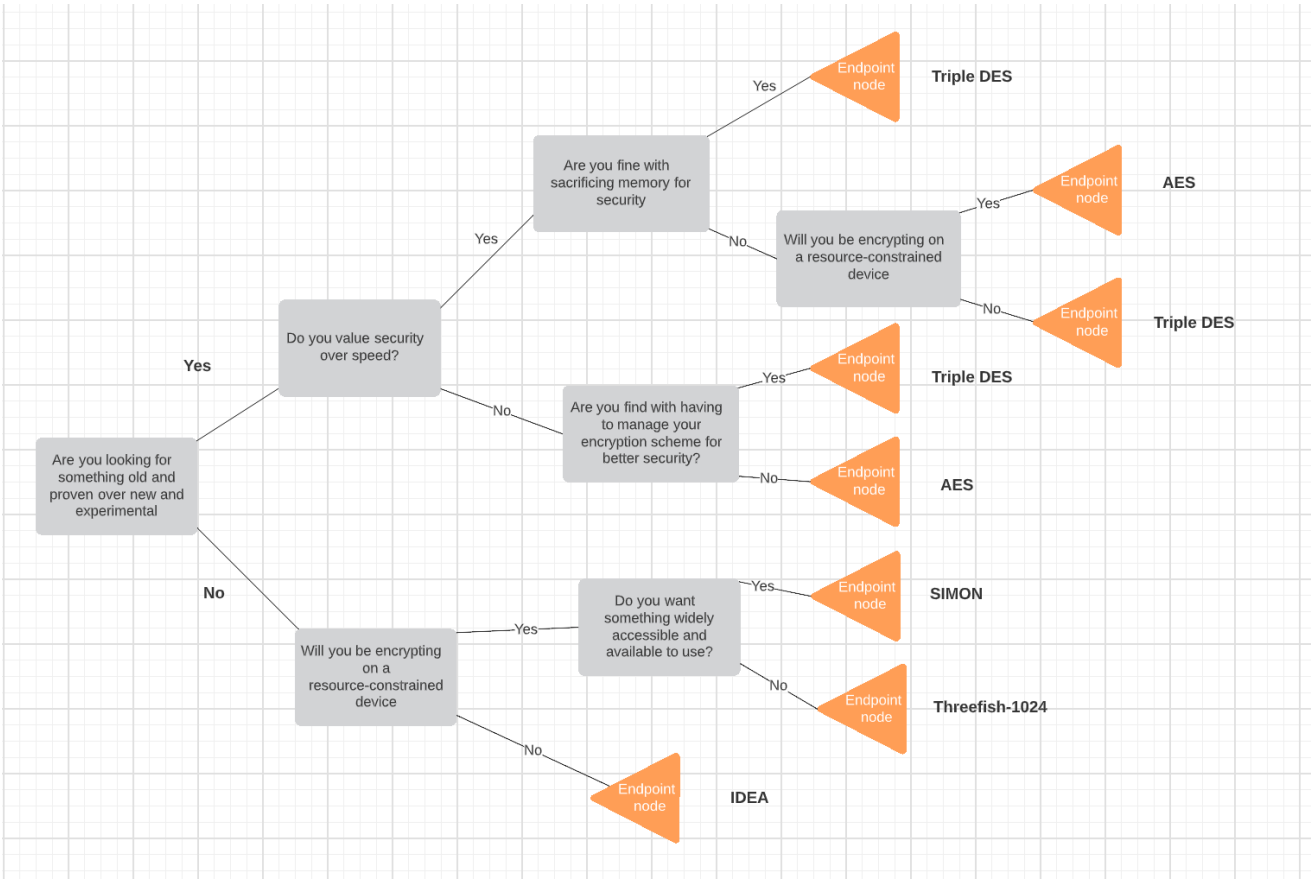
The table above was imported into the decision tree algorithm. In order to test the possible desired outcomes, two different decision tree algorithms were created with

their own sets of data sets to train the algorithms. The tree shown below was generated from a decision tree generator library created using the scikit Decision Tree Library. However, the ML tree was converted and modified in order to allow every feature to be represented as binary values. As a result, certain outputs might have contradictory attributes such as SIMON being allowed for resource-constrained and non resource-constrained devices. In addition, the parent nodes in the ML tree were converted from the attributes being compared into the questions that would be asked regarding the attribute.



This was created through a data set that identified each of the five block cipher schemes based on age, widespread support, resource-constraints, security vs. speed, simplicity vs. complexity, and memory usage. To further add deviation, multiple entries leading to the same scheme were created as a means to show acceptable deviations from the user expectation. For instance, while AES is marked as prioritizing speed over security, the data set will allow some entries prioritizing security to decide upon AES as the encryption scheme has an acceptable balance of both speed and security.

Based on the newly created data trees and based on the data gathered from research and benchmarking, the following decision tree was created by hand. This tree also carries the same methodology of accounting for acceptable deviations from expectations based on expectations.



To evaluate the trees, we sampled a random scheme and stepped through the tree, answering the questions according to the characteristics that the chosen scheme has. Both trees produced the correct results for all schemes, meaning that both trees have 100% accuracy. The ML tree has a total of 6 question nodes, while the hand made tree has 7 question nodes. Therefore, the ML tree is more efficient, since it has the same amount of information encoded with less space.

The ML tree was originally generated using Scikit learn, which does not provide functionality to traverse the tree, question by question. This means that the Scikit learn tree can not be used as a Q/A interface. In order to resolve this, we implemented the Scikit learn tree in Javascript and added this functionality. The result is a QA interface based on the ML tree that a user can interact with at each step.

When interacting with the decision tree itself, it takes, on average, 3 questions before outputting the ideal encryption scheme output.

Test Run:

Desired output: AES

Do you want a scheme that is old and battle-tested? -> Yes

Are you willing to sacrifice speed for better security? -> No

Are you encrypting on a resource constrained device? -> Yes

Output: AES

Desired Scheme: Threefish

Do you want a scheme that is old and battle-tested?**->YES**

Are you willing to sacrifice speed for better security?**->YES**

For this trial run, the user wanted to find the Threefish output scheme, but the user also specified that he was looking for an old and battle-tested encryption scheme despite Threefish being relatively new.

OUTPUT: AES

Sources:

AES encryption method and properties:

https://ro.uow.edu.au/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1176&context=theses

AES vs. IDEA

https://www.sciencedirect.com/science/article/pii/S0026269208005661#:~:text=The%20AES%20implementation,implement%2016%20SBox%20by%20phase.

José M. Granado, Miguel A. Vega-Rodríguez, Juan M. Sánchez-Pérez, Juan A. Gómez-Pulido,
IDEA and AES, two cryptographic algorithms implemented using partial and dynamic reconfiguration,
Microelectronics Journal,
Volume 40, Issue 6,
2009,
Pages 1032-1040,
ISSN 0026-2692,
https://doi.org/10.1016/j.mejo.2008.11.044.
(https://www.sciencedirect.com/science/article/pii/S0026269208005661)

SIMON resources:

https://github.com/nsacyber/simon-speck

Linux announcing support for SPECK and SIMON

https://www.mail-archive.com/linux-crypto@vger.kernel.org/msg30853.html

IDEA on the cryptowiki:

https://www.cryptopp.com/wiki/IDEA

AES vs. SIMON

https://csrc.nist.gov/csrc/media/events/lightweight-cryptography-workshop-2015/documents/papers/session1-shors-paper.pdf

https://www.mail-archive.com/linux-crypto@vger.kernel.org/msg30853.html

Triple DES vs. AES
https://symbiosisonlinepublishing.com/computer-science-technology/computerscience-information-technology32.php

https://www.baeldung.com/cs/des-vs-3des-vs-blowfish-vs-aes

Triple DES Sweet-32 attack

https://sweet32.info/

IDEA encryption attack

Biham, E., Dunkelman, O., Keller, N. *et al.* New Attacks on IDEA with at Least 6 Rounds. *J Cryptol* 28, 209–239 (2015). https://doi.org/10.1007/s00145-013-9162-9

Co-author of Threefish discusses Threefish's security:

https://crypto.stackexchange.com/questions/11725/has-threefish-successfully-been-attacked-practically-or-theoretically

Comprehensive comparison of symmetric block cipher schemes:

http://paper.ijcsns.org/07_book/201812/20181218.pdf

https://eprint.iacr.org/2010/538.pdf

http://article.nadiapub.com/IJSIA/vol9_no4/27.pdf

Threefish's implementation issues:

https://crypto.stackexchange.com/questions/42463/what-encryption-should-i-use-blowfish-twofish-or-threefish

Skein-1024 vs. Skein 256 (Threefish's algorithm):

https://crypto.stackexchange.com/questions/16029/skein-state-size-advantages

Skein implementation benchmarking

https://www.schneier.com/wp-content/uploads/2015/01/skein.pdf

Importance of Block Size for Encryption:

https://crypto.stackexchange.com/questions/35450/how-does-blocksize-affect-security

NIST report on Block Ciphers:

https://csrc.nist.gov/CSRC/media/Publications/sp/800-131a/rev-2/draft/documents/sp800-131Ar2-draft.pdf

New Weak-Key Classes of IDEA

https://www.esat.kuleuven.be/cosic/publications/article-189.pdf

Alternative block ciphers:

- https://en.wikipedia.org/wiki/IDEA_NXT
- https://en.wikipedia.org/wiki/Serpent_(cipher)