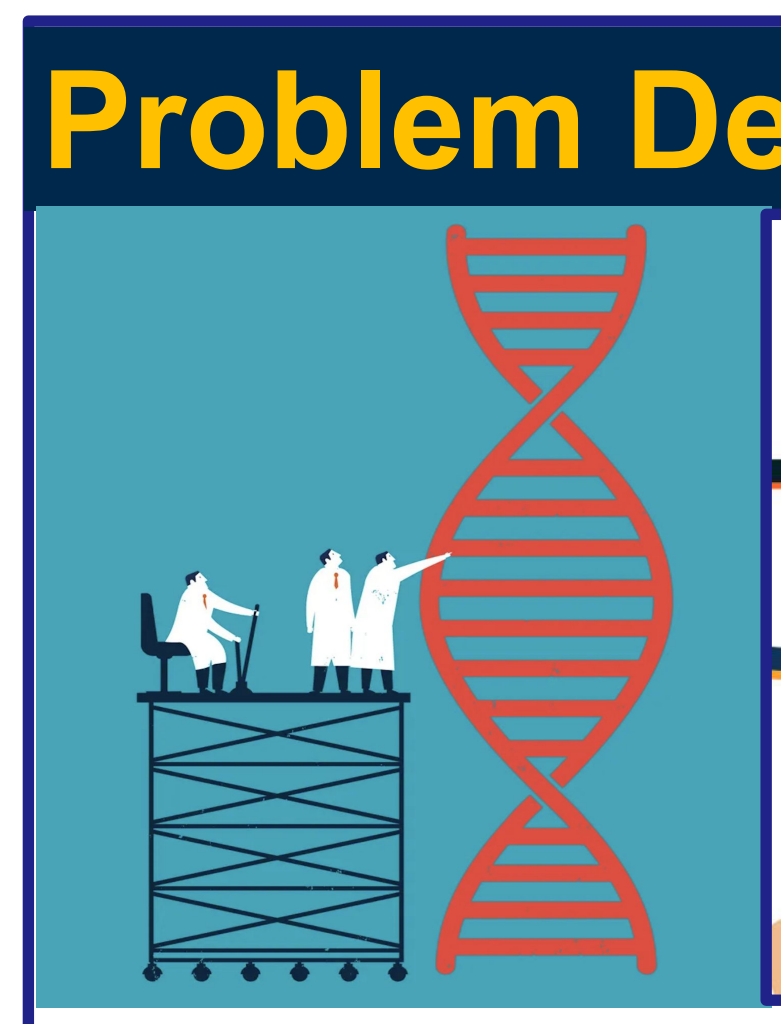
The Most Impenetrable Castle(TMIS): Genome Sequencing with Side-Channel Attack Protection inside INTEL SGX

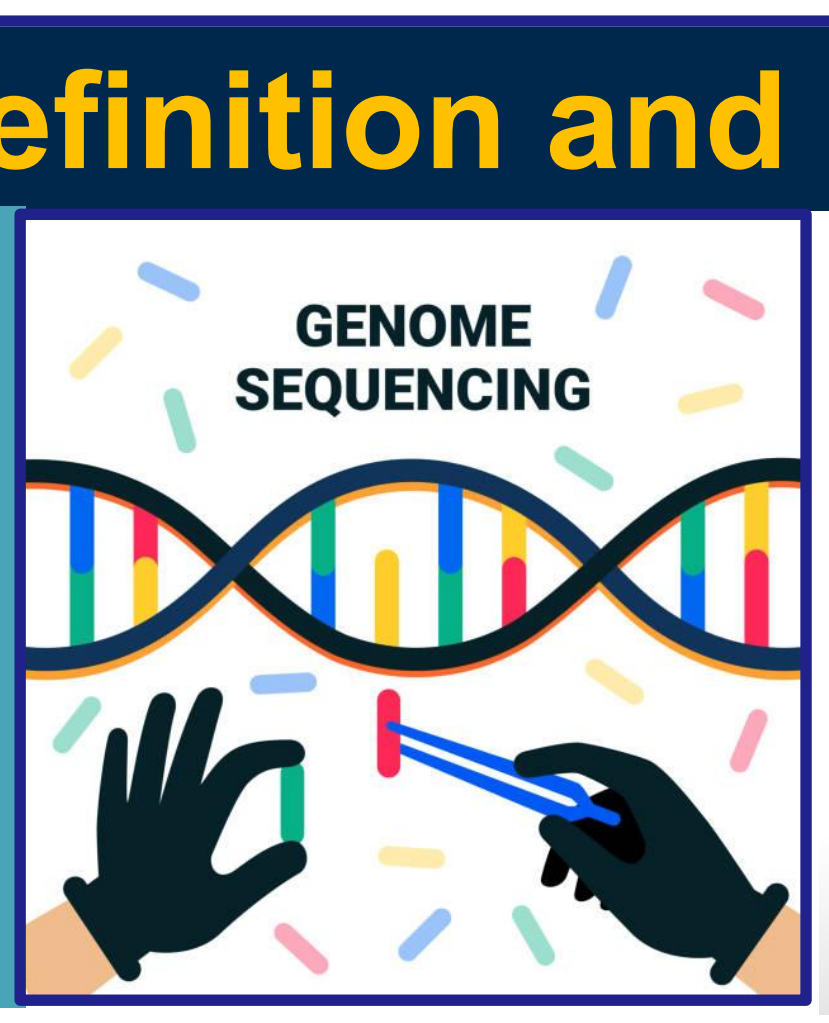
Donayam Nega Benti, Aymen Jelaludin Ahmed, Suman Kumar Mallik, Jack Andrew Skupski, Xiyu Tian
{donayam, ahmedaj, mallik, jaskupsk, xiyutian}@umich.edu
Electrical Engineering and Computer Science, University of Michigan, Ann Arbor

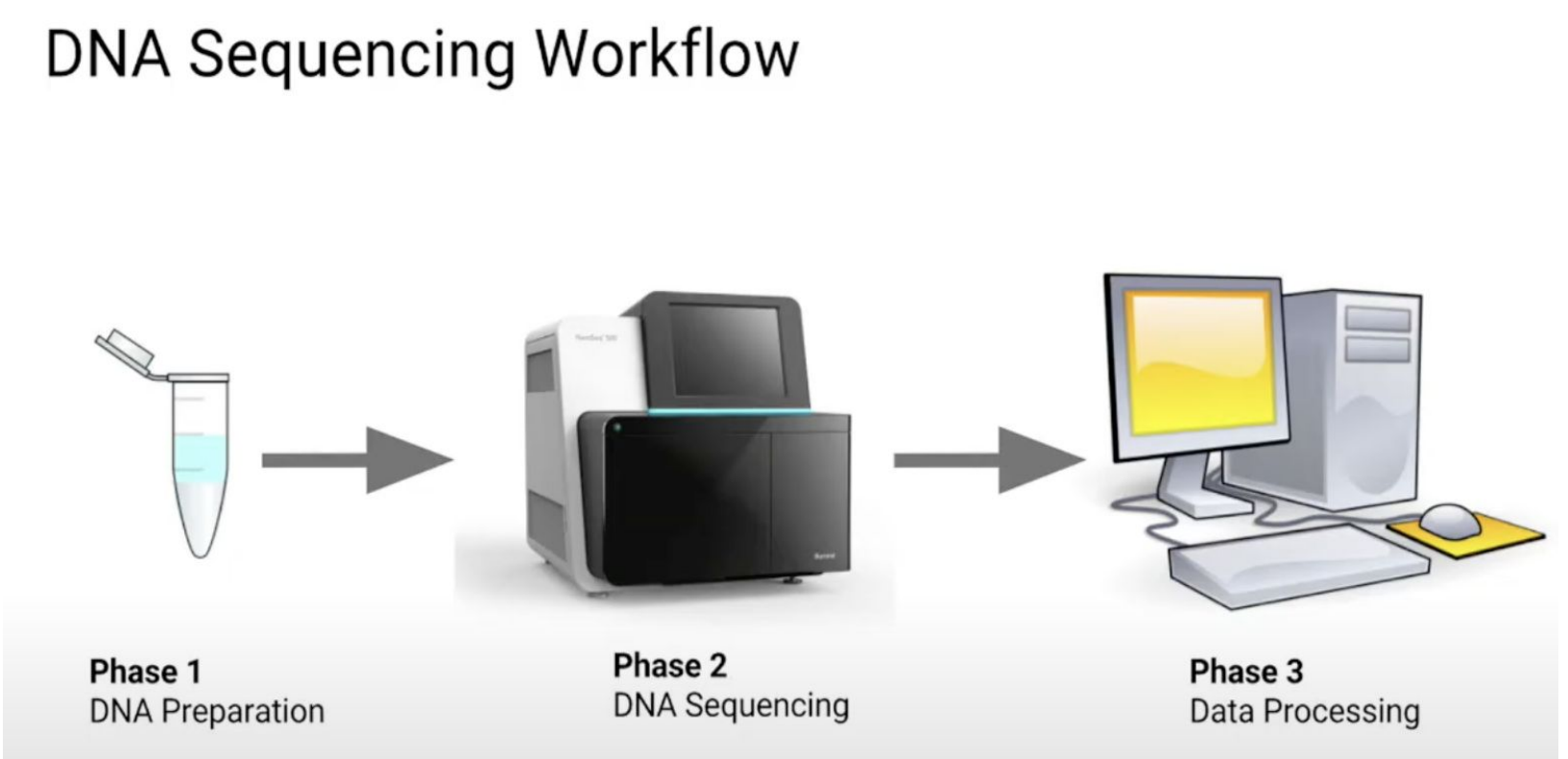
EECS 570, W22
Parallel Computer Architecture
Prof. Yatin A. Manerkar

GitHub Repo and ePoster

Problem Definition and Motivation






DNA Sequencing Workflow

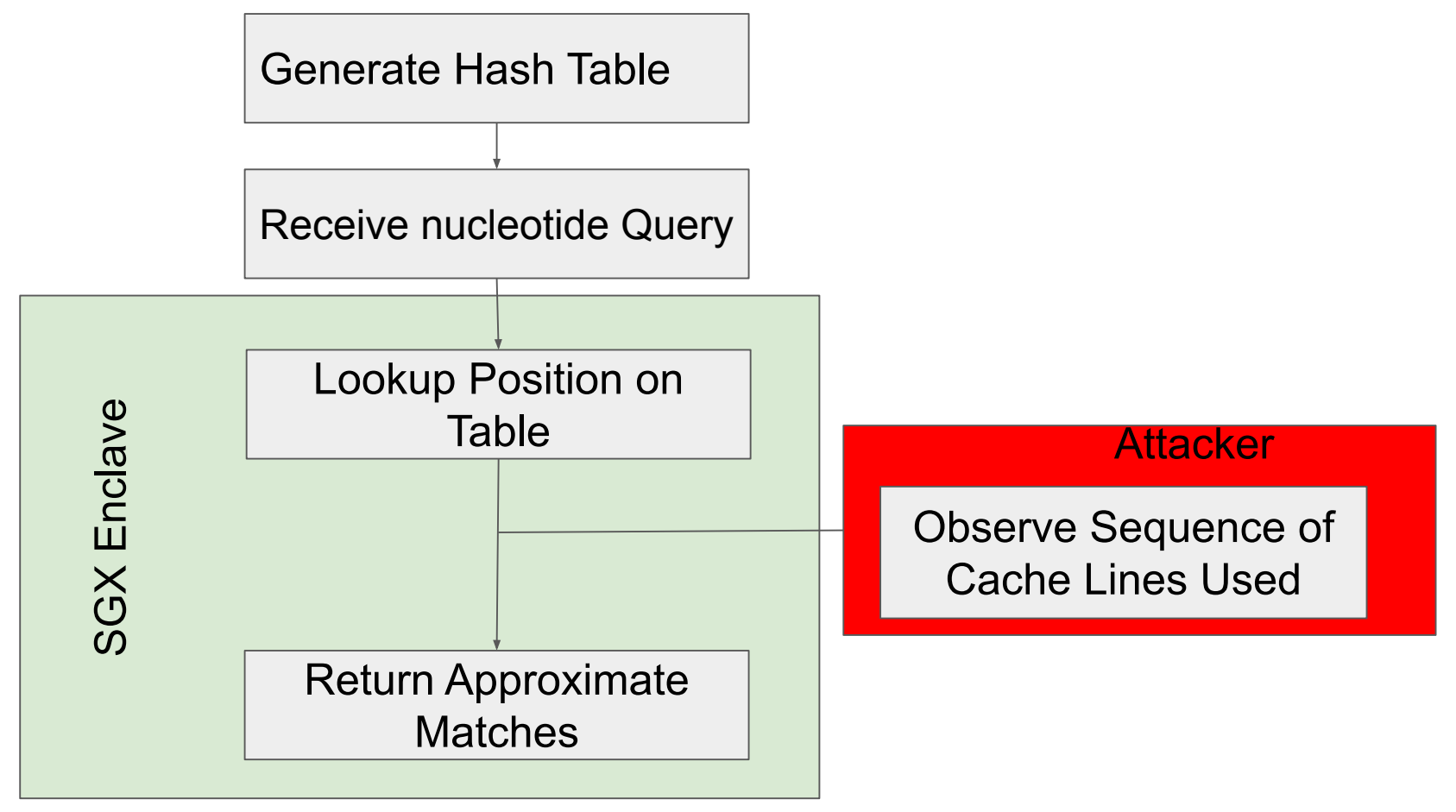
- Genome sequencing is an important and privacy-sensitive computation; therefore, it is imperative to protect the data using a Trusted Execution Environment (TEE), such as: Intel SGX, ARM's TrustZone, AMD's Secure Execution Environment, and Apple's Secure Enclave.
- Intel Software Guard Extensions (SGX) were designed to encrypt memory sections natively and perform computation inside hardware-encrypted enclaves. Genome sequencing applications can use Intel SGX to protect user data privacy; however, much research has been done showing that Intel SGX is prone to both timing and cache related side channel attacks. The purpose of our research is both protect genome sequencing using SGX, while additionally preventing what SGX cannot: cache side channel attacks.

Prior Work

- **HySec-Flow**: Implemented full genome sequencing inside an SGX enclave. [4]
- **Software Grand Exposure**: Demonstrated cache side channel attack on SGX. [5]
- **Foreshadow**: Demonstrated speculative attack on SGX. [6]
- **Data Oblivious Genome Variants Search**: Memory oblivious implementation to prevent memory and cache-based side channel attacks. [7]
- **Time and Order:** Presented ANABLEPS that detects side-channel vulnerabilities in enclave binaries, considering both memory access order and time. [8]

Primex-nucleotide search

- Genome sequencing applications are prone to cache side channel attacks:



Experimental Methodology

- Intel SGX Size:
  - 128 MB (typically)
- Enclave Page Cache(EPC):
  - 96 MB
- Primex: a program that creates a lookup table from DNA sequence and receives nucleotide queries to search for matches

Level 1:Intel SGX + Primex

- ➕ Leverages trusted hardware to establish a secure container
- ➕ Ensures data integrity and **confidentiality**
- ✖ No Side Channel Protection(no confidentiality)

Level 2: Intel SGX + Side Channel Protection + Primex

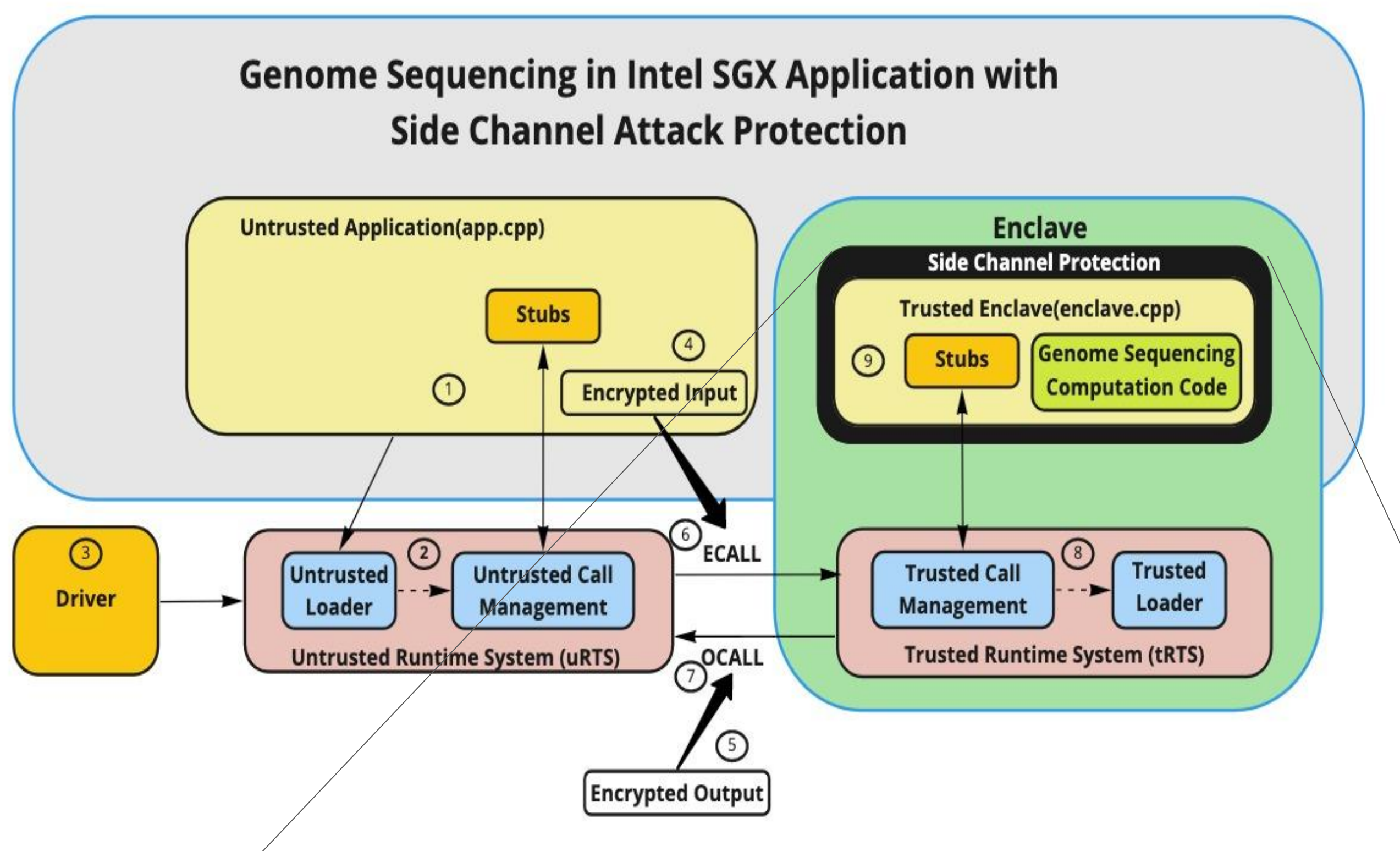
- ➕ Protections from Level 1
- ➕ Side Channel Protection
- ✖ Input query is unprotected

Level 3: Intel SGX + Side Channel Protection + Encryption + Primex

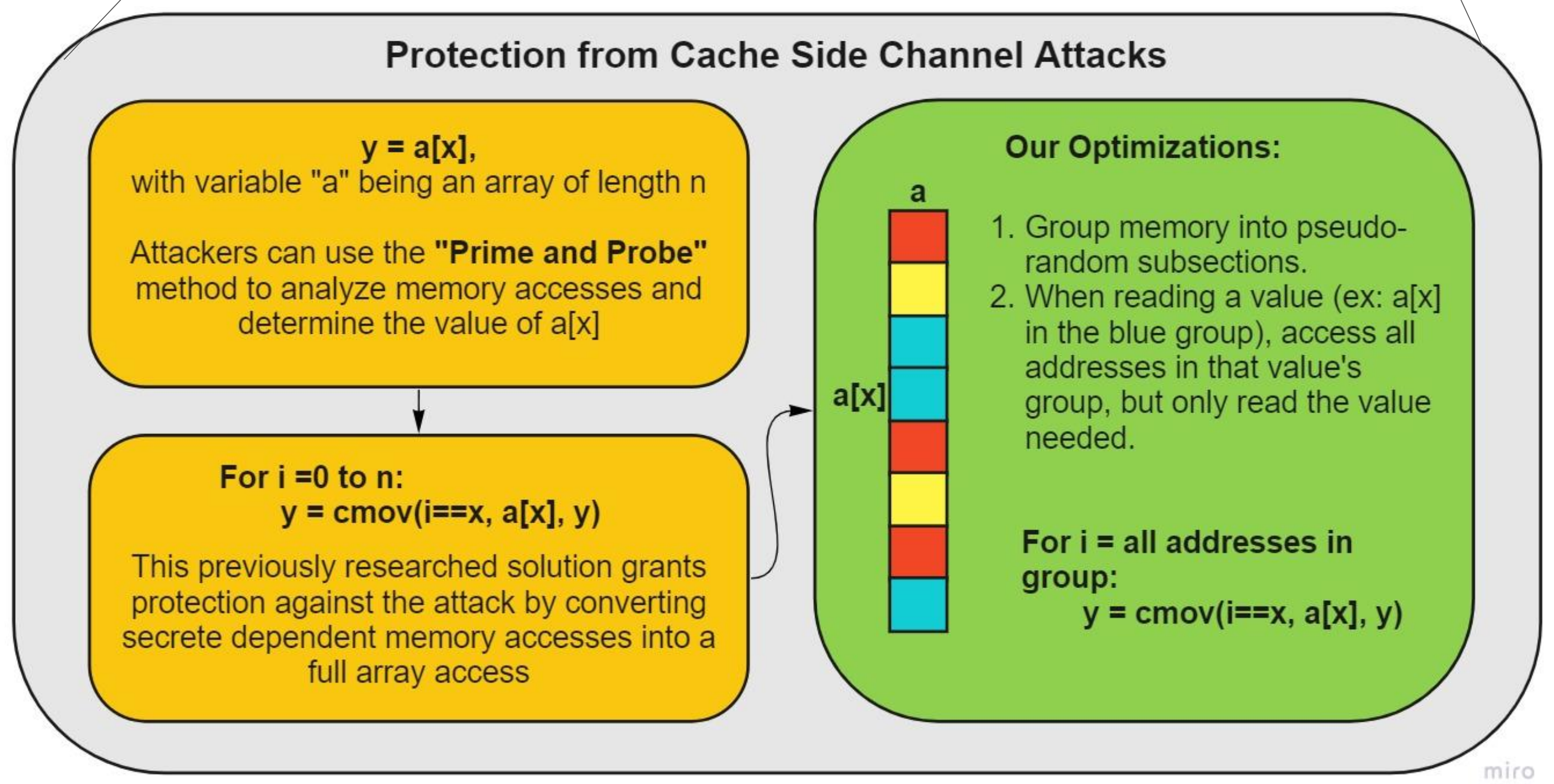
- ➕ Protections from Level 1 and 2
- ➕ Input query is encrypted before entering enclave

Our Design Approach

- **We propose to implement a cache side-channel attack-resistant nucleotide search application using Intel SGX:**


Genome Sequencing in Intel SGX Application with Side Channel Attack Protection
Untrusted Application(app.cpp)
Stubs
1
4
Encrypted Input
Enclave
Side Channel Protection
Trusted Enclave(enclave.cpp)
9
Stubs
Genome Sequencing Computation Code
3
Driver
Untrusted Loader
2
Untrusted Call Management
Untrusted Runtime System (uRTS)
6
ECALL
OCALL
7
5
Encrypted Output
Trusted Call Management
8
Trusted Loader
Trusted Runtime System (tRTS)

1. **Untrusted** : Application environment outside the enclave.
2. **uRTS** :Code that executes outside the enclave environment and performs functions such as:
   a. Loading and manipulating an enclave(Ex: destroying an enclave)
   b. Making calls(ECALLS) to an enclave and receiving calls(OCALLS) from an enclave
3. **Driver** : PSW and DCAP for supporting miscellaneous operation and functionalities.
4. **Encrypted Input** : For additional security.
5. **Encrypted Output** : For additional security.
6. **ECALL** :A call from the application into an interface function within the enclave.
7. **OCALL** :A call made from within the enclave to the application.
8. **tRTS** :Code that executes within the enclave environment and performs functions such as:
   a. Receiving calls (ECALLs) from the application and making calls outside (OCALLs) the enclave.
   b. Managing the enclave itself.
9. **Trusted** : Code or construct that runs in the Trusted Execution Environment inside the enclave.


Protection from Cache Side Channel Attacks

y = a[x],
with variable "a" being an array of length n

Attackers can use the **"Prime and Probe"** method to analyze memory accesses and determine the value of a[x]

For i =0 to n:
y = cmov(i==x, a[x], y)

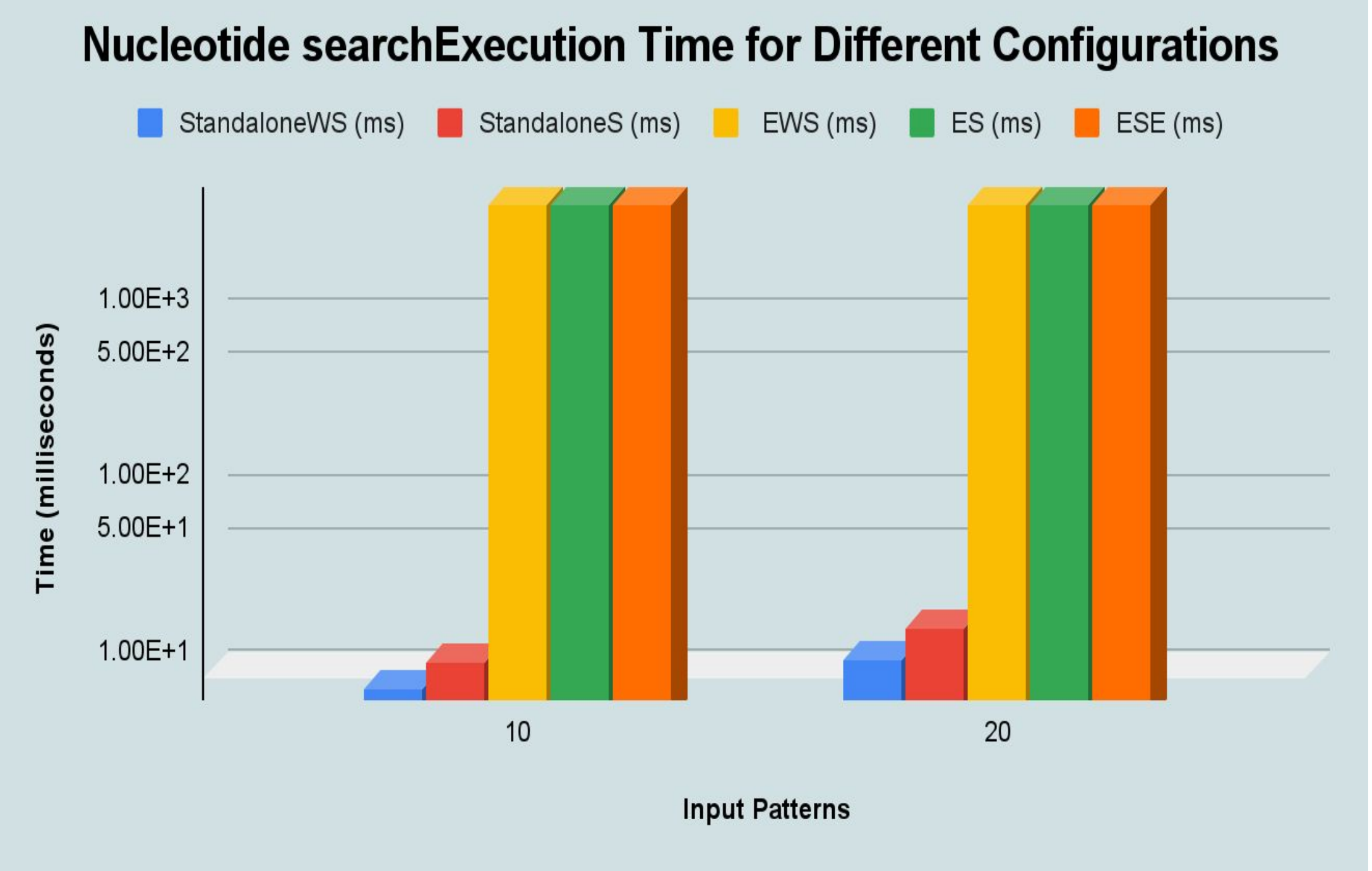
This previously researched solution grants protection against the attack by converting secrete dependent memory accesses into a full array access

Our Optimizations:

1. Group memory into pseudo-random subsections.
2. When reading a value (ex: a[x] in the blue group), access all addresses in that value's group, but only read the value needed.

For i = all addresses in group:
y = cmov(i==x, a[x], y)

Results

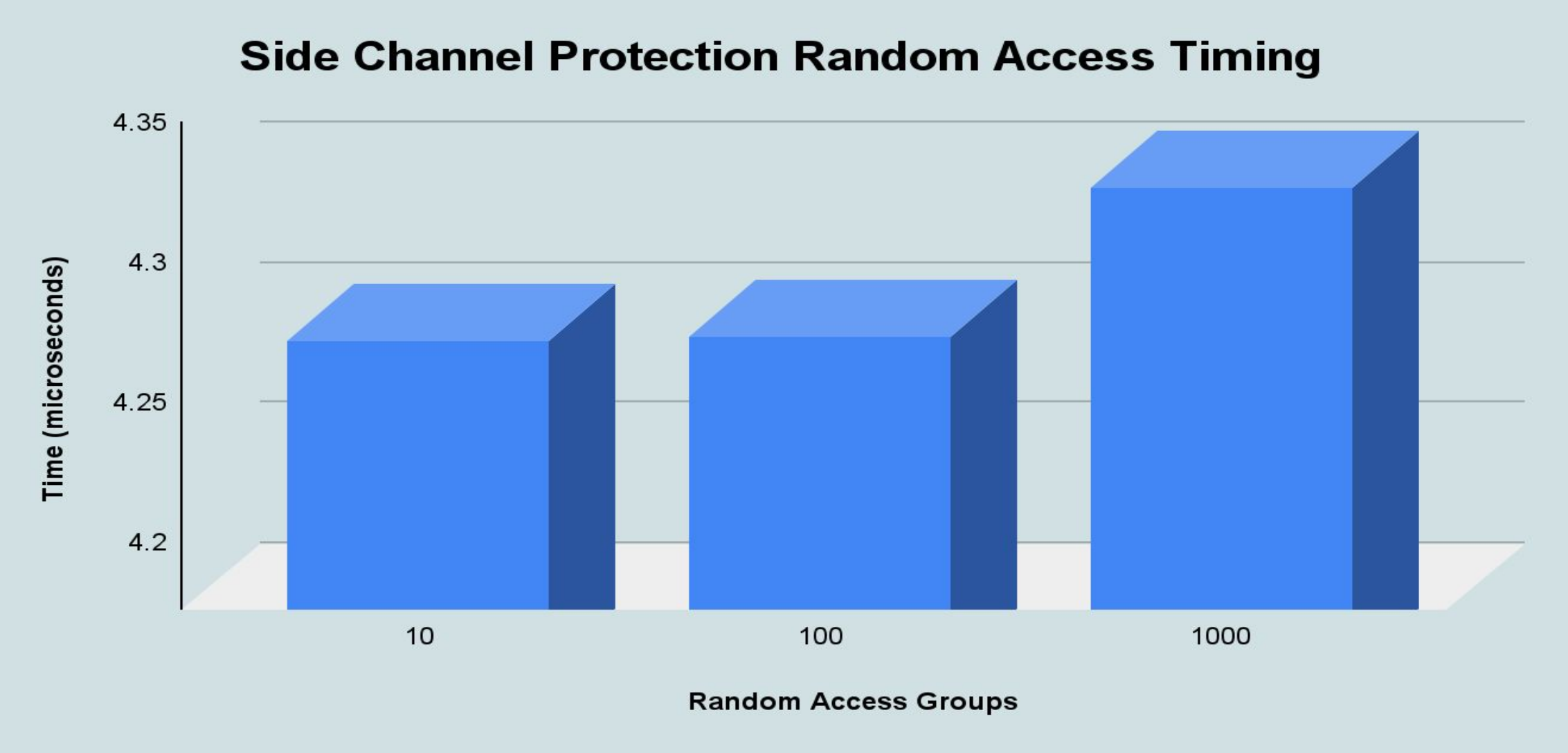

Nucleotide searchExecution Time for Different Configurations

- **StandaloneWS**: Genome Sequencing code without enclave and side channel protection
- **StandaloneS**: Genome Sequencing code without enclave
- **EWS**: Enclave without side channel protection
- **ES**:Enclave with side channel protection
- **ESE**:Enclave with side channel protection using encrypted input

Results

Input Length	StandaloneWS (ms)	StandaloneS (ms)	EWS (ms)	ES (ms)	ESE (ms)
10	7.63	10.83	4291.22	4301.10	4305.17
20	11.21	17.21	4293.30	4308.06	4293.53

We observe that changing the input data set doesn't considerably impact the execution time.


Side Channel Protection Random Access Timing

Increasing the size of random access in order to provide side channel protection increases execution time of the program.

Challenges

- Intel SGX is not supported in Mac OS or any virtual machine.
  - Supported only on Intel's 6th-10th generation processors.
- Many standard C/C++ libraries are not supported inside an SGX enclave.
- Genome sequencing algorithms are complex to debug, edit and understand.

Conclusion

- Attacks like Foreshadow are impossible to protect with only source code modification, even when using Intel SGX.
- Also intel SGX fails to provide confidentiality, application can still use it to provide Integrity

Future Work

- Include remote attestation.
- Add client and server application communication logic over secure TCP network.
- Add task partition so our implementation can be scaled to larger applications using multiple enclaves.

Acknowledgements

We want to extend special thanks to Professor Yatin A. Manerkar for constant support and encouragement, and for requesting DCO for a server. And lastly, we want to thank DCO, especially Elizabeth Z for her prompt supports.

References

[1] Intel(R) Software Guard Extensions for Linux* OS
[2] Software Grand Exposure: SGX Cache Attacks Are Practical
[3] GenomicsBench: A Benchmark Suite for Genomics*ISPASS 2021
[4] HySec-Flow: Privacy-Preserving Genomic Computing with SGX-based Big-Data Analytics Framework
[5] Software grand exposure: SGX cache attacks are practical
[6] FORESHADOW: Extracting the Keys to the Intel SGX Kingdom with Transient Out-of-Order Execution
[7] Data Oblivious Genome Variants Search on Intel SGX
[8] Time and Order: Towards Automatically Identifying Side-Channel Vulnerabilities in Enclave Binaries