Technische Universität Braunschweig

Funded by DFG

SGXoMeter: Open and Modular Benchmarking for Intel SGX

EuroSec 2021

Mohammad Mahhouk, Nico Weichbrodt, Rüdiger Kapitza
mahhouk@ibr.cs.tu-bs.de
@m_mahhouk

TU Braunschweig, Germany

The Rise of Trusted Execution Environments

- Increasing interest and rapid development of TEEs
- Confidentiality and integrity protection against a strong threat-model
- Hardware-based technologies:
  - Intel SGX
  - AMD SEV-VMs
- Commercial secure clouds:
  - Microsoft Azure (Intel SGX)
  - Google confidential VMs (AMD SEV)

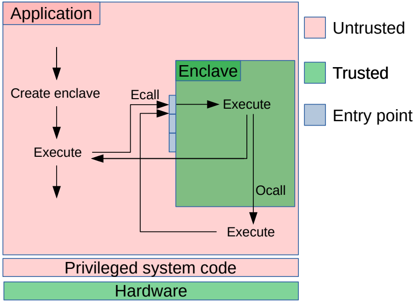
The Rise of Trusted Execution Environments

- Increasing interest and rapid development of TEEs
- Confidentiality and integrity protection against a strong threat-model
- Hardware-based technologies:
  - Intel SGX
  - AMD SEV-VMs
- Commercial secure clouds:
  - Microsoft Azure (Intel SGX)
  - Google confidential VMs (AMD SEV)

How do the frequent changes and enhancement of these TEEs affect the performance?

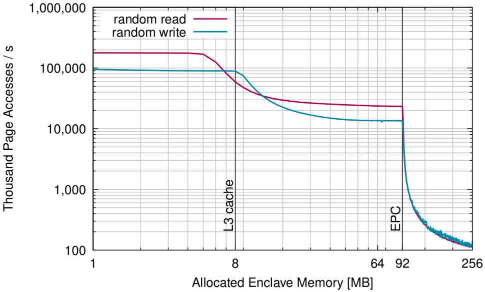
Brief Introduction into Intel SGX

- Extension of the x86 instruction set
- Trusted execution environments called *Enclaves*
- Enclaves characteristics:
  - Isolated memory regions
  - Stored in the Enclave Page Cache (EPC)
  - Part of the application’s address space
  - Contains the sensitive data and critical code
- SGX software development kit (SDK)
  - Eases the work with the enclaves



Performance Overhead Factors

- Enclave size limitation [1,2]
  - Exceeding EPC size limitation ≈ $93 MiB / 188 MiB$
  - En/Decryption upon loading uncached buffer
- Enclave transitions [3]
  - Security checks
  - Buffer copy



Source: [2]

[1] Brenner et al., SecureKeeper, Middleware ’16

[2] Arnautov et al., SCONE, OSDI ’16

[3] Weisse et al., Hotcalls, ISCA ’17

TEE Evolution Impact on the Performance

- *μcode* updates to mitigate side-channel attacks such as Spectre and Foreshadow
  - Increase overhead of enclave transition [Weichbrodt et al., sgx-perf, Middleware '18]

- Mitigations against controlled-channel attacks
  - Mitigating page-fault [Fu et al., SGX-LAPD, RAID '17]
  - Mitigating branch shadowing attacks for Intel SGX [Hosseinzadeh et al., SysTEX '18]

- Progression of the SGX SDK development
  - Enhancements and mitigations

TEE Evolution Impact on the Performance

- $\mu code$ updates to mitigate side-channel attacks such as Spectre and Foreshadow
  - Increase overhead of enclave transition [Weichbrodt et al., sgx-perf, Middleware '18]
- Mitigations against controlled-channel attacks
  - Mitigating page-fault [Fu et al., SGX-LAPD, RAID '17]
  - Mitigating branch shadowing attacks for Intel SGX [Hosseinzadeh et al., SysTEX '18]
- Progression of the SGX SDK development
  - Enhancements and mitigations

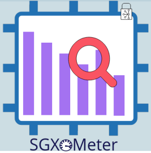
Only micro-benchmarks and no widely used application benchmark tool for Intel SGX

Motivation & Our Goal

- Lack of application benchmark tools dedicated for Intel SGX
- Analysis of SGX-NBench
  - Suitability as an SGX benchmark
  - Highlight the existing flaws

Development of an application benchmark framework for Intel SGX

- Extensible and easy to use
- SGX-suitable benchmark applications
- Reproducibility and comparability of research results



Contents

- **Brief Analysis and Evaluation of SGX-NBench**
- **The Development of SGXoMeter, a Benchmarking Framework Dedicated for Intel SGX**
  - Architecture and Workflow
- **Performance Overhead Evaluation of 2 Different SGX-SDK Versions with SGXoMeter**
  - Old SDK Version 2.7 vs New SDK Version 2.12
- **Insight on the Upcoming Plans for SGXoMeter Framework**

SGX-NBench in a 🌰

- Port of nbench-byte [4]
- Nbench is developed in the mid-90s
- CPU, FPU and memory speed benchmarks
- Single threaded
- Open-source[1]
- Used in other research papers [5,6]

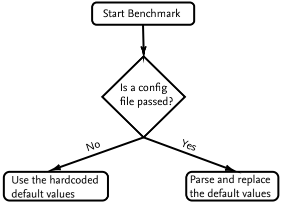
[4] Fu et al., SGX-LAPD, RAID '17

[5] Shih et al., T-SGX, NDSS '17

[6] Hosseinzadeh et al., Mitigating Branch-Shadowing attacks, SysTEX '18

[1] https://github.com/utds3lab/sgx-nbench

Workflow and Evaluation


Start Benchmark
Is a config file passed?
No
Yes
Use the hardcoded default values
Parse and replace the default values

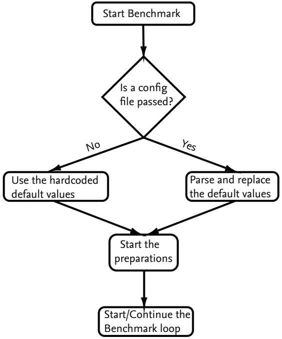
Usability:

- ✗ Missing documentation
- ✗ Unknown hardcoded default values

Workflow and Evaluation

Usability:

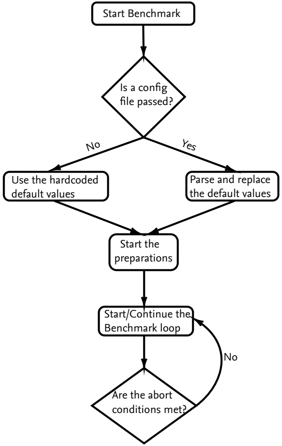
- ✗ Missing documentation
- ✗ Unknown hardcoded default values
- ✗ No warm-up phase


Start Benchmark
Is a config file passed?
No
Yes
Use the hardcoded default values
Parse and replace the default values
Start the preparations
Start/Continue the Benchmark loop

Workflow and Evaluation

Usability:

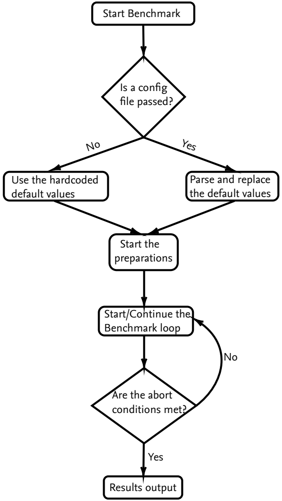
- ✗ Missing documentation
- ✗ Unknown hardcoded default values
- ✗ No warm-up phase
- ✗ $5 \leq$ result samples per benchmark ≤ 30


Start Benchmark
Is a config file passed?
No
Yes
Use the hardcoded default values
Parse and replace the default values
Start the preparations
Start/Continue the Benchmark loop
No
Are the abort conditions met?

Workflow and Evaluation

Usability:

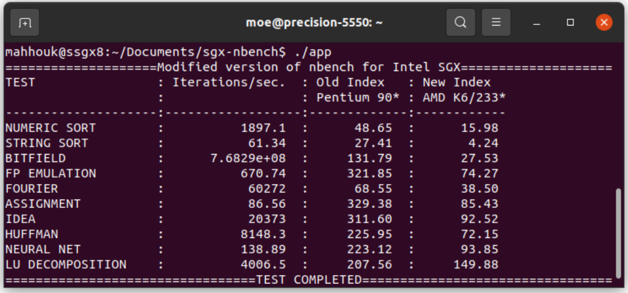
- ✗ Missing documentation
- ✗ Unknown hardcoded default values
- ✗ No warm-up phase
- ✗ $5 \leq$ result samples per benchmark ≤ 30
- ✗ No sensible baseline


Start Benchmark
Is a config file passed?
No
Yes
Use the hardcoded default values
Parse and replace the default values
Start the preparations
Start/Continue the Benchmark loop
No
Are the abort conditions met?
Yes
Results output

Workflow and Evaluation

SGX-Suitability:

- ✗ Enclave transitions
- ✗✓ Suitable benchmarks
- ? Reliability

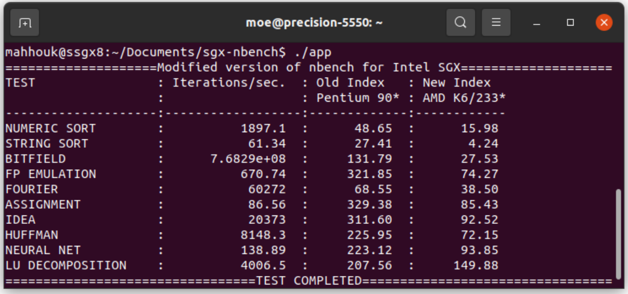


```
mahhouk@ssgx8:~/Documents/sgx-nbench$ ./app
====================Modified version of nbench for Intel SGX====================
TEST                : Iterations/sec.  : Old Index   : New Index
                    :                  : Pentium 90* : AMD K6/233*
--------------------:------------------:-------------:------------
NUMERIC SORT        :          1897.1  :      48.65  :      15.98
STRING SORT         :           61.34  :      27.41  :       4.24
BITFIELD            :      7.6829e+08  :     131.79  :      27.53
FP EMULATION        :          670.74  :     321.85  :      74.27
FOURIER             :           60272  :      68.55  :      38.50
ASSIGNMENT          :           86.56  :     329.38  :      85.43
IDEA                :           20373  :     311.60  :      92.52
HUFFMAN             :          8148.3  :     225.95  :      72.15
NEURAL NET          :          138.89  :     223.12  :      93.85
LU DECOMPOSITION    :          4006.5  :     207.56  :     149.88
=================================TEST COMPLETED=================================
```

Workflow and Evaluation

SGX-Suitability:

- ✗ Enclave transitions
- ✗✓ Suitable benchmarks
- ? Reliability



```
mahhouk@ssgx8:~/Documents/sgx-nbench$ ./app
====================Modified version of nbench for Intel SGX====================
TEST                : Iterations/sec.  : Old Index   : New Index
                    :                  : Pentium 90* : AMD K6/233*
--------------------:------------------:-------------:------------
NUMERIC SORT        :          1897.1  :      48.65  :      15.98
STRING SORT         :           61.34  :      27.41  :       4.24
BITFIELD            :      7.6829e+08  :     131.79  :      27.53
FP EMULATION        :          670.74  :     321.85  :      74.27
FOURIER             :           60272  :      68.55  :      38.50
ASSIGNMENT          :           86.56  :     329.38  :      85.43
IDEA                :           20373  :     311.60  :      92.52
HUFFMAN             :          8148.3  :     225.95  :      72.15
NEURAL NET          :          138.89  :     223.12  :      93.85
LU DECOMPOSITION    :          4006.5  :     207.56  :     149.88
=================================TEST COMPLETED=================================
```

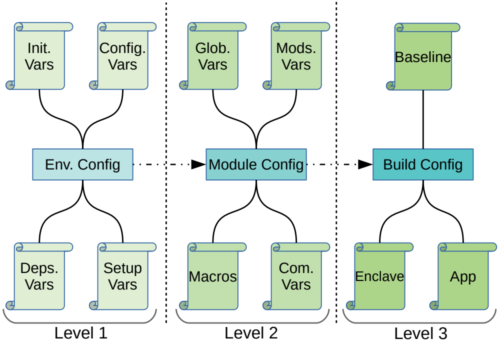
We need a dedicated application benchmark tool for Intel SGX
to reliably reproduce and compare research results

About SGXoMeter

- Macro-benchmark tool for Intel SGX
- Extensible, everything is a module
- No enclave transition during benchmark
- User friendly

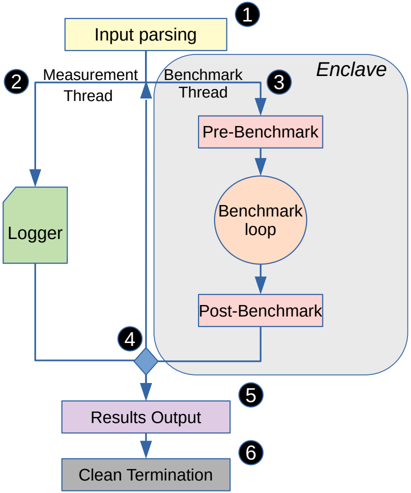
Framework Structure

- Level 1
  - Environment and dependencies setup
- Level 2
  - Modules selection and configuration
  - GUI option
- Level 3
  - Generation of binaries
  - Shared source files for multiple executables



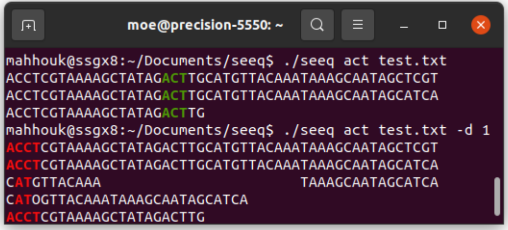
Workflow – Runtime Phase

1. Default values configuration
2. Measurement thread
3. Benchmark thread
4. Next module selection point
5. Result output
6. Clean up and terminate



Implemented Benchmark Modules

- **Seeq** module
  - DNA/RNA pattern matching algorithm[2]
  - CPU and memory heavy workload
  - Operates on security-sensitive data
  - Several flags available for different purposes:
    - Matching options
    - Format options
    - Misc options



```
mahhouk@ssgx8:~/Documents/seeq$ ./seeq act test.txt
ACCTCGTAAAAGCTATAGACTTGCATGTTACAAATAAAGCAATAGCTCGT
ACCTCGTAAAAGCTATAGACTTGCATGTTACAAATAAAGCAATAGCATCA
ACCTCGTAAAAGCTATAGACTTG
mahhouk@ssgx8:~/Documents/seeq$ ./seeq act test.txt -d 1
ACCTCGTAAAAGCTATAGACTTGCATGTTACAAATAAAGCAATAGCTCGT
ACCTCGTAAAAGCTATAGACTTGCATGTTACAAATAAAGCAATAGCATCA
CATGTTACAAA                       TAAAGCAATAGCATCA
CATOGTTACAAATAAAGCAATAGCATCA
ACCTCGTAAAAGCTATAGACTTG
```

[2] https://github.com/ezorita/seeq

Implemented Benchmark Modules

- **Intel SGX SSL**[3] based on **OpenSSL**
  - **RSA** module
    - Keypair generation
    - Encryption & decryption
    - Signing & verification
  - **SHA256** module
- Other cryptographic modules
  - Diffie Hellman
  - Elliptic curve, also combination with DH, DSA
  - Encryption & decryption using AES-GCM of the SGX SDK

OpenSSL
Cryptography and SSL/TLS Toolkit

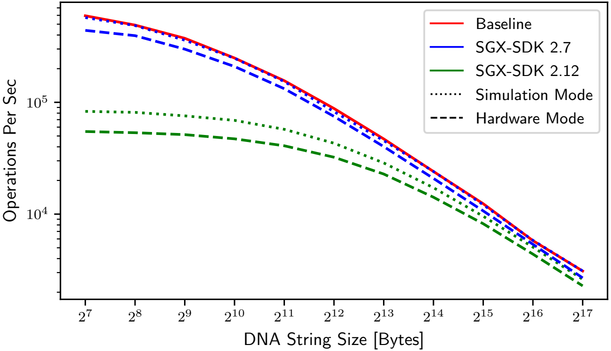
[3] https://github.com/intel/intel-sgx-ssl

Testbed Configuration

- SGX SDK versions 2.7 vs 2.12
- Baseline and SGX in hardware & simulation mode
  - The baseline runs the same benchmarks without any SGX primitives
- 10s Warm-up and 60s runtime
- System specification
  - Intel Xeon E-2176G @ 3.70GHz
  - 32 GB @ 2666MHz RAM
  - Ubuntu 18.04 LTS
  - CPU μcode `0xde`
  - SGX driver 2.11

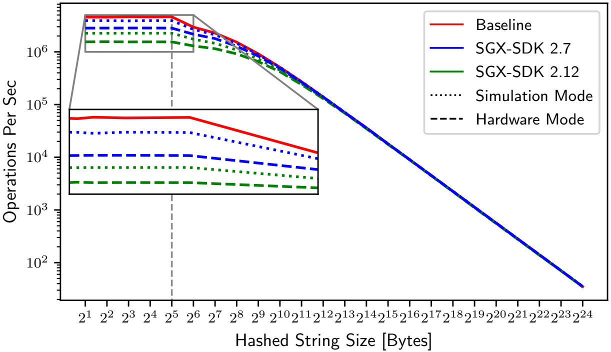
DNA/RNA Pattern Matching Algorithm

- Overhead compared to the baseline
  - SDK 2.7 up to ≈×1.3
  - SDK 2.12 up to ≈×10
- Overhead between the SDK versions
  - SDK 2.12 up to ≈×8 slower than SDK 2.7



Hashing with SHA256

- Overhead compared to the baseline
  - SDK 2.7 up to ≈×1.6
  - SDK 2.12 up to ≈×3
- Overhead between the SDK versions
  - SDK 2.12 up to ≈×1.8 slower than SDK 2.7

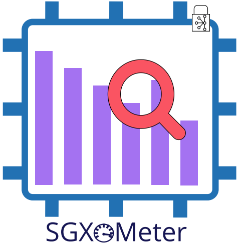


Upcoming Next

- More benchmark modules
  - Port of specific benchmarks from SGX-NBench
  - Different hardware-accelerated cryptographic algorithms
  - Other ports like OpenCV or QuickJS modules
- Support for other frameworks
  - Open-enclave
  - Graphene-SGX [Tsai et al., ATC '17]
- Extra features and better usability
  - Opting-in the enclave transitions in the measurements
  - Setting up the enclave's configuration from the GUI

Summary and Takeaways

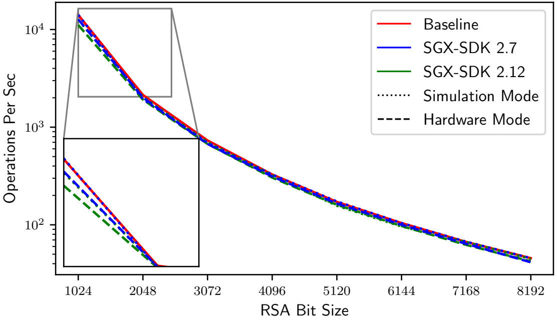
- Need for a practical and reliable SGX benchmark tool
- Identified flaws in SGX-NBench
  - No reasonable baseline
  - Unknown hardcoded default values
- Novel benchmark framework: SGXoMeter
  - Open-source[4]
  - High modularity
  - Reproducibility of results
  - Reasonable baseline configuration



[4] https://github.com/ibr-ds/SGXoMeter

Encryption & Decryption with RSA

- Overhead compared to the baseline
  - SDK 2.7 up to ≈×1.1
  - SDK 2.12 up to ≈×1.3
- Overhead between the SDK versions
  - SDK 2.12 up to ≈×1.14 slower than SDK 2.7



Internal Components

❶ Untrusted part

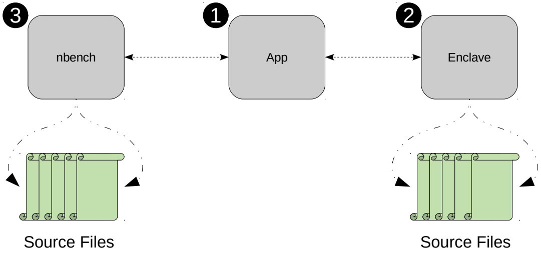
- Wrappers for ecalls
- Ocalls definitions

❷ Trusted part

- Benchmark programs' implementation
- Necessary buffers and helper functions

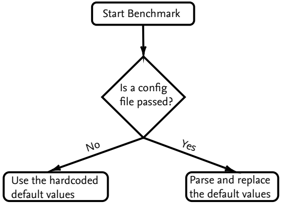
❸ Transition part

- Input parsing and pre/post-preparation
- Gather results, statistics calculation & output



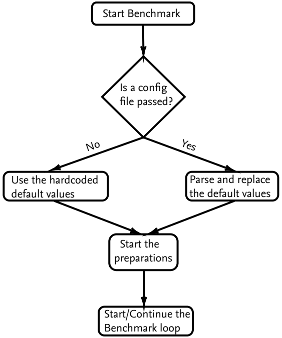
SGX-Nbench Workflow

1. Input parsing and setting global configuration
   - Default hard coded values or a configuration file


Start Benchmark
Is a config file passed?
No
Yes
Use the hardcoded default values
Parse and replace the default values

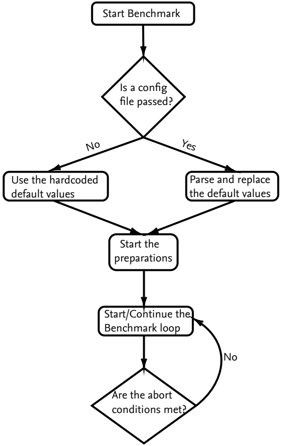
SGX-Nbench Workflow

1. Input parsing and setting global configuration
   - Default hard coded values or a configuration file
2. Executing every single benchmark including
   - Associated pre/post-preparations
   - Between [5, 30] executions per benchmark program
   - Actual execution loop is fixed by a time constraint (5s)
   - intermediate results in form of Iterations per second
   - 95% confidence-interval using student-t-distribution


Start Benchmark
Is a config file passed?
No
Yes
Use the hardcoded default values
Parse and replace the default values
Start the preparations
Start/Continue the Benchmark loop

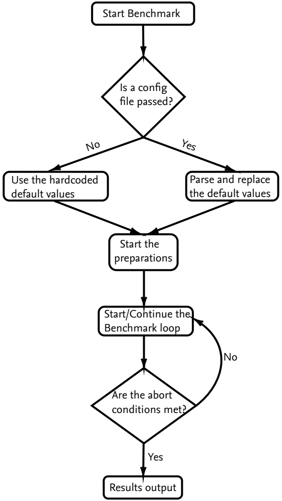
SGX-Nbench Workflow

1. Input parsing and setting global configuration
   - Default hard coded values or a configuration file
2. Executing every single benchmark including
   - Associated pre/post-preparations
   - Between $[5, 30]$ executions per benchmark program
   - Actual execution loop is fixed by a time constraint (5s)
   - intermediate results in form of Iterations per second
   - 95% confidence-interval using student-t-distribution
3. abort conditions are
   - 30 executions are reached
   - $\frac{CI}{\overline{R}} < 1\%$; CI:=Confidence-Interval, $\overline{R}$:=Results' mean value


Start Benchmark
Is a config file passed?
No
Yes
Use the hardcoded default values
Parse and replace the default values
Start the preparations
Start/Continue the Benchmark loop
No
Are the abort conditions met?

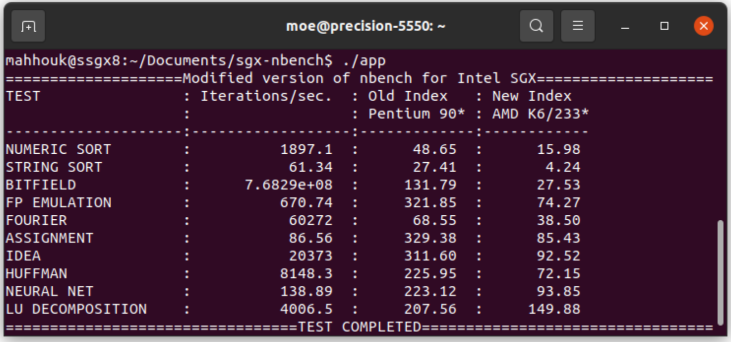
SGX-Nbench Workflow

1. Input parsing and setting global configuration
   - Default hard coded values or a configuration file
2. Executing every single benchmark including
   - Associated pre/post-preparations
   - Between $[5, 30]$ executions per benchmark program
   - Actual execution loop is fixed by a time constraint (5s)
   - intermediate results in form of Iterations per second
   - 95% confidence-interval using student-t-distribution
3. abort conditions are
   - 30 executions are reached
   - $\frac{CI}{\overline{R}} < 1\%$; CI:=Confidence-Interval, $\overline{R}$:=Results' mean value
4. Results output


Start Benchmark
Is a config file passed?
No
Yes
Use the hardcoded default values
Parse and replace the default values
Start the preparations
Start/Continue the Benchmark loop
Are the abort conditions met?
No
Yes
Results output

SGX-Nbench Workflow

4. Results output
   - Textual form in the console
   - Result's mean value
   - Comparison against two different machines



```
mahhouk@ssgx8:~/Documents/sgx-nbench$ ./app
====================Modified version of nbench for Intel SGX====================
TEST                : Iterations/sec.  : Old Index   : New Index
                    :                  : Pentium 90* : AMD K6/233*
--------------------:------------------:-------------:------------
NUMERIC SORT        :          1897.1  :      48.65  :      15.98
STRING SORT         :           61.34  :      27.41  :       4.24
BITFIELD            :      7.6829e+08  :     131.79  :      27.53
FP EMULATION        :          670.74  :     321.85  :      74.27
FOURIER             :           60272  :      68.55  :      38.50
ASSIGNMENT          :           86.56  :     329.38  :      85.43
IDEA                :           20373  :     311.60  :      92.52
HUFFMAN             :          8148.3  :     225.95  :      72.15
NEURAL NET          :          138.89  :     223.12  :      93.85
LU DECOMPOSITION    :          4006.5  :     207.56  :     149.88
=================================TEST COMPLETED=================================
```

Enclave Interface

```
enclave {
    from "sgx_tsgxssl.edl" import *;
    trusted {
        public void ecall_start_bench();
        public void ecall_pause_bench();
        public void ecall_stop_bench();
        public void ecall_run_bench(int test_id);
        public void ecall_set_config([user_check] uint64_t *ctr,
                                     [user_check] void *globalConfig );
    };
};
```

Benchmark Interface

```c
/* Pre-preparation function called before the benchmark loop */
void pre_custom_test(globalConfig_t *globalConfig);

/* Clean termination function after the benchmark loop */
void post_custom_test();

/* The actual benchmarked function */
int custom_test();
```