**Intel Software Guard Extensions** (**SGX**) is a set of [security-related](https://en.wikipedia.org/wiki/Computer_security) [instruction codes](https://en.wikipedia.org/wiki/Instruction_code) that are built into some modern [Intel](https://en.wikipedia.org/wiki/Intel) [central processing units](https://en.wikipedia.org/wiki/Central_processing_unit) (CPUs). They allow [user-level](https://en.wikipedia.org/wiki/User-level) as well as [operating system](https://en.wikipedia.org/wiki/Operating_system) code to define private regions of memory, called *enclaves*, whose contents are protected and unable to be either read or saved by any process outside the enclave itself, including processes running at higher [privilege levels](https://en.wikipedia.org/wiki/Privilege_level).[[1]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-1)[[2]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-2)

SGX involves [encryption](https://en.wikipedia.org/wiki/Encryption) by the CPU of a portion of memory. The enclave is decrypted [on the fly](https://en.wikipedia.org/wiki/On_the_fly) only within the CPU itself, and even then, only for code and data running from within the enclave itself.[[3]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-sd-3) The processor thus protects the code from being "spied on" or examined by other code.[[3]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-sd-3) The code and data in the enclave utilize a [threat model](https://en.wikipedia.org/wiki/Threat_model) in which the enclave is trusted but no process outside it can be trusted (including the [operating system](https://en.wikipedia.org/wiki/Operating_system) itself and any [hypervisor](https://en.wikipedia.org/wiki/Hypervisor)), and therefore all of these are treated as potentially hostile. The enclave contents are unable to be read by any code outside the enclave, other than in its encrypted form.[[3]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-sd-3). Applications running inside of SGX must be written to be side channel resistant as SGX does not protect against side channel measurement or observation.[[4]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-4)

SGX is designed to be useful for implementing secure [remote computation](https://en.wikipedia.org/wiki/Remote_computing), secure [web browsing](https://en.wikipedia.org/wiki/Web_browsing), and [digital rights management](https://en.wikipedia.org/wiki/Digital_rights_management) (DRM).[[5]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-5) Other applications include concealment of [proprietary algorithms](https://en.wikipedia.org/wiki/Proprietary_software) and of [encryption keys](https://en.wikipedia.org/wiki/Encryption_key).[[3]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-sd-3)

Details[[edit](https://en.wikipedia.org/w/index.php?title=Software_Guard_Extensions&action=edit&section=1)]

SGX was first introduced in 2015 with the sixth generation [Intel Core](https://en.wikipedia.org/wiki/Intel_Core) microprocessors based on the [Skylake](https://en.wikipedia.org/wiki/Skylake_(microarchitecture)) [microarchitecture](https://en.wikipedia.org/wiki/Microarchitecture).

Support for SGX in the CPU is indicated in [CPUID](https://en.wikipedia.org/wiki/CPUID) "Structured Extended feature Leaf", EBX bit 02,[[6]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-6) but its availability to applications requires [BIOS](https://en.wikipedia.org/wiki/BIOS)/[UEFI](https://en.wikipedia.org/wiki/Unified_Extensible_Firmware_Interface) support and opt-in enabling which is not reflected in CPUID bits. This complicates the feature detection logic for applications.[[7]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-7)

Emulation of SGX was added to an experimental version of the [QEMU](https://en.wikipedia.org/wiki/QEMU) system emulator in 2014.[[8]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-8) In 2015, researchers at the [Georgia Institute of Technology](https://en.wikipedia.org/wiki/Georgia_Institute_of_Technology) released an open-source simulator named "OpenSGX".[[9]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-9)

One example of SGX used in security was a demo application from [wolfSSL](https://en.wikipedia.org/wiki/WolfSSL)[[10]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-10) using it for cryptography algorithms.

Intel [Goldmont Plus](https://en.wikipedia.org/wiki/Goldmont_Plus) (Gemini Lake) microarchitecture also contains support for Intel SGX.[[11]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-11)

## Attacks[[edit](https://en.wikipedia.org/w/index.php?title=Software_Guard_Extensions&action=edit&section=2)]

### Prime+Probe attack**[**[**edit**](https://en.wikipedia.org/w/index.php?title=Software_Guard_Extensions&action=edit&section=3)**]**

On 27 March 2017 researchers at Austria's [Graz University of Technology](https://en.wikipedia.org/wiki/Graz_University_of_Technology) developed a proof-of-concept that can grab [RSA](https://en.wikipedia.org/wiki/RSA_(cryptosystem)) keys from SGX enclaves running on the same system within five minutes by using certain CPU instructions in lieu of a fine-grained timer to exploit [cache](https://en.wikipedia.org/wiki/Cache_memory) [DRAM](https://en.wikipedia.org/wiki/DRAM) side-channels.[[12]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-12)[[13]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-13) One countermeasure for this type of attack was presented and published by Daniel Gruss et al. at the [USENIX](https://en.wikipedia.org/wiki/USENIX) Security Symposium in 2017.[[14]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-14) Among other published countermeasures, one countermeasure to this type of attack was published on September 28, 2017, a compiler-based tool, DR.SGX,[[15]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-15) that claims to have superior performance with the elimination of the implementation complexity of other proposed solutions.

### Spectre-like attack**[**[**edit**](https://en.wikipedia.org/w/index.php?title=Software_Guard_Extensions&action=edit&section=4)**]**

*Main article:*[*Spectre (security vulnerability)*](https://en.wikipedia.org/wiki/Spectre_(security_vulnerability))

The LSDS group at Imperial College London showed a proof of concept that the [Spectre](https://en.wikipedia.org/wiki/Spectre_(security_vulnerability)) speculative execution security vulnerability can be adapted to attack the secure enclave.[[16]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-16) The [Foreshadow](https://en.wikipedia.org/wiki/Foreshadow_(security_vulnerability)) attack, disclosed in August 2018, combines speculative execution and buffer overflow to bypass the SGX.[[17]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-17)

### Enclave attack**[**[**edit**](https://en.wikipedia.org/w/index.php?title=Software_Guard_Extensions&action=edit&section=5)**]**

On 8 February 2019, researchers at Austria's [Graz University of Technology](https://en.wikipedia.org/wiki/Graz_University_of_Technology) published findings, which showed that in some cases it is possible to run malicious code from within the enclave itself.[[18]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-18) The exploit involves scanning through process memory, in order to reconstruct a payload, which can then run code on the system. The paper claims that due to the confidential and protected nature of the enclave, it is impossible for [Antivirus software](https://en.wikipedia.org/wiki/Antivirus_software) to detect and remove malware residing within it. However, since modern anti-malware and antivirus solutions monitor system calls, and the interaction of the application with the operating system, it should be possible to identify malicious enclaves by their behavior, and this issue is unlikely to be a concern for state-of-the-art antiviruses. Intel issued a statement, stating that this attack was outside the threat model of SGX, that they cannot guarantee that code run by the user comes from trusted sources, and urged consumers to only run trusted code.[[19]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-19)

### MicroScope replay attack**[**[**edit**](https://en.wikipedia.org/w/index.php?title=Software_Guard_Extensions&action=edit&section=6)**]**

There is a proliferation of [Side-channel attack](https://en.wikipedia.org/wiki/Side-channel_attack) plaguing modern computer architecture. Many of these attacks measure slight, nondeterministic variations in the execution of some code, so the attacker needs many, possibly tens of thousands, of measurements to learn secrets. However, the Microscope attack allows a malicious OS to replay code an arbitrary number of times regardless of the programs actual structure, enabling dozens of side-channel attacks.[[20]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-20)

### Plundervolt**[**[**edit**](https://en.wikipedia.org/w/index.php?title=Software_Guard_Extensions&action=edit&section=7)**]**

Security researchers were able to inject timing specific faults into execution within the enclave, resulting in leakage of information. The attack *can* be executed remotely, but requires access to the privileged control of the processor's voltage and frequency.[[21]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-21)

### LVI**[**[**edit**](https://en.wikipedia.org/w/index.php?title=Software_Guard_Extensions&action=edit&section=8)**]**

*Main article:*[*Load value injection*](https://en.wikipedia.org/wiki/Load_value_injection)

Load Value Injection[[22]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-22)[[23]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-23) injects data into a program aiming to replace the value loaded from memory which is then used for a short time before the mistake is spotted and rolled back, during which LVI controls data and control flow.

### SGAxe**[**[**edit**](https://en.wikipedia.org/w/index.php?title=Software_Guard_Extensions&action=edit&section=9)**]**

SGAxe,[[24]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-24) a SGX vulnerability, extends a [speculative execution attack](https://en.wikipedia.org/wiki/Speculative_execution#Security_vulnerabilities) on cache [[25]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-25), leaking content of the enclave. This allows an attacker to access private CPU keys used for remote attestation[[26]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-26). In other words, a threat actor can bypass Intel's countermeasures to breach SGX's enclaves confidentiality. The [SGAxe attack](https://www.hackreports.com/sgaxe-crosstalk-attacks-intel-sgx-vulnerability/) is carried out by extracting attestation keys from SGX's private quoting enclave, that are signed by Intel. The attacker can then masquerade as legitimate Intel machines by signing arbitrary SGX attestation quotes[[27]](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_note-27).

## See also[[edit](https://en.wikipedia.org/w/index.php?title=Software_Guard_Extensions&action=edit&section=10)]

* [Intel MPX](https://en.wikipedia.org/wiki/Intel_MPX)
* [Spectre-NG](https://en.wikipedia.org/wiki/Spectre-NG)
* [Trusted execution environment](https://en.wikipedia.org/wiki/Trusted_execution_environment) (TEE)

See also[[edit](https://en.wikipedia.org/w/index.php?title=Software_Guard_Extensions&action=edit&section=10)]

* [Intel MPX](https://en.wikipedia.org/wiki/Intel_MPX)
* [Spectre-NG](https://en.wikipedia.org/wiki/Spectre-NG)
* [Trusted execution environment](https://en.wikipedia.org/wiki/Trusted_execution_environment) (TEE)

References[[edit](https://en.wikipedia.org/w/index.php?title=Software_Guard_Extensions&action=edit&section=11)]

* 1. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-1) [*"Intel SGX for Dummies (Intel SGX Design Objectives)"*](https://software.intel.com/en-us/blogs/2013/09/26/protecting-application-secrets-with-intel-sgx)*. intel.com. 2013-09-26.*
  2. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-2) *johnm (2017-08-08).*[*"Properly Detecting Intel® Software Guard Extensions (Intel® SGX) in Your Applications"*](https://software.intel.com/en-us/articles/properly-detecting-intel-software-guard-extensions-in-your-applications)*. software.intel.com. Retrieved 2019-02-15.*
  3. ^ [Jump up to:***a***](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-sd_3-0) [***b***](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-sd_3-1) [***c***](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-sd_3-2) [***d***](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-sd_3-3) [*"Researchers Use Intel SGX To Put Malware Beyond the Reach of Antivirus Software - Slashdot"*](https://it.slashdot.org/story/19/02/12/222254/researchers-use-intel-sgx-to-put-malware-beyond-the-reach-of-antivirus-software?sbsrc=developers)*. it.slashdot.org.*
  4. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-4) [*"Intel SGX and Side-Channels"*](https://software.intel.com/en-us/articles/intel-sgx-and-side-channels)*. intel.com. 2020-02-28.*
  5. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-5) [*"Intel SGX Details"*](https://software.intel.com/en-us/sgx/details)*. intel.com. 2017-07-05.*
  6. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-6) [Intel Architecture Instruction Set Extensions Programming Reference](https://software.intel.com/sites/default/files/managed/07/b7/319433-023.pdf#page=36), Intel, AUGUST 2015, page 36 "Structured Extended feature Leaf EAX=07h, EBX Bit 02: SGX"
  7. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-7) [*"Properly Detecting Intel Software Guard Extensions in Your Applications"*](https://software.intel.com/en-us/articles/properly-detecting-intel-software-guard-extensions-in-your-applications)*. intel.com. 2016-05-13.*
  8. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-8) [*"Intel SGX Emulation using QEMU"*](https://tc.gtisc.gatech.edu/bss/2014/l/final/pjain43.pdf)*(PDF). tc.gtisc.gatech.edu. Retrieved 2018-11-02.*
  9. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-9) [*"sslab-gatech/opensgx"*](https://github.com/sslab-gatech/opensgx)*.*[*GitHub*](https://en.wikipedia.org/wiki/GitHub)*. Retrieved 2016-08-15.*
  10. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-10) [*"wolfSSL At IDF"*](https://www.wolfssl.com/wolfssl-at-idf/)*. wolfssl. 2016-08-11.*
  11. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-11) [*"Intel® Pentium® Silver J5005 Processor"*](https://ark.intel.com/content/www/us/en/ark/products/128984/intel-pentium-silver-j5005-processor-4m-cache-up-to-2-80-ghz.html)*. Retrieved 2020-07-10.*
  12. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-12) *Chirgwin, Richard (March 7, 2017).*[*"Boffins show Intel's SGX can leak crypto keys"*](https://www.theregister.co.uk/2017/03/07/eggheads_slip_a_note_under_intels_door_sgx_can_leak_crypto_keys/)*. The Register. Retrieved 1 May 2017.*
  13. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-13) *Schwarz, Michael; Weiser, Samuel; Gruss, Daniel; Maurice, Clémentine; Mangard, Stefan (2017). "Malware Guard Extension: Using SGX to Conceal Cache Attacks".*[*arXiv*](https://en.wikipedia.org/wiki/ArXiv_(identifier))*:*[*1702.08719*](https://arxiv.org/abs/1702.08719)*[*[*cs.CR*](https://arxiv.org/archive/cs.CR)*].*
  14. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-14) [*"Strong and Efficient Cache Side-Channel Protection using Hardware Transactional Memory"*](https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-gruss.pdf)*(PDF). USENIX. 2017-08-16.*
  15. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-15) *Brasser, Ferdinand; Capkun, Srdjan; Dmitrienko, Alexandra; Frassetto, Tommaso; Kostiainen, Kari; Müller, Urs; Sadeghi, Ahmad-Reza (2017-09-28). "DR.SGX: Hardening SGX Enclaves against Cache Attacks with Data Location Randomization".*[*arXiv*](https://en.wikipedia.org/wiki/ArXiv_(identifier))*:*[*1709.09917*](https://arxiv.org/abs/1709.09917)*.*[*doi*](https://en.wikipedia.org/wiki/Doi_(identifier))*:*[*10.1145/3359789.3359809*](https://doi.org/10.1145%2F3359789.3359809)*.*[*S2CID*](https://en.wikipedia.org/wiki/S2CID_(identifier))[*19364841*](https://api.semanticscholar.org/CorpusID:19364841)*.*
  16. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-16) [*Sample code demonstrating a Spectre-like attack against an Intel SGX enclave.*](https://github.com/lsds/spectre-attack-sgx)
  17. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-17) *Peter Bright - Jul 10, 2018 9:00 pm UTC (2018-07-10).*[*"New Spectre-like attack uses speculative execution to overflow buffers"*](https://arstechnica.com/gadgets/2018/07/new-spectre-like-attack-uses-speculative-execution-to-overflow-buffers/)*. Ars Technica. Retrieved 2018-11-02.*
  18. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-18) *Schwarz, Michael; Weiser, Samuel; Gruss, Daniel (2019-02-08). "Practical Enclave Malware with Intel SGX".*[*arXiv*](https://en.wikipedia.org/wiki/ArXiv_(identifier))*:*[*1902.03256*](https://arxiv.org/abs/1902.03256)*[*[*cs.CR*](https://arxiv.org/archive/cs.CR)*].*
  19. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-19) *Bright, Peter (2019-02-12).*[*"Researchers use Intel SGX to put malware beyond the reach of antivirus software"*](https://arstechnica.com/gadgets/2019/02/researchers-use-intel-sgx-to-put-malware-beyond-the-reach-of-antivirus-software/)*. Ars Technica. Retrieved 2019-02-15.*
  20. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-20) *Skarlatos, Dimitrios; Yan, Mengjia; Gopireddy, Bhargava; Sprabery, Read; Torrellas, Josep; Fletcher, Christopher W. (2019).*[*"MicroScope: enabling microarchitectural replay attacks"*](http://dl.acm.org/citation.cfm?doid=3307650.3322228)*. Proceedings of the 46th International Symposium on Computer Architecture - ISCA '19. Phoenix, Arizona: ACM Press: 318–331.*[*doi*](https://en.wikipedia.org/wiki/Doi_(identifier))*:*[*10.1145/3307650.3322228*](https://doi.org/10.1145%2F3307650.3322228)*.*[*ISBN*](https://en.wikipedia.org/wiki/ISBN_(identifier))[*978-1-4503-6669-4*](https://en.wikipedia.org/wiki/Special:BookSources/978-1-4503-6669-4)*.*
  21. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-21) [*"Plundervolt steals keys from cryptographic algorithms"*](https://www.rambus.com/blogs/plundervolt-steals-keys-from-cryptographic-algorithms/)*. Rambus Blog. 2019-12-11. Retrieved 2020-03-20.*
  22. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-22) [*"LVI: Hijacking Transient Execution with Load Value Injection"*](https://lviattack.eu/)*. lviattack.eu. Retrieved 2020-03-12.*
  23. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-23) [*"Load Value Injection"*](https://software.intel.com/security-software-guidance/software-guidance/load-value-injection)*. software.intel.com. Retrieved 2020-03-12.*
  24. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-24) [*"SGAxe: How SGX Fails in Practice"*](https://sgaxe.com/)*.*
  25. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-25) [*"CacheOut: Leaking Data on Intel CPUs via Cache Evictions"*](https://cacheoutattack.com/)*.*
  26. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-26) [*"Towards Formalization of Enhanced Privacy ID (EPID)-based Remote Attestation in Intel SGX"*](https://www.researchgate.net/publication/342833256_Towards_Formalization_of_Enhanced_Privacy_ID_EPID-based_Remote_Attestation_in_Intel_SGX)*.*
  27. [**^**](https://en.wikipedia.org/wiki/Software_Guard_Extensions#cite_ref-27) [*"SGAxe & CrossTalk Attacks: New Intel SGX Vulnerability Leaks Data"*](https://www.hackreports.com/sgaxe-crosstalk-attacks-intel-sgx-vulnerability/)*. Hack Reports. 2020-06-12. Retrieved 2020-06-12.*

External links[[edit](https://en.wikipedia.org/w/index.php?title=Software_Guard_Extensions&action=edit&section=12)]

* [Intel Software Guard Extensions (Intel SGX)](https://software.intel.com/en-us/isa-extensions/intel-sgx) / ISA Extensions, Intel
  + [Intel Software Guard Extensions (Intel SGX) Programming Reference](https://software.intel.com/sites/default/files/managed/48/88/329298-002.pdf), Intel, October 2014
  + [IDF 2015 - Tech Chat: A Primer on Intel Software Guard Extensions](https://software.intel.com/sites/default/files/managed/3e/b9/SF15_ISGC003_81_SGX_DL_100_small.pdf), Intel (poster)
  + [ISCA 2015 tutorial slides for Intel SGX](https://software.intel.com/sites/default/files/332680-002.pdf), Intel, June 2015
* McKeen, Frank, et al. (Intel), [Innovative Instructions and Software Model for Isolated Execution](http://web.cs.wpi.edu/~guttman/cs557_website/papers/trusted_computing/sgx1.pdf)// Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy. ACM, 2013.
* [Joanna Rutkowska](https://en.wikipedia.org/wiki/Joanna_Rutkowska), [Thoughts on Intel's upcoming Software Guard Extensions (Part 1)](http://theinvisiblethings.blogspot.ru/2013/08/thoughts-on-intels-upcoming-software.html), August 2013
* [SGX: the good, the bad and the downright ugly](https://www.virusbtn.com/virusbulletin/archive/2014/01/vb201401-SGX) / Shaun Davenport, Richard Ford (Florida Institute of Technology) / Virus Bulletin, 2014-01-07
* Victor Costan and Srinivas Devadas, [Intel SGX Explained](https://eprint.iacr.org/2016/086.pdf), January 2016.
* [wolfSSL](https://github.com/wolfSSL/wolfssl), October 2016.
* [The Security of Intel SGX for Key Protection and Data Privacy Applications](https://cdn2.hubspot.net/hubfs/1761386/security-of-intelsgx-key-protection-data-privacy-apps.pdf) / Professor Yehuda Lindell (Bar Ilan University & Unbound Tech), January 2018
* [Intel SGX Technology and the Impact of Processor Side-Channel Attacks](https://fortanix.com/blog/2020/03/intel-sgx-technology-and-the-impact-of-processor-side-channel-attacks/), March 2020