A Technical Analysis On The Feasibility Of Implementing Location Verification In The 'Chip Security Act'

September 26th, 2025

Hardware enabled AI governance mechanisms have been proposed as a means for deterring and tracking the diversion of high end AI compute. Achieving this will require combining remote attestation techniques with new technology and infrastructure. These new mechanisms will be the target of sustained research to break them by motivated and technically capable adversaries. We describe the types of components that would be required for building one potential solution in enough detail to consider the costs, timelines, and vulnerability to attack. **We find that while the cost for building this system is low in relative terms, it requires access to cryptographic primitives and operations performed on chip that are not available today, but even with these new capabilities the system is unlikely to withstand persistent research efforts to defeat it by a capable adversary.**

The Department of Commerce's Bureau of Industry and Security (BIS) remains the primary agency responsible for administering and enforcing export controls on advanced semiconductors, but it operates with limited investigative and enforcement capacity. As a result policymakers have focused on preventing the diversion of AI chips and limiting high end compute access via hardware enabled governance mechanisms. Since early 2024, the policy environment has shifted considerably: Chinese firms such as DeepSeek have made rapid advances in large scale model development, a new U.S. administration has reframed the strategic objectives of AI export controls, and Congress has introduced the Chip Security Act as a legislative means for embedding technical safeguards at the hardware level as one option for enforcing them.

The proposed Chip Security Act seeks to *"prevent diversion of advanced chips to America's adversaries and protect U.S. product integrity."* It covers integrated circuits subject to ECCNs 3A090 and 4A090 (and successor classifications). For the casual reader, these ECCN's cover high end GPUs used for AI training and inference workloads above a

certain performance specification. First, the Act proposes location verification as one potential solution of covered chips, along with mandatory reporting if devices are detected in unauthorized jurisdictions. Second, it requires the study of additional enhanced chip security mechanisms for detecting and deterring smuggling of covered chips while preserving confidentiality.

Hardware level AI governance mechanisms provide a potential solution for selectively constraining AI capabilities, shaping distribution, and partially advancing policy goals similar to those defined in the now defunct AI diffusion rules. Embedding enforcement directly within silicon offers remote programmability and precise compliance controls that can scale across global deployments. The threat model for this use case is complex and demanding: protecting millions of individual assets (AI chips) possibly under physical control by a sophisticated nation state threat actor with near unlimited resources and technical capabilities. If these security controls are defeated in order to fool the auditing and oversight mechanism, the result is not simply a technical failure but could undermine enforcement credibility and ultimately national security policy objectives.

This paper focuses on defining a secure design that may not be possible to implement today without chip designers making some cryptographic primitives available via API in their on chip root of trust. Establishing and sustaining this level of protection is technically difficult and comes with an ongoing financial cost. Long term resilience will require rigorous secure hardware and firmware design, ongoing operational security investments, systematic use of cryptographic verification protocols, and comprehensive supply chain integrity measures. Without this foundation and ongoing commitment, even well designed hardware governance features are unlikely to remain reliable in the face of persistent and sophisticated adversaries. We briefly explore alternative solutions that offer fewer security guarantees but may provide pragmatic and acceptable solutions to policy makers.

Location Verification

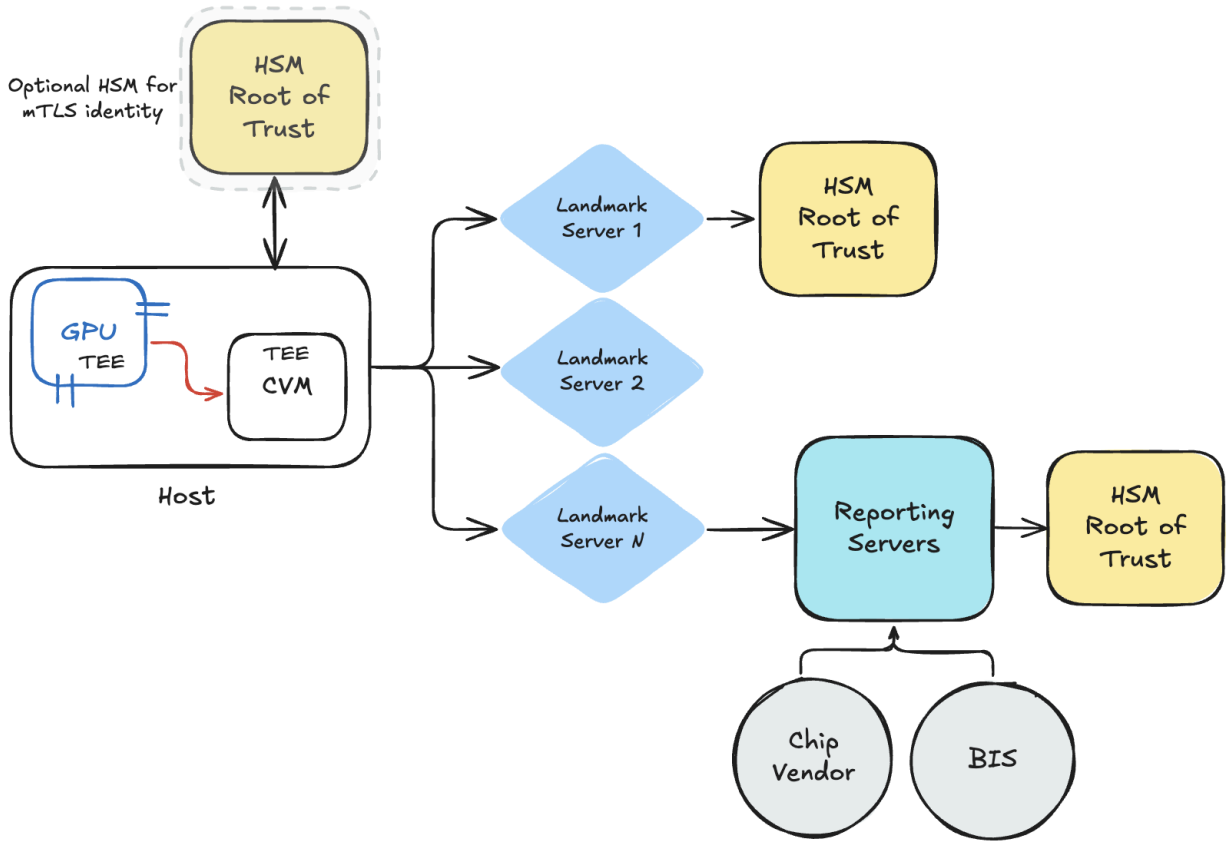
The first requirement of the Chip Security Act is a security mechanism such as the ability to verify where a chip is physically located. This has the benefit of knowing where the chip is not located, as well as when and where it may have been moved to. It is assumed that the intent of this control is to ensure the chip designer and the U.S. Government can verify that chip shipments remain at the customer locations they are reported to be at. Implementing this control may sound straightforward, but in practice it is complex. Various timing based ping schemes have been proposed in which several trusted "landmark" servers, deployed to strategic geographic locations around the world, would verify round trip times (RTT) to chips hosted in data centers at customer locations. This is based on the principle that chips located in geographic locations further away from the landmark servers would result in a longer RTT. These location measurements may not be precise, but they are accurate enough to roughly assess within a tens to hundreds of miles radius where the chip may be located. Even without precise geolocation, these RTT measurements can be used in a trust on first use (TOFU) way where the first few RTT measurements are considered the baseline and future RTT measurements are compared against them to determine if chips have moved. This capability has the potential to reduce manual investigations of chip diversions to restricted countries. Policymakers should note that Location Verification System (LVS) accuracy depends heavily on the diversity and trustworthiness of the global landmark infrastructure and that it can be degraded by adversarial techniques such as traffic relays or network path spoofing. As a result, LVS should be viewed as a coarse grained but useful enforcement signal, not as a foolproof tool or method for geolocation with pinpoint accuracy.

High Level System Design

In order to build this system the chips must first include a cryptographic hardware module on an on-die root of trust where secrets can be stored or optionally derived through a Physical Uncloneable Function (PUF). These modules must also support basic cryptographic operations such as symmetric key algorithms, HMAC generation and verification, and secure random number generation. These operations enable

encryption, decryption, and integrity verification of both data and code, and enforce a secure boot chain. All of this functionality runs inside the AI chip Trusted Execution Environment (TEE), which is already supported in most widely deployed advanced AI chips.

When a chip with this capability is paired with a CPU that supports its own TEE, sometimes referred to as a secure enclave, the two environments can establish a trusted cryptographic channel. The CPU side of the system is known as a Confidential Virtual Machine (CVM). The operating system (OS) inside the CVM resembles a standard OS but runs with encrypted memory that is inaccessible to untrusted code, unprivileged users, even those running in the server host operating system above where standard workloads are executed. The CVM itself has a secure boot chain verified by the CPU root of trust and can be launched stateless, with no access to writeable disks. The CVM is a critical component because it communicates directly with the AI chip TEE, and its primary purpose is to forward signed payloads to and from the chip itself. Once securely booted, it can present a cryptographically signed boot attestation to a networked Hardware Security Module (HSM) to obtain a short lived certificate for creating a mutual TLS (mTLS) tunnel to landmark servers. Storing the root key for mTLS inside an HSM protects it from theft in the event the CVM is compromised and allows for flexible revocation. This HSM can be considered an optional defense in depth control. Excluding it simplifies the architecture and lowers cost for compliant deployers, but at the expense of reducing secure isolation for mTLS key storage. In this alternative scenario this key would be generated within the CPU TEE or CVM itself. Landmark and reporting servers would run their own stateless CVMs within a CPU TEE and communicate with optional HSM clusters for secure key storage, and other cryptographic operations.

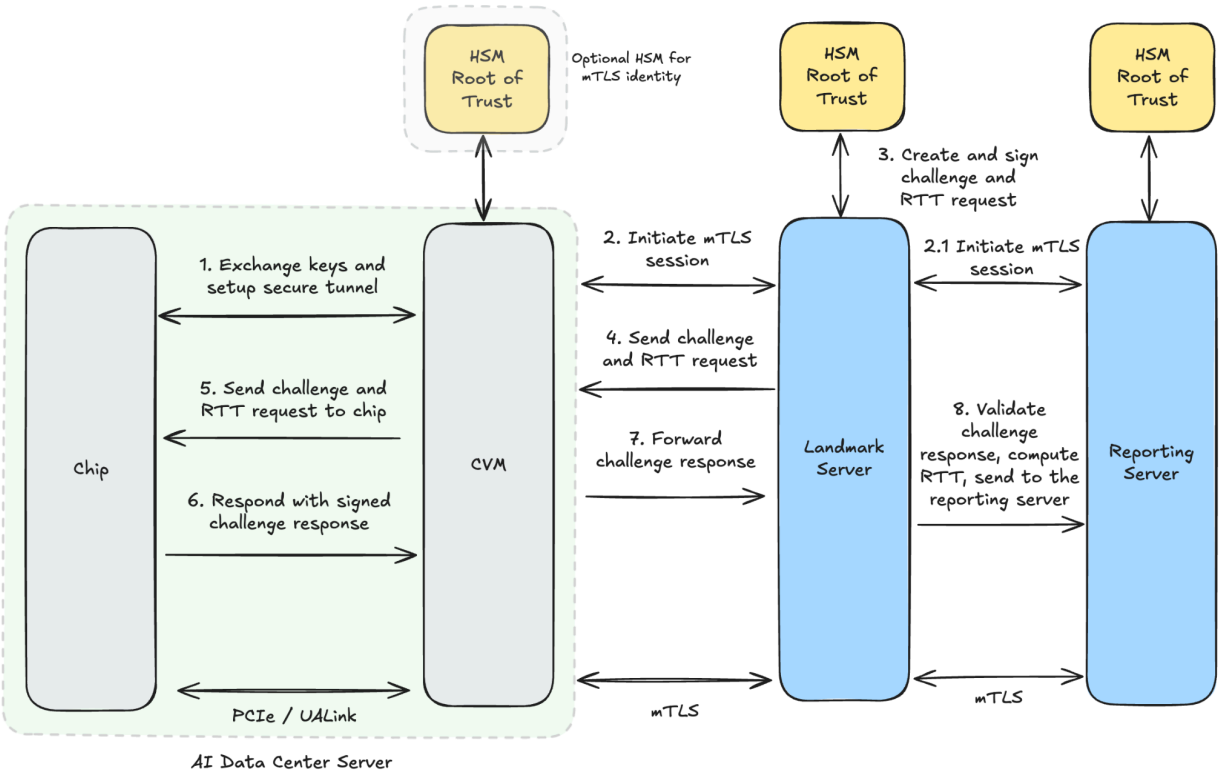


High Level LVS

The diagram above is simplified in order to illustrate the core components of the system. For a secure and reliable implementation, however, the system must account for dozens to hundreds of geographically distributed landmark servers and tens of millions of AI chips. Each deployment of landmark servers and data centers housing these chips would require access to dedicated, high-availability HSM clusters. Reporting servers would also need redundancy and high availability. While this scale is comparable to existing distributed system deployments managed by hyperscalers, it is still non-trivial to implement, operate, and maintain. A potential alternative to these reporting servers might be a cryptographic append-only transparency log (more commonly known and implemented as a blockchain) through which multiple systems could host in a p2p manner. Note, this alternative is not explored further in this paper.

Location Verification Protocol

In order for an LVS to function, the protocol between chips and landmark servers must be carefully designed. High level descriptions of this system often suggest that landmark servers would directly ping chips to calculate round-trip time (RTT). This approach would require exposing network interfaces on the CVM directly to the internet, which would unnecessarily increase the attack surface for adversaries to probe and attack. A more secure design is for the CVM to create a mutual TLS (mTLS) tunnel to one or more landmark servers. The landmark server then issues a cryptographic challenge and RTT request through this channel. The challenge message would contain fields such as a timestamp, a unique landmark ID, and a nonce to prevent replay or precomputation by a compromised AI chip key. The landmark would sign the challenge using keys protected by its HSM, rather than relying on those generated within the CVM running within its CPU TEE. After verifying the challenge, the AI chip responds with a signed message that the CVM forwards back to the landmark. The landmark then verifies the AI chip signature, computes the RTT based on its own authoritative timestamps, and records the result.



High Level LVS Protocol

The protocol outlined above is a simplified example that has not been independently verified. It is included only to demonstrate the types of sensitive cryptographic primitives and message exchanges required to construct a secure system end to end. There are additional technical challenges that are not reflected in the example flow. For instance, the AI chip connected CVM must know which landmark servers to contact before an mTLS tunnel can be established. This can be addressed by the chip designer securely distributing and updating the list of valid landmark servers on a recurring basis. Even so, it illustrates that the design of such a protocol must consider issues of scalability, resilience, and manageability in order to support secure deployments of AI chips that would number in the millions. This would also affect the cost to design the system considerably.

Threat Model

Before diving into the technical security challenges it is useful to frame this system through the lens of a threat model. For this paper I have chosen the STRIDE model which is an acronym for "**S**poofing, **T**ampering, **R**epudiation, **I**nformation disclosure, **D**enial of service and **E**levation of privilege". This model breaks down our system into 6 distinct categories through which we can build a list of the attacks on assets within the system and their mitigations.

The primary assets in this threat model are the AI chip, CVM, the landmark servers, and the reporting servers. The threat actor is assumed to be an advanced nation state with highly sophisticated offensive cyber capabilities. For the purpose of analysis, we assume this adversary has virtually unlimited resources, including human and signals intelligence capacity, and may obtain physical access to AI chips, CVMs, landmark servers, and reporting servers.

Category	Attack	Mitigation
Spoofing	• Impersonation of any asset or actor • Forging messages or logs from any asset • Replaying messages or logs from any asset	• Each asset must have a cryptographically verifiable identity and messages signed and verified with a hardware root of trust • All protocol messages should include anti-replay mechanisms such as a nonce
Tampering	• Firmware or software rollback • Hardware modification • Modifying messages or logs from any asset either through network Man In The Middle (MITM), PCI(e) bus, host or CVM compromise through arbitrary code execution or a supply chain attack.	• Secure boot and anti-rollback protections • Disable debug fuses and other hardware debug mechanisms • Each asset must have a cryptographically verifiable identity and messages signed and verified with a hardware root of trust
Repudiation	• Dropping messages such as failed challenge requests or audit logs • Claiming a chip did or did not send a	• Audit logs for all successful and unsuccessful cryptographic operations • Public append-only logging / Merkle trees

	message or produce an audit log	
Information Disclosure	• Leaking or exposure of cryptographic secrets • Over collection of system identifiers or other confidential information • Side channels via observation of network protocols and system behavior	• Physically hardened TEE/HSM, networking and data center infrastructure • Segment or otherwise obscure LVS results based on Reporting server user privilege level • Network traffic shaping • Least privilege access controls
Denial of Service	• LVS protocol based resource consumption • Network path degradation or spoofing	• API rate limiting and quotas • Early authentication of protocols
Elevation of Privilege	• Compromise of highly trusted hardware roots of trust in TEE, or HSM • Escape or bypass of security boundaries (e.g. CVM into chip firmware)	• Least privilege access controls • Memory safe languages and runtimes • Attack surface reduction • Exploit mitigations (ASLR, CFI etc)

Example Location Verification Threat Model

Designing a complete LVS is a complex undertaking. While the threat model described here should not be viewed as comprehensive or exhaustive, it should be considered a starting point for analyzing a subset of the system. This model can also be extended to identify the broader security requirements of a fully designed and specified system.

Technical Security Challenges

Embedding effective hardware enabled governance policies over AI compute requires a strong security design across all components including those that only support that monitoring and enforcement. It is not sufficient to mandate in policy that these systems are secure. The technical solution must be designed and implemented carefully, down to the last line of code. If even a single component is vulnerable, an adversary may be able to bypass or disable the system and thereby render it ineffective as a detection and enforcement mechanism. Adversaries will likely have unrestricted physical access to a

large number of deployed GPUs. Many researchers and advanced persistent threat (APT) groups have repeatedly demonstrated their ability to discover and exploit complex vulnerabilities in low-level software and firmware. **Even with security controls in place, the probability of vulnerabilities existing in any LVS implementation is high, and the probability that a technically sophisticated adversary will eventually discover and exploit those vulnerabilities is also high.**

- On-die chip security mechanisms such as TEEs should ideally include physical seals or sensors that trigger zeroization of firmware upon detection of tampering. They should be hardened against decapping and glitching attacks, and they should include a Physical Uncloneable Function (PUF) on-die that can be used as a Key Derivation Function (KDF) during power on. While not strictly necessary, this significantly increases the difficulty of extracting cryptographic keys and prevents attackers from scaling successful physical attacks across thousands of devices. However, it is likely to raise the cost of hardware design and fabrication.
- Each critical asset in the system should implement a secure boot mechanism that enforces strict cryptographic signature checks on firmware and operating system images to prevent the introduction of malicious or vulnerable code. This verified boot chain must include anti-rollback protections and be designed to withstand side channel and microcode level attacks from untrusted workloads that may be running on the server. On the AI chip side it should be assumed that adversaries can achieve close execution proximity to the hardware, as common AI inference workloads often involve generating arbitrary code from untrusted natural language prompts. In these systems arbitrary code execution influenced by these untrusted prompts is a feature. In poorly designed agentic inference systems, a remote attacker could use this pathway to attempt a compromise of the LVS system meant to gather location data from the same AI chip.
- Firmware and lower level components found in hardware roots of trust are typically written in memory unsafe languages such as C and C++. This choice is often driven by the need for low memory consumption, low latency, and high performance in resource constrained environments. These languages provide

advantages such as small binary size, a largely deterministic runtime, direct access to hardware operations without abstraction, easy integration with assembly code, and precise control of memory layout. The drawback is that they provide no memory safety guarantees, which frequently results in exploitable vulnerabilities. Firmware for these embedded architectures is often built with non standard toolchains, which results in binaries that ship without common exploit mitigations. Such mitigations, however, should be enabled on these firmware images, including Control Flow Integrity (CFI), Memory Tagging Extension (MTE), and stack canaries. These technical attributes and challenges aside, there is little incentive for companies to spend scarce resources rewriting existing functional code when the challenge of training engineers on newer memory safe languages with similar attributes is not free in either dollars or opportunity costs.

- Higher level operating systems, such as those that may run on the CVM, typically expose a large attack surface and include language runtimes like Javascript or Python that are not formally verified. Although the CVM is designed to be stateless, vulnerabilities in it could still allow an attacker to gain a temporary foothold in this privileged component, even if the code is only resident in memory. The CVM should not be able to extract cryptographic secrets from the chip, but it must communicate with privileged interfaces on the AI chip. This creates the possibility of executing privileged operations and of privilege escalation through shared memory buffers that are also operated on by AI chip firmware. A compromised CVM could also interfere with the update process by dropping or denying firmware updates, leaving the chip in a perpetually vulnerable state. Mitigating this can be achieved using the LVS protocol by enforcing a minimum firmware version in its challenge messages.
- Securing the software supply chain and managing vulnerabilities is challenging even in the most well orchestrated infrastructures. Maintaining accurate and up to date registries of code provenance and assessing the trustworthiness of third party code remains an unsolved problem. Although the CVM may be stateless

and its components cryptographically signed, if it unknowingly incorporates malware from an untrusted source, the integrity of that component is compromised. These risks are compounded when a product vendor and the organization deploying the product have different threat models and compliance expectations. Addressing known CVEs within short time windows is necessary to maintain the security of any AI governance system, but this requires complex vendor testing and may introduce downtime when updates are deployed.

- Landmark servers must be strategically deployed at data centers across the world. They require low latency access to high availability HSM clusters and should be reduced to the bare minimum functionality needed to support LVS. At the same time, they must be securely administered and operated remotely. Strong authentication and authorization are essential because adversary access could be used to spoof or tamper with RTT measurements. Logs from these systems, especially those generated by authentication events or system failures, should be written remotely to append only Merkle trees by default and actively monitored. These servers must also be hosted on secure networks where malicious network path protocols (e.g. spoofable BGP routes) and other kinds of denial of service attacks are mitigated to ensure they can report accurate timing information to reporting servers. Administration of these systems should be restricted to trusted personnel and/or U.S. citizens.
- Reporting servers may be the most targeted component in the entire system since final determinations of violations occur here and they have access to raw data about deployed AI systems worldwide. These servers likely need to expose a web interface for auditors to log in and view anomalies and audit findings. This interface must be hardened against common web application vulnerabilities such as SQL injection, XSS, CSRF, and SSRF. Reducing attack surface on these systems is important but challenging because their purpose is to serve sensitive data to end users and auditors. They should be heavily segmented and isolated from other systems operated by the chip vendor, with fine grained access controls and permissions. Logs from these systems, especially those related to

authentication events and system failures, should be actively monitored. Access to these systems and the data they store should be restricted to specific personnel, similar to requirements for legal collection systems.

- All cryptographic primitives used across the system, including those in TEEs, on-die roots of trust, and networked HSMs, should be revocable. Revocability provides a lever for denying access to the overall infrastructure even if the AI chip functionality itself is not directly disabled.
- Even when the system is deployed correctly and free of compromise, an attacker with sufficient long term network visibility and a favorable network position can passively observe traffic to and from landmark servers. This information may allow the adversary to approximate the number of chips operating in a datacenter. Such traffic analysis attacks can often be mitigated through network shaping techniques that introduce random padding or timing jitter to network packets. These mitigations may not be sufficient against an attacker with the ability to monitor network traffic over long periods of time in order to gather data for statistical analysis. Aggregating sessions through a central proxy can provide additional obfuscation, but this must be carefully designed so that it does not introduce extra processing time or network latency that would interfere with RTT computation.

If these attacks and their mitigations are not considered during system design, the cost of bypassing LVS will remain low. A well positioned adversary could rely on simple techniques such as traffic analysis, replay, or weak binding in the protocol to spoof chip locations. The defender's objective should be to raise the cost and reduce the reliability of attack so that adversaries cannot scale their efforts to smuggle entire clusters of chips. The best outcome is to force adversaries into developing complex exploit chains that target memory corruption, logic flaws in the chip TEE and CVM interconnect, or secure enclave escapes. These attacks are expensive to discover, difficult to exploit or operationalize at scale, and carry a higher risk of exposure. The cost curve for defeating the system en masse becomes steep, and the cheapest available option shifts from passive observation to advanced exploitation with limited reach.

Technical Feasibility

It is difficult to assess the feasibility of securely building and deploying a LVS that can withstand adversarial attempts to subvert it. While the fundamental building blocks for such a system exist today, implementing and operating them correctly over a sustained period of time is costly and will be challenged by highly capable adversaries. These assessments do not include potential silicon changes such as the integration of PUFs.

- We have **high confidence** it is feasible to specify and design a secure LVS protocol and general system architecture. The cryptographic primitives needed have been studied, available in open source code today and well tested. The timing based algorithms for geolocation are generally sound and can likely be improved upon with time.
- We have **high confidence** that a LVS can be implemented today within the timeline and cost estimates outlined here that can withstand low cost attacks that don't defeat the entire system but rather offer limited compromise of specific components. These kinds of attacks can be thought of as those implicated in typical incidents and breaches (e.g. weak or missing authentication and/or authorization, no encryption, known but unpatched vulnerabilities etc) and not specific to LVS. While these kinds of attacks are technically easy and cheap to scale across all chips, they are also easily mitigated with well understood techniques and technologies.
- We have **low to moderate confidence** that a LVS can be implemented today within the timeline and cost estimates outlined above that will withstand persistent and technically capable adversaries with access to the resources required to find and exploit LVS specific vulnerabilities that compromise the most trusted components in the system. These attacks may not inherently scale without significant investment but given the attack surface required to implement the system, and the adversaries incentive to defeat it, we assess the system will likely be bypassed or defeated within 9 - 12 months.

Estimated Location Verification System Development and Deployment Timeline and Costs

While it is not possible to precisely measure the total cost of implementing, maintaining, and operating such a system given the number of variables, cost estimates can be provided for individual components and activities. The time required to implement this system will vary by chip designer and will depend heavily on their existing hardware and operational capabilities. We estimate that designing, implementing, and securing a comparable LVS protocol would take approximately 9 to 12 months, assuming that some of the work is performed concurrently. It is important to note that there is no technical standard requiring chip designers to implement uniform security features, which means they could choose to eliminate or weaken specific elements. The removal of any critical security control or component could significantly affect these cost estimates. Note that these timelines are somewhat aggressive and assume they are prioritized in order to meet U.S. law and regulatory requirements.

This estimate excludes hardware modifications such as on-die PUFs. The cost of designing and deploying these features is likely in the tens of millions of dollars and would take multiple years. Where possible we have estimated costs in time but not dollars.

CAPEX

- 1 - 3 months: Design and verification of the overall LVS protocol including cryptographic primitives and accurate RTT algorithm. Timeline assumes there is existing literature and protocols to draw from.
- 4 - 6 months: Concurrent implementation and testing of all components to include writing new code in order to meet system requirements.
- 2 - 3 months: One time third party security verification of the protocol design, implementation, and deployment. Estimate based on 3-4 security subject matter experts in cryptography, memory safety, confidential computing and hardware based security.
- Initial acquisition costs for geographically distributed Landmark and Reporting servers. Depending on how often AI chips are required to ping the Landmark

Servers the number of physical servers required will vary. For example, if 250,000 AI chips are only required to ping the Landmark Servers every 24 hours this could be handled by 2 (for redundancy and capacity) Landmark servers per geographic installation.
- Dedicated high availability FIPS-140 certified HSM clusters are needed to support cryptographic operations.

OPEX

- Long term cost of log monitoring and incident response, maintaining and operationalizing these systems. Factors such as system uptime, vulnerability patching, log monitoring will also have costs. Additional engineers would be needed for new feature development and bug fixes.
- Long term cost of data center colocation, network bandwidth, log data stores and general server maintenance.

Alternative Designs

There are alternative 3rd party designs currently emerging in the market. While the author has not evaluated any of these solutions, at the time of this writing none of these offer a comprehensive technical whitepaper to review, based on API documentation they all appear to follow a similar design pattern. Because these solutions do not involve the chip designer they rely on secure or 'confidential compute' mode APIs exposed by the chip designer to inject an untrusted AI kernel into the confidential compute runtime. That kernel can theoretically generate an asymmetric cryptographic public/private keypair while running in secure mode but the AI chip won't provide cryptographic signing or secure storage APIs based on a hardware root of trust located in the GPU. Therefore this keypair would remain ephemeral as these kernels are intended to be stateless. This will likely require signing the key with an external certificate authority and moving it to the CPU TEE where it can be sealed for long term storage. This is not as secure of a design as the one covered in this paper. The CPU TEE

may not come with as strong of security guarantees as the AI chip, and the transfer of the key into the CPU TEE may leave it susceptible to interception even with encrypted DRAM. Furthermore this design requires all good actors to opt into this solution so that the lack of signal from bad actors can be actioned. While these 3rd party solutions may help good actors ensure their compute isn't physically diverted it can only scale with the involvement of AI chip resellers who package them in servers to be deployed in data centers. These resellers would be able to install these 3rd party LVS solutions in every server they sell, and stand up the infrastructure to implement the remainder of the LVS system similar to what is described in this paper. This would help detect large scale diversion of chips but would not prevent smaller amounts purchased from smaller resellers. It would also come with its own technical challenges and ultimately require the end user to keep it enabled and operational.

Another alternative design with lower costs and complexity is attestation to an on-site secure auditor system. In this alternative model a threshold would be set (e.g. deployments of 1,000 chips or more) would require a locally deployed auditor system. In this technical design a locally deployed system with a TEE would frequently and randomly present a cryptographic challenge to chips deployed in the same data center and record how long it takes for the chip to respond. The expected RTT would be very short given local physical proximity to the chip itself. The RTT would be recorded in batches and reported to an auditing server. This would still require the chip designer to extend their firmware but greatly reduce the complexity and attack surface of the overall system. While the local auditor system might be under physical attacker control it would require defeating the CPU TEE, but the physical and digital security of this system could be strengthened at low cost given the relatively low number of auditor systems required.

Enhanced Requirements

The Chip Security Act calls for studying enhancements to chip security that could improve or enforce protections against diversion. The LVS proposals discussed here and in other papers generally assume good actors deploying these systems and do not address what happens when a bad actor refuses to participate in the LVS system with diverted chips that were intended to be in a country exempt from the controls. They also do not address what technical levers could be used to disable diverted chips, even if only as a deterrence mechanism.

Using the cryptographic primitives described earlier for LVS enforcement, it is theoretically possible to implement digital licensing schemes that remotely enable or disable functionality. These mechanisms are often referred to as "kill switches." This idea has some precedent: while vendors did not ship kill switches, semiconductor designers in the 2010s experimented with remote microcode updates that unlocked additional CPU functionality after purchase. No publicly available AI chip documentation describes such functionality today, but it is technically feasible to build.

Additional telemetry such as compute and memory usage, power consumption, uptime, integrity check failures, and general AI chip activity could be uploaded at regular intervals from the TEE, through the CVM, to a reporting server. This would allow for operational insight and anomaly detection. However, mandatory reporting of this kind would likely be viewed as intrusive. One possible avenue for exploration is the use of privacy preserving aggregation techniques that derive insights without deanonymizing sources, though this approach may limit the usefulness of the data for export control enforcement.

As noted earlier, it is possible to design falsifiable protocols that allow chips to prove they are operating inside the U.S. or other approved countries. One option would be to deploy trusted auditor systems within authorized data centers that co-sign a chip's attestation, thereby vouching for its physical presence on a dedicated link. An adversary would need to physically steal or relocate both the chip and the colocated auditor to bypass this control. To reduce the risk of theft, these auditor systems should rely on

revocable cryptographic keys so that once reported missing or compromised, their signatures would no longer be trusted by the governance infrastructure.

Ensuring that global markets, including potential adversaries, rely on U.S. technology provides both economic benefits and soft power by reinforcing U.S. influence over technical ecosystems and standards. AI chip designers have an opportunity to design security features that restrict on-die cryptographic links so they can only be established with CPU TEEs of trusted or allied origin. This would preclude adversaries from developing competing parts of the overall system and would weaken their ability to influence the broader AI stack. With these kinds of foundational guarantees in place, it becomes possible to extend trust to the inference process and even to every tensor operation. In practice, this could mean requiring each kernel or operator invoked by the AI chip to be cryptographically signed and executed inside an attested TEE context, producing verifiable inference logs. These logs could then be aggregated by the CVM to a reporting server, enabling auditors to verify not just the location of the chip but also the integrity of individual tensor computations in a privacy preserving way that does not reveal the inference workload itself. However, the infrastructure, in both hardware and software, is not yet mature enough to realize these kinds of guarantees.

Open Questions

There are many unanswered questions around developing and deploying a hardware enabled AI governance system. Some of the answers will depend heavily on the direction of U.S. export controls, AI scaling laws, and other factors outside the scope of cyber security. This non-exhaustive list is intended to give policymakers a starting point when debating the merits, strengths and limitations of these kinds of controls:

- Consumers of these covered chips inside the U.S. are not subject to export control restrictions. This creates a difficult policy and technical challenge. The challenge for chip designers and deployers is how to exempt domestic users from LVS requirements without creating vulnerabilities or loopholes to the system. A design that is disabled for U.S. consumers, or that permits remote disablement

for U.S. based consumers, could be exploited if a chip is later smuggled abroad. This functionality also cannot be selectively applied at manufacturing or shipment time because any chip may later be resold and smuggled. If disabled for chips intended to be sold in the U.S. market, a chip smuggling operation only needs to find a source within the U.S. willing to sell them chips. A more robust solution is to have the feature enabled for all chips and develop falsifiable protocols that provide verifiable evidence of domestic use while avoiding remotely triggerable on/off functionality that might be repurposed if implemented insecurely. Such protocols would allow regulators and vendors to adhere to legal exemptions for U.S. customers while preserving the integrity of hardware based governance mechanisms. This will still impose a cost burden on domestic users since a functional LVS system requires them to attest to their location or risk being flagged as anomalous. Another alternative mitigation involves a TOFU approach where, in addition to the design in this document, all AI chips deployed within the U.S. (or other exempt countries) perform RTT measurements to U.S. based landmark servers in order to establish a baseline. This baseline would first be established when the chip is initialized and passes a predefined threshold of compute operations. All future RTT measurements are compared against this baseline in order to determine if the chip has potentially moved a measurable geographical distance. The relative geographic isolation of the U.S. may lend itself to higher accuracy of anomaly detection using this baseline in future RTT measurements.

- It is unclear how long the security of an LVS or otherwise enhanced chip security controls are expected to withstand adversarial attack. Current research and deployment experience suggest that designing a system able to resist years of focused research efforts to defeat it from a technically sophisticated nation-state adversary is unlikely. Policymakers should therefore set clear expectations for chip vendors on the intended security lifetime of such mechanisms or require compensating controls. These may include procedures for revoking cryptographic credentials, disabling compromised devices, or issuing updated

firmware and protocols when vulnerabilities are discovered with strict timelines for adoption. This kind of timeline would be an additional factor to consider in addition to TFLOPS, interconnect bandwidth, and performance density thresholds currently used to determine which semiconductors fall under existing ECCNs such as 3A090 and 4A090.

- There is substantial public evidence that chip smuggling remains a persistent and large scale problem potentially as high as 30% of inference chips and 40% of training chips of the global supply according to recent reports. Without adequate enforcement, a LVS might flag suspicious chip deployments but unless accompanied by effective policies and on the ground deterrents (such as revocation of cryptographic credentials, physical seizures, or legal action), the system's impact may be limited. In other words, LVS generated telemetry is only as useful as the operational and regulatory infrastructure that follows up on them.
- While the Chip Security Act is specifically written for chips covered under ECCNs 3A090 and 4A090 (and successors) it may create a desire for other export controlled technologies to include and enforce on similar remote attestation systems. Some of these technologies (such as robotics, avionics and sensors for example) include semiconductors but likely cannot implement similar controls. High end AI chips have dedicated TEEs, and sit in data centers with access to significant power, compute and reliable networking infrastructure. Policymakers should recognize these limitations, although they do not diminish the case for the Chip Security Act.
- The current draft of the bill is unclear about who is responsible for collecting attestation data from chips and standing up a LVS as described here. If the burden falls on the chip deployer (e.g. a company outside the U.S.) or a third party reseller, then resale of chips must be contemplated in the bill. From a technical perspective, the cryptographic and security guarantees are easier to build and enforce if the chip designer itself is responsible for building and enforcing the system no matter who deploys the chips.

Conclusion

Implementing a LVS that can withstand persistent adversarial attempts to subvert it is technically feasible but will be difficult to achieve. A single exploitable security vulnerability could potentially allow the system to be subverted for an indefinite amount of time, or reduce trust in its efficacy and accuracy, which could have a negative impact on U.S. national security. Without specific technical security requirements, or a baseline expectation for survivability, the system risks being undermined before it can provide enduring national security benefits. We suggest policymakers consider requiring BIS and/or the Secretary of Commerce to provide guidance to chip designers on how long the system is expected to remain effective in the face of persistent research by adversaries to defeat it.

Aggressive telemetry collection, or embedding remote kill switch functionality, poses a potential reputational risk as it may weaken trust in American technology. This can be partially mitigated through reproducible builds, binary transparency and open sourcing of code. However, even if designed with benign intent, features such as kill switches could be repurposed by adversaries to disable critical chips at scale. If this functionality is implemented then chip designers will certainly be targeted through cyber and malicious insider techniques as a means to gain access to this lever. This creates a single point of failure in perhaps what is the most important technology in the world, and a fundamental building block of the American economy in the 21st century. These risks should be weighed heavily by policymakers in determining the requirements of the Chip Security Act.

Instead of relying on fragile trust in embedded tracking and disablement, policymakers should consider shifting focus toward cloud centric deployment models for AI workloads. Treating these cloud compute providers as "regulatory intermediaries" and requiring them to perform robust Know Your Customer (KYC) checks and report

detailed workload telemetry, thereby offering better visibility and control over foreign AI compute without compromising chip integrity. This still achieves the goal of having the broader market, including adversaries, built on an American AI stack but without the risk of diverted AI chips that cannot be tracked.

References

https://mileskellerman.substack.com/p/this-chip-will-self-destruct - This Chip will Self-Destruct

https://www.rand.org/pubs/working_papers/WRA3056-1.html - RAND Hardware-Enabled Governance Mechanisms

https://s3.us-east-1.amazonaws.com/files.cnas.org/documents/CNAS-Report-Tech-Secure-Chips-Jan-24-finalb.pdf - CNAS Secure, Governable Chips

https://ai-frontiers.org/articles/location-verification-ai-chips - Can "Location Verification" Stop AI Chip Smuggling?

https://www.iaps.ai/research/location-verification-for-ai-chips - Location Verification for AI Chips

https://www.thefai.org/posts/the-chip-security-act-a-bipartisan-solution-to-chip-smuggling - Chip Security Act: A Bipartisan Solution to Chip Smuggling

https://www.nationalsecurity.ai/chapter/nonproliferation - Super Intelligence Strategy

https://arxiv.org/pdf/2503.05628 - Super Intelligence Strategy (Extended Paper)

https://www.rebuilding.tech/posts/conditional-export-controls-on-ai-chips - Conditional Export Controls on AI Chips

https://struct.github.io/hardware_exc.html - AI and Hardware Enabled Governance Mechanisms

https://papers-pdfs.assets.alphaxiv.org/2507.15916v1.pdf - Verifying International Agreements on AI

https://arxiv.org/pdf/2403.13230 - BFT-PoLoc: A Byzantine Fortified Trigonometric Proof of Location Protocol using Internet Delays

Thank You

This paper went through several iterations. The following people provided invaluable feedback: Hanna Dohmen, Jacob Feldgoise, Lennart Heim, Yan Ivnitskiy, Drew Lohn, Igor Mikolic-Torreira, Kyle Miller, and other anonymous experts. All remaining inaccuracies belong to the author.

About the Author

Chris Rohlf is a cyber security expert with over 20 years of experience working in tech, and advising national security focused think tanks and policymakers on cyber and emerging technology issues. The views expressed in this paper are his alone and do not represent any company, organization, or official U.S. Government policy positions.