# VensaSIM: A Foundational Analysis for a Sovereign SIM Card Standard

## 1. The Philosophical and Strategic Imperative

The architecture of global telecommunications infrastructure is not merely a technical arrangement of silicon, copper, and fiber; it is a manifestation of political power. The contemporary landscape is defined by a profound asymmetry between the users of technology—who rely on connectivity for economic survival, social organization, and political expression—and the entities that control the "chokepoints" of that connectivity. The "VensaSIM" initiative, inspired by the "open silicon" principles articulated by Vitalik Buterin and the hardware sovereignty advocacy of Bunnie Huang, emerges as a strategic counter-measure to this asymmetry. It seeks to transpose the cryptographic self-sovereignty pioneered in the blockchain domain—"not your keys, not your coins"—to the fundamental identity layer of the mobile network: the Subscriber Identity Module (SIM).

### 1.1 The Failure of Liberal Institutionalism and the Rise of Weaponized Interdependence

To understand the strategic necessity of a sovereign SIM, one must first interrogate the geopolitical context of the current digital order. For decades, the dominant framework in international relations and technology policy was "liberal institutionalism." This theory posited that complex interdependence—the dense web of global trade, communication, and supply chains—would inevitably foster cooperation and peace.1 The assumption was that because conflict would disrupt these mutually beneficial networks, state actors would be deterred from aggression. The networks themselves were viewed as neutral substrates—flat, decentralized meshes that facilitated the free flow of information and capital.2

However, this view has been shattered by the reality of "weaponized interdependence," a theoretical framework rigorously developed by political scientists Henry Farrell and Abraham Newman.1 Their analysis reveals that global networks do not evolve into flat meshes but rather into highly centralized "hub-and-spoke" topologies. Driven by network effects and efficiency gains, critical nodes—whether they are financial clearinghouses like SWIFT, internet exchange points, or the root certificate authorities of the telecommunications sector—become concentrated in a few jurisdictions.1

This centralization creates structural power that can be exploited by the states that host these hubs. Farrell and Newman identify two distinct mechanisms of this power, both of which are directly relevant to the architecture of the SIM card:

#### 1.1.1 The Panopticon Effect

The "Panopticon effect" refers to the surveillance capability granted to the state or entity that controls a central hub. Because information flows must pass through these central nodes to reach their destination, the hub controller gains a comprehensive view of the network's activity.3 In the context of mobile telecommunications, the SIM card is the authentication token that grants access to the network. Under the current paradigm, the cryptographic keys (Ki) used for this authentication are generated by the Mobile Network Operator (MNO) and stored in the Home Subscriber Server (HSS). The user possesses the physical card, but the *secret* that defines their identity is shared with the operator.4 This creates a permanent Panopticon. The operator—and by extension, intelligence agencies with legal or extralegal access to the operator's backend—can intercept communications (in 2G/3G contexts), track physical location via cell tower triangulation, and map social graphs via metadata, all without the user's consent or awareness.5 The user is "transparent" to the network, while the network remains opaque to the user.

#### 1.1.2 The Chokepoint Effect

The "Chokepoint effect" is the ability to deny access to the network entirely.1 States can exploit their jurisdiction over key hubs to cut off adversaries from critical flows. In the financial system, this is manifested in sanctions. In telecommunications, it is manifested in the remote deactivation of SIM cards or the revocation of service. The evolution toward the eSIM (embedded SIM) exacerbates this effect. The eSIM ecosystem relies on a centralized Public Key Infrastructure (PKI) governed by the GSMA (GSM Association). The Root of Trust (RoT) for the consumer eSIM is the GSMA Certificate Issuer (CI).6 If a device or a profile does not comply with the policies of this central authority, it can be effectively "bricked" or denied connectivity. This transforms the SIM from a piece of property owned by the user into a licensed service that can be revoked at a distance—a quintessential chokepoint.8

The VensaSIM project is a direct response to these structural vulnerabilities. By moving the generation and control of cryptographic keys from the central hub (the operator/GSMA) to the edge (the user's device), VensaSIM aims to dismantle the Panopticon and neutralize the Chokepoint, restoring the balance of power in favor of the individual.

### 1.2 The "Black Box" of Trust: Common Knowledge and Open Silicon

The strategic imperative for VensaSIM is further illuminated by Vitalik Buterin’s analysis of trust and "common knowledge" in security systems. Buterin argues that for a system to be truly secure, it is not enough for it to be mathematically sound; it must be *known* to be sound by the community of users. This requires "common knowledge"—a state where everyone knows that the system is secure, and everyone knows that everyone else knows it.1

The current SIM card ecosystem is the antithesis of this principle. It is a regime of "security through obscurity."

* **Proprietary Hardware**: The silicon layout of the UICC (Universal Integrated Circuit Card) is a trade secret of vendors like Thales (Gemalto), IDEMIA, and Giesecke+Devrient.
* **Closed Firmware**: The Operating System (typically a proprietary implementation of Java Card) is closed source.
* **Opaque Cryptography**: The algorithms used for key generation and authentication are often proprietary or standard algorithms (like Milenage) running in black-box environments where implementation flaws cannot be audited.9

Buterin’s "Vensa" initiative advocates for "open silicon"—hardware designs that are open-source and verifiable. While fully manufacturing a transparent chip is a formidable challenge, VensaSIM applies this philosophy to the *logic* layer of the SIM. By utilizing the programmable nature of the UICC and the Java Card standard, VensaSIM seeks to replace the proprietary authentication applets with open-source code that can be audited, compiled, and loaded by the user. This shifts the trust model from "trust the vendor" to "verify the code," aligning with the broader movement toward verifiable computing.

### 1.3 Hardware Trust and the "Time of Check vs. Time of Use" Problem

The work of hardware hacker and researcher Andrew "bunnie" Huang provides the technical grounding for VensaSIM’s approach to hardware verification. Huang’s projects, such as *Precursor* and *Betrusted*, grapple with the immense difficulty of establishing trust in physical devices.11 Unlike software, which can be hash-verified against a source code repository, hardware is physical and mutable. It can be tampered with during manufacturing, interdicted during shipping (evil maid attacks), or modified by sophisticated adversaries after deployment.

Huang identifies a critical vulnerability known as the "Time of Check vs. Time of Use" (TOCTOU) problem.12 A user might verify that a device is secure at a specific moment (Time of Check), but they have no guarantee that it remains secure when they actually use it (Time of Use). A "clean" device can be compromised between the audit and the operation.

The current SIM card suffers acutely from TOCTOU issues. Even if the specifications are public, the user receives a card that has already been provisioned with keys and applets in a secure facility to which they have no access. They must trust that the keys on the card are the only copies in existence, despite the fact that the operator generated them.

VensaSIM integrates Huang’s principle of "self-sealing" systems.11 A trusted device should be able to generate its own secrets internally, without relying on external provisioning tools that might be compromised. "Three can keep a secret, if two of them are dead," Huang quotes Benjamin Franklin, emphasizing that the only way to ensure a key is private is if it is generated by the user on the device itself and never leaves the secure boundary.11 The VensaSIM architecture mandates that the SIM card must generate the authentication keys (Ki) on-chip using a high-entropy RNG, ensuring that the operator never possesses the private key, only the public key required for verification.

### 1.4 The Cypherpunk Lineage: From the Clipper Chip to VensaSIM

VensaSIM is not an isolated technical endeavor; it is the latest iteration of a decades-long struggle for digital sovereignty known as the Cypherpunk movement. In the 1990s, the "Crypto Wars" were fought over the US government’s attempt to introduce the **Clipper Chip**.13 The Clipper Chip was a hardware encryption device intended for telecommunications that used a classified algorithm (Skipjack) and, crucially, included a "key escrow" mechanism. This mechanism gave the government a "backdoor" to decrypt all communications, ostensibly with a warrant.14

The Cypherpunks—a loose collective of mathematicians, cryptographers, and activists including Eric Hughes, Tim May, and later Matt Blaze—vehemently opposed this proposal. They argued that "privacy is necessary for an open society in the electronic age" and that building backdoors for the "good guys" inevitably creates vulnerabilities for the "bad guys".15 Their resistance, bolstered by Matt Blaze’s discovery of a technical flaw in the Clipper Chip’s escrow system, eventually led to the project's abandonment.16

However, while the Clipper Chip failed as a specific product, its spirit lives on in the modern SIM card. The standard SIM is, in effect, a Clipper Chip: it utilizes encryption (A5/1, A5/3), but the keys are "escrowed" by the network operator, who retains full access to user communications. The user has no choice in the algorithm and no control over the keys.

VensaSIM is the ideological successor to the Cypherpunk resistance. Just as the Cypherpunks wrote code (PGP, remailers) to make privacy a reality, VensaSIM proposes to write code (Java Card applets, host tools) to make hardware sovereignty a reality. It rejects the "key escrow" model of the GSMA in favor of the "forward secrecy" and "public key" models championed by the open privacy community.17 By allowing users to define their own cryptographic parameters—specifically avoiding government-standardized curves that may harbor kleptographic backdoors—VensaSIM reclaims the "digital sovereignty" that was the original promise of the internet.19

## 2. Incumbent Technology Deep Dive: Anatomy of the SIM/UICC/eSIM

To engineer a sovereign alternative, it is necessary to rigorously dissect the incumbent technology. The device commonly known as a "SIM card" is technically a **UICC (Universal Integrated Circuit Card)** running a **USIM (Universal Subscriber Identity Module)** application. It is a sophisticated, secure cryptoprocessor that acts as the anchor of trust in the cellular network.

### 2.1 Architectural Breakdown of the UICC

The UICC is a complete computer system on a single chip, engineered to withstand physical and logical attacks. Its architecture is defined by ISO/IEC 7816 standards and ETSI/3GPP specifications.

#### 2.1.1 The Hardware Components

The physical layer of the UICC consists of a secure microcontroller with specific memory and I/O subsystems:

* **CPU**: A secure 8-bit, 16-bit, or 32-bit processor (often based on ARM SecurCore, 8051, or proprietary architectures like Infineon’s SLE series). It includes hardware accelerators for cryptographic operations (DES, AES, RSA, ECC) and mechanisms to detect fault injection (e.g., voltage glitches, laser shots).9
* **ROM (Read-Only Memory)**: Contains the immutable Mask ROM code, including the Operating System (OS), the Java Card Virtual Machine (JCVM), and the fundamental API libraries. This code is burned into the silicon during fabrication and cannot be altered.9
* **EEPROM / Flash**: Non-volatile memory used for mutable data. This stores the file system (Master File, Dedicated Files, Elementary Files), installed applets (Load Files), and persistent cryptographic keys (Ki, OTA keys). This is the primary storage area that VensaSIM seeks to repurpose.
* **RAM**: Volatile memory used for the stack, heap, and transient objects during transaction processing.
* **I/O Interfaces**:
  + **ISO 7816-3**: The traditional contact interface providing VCC (Power), RST (Reset), CLK (Clock), and I/O (Serial Data).
  + **SWP (Single Wire Protocol)**: A single-pin interface (C6) used for communication with the NFC controller in the phone, enabling contactless payments and secure element functions.20

#### 2.1.2 The Software Stack: Java Card OS

The dominant operating system for modern UICCs is **Java Card**, a subset of the Java language optimized for smart cards. It enables "Write Once, Run Anywhere" portability for card applications (applets).

* **JCVM (Java Card Virtual Machine)**: The heart of the software stack. Unlike standard Java, the compilation process is split. The source code (.java) is compiled to class files (.class) and then converted off-card into a **CAP (Converted Applet)** file. The JCVM on the card executes the bytecode within this CAP file.9
* **JCRE (Java Card Runtime Environment)**: Manages the card's lifecycle (SELECT, PROCESS, DESELECT), memory allocation, and transaction atomicity.
* **Applet Firewall**: This is the primary security mechanism of the OS. It enforces strict isolation between applications. An applet in one context cannot access the objects or fields of an applet in another context unless they explicitly share data via **Shareable Interfaces**.21 This prevents a malicious game applet from reading the banking applet's keys.
* **GlobalPlatform API**: A standardized management layer that handles the secure loading, installation, and deletion of applets. It uses **Security Domains** (SD) to manage keys. The **Issuer Security Domain (ISD)** is the root administrator of the card.22

### 2.2 GSM/LTE Authentication: The Mathematical Mechanics

The core function of the SIM is **Authentication and Key Agreement (AKA)**. This process proves the subscriber's identity to the network and generates the session keys for encryption.

#### 2.2.1 The Evolution of Algorithms

The security of the cellular network has evolved through several generations of algorithms, each attempting to fix the flaws of the previous:

* **COMP128-1 (2G)**: The original GSM algorithm. It was a closed, proprietary design. In 1998, it was reverse-engineered and found to be catastrophically weak. The "narrow pipe" vulnerability allowed attackers to recover the master key (Ki) by querying the SIM with chosen challenges (RAND) roughly 150,000 times. This enabled the cloning of SIM cards.4
* **COMP128-2/3**: Updated versions that fixed the leakage but remained proprietary.
* **Milenage (3G/4G)**: A standardized algorithm set (3GPP TS 35.205) based on the **AES (Rijndael)** block cipher. It is open, scrutinized, and widely considered secure. It uses the 128-bit AES core to derive the authentication response and session keys.24
* **TUAK (4G/5G)**: Introduced as a backup to Milenage, TUAK is based on the **Keccak** permutation (the basis of SHA-3). It was developed to ensure that if a fundamental flaw were found in AES, the telecom industry could switch to a completely different cryptographic primitive without changing the hardware (assuming software implementation is possible).25

#### 2.2.2 The AKA Protocol Flow

The authentication process is a Challenge-Handshake protocol:

1. **Identity**: The User Equipment (UE) sends the IMSI to the network.
2. **Vector Generation**: The Home Subscriber Server (HSS) retrieves the user's secret key () and generates an Authentication Vector (AV).
   * It generates a 128-bit Random Number ().
   * It computes the Expected Response ().
   * It computes the Cipher Key () and Integrity Key ().
   * It computes the Authentication Token () to authenticate the network to the user (preventing false base stations).
3. **Challenge**: The network sends  and  to the UE.
4. **Verification**: The USIM verifies the  (using the sequence number  to prevent replay attacks). If valid, it computes its own response () using the stored  and the .
5. **Response**: The UE sends  to the network. If , the user is authenticated.27

### 2.3 Vulnerabilities and Attacks

Despite these improvements, the SIM ecosystem remains vulnerable due to legacy support and implementation complexity.

#### 2.3.1 Simjacker and the S@T Browser

**Simjacker** is a vulnerability disclosed in 2019 that exploits the **S@T (SIMalliance Toolbox) Browser**. This is a legacy application residing on many SIM cards, originally intended for WAP services. Attackers can send a specially crafted **Binary SMS** (SMS-PP Data Download) to the victim's phone number. The modem passes this SMS directly to the SIM card without displaying it to the user. The SMS contains bytecode instructions for the S@T Browser.

* **Mechanism**: The attacker sends a "Proactive Command" instruction (e.g., PROVIDE LOCAL INFORMATION). The S@T Browser executes this, queries the phone's OS for location (Cell ID) or IMEI, and sends the data back to the attacker via another silent SMS.
* **Impact**: This allows for remote tracking and espionage without the user's knowledge, bypassing the smartphone's OS security entirely.5

#### 2.3.2 The WIB Attack

Similar to Simjacker, the **WIB (Wireless Internet Browser)** attack exploits a proprietary browser created by SmartTrust. It allows for similar remote control capabilities, including initiating calls, sending SMS, and launching web browsers.5 Both attacks highlight the danger of "bloatware" on the SIM—hidden applications that the user cannot see or remove.

#### 2.3.3 Side-Channel Analysis

Hardware attacks on SIM cards have also proven effective. Differential Power Analysis (DPA) involves monitoring the power consumption of the SIM card while it performs cryptographic operations. Since the power usage of the transistors correlates with the data being processed (Hamming weight), statistical analysis of thousands of traces can reveal the secret key (). While modern cards have countermeasures (random wait states, masking), older or cheaper implementations remain vulnerable.10

### 2.4 Critical Analysis of the eSIM (eUICC) Standard

The industry's move to **eSIM (embedded SIM)** is often touted as a user-friendly evolution, but from a sovereignty perspective, it represents a significant regression.

#### 2.4.1 The SGP.22 Architecture

The Consumer eSIM standard (GSMA SGP.22) introduces a complex remote provisioning architecture:

* **SM-DP+ (Subscription Manager Data Preparation)**: A server controlled by the operator that stores the "Profiles" (virtual SIM cards).
* **SM-DS (Discovery Server)**: A directory service that helps the device find waiting profiles.
* **LPA (Local Profile Assistant)**: Software on the device that downloads and installs the profile.

#### 2.4.2 The Root of Trust Chokepoint

The critical flaw in this model is the **PKI (Public Key Infrastructure)**. The security of the entire ecosystem relies on the **GSMA Certificate Issuer (CI)**.

* Every eUICC chip has a certificate signed by a manufacturer (EUM), whose certificate is signed by the GSMA Root CI.
* The SM-DP+ also has a certificate signed by the GSMA Root CI.
* **The Lock-In**: The eUICC will *only* accept a profile download if it comes from an SM-DP+ with a valid GSMA signature. This means that an independent developer, a private company, or a community mesh network *cannot* issue their own eSIM profiles to standard consumer devices without paying for GSMA certification and hosting an accredited SM-DP+. This effectively bans "homebrew" SIMs and cements the GSMA as the ultimate gatekeeper of connectivity.6

This architecture creates a global chokepoint. If the GSMA revokes a certificate, or if a government pressures the GSMA to deny certification to a specific entity (e.g., an encrypted messaging service provider), those entities are functionally locked out of the global cellular ecosystem.

## 3. Cryptographic Sovereignty: The Core Innovation

The VensaSIM project addresses the "Black Box" nature of the SIM by introducing **Cryptographic Sovereignty**. This is the capability for the user to select, verify, and generate the cryptographic primitives used for their identity, rather than accepting the defaults provided by the vendor. This necessity is driven by the history of cryptographic subversion by state actors.

### 3.1 The NIST Controversy and Kleptography

Trust in standardized cryptography was irrevocably damaged by the **Dual\_EC\_DRBG** scandal. In 2006, NIST (National Institute of Standards and Technology) standardized a random number generator based on elliptic curves (SP 800-90A). Researchers later discovered that the algorithm relied on two points,  and . If an attacker knew a secret relationship between these points—specifically, if they generated  and  such that  for a secret integer —they could predict the output of the RNG after observing just a small amount of data.29

* **The Subversion**: This was a classic "Kleptographic" backdoor. It was indistinguishable from a secure system to an outside observer, but transparent to the keyholder.
* **The Evidence**: Snowden leaks revealed that the NSA had engineered this backdoor and paid RSA Security $10 million to make it the default in their BSAFE toolkit.30

This scandal casts a long shadow over other NIST standards, particularly the **NIST Elliptic Curves (P-256, P-384, P-521)**. These curves are defined by "seed" values—random hexadecimal strings used to generate the curve coefficients. The origin of these seeds is unexplained ("verifiably random" in name only). Skeptics argue that the seeds could have been brute-forced to find a curve with specific, hidden weaknesses (e.g., related to the endomorphism ring structure) known only to the NSA.31 While no attack has been proven, the *possibility* violates the principle of "nothing up my sleeve" numbers.

### 3.2 Comparative Analysis: secp256k1 vs. Curve25519

VensaSIM advocates for the use of "Rigid" curves—curves whose parameters are generated by a transparent mathematical process that leaves no room for manipulation. We compare the two leading candidates against the NIST standard.

| **Feature** | **NIST P-256** | **secp256k1 (Bitcoin)** | **Curve25519 (DJB)** |
| --- | --- | --- | --- |
| **Form** | Short Weierstrass | Short Weierstrass | Montgomery |
| **Equation** |  |  |  |
| **Coefficients** | Derived from unexplained seed | (Simple integers) | Smallest integer satisfying security criteria |
| **Field** | Prime | Prime | Prime |
| **Backdoor Risk** | High (Unexplained seed) | Low (Parameters too simple to hide trapdoor) | Minimal (Fully rigid generation) |
| **Side-Channel Safety** | Difficult (Requires complete addition formulas) | Moderate (Supports GLV method) | Excellent (Montgomery Ladder is constant-time) |
| **Adoption** | TLS, Government, Banking | Bitcoin, Ethereum | Signal, SSH, TLS 1.3, WhatsApp |

#### 3.2.1 secp256k1

Used by Bitcoin, this curve was defined by Certicom. Its parameters are incredibly simple (). The simplicity is its defense; it is highly improbable that the number "7" conceals a complex mathematical backdoor. It also supports the Gallant-Lambert-Vanstone (GLV) endomorphism, allowing for faster scalar multiplication than P-256.32

#### 3.2.2 Curve25519

Designed by Daniel J. Bernstein (DJB), Curve25519 is the gold standard for sovereign cryptography.

* **Rigidity**: The coefficient 486662 is the smallest integer that satisfies the curve's security requirements (twist security, prime order). There was no "wiggle room" for DJB to choose a weak parameter.33
* **Montgomery Ladder**: The arithmetic on Montgomery curves allows the use of the "Montgomery Ladder" algorithm for scalar multiplication. This algorithm always executes in the same number of steps and accesses memory in the same pattern, regardless of the key bits. This eliminates the **timing** and **power analysis** side-channels that plague Weierstrass curves like P-256.33
* **Twist Security**: It is secure against "Invalid Curve Attacks," where an attacker sends a point that lies on the "twist" of the curve rather than the curve itself to extract information.31

### 3.3 SafeCurves Criteria for User-Defined Parameters

VensaSIM proposes a radical feature: **User-Defined Curves**. The user should be able to load their own curve parameters into the SIM. However, allowing arbitrary curves is dangerous (a user might choose a weak curve). Therefore, the VensaSIM applet must verify that the user's curve meets the **SafeCurves** criteria established by Bernstein and Lange.31

The applet must check:

1. **Rho Hardness**: The order of the base point must be a sufficiently large prime (or near-prime) to resist Pollard's Rho attack ().
2. **Transfer Resistance**: The embedding degree must be large enough to prevent Menezes-Okamoto-Vanstone (MOV) attacks (transferring the discrete log problem to a finite field where it is easier to solve).
3. **Discriminant**: The complex multiplication (CM) field discriminant must be large enough () to ensure the endomorphism ring is generic.
4. **Rigidity**: The user must provide the seed string that generates the coefficients via a standard hash function (e.g., SHA-256), proving that the coefficients were not cherry-picked.

By implementing these checks within the open-source VensaSIM applet, we empower the user to achieve cryptographic agility without compromising security.

## 4. The Path to VensaSIM: Landscape Analysis & Technical Implementation

The realization of VensaSIM requires bridging the gap between high-level cryptographic theory and the low-level constraints of telecommunications hardware. We look to existing open-source projects for the building blocks.

### 4.1 The Open Telecom Ecosystem: Osmocom

The **Osmocom (Open Source Mobile Communications)** project is the bedrock of open cellular research. They have successfully reverse-engineered the entire GSM stack.

* **OsmoMSC (Mobile Switching Center)** and **OsmoHLR (Home Location Register)**: These represent the "Core Network." They hold the database of subscribers (IMSI, Ki).
* **OsmoBTS (Base Transceiver Station)**: The software that runs the radio access network.
* **Integration**: To test VensaSIM, one does not need a commercial spectrum license. One can run a "Network-in-a-Box" using a Software Defined Radio (like a LimeSDR) and the Osmocom stack.36 This allows the user to act as *both* the Subscriber and the Operator, verifying the end-to-end authentication flow.

### 4.2 Programmable SIMs: The Hardware Platform

Commercial SIMs are locked. The ADM (Administrative) keys required to overwrite the Ki or install new applets are held by the issuer. For VensaSIM, we utilize **Programmable SIMs**, such as those manufactured by **Sysmocom** (e.g., sysmoISIM-SJA2).

* **Open Access**: These cards are shipped with the ADM keys provided to the buyer. This gives the user full write access to the file system (MF, DF, EF).38
* **OPc on Card**: A crucial feature of Sysmocom cards is the ability to compute the OPc (Operator Variant Algorithm Configuration Field) *on the card*. Typically, OPc is derived from the master key OP and Ki off-card. By doing it on-card, the raw OP key never needs to be exposed or stored in the HLR, reducing the attack surface.39
* **Applet Support**: These cards comply with GlobalPlatform specs and allow the loading of custom Java Card CAP files. This is the deployment target for the Vensa applet.

### 4.3 Hardware Security Lessons: YubiKey vs. Nitrokey

The architectural choice between "Secure Element" and "Open MCU" is illustrated by the market leaders in hardware tokens.

| **Feature** | **YubiKey (Yubico)** | **Nitrokey 3 (Nitrokey)** |
| --- | --- | --- |
| **Architecture** | Secure Element (Closed Source) | MCU (Open Source) + Secure Element |
| **Firmware** | Proprietary | Open Source (Rust / Trussed) |
| **Verification** | Trust Vendor | Audit Code + Verify Hardware |
| **Applets** | Fixed at Factory | User Updateable (FIDO, OpenPGP) |

**YubiKey** relies on the "Black Box" model. The firmware is proprietary and runs on a secure element. It is secure, but unverifiable.40 **Nitrokey 3** uses a general-purpose microcontroller (nRF52) for the application logic and a Secure Element (SE050) only for key storage. The firmware is open-source **Rust**, allowing the community to audit the logic.41

**Lesson for VensaSIM**: While we currently must use the closed hardware of the UICC (similar to YubiKey), we can emulate the Nitrokey model by ensuring the *application logic* (the Applet) is open source. The ultimate goal is a RISC-V based open-silicon SIM, but the intermediate step is "Open Code on Closed Silicon."

### 4.4 The Enabler: JCMathLib and Software-Defined Crypto

A major technical barrier to VensaSIM is that standard Java Card APIs (e.g., JC 3.0.4) often lack native support for Curve25519 or custom curves. The hardware accelerators are hardwired for RSA and NIST P-256.

The solution is **JCMathLib**, an open-source library developed by the OpenCrypto Project.42

* **BigNat**: It implements a Big Integer library in pure Java Card bytecode, allowing operations on integers larger than the native register size.
* **Soft-Crypto**: It implements Elliptic Curve Point Addition and Scalar Multiplication in software.
* **Optimization**: To make this performant enough for authentication (under 2 seconds), JCMathLib uses the card's RSA coprocessor (specifically the ALG\_RSA\_NOPAD engine) to perform the heavy modular multiplications required for ECC.44
* **Capability**: This allows a VensaSIM applet to perform a handshake using **Curve25519** or **secp256k1** on a standard card that officially only supports NIST curves. This breaks the vendor lock-in on cryptographic primitives.

### 4.5 The VensaSIM Architecture Proposal

Synthesizing these elements, we define the VensaSIM Standard Architecture.

#### 4.5.1 Component Stack

1. **Hardware**: Sysmocom sysmoISIM-SJA2 (or any unlocked GP 2.2+ card).
2. **Card Side (The Vensa Applet)**:
   * **Core**: Java Card 3.0.4 Applet.
   * **Crypto Layer**: JCMathLib integration for Curve25519/secp256k1 support.
   * **Storage**: Secure storage for User-Generated Keys (PrK) and Curve Parameters.
   * **Logic**: Implementation of a custom AKA protocol (Vensa-AKA) that uses the custom curve for the handshake.
3. **Host Side (The Manager)**:
   * **Tooling**: A fork of **pySim** extended with GlobalPlatformPro capabilities.45
   * **Functions**:
     + vensa-init: Compiles the applet, generates keys on the host (or requests on-card generation), and installs the CAP file.
     + vensa-verify: Audits the installed applet against the source code hash.
     + vensa-parameter-load: Injects user-defined curve parameters (A, B, P, G, N) after verifying SafeCurves compliance.

#### 4.5.2 The Vensa-AKA Protocol

Instead of the standard MILENAGE flow, the VensaSIM operates a Public Key Authenticated Key Exchange:

1. **Registration**: User installs VensaSIM. The Applet generates a keypair () on Curve25519. The User sends the Public Key  to their provider (e.g., a private Osmocom network).
2. **Challenge**: Network sends a random challenge .
3. **Response**:
   * VensaSIM signs  with private key  (using Ed25519 signature).
   * VensaSIM performs ECDH with the Network's Ephemeral Key to derive session keys ().
4. **Result**: Perfect Forward Secrecy is achieved. The operator never holds the user's private key. The Panopticon is blinded.

## 5. Conclusion

The VensaSIM project is more than a technical specification; it is a declaration of independence for the digital age. It bridges the philosophical chasm between the centralized, coercive architectures identified by the theory of Weaponized Interdependence and the decentralized, sovereign ideals of the Cypherpunk movement.

By leveraging the existing tools of the open telecom community—Osmocom, pySim, and programmable hardware—and combining them with the cryptographic agility enabled by JCMathLib, VensaSIM demonstrates that the "Black Box" of the SIM card can be opened. We need not wait for a benevolent corporation to manufacture a transparent chip; we can reprogram the existing infrastructure to serve the user rather than the network.

The path forward requires a concerted effort to build the "Vensa-AKA" protocol and verify the performance of soft-crypto implementations on commodity hardware. But the foundational analysis is clear: Cryptographic Sovereignty is the only viable defense against the Panopticon, and VensaSIM is the architecture to achieve it.

*(End of Report)*

#### Источники

1. Weaponized interdependence - Wikipedia, дата последнего обращения: января 28, 2026, <https://en.wikipedia.org/wiki/Weaponized_interdependence>
2. Henry Farrell and Abraham Newman: Weaponized Interdependence | University Consortium, дата последнего обращения: января 28, 2026, <https://uc.web.ox.ac.uk/article/henry-farrel-and-abraham-newman-weaponized-interdependence>
3. Weaponized Interdependence: How Global Economic Networks Shape State Coercion | International Security - MIT Press Direct, дата последнего обращения: января 28, 2026, <https://direct.mit.edu/isec/article/44/1/42/12237/Weaponized-Interdependence-How-Global-Economic>
4. GSM and 3G Security Summary and Vocabulary Help, © Nicolas T. Courtois, 2006-2010 University College London Telco, дата последнего обращения: января 28, 2026, <http://www0.cs.ucl.ac.uk/staff/n.courtois/GSM_vocab_Courtois.pdf>
5. Mobile Cyberattacks Conducted by U.S. Intelligence Agencies, дата последнего обращения: января 28, 2026, <https://www.china-cia.org.cn/WorkDetail/Mobile%20Cyberattacks%20Conducted%20by%20US%20Intelligence%20Agencies.pdf>
6. SGP.14-v2.2.docx - GSMA, дата последнего обращения: января 28, 2026, <https://www.gsma.com/solutions-and-impact/technologies/esim/wp-content/uploads/2025/01/SGP.14-v2.2.docx>
7. EMBEDDED SIM ECOSYSTEM, SECURITY RISKS AND MEASURES - ENISA, дата последнего обращения: января 28, 2026, <https://www.enisa.europa.eu/sites/default/files/publications/Embedded%20Sim%20Ecosystem%20Security%20Risks%20and%20Measures.pdf>
8. A few thoughts on Ray Ozzie's “Clear” Proposal, дата последнего обращения: января 28, 2026, <https://blog.cryptographyengineering.com/2018/04/26/a-few-thoughts-on-ray-ozzies-clear-proposal/>
9. Interoperability Stepping Stones - Trusted Connectivity Alliance, дата последнего обращения: января 28, 2026, <https://trustedconnectivityalliance.org/wp-content/uploads/2020/01/SteppingStones_R7_v100.pdf>
10. embedded UICC (eUICC) - NCCS, дата последнего обращения: января 28, 2026, <https://nccs.gov.in/public/itsar/ITSAR409022411.pdf>
11. Our Core Principles | betrusted.io, дата последнего обращения: января 28, 2026, <https://betrusted.io/>
12. SLP241 Bunnie – Precursor: Open Source Hardware Development Platform (Feat. Nicolas Dorier as guest host) - Stephan Livera, дата последнего обращения: января 28, 2026, <https://stephanlivera.com/episode/241/>
13. The First Crypto War | CoinMarketCap, дата последнего обращения: января 28, 2026, <https://coinmarketcap.com/academy/article/the-first-crypto-war>
14. Cypherpunk - Wikipedia, дата последнего обращения: января 28, 2026, <https://en.wikipedia.org/wiki/Cypherpunk>
15. Cypherpunk - Internet Policy Review, дата последнего обращения: января 28, 2026, <https://policyreview.info/glossary/cypherpunk>
16. How 70s Cryptography Became the Foundation of Bitcoin - Digital Ascension Group, дата последнего обращения: января 28, 2026, <https://www.digitalfamilyoffice.io/how-70s-cryptography-became-the-foundation-of-bitcoin/>
17. The History and Philosophy of Cypherpunk | Devcon SEA - YouTube, дата последнего обращения: января 28, 2026, <https://www.youtube.com/watch?v=OZwG_Tx1hdA>
18. (PDF) The legacy of cypherpunk in the modern economy - ResearchGate, дата последнего обращения: января 28, 2026, <https://www.researchgate.net/publication/391919627_The_legacy_of_cypherpunk_in_the_modern_economy>
19. The Path to Digital Sovereignty. What is Decentralized Identity? How… | by Guido Sirna | Medium, дата последнего обращения: января 28, 2026, <https://medium.com/@guidosirna/the-path-to-digital-sovereignty-0beb42543172>
20. ETSI TS 103 666-1 V15.3.0 (2020-09), дата последнего обращения: января 28, 2026, <https://www.etsi.org/deliver/etsi_ts/103600_103699/10366601/15.03.00_60/ts_10366601v150300p.pdf>
21. Java Card - Wikipedia, дата последнего обращения: января 28, 2026, <https://en.wikipedia.org/wiki/Java_Card>
22. martinpaljak/GlobalPlatformPro: Manage applets and keys on JavaCard-s like a pro - GitHub, дата последнего обращения: января 28, 2026, <https://github.com/martinpaljak/GlobalPlatformPro>
23. On the Need of Physical Security for Small Embedded Devices: A Case Study with COMP128-1 Implementations in SIM Cards - ResearchGate, дата последнего обращения: января 28, 2026, <https://www.researchgate.net/publication/284569428_On_the_Need_of_Physical_Security_for_Small_Embedded_Devices_A_Case_Study_with_COMP128-1_Implementations_in_SIM_Cards>
24. TSGS#18(02)0698 - 3GPP, дата последнего обращения: января 28, 2026, <https://www.3gpp.org/ftp/TSG_SA/TSG_SA/TSGS_18/Docs/pdf/SP-020698.pdf>
25. 3GPP Mechanisms for 5G Mobile Networks - TUAK - Thales Docs, дата последнего обращения: января 28, 2026, <https://www.thalesdocs.com/gphsm/luna/7/docs/network/Content/sdk/extensions/3GPP.htm>
26. A Comprehensive Security Analysis of the TUAK Algorithm Set - Centre For Applied Cryptographic Research, дата последнего обращения: января 28, 2026, <https://cacr.uwaterloo.ca/techreports/2016/cacr2016-02.pdf>
27. ETSI TS 133 220 V17.3.0 (2022-07), дата последнего обращения: января 28, 2026, <https://www.etsi.org/deliver/etsi_ts/133200_133299/133220/17.03.00_60/ts_133220v170300p.pdf>
28. Threat Wire - Amazon S3, дата последнего обращения: января 28, 2026, <https://s3.amazonaws.com/feed.podbean.com/shannonmorse/feed.xml>
29. Dual\_EC\_DRBG - Wikipedia, дата последнего обращения: января 28, 2026, <https://en.wikipedia.org/wiki/Dual_EC_DRBG>
30. How the NSA (may have) put a backdoor in RSA's cryptography: A technical primer, дата последнего обращения: января 28, 2026, <https://blog.cloudflare.com/how-the-nsa-may-have-put-a-backdoor-in-rsas-cryptography-a-technical-primer/>
31. Rigidity - SafeCurves, дата последнего обращения: января 28, 2026, <http://safecurves.cr.yp.to/rigid.html>
32. Two glaring exceptions. Bitcoin and Ethereum both use secp256k1. As well as a ho... | Hacker News, дата последнего обращения: января 28, 2026, <https://news.ycombinator.com/item?id=34250604>
33. Curves with a Twist - XRP Ledger, дата последнего обращения: января 28, 2026, <https://xrpl.org/ja/blog/2014/curves-with-a-twist>
34. On cryptographic weaknesses related to elliptic curves - IRIS, дата последнего обращения: января 28, 2026, <https://iris.unive.it/retrieve/9a6cda7e-bd55-4457-98ff-210886e00671/956406-1237431.pdf>
35. SafeCurves: Introduction, дата последнего обращения: января 28, 2026, <https://safecurves.cr.yp.to/>
36. GSM with Osmocom Part 7: The HLR – Home Location Register (and Friends), дата последнего обращения: января 28, 2026, <https://nickvsnetworking.com/gsm-with-osmocom-part-7-the-hlr-home-location-register-and-friends/>
37. Homebrewing a basic circuit-switched GSM (2G) base station! | krvprashanth, дата последнего обращения: января 28, 2026, <https://krvprashanth.in/posts/homebrew-gsm-base-station/>
38. PySIM - SIM card management tool used in Web UI to write SIM cards - YateBTS, дата последнего обращения: января 28, 2026, <https://yatebts.com/documentation/pysim-documentation/>
39. Sysmocom - S.F.M.C. GMBH: Sysmousim / Sysmoisim User Manual | PDF - Scribd, дата последнего обращения: января 28, 2026, <https://www.scribd.com/document/665605978/01-29>
40. YubiKey Technical Manual - Yubico Product Documentation, дата последнего обращения: января 28, 2026, <https://docs.yubico.com/hardware/yubikey/yk-tech-manual/webdocs.pdf>
41. U2F2 : Prévenir la menace fantôme sur FIDO/U2F - Sstic, дата последнего обращения: января 28, 2026, <https://www.sstic.org/media/SSTIC2021/SSTIC-actes/u2f2__prvenir_la_menace_fantme_sur_fidou2f/SSTIC2021-Article-u2f2__prvenir_la_menace_fantme_sur_fidou2f-thierry_benadjila.pdf>
42. JCMathLib: Wrapper Cryptographic Library for Transparent and Certifiable JavaCard Applets - UCL Discovery, дата последнего обращения: января 28, 2026, <https://discovery.ucl.ac.uk/10117901/1/JCMathLib__Towards_Transparent_and_Auditable_JavaCard_Applets%20%283%29.pdf>
43. OpenCryptoProject/JCMathLib: Implementation of mathematical operations with big numbers and elliptic curve points for smart cards with JavaCard platform. - GitHub, дата последнего обращения: января 28, 2026, <https://github.com/OpenCryptoProject/JCMathLib>
44. Enabling Efficient Threshold Signature Computation via Java Card API - CRoCS, дата последнего обращения: января 28, 2026, <https://crocs.fi.muni.cz/_media/publications/pdf/2023-ares-dufka.pdf>
45. osmopysim-usermanual - Index of /, дата последнего обращения: января 28, 2026, <https://downloads.osmocom.org/docs/pysim/master/osmopysim-usermanual.pdf>
46. Home · martinpaljak/GlobalPlatformPro Wiki - GitHub, дата последнего обращения: января 28, 2026, <https://github.com/martinpaljak/GlobalPlatformPro/wiki/Home/271e9ab3567e35329b72408e6a0ea5afba2035e4>