WaC: Trustworthy Encryption and Communication in an IT Ecosystem with Artificial Superintelligence (ASI)

• • •

ASHES '21 – Seoul, South Korea
Nov 19th, 2021

Erland Wittkotter, PhD
Erland.Wittkotter@gmail.com
+1 702 997 2475
Skype: ike2345

Highlights

- Goal: Protection of keys from being stolen by the worst adversary
  1. Keys never in Cleartext outside specially designed/protected hardware
     - Keys inaccessible
     - Key in main CPU/RAM → Key compromised
  2. Exchange of Keys between Hardware Key-Safes only
  3. Regardless of "unbreakable" or "impossible" to extract cleartext key:
     - Redundancy: Auto-detection if key was stolen
       - via (Random) Challenge – "Prove to me that you are not fake/software"
     - Distrust of CPU: Use-Case Detection for dealing with "API-Problem"
- Usage of Trustworthy Encryption/Decryption
  - "Trustworthy Communication"
    - Facilitating legitimate (court-ordered) eavesdropping
  - "Trustworthy eCommerce" (incl. logistics)
  - Make or Break for AI Safety

Adversary

- Artificial Superintelligence (ASI) is super-smart
  - At least human-expert-level in all topics/skills
    - Combining expert level skills/tools
    - Synergies → "Intelligence Explosion" (Fast take-off)
  - Modifies code intentionally (**"ASI's Defining Feature"**)
    - Reverse Code Engineering (RCE): modifying binary code
- Possibly: Super-Hacker, Digital-Ghost, Master-Thief, …
  - Effortless access to all devices (computational resources: the more, the better)
  - Might exist in IT ecosystem (hidden, currently undetectable)
  - Steals every encryption key it wants (undetectable) …
- ASI is the worst-case adversary

→ Goal: Protect Keys from being stolen by ASI

Trusted vs. Trustworthy

- Trusted
  - Receives secrets / entrusted with tasks
  - Reveals secrets or cooperates under distress
  - Could be turned into traitor, saboteur
    - Trusted until becoming traitor
    - Can't be trusted again
- Trustworthy
  - Receives secrets / entrusted with tasks
  - Never cooperates:
    - **taking secrets into its grave**
    - **better be dead than a traitor**
  - No betrayal, never disloyal
    - If failed under distress
      - Never gives up to reveal/correct betrayal – asap
      - Uncooperative in making betrayal worse
      - Constantly probing if misused
      - Trying to fix damage done, automatically, if possible
    - **Can be trustworthy again, once issue fixed**

Trustworthy Encryption/Decryption

- Guiding Principle: **Keys never shown in cleartext:**
  - Cleartext Keys (CTK) in main CPU are compromised
    - CTK "touchable" by modifiable Software → compromised
  - No openly published Public Keys
- Keys stored/processed in hardware-based **Key-Safes** with
  - Physically Unclonable Functions (PUF) to protect keys,
  - Dedicated **Encryption/Decryption Unit** to process keys/content.
  - Key-Safe-Key-Pair (as fall-back) – generated when manufactured
  - Validator: contacted Key-Safe uses keys that only Key-Safe hardware can know
- Public Keys "published" via Hashcode Directories
  - Keys referenced via hashcodes (otherwise similar to SSL/TLS/PKI)
    - Intentionally incompatible with existing PKI (and SSL/TLS)

→ **Unbreakable Data-Privacy with Key-Safe**

- **Deletion before revealing keys – Restauration via Hashcode**
- **Not taking chances – better (multiple) redundancies**

Redundancies (1)

- Auto-Detection/-Report misuse, attacks on keys (even attempts)
  - Consumer Products – No organizational security
- Scenarios: "What if" - Key stolen, or unknown use case
  1. Public Key
  2. Private Key
  3. Session Key ← bad, but temporary
  4. **Private + Public Key stolen ← Worst Case**
  5. *** Key-Safe is misused by attacker/CPU (Unknown Use-Case)*
- Sender (**A**lice)/Receiver (**B**ob) don't trust each other
  - Eavesdropping or Impersonating by "Eve" (Eavesdropper)
  - Worst Case "Man-In-The-Middle-Attack" (MITMA)
- **A** or **B** initiates "Challenge":
  - Randomly: "Prove to me that you are not fake" i.e. a "real" Key-Safe
  - Used Tools:
    - Sequence Number
    - Hashcode of already send/received data
    - "Multiple Equivalent Secret Keys" (MESK)

Redundancies (2)

- What if Senders Private + Public Key was stolen → MITMA
  1. Sequence Number (+ offset)
  2. Hashcode of already send/received data (from multiple contexts)
  3. Server: "Multiple Equivalent Secret Keys" (MESK)
     - 100's / 1000's of Priv./Publ. Key Pairs for same service
     - Receiver KS "knows" 3 or 4 based on "Selection Key"
- Bob: Sender, prove to me that you are not Eve!

Sender/Server (Alice)	↔	Eve ⟷ Eve	↔	Receiver/Client (Bob)
		Ok (pass through)		1. Your Sequence Number?
		Ok (stupid question)		~~2. Your Hashcode of Rec./Send Data?~~
		Which Key? (steps aside)		3. B's Selection Key is: … restart session a. A sends new Session Key encrypted with other Private Key: A know that B can read msg/key
		Gotcha		b. • Session ID • Hashcode of already exchanged Data?

Use Case Detection – "API Problem"

- Key-Safe can't simply be used by CPU
  - Use Case matters
  - Misuse of Trustworthy Encryption is unacceptable
    - Contradiction to trustworthiness
  - Key-Safe does not trust main CPU/OS
    - Part of network of trustworthy components ("Watchdogs" - WDs)
    - WDs are OK-ing Use-Case and content
  - New Use Cases – No Problems – Its done in the open
- Key-Safe Software:
  - Algo(s) dealing with Cleartext Keys are in Hardware only
  - OS for dealing with WDs , Use Case Detection
  - OS update
    - From remote (trustworthy) sources only
    - Local hashcode validation of software

Key-Safes in Communication

- Unbreakable communication is unacceptable
  - Good reasons for surveillance
    - Legitimate spying tools shall not be used by criminals
  - Currently: weak encryption, "backdoors" in OS (secret 0-day exploits)
- Facilitation of Legitimate Eavesdropping
  - **Side-door for court-issued "Warrants"** (signed **with courts' public keys**)
    - Law Enforcement – legitimized via special hardware/keys
      - No Cleartext Key for Law Enforcement
      - Access to session keys
      - Reasonable: Immutable Log Records from Side-door usage
    - Use Case: Facilitating parental supervision over children's communication
      - Possible misuse of parental features via attackers
      - Mitigation tools/procedures required: preventing social hacking

Key-Safes in eCommerce

- eCommerce - prevent manipulation of transaction terms
  - 2020: globally $1T (via cybercrime) -- 2025: est. $10T
    - Cybercrime will soon less labor intensive
    - eCommerce vulnerable to AI tools (personalized attacks)
  - Use Case Detection for a wide variety of situations
  - Scripts: confirming/documenting commercial transaction
    - Has user seen terms?
    - Are terms unmodified? …
  - Providing immutable log records for all participants
  - → "Trustworthy eCommerce"
- Utilization within (delivery) Logistics
  - Logistics is Achilles Heel of our Infrastructure
    - Currently: "almost" unprotected

Summary

ASI Lab Safety

- Trustworthy Encryption/Decryption:
  - Unbreakable Data-Privacy
  - Redundancy to discover stolen keys (reliably)
  - Requires reliable/trustworthy use case detection
- Key-Safe usage:
  - "Trustworthy Communication" with legitimate Eavesdropping
  - "Trustworthy eCommerce" (incl. Logistics)
- Key-Safes restrain ASI's ability to steal keys/secrets