최신 컴퓨터보안 연구 소개

서울대학교
공과대학 전기정보공학부
이병영
byoungyoung@snu.ac.kr

Speaker: 이병영

- 연구분야: Hacking, Systems Security, Software Security
  - Microsoft Research, Research Intern (2012)
  - Google, Software Engineering Intern (2014)
  - Purdue University, Assistant Professor (2016-2018)
  - Seoul National University, Assistant Professor (2018-Current)
- Three times DEFCON CTF Finalist (2007,2009, and 2011)
- Internet Defense Prize by Facebook and USENIX (2015)
- DARPA Cyber Grand Challenge (CGC) Finalist (2016)
- Google ASPIRE Awards (2019)
- Found 100++ vulnerabilities from Windows kernel, Linux kernel, Chrome, Firefox, etc.

전반적 보안 연구 분야

- **Systems Security**
  - **Hacking (software/hardware security)**
  - **Data security**

- Network Security
  - Anonymity network (Tor)
  - Blockchain

- Cryptography
  - Homomorphic encryption (동형암호)
  - Post-quantum cryptography

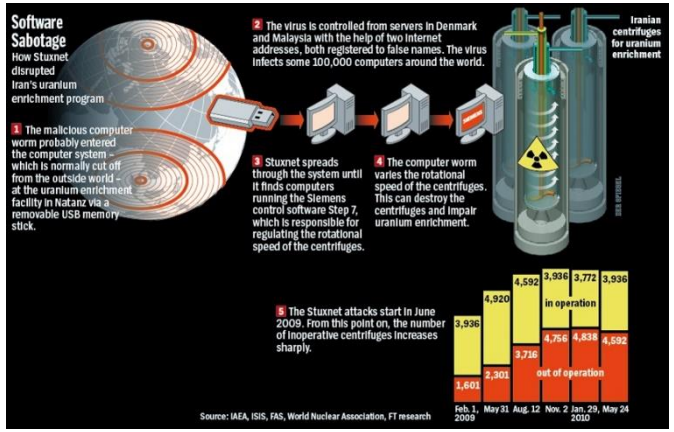
Hacking

Hacking

- **Hacking 101: Hacking in two steps**

- **Step1. Find vulnerability (developer's mistake)**
  - Look at the source code of implementation
  - Reverse-engineer the implementation
- **Step2. Attack the vulnerability**
  - Understand how the system work
  - Bypass all protection components

Hacking in Real-world



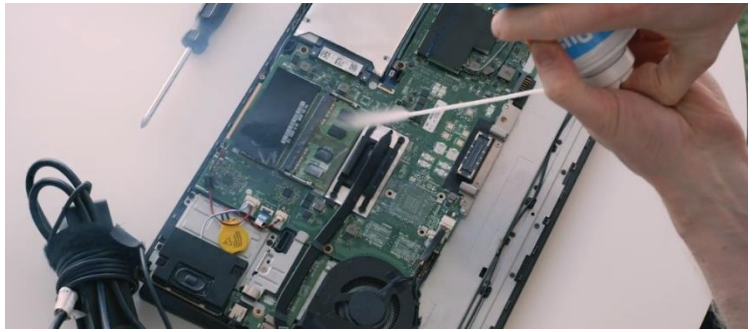
Stuxnet

Automobile Hacking (Jeep)

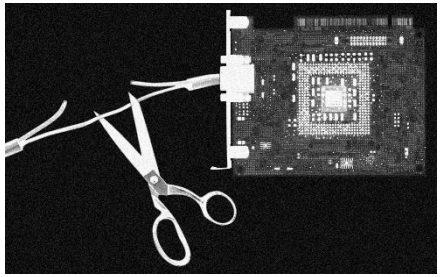
Mobile hacking (iOS, Android)

New Hacking Trends (1)

- **From cyber to physical attacks**
  - Cold-boot attacks

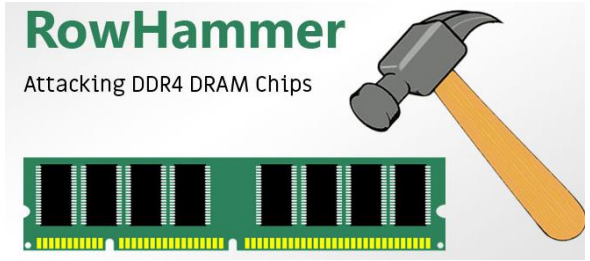


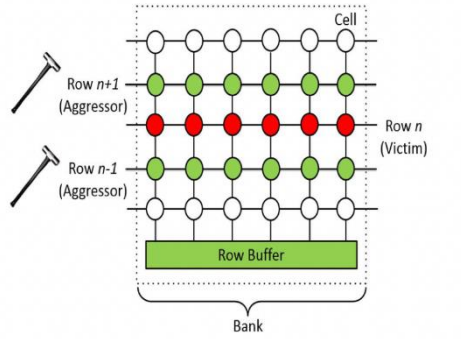
  - Plunder volt attacks



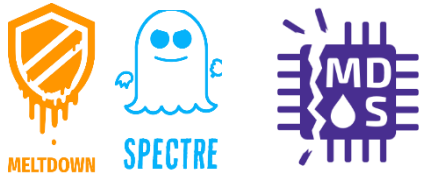
New Hacking Trends (2)

- **From software to hardware attacks**
  - Attack vulnerabilities in DRAM: RowHammer





  - Attack vulnerabilities in CPU: Spectre, Meltdown, MDS



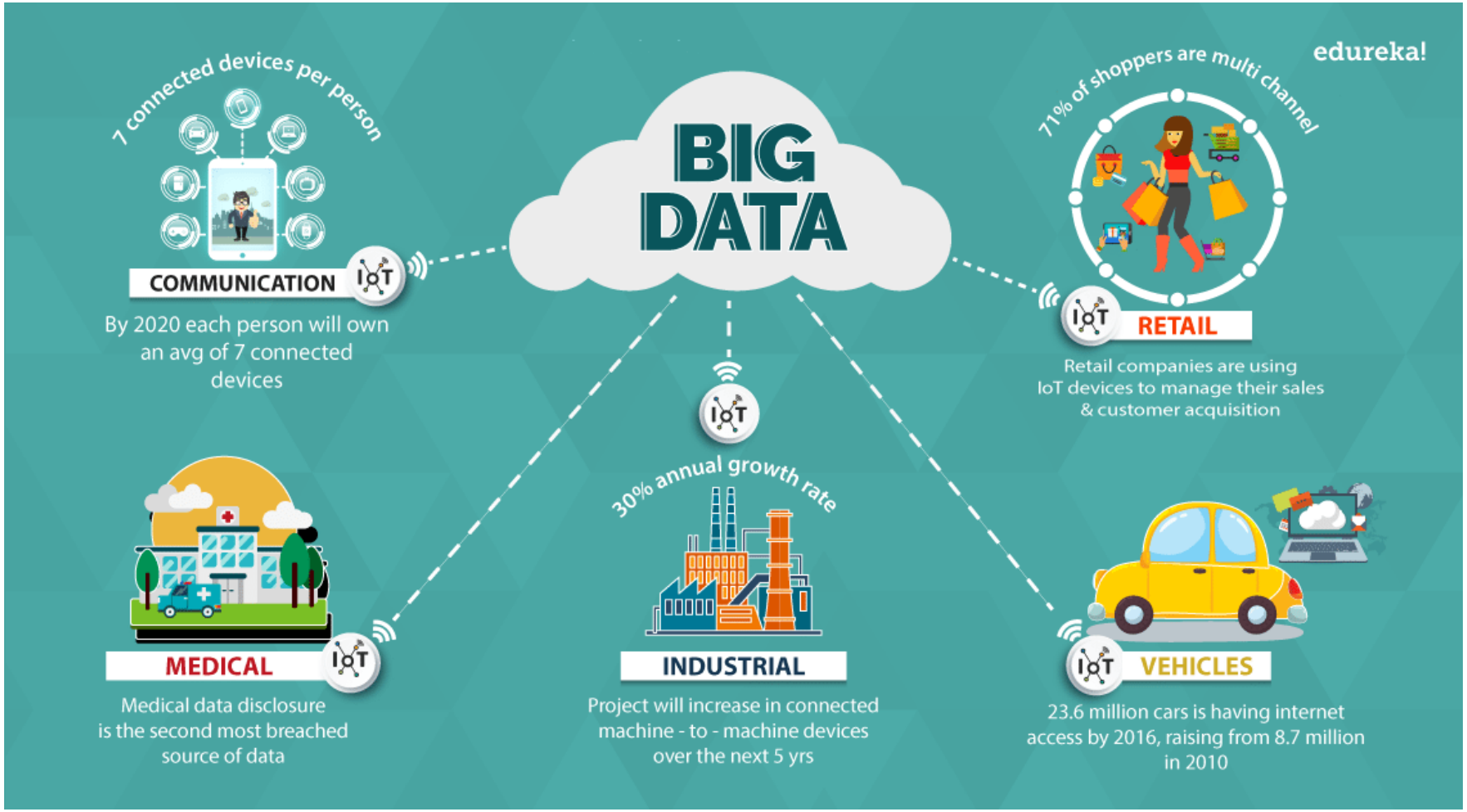
How to Stop Hacking

- **So how do we stop hacking?**
  - **No single solution cannot solve all security problems**
    - Tradeoffs: Performance vs. Security
    - Tradeoffs: Usability vs. Security

- Following three approaches are **always used together**
  - 1) Vulnerability finding
  - 2) Protection by runtime enforcement (detection)
  - 3) Prevention by design

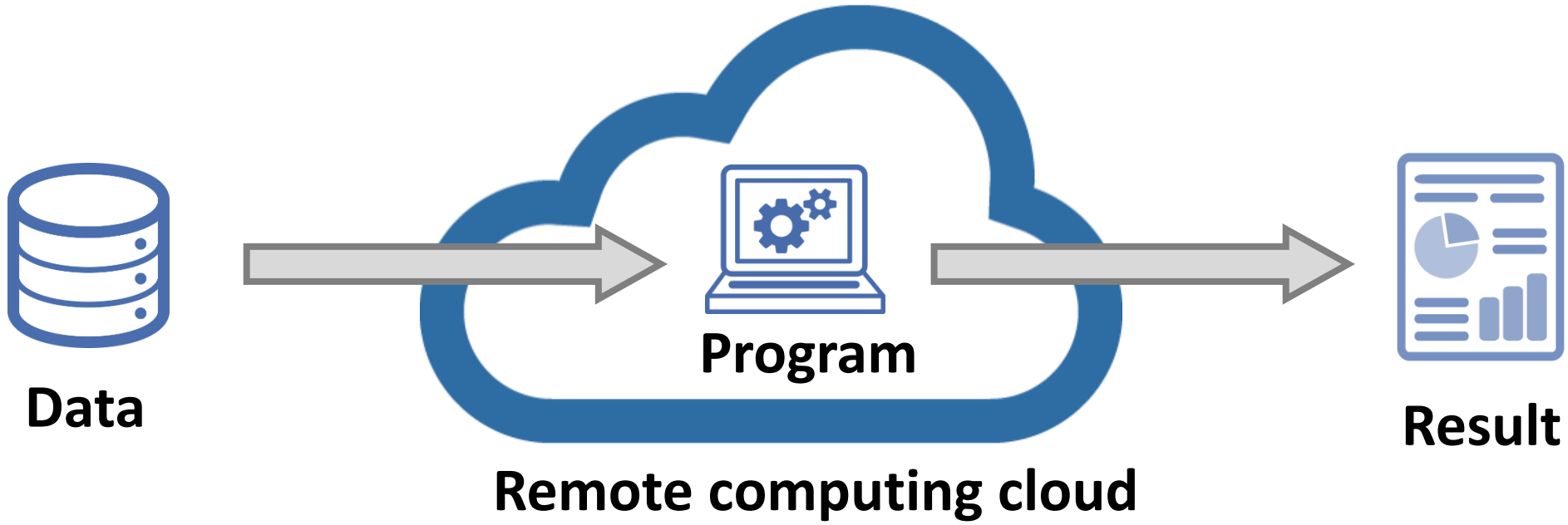
Data Security

The Era of AI/ML/Big Data

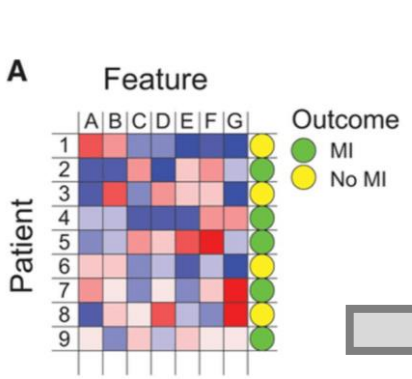


https://www.edureka.co/blog/big-data-applications-revolutionizing-various-domains/

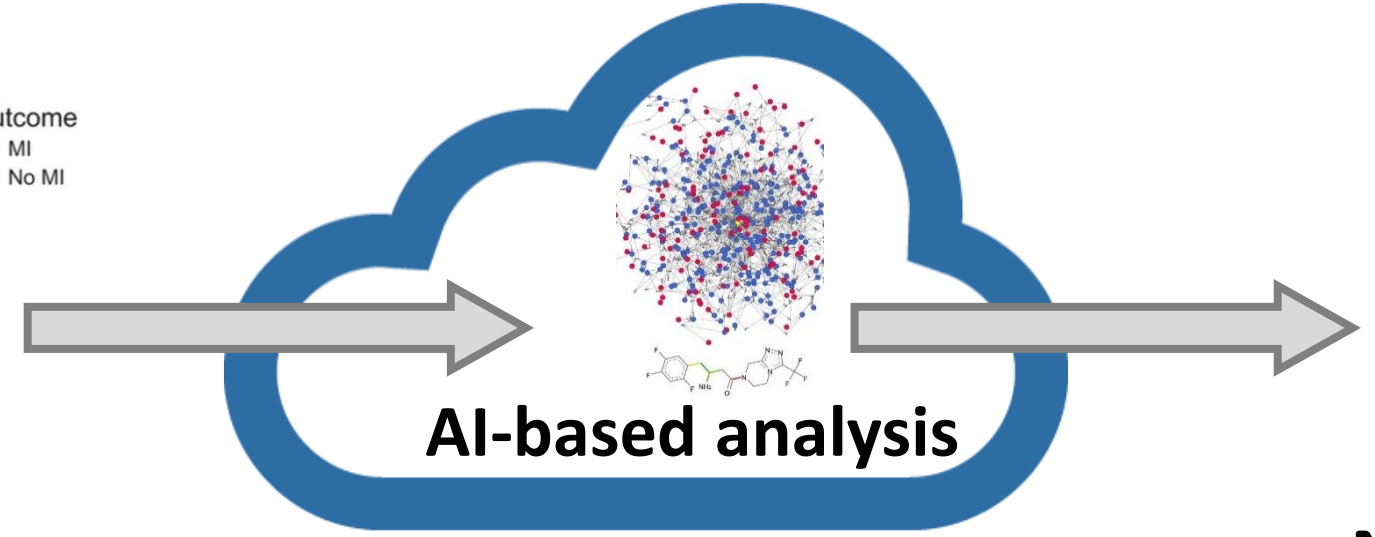
Basic AI/ML Service Architecture



Basic AI/ML Service Architecture



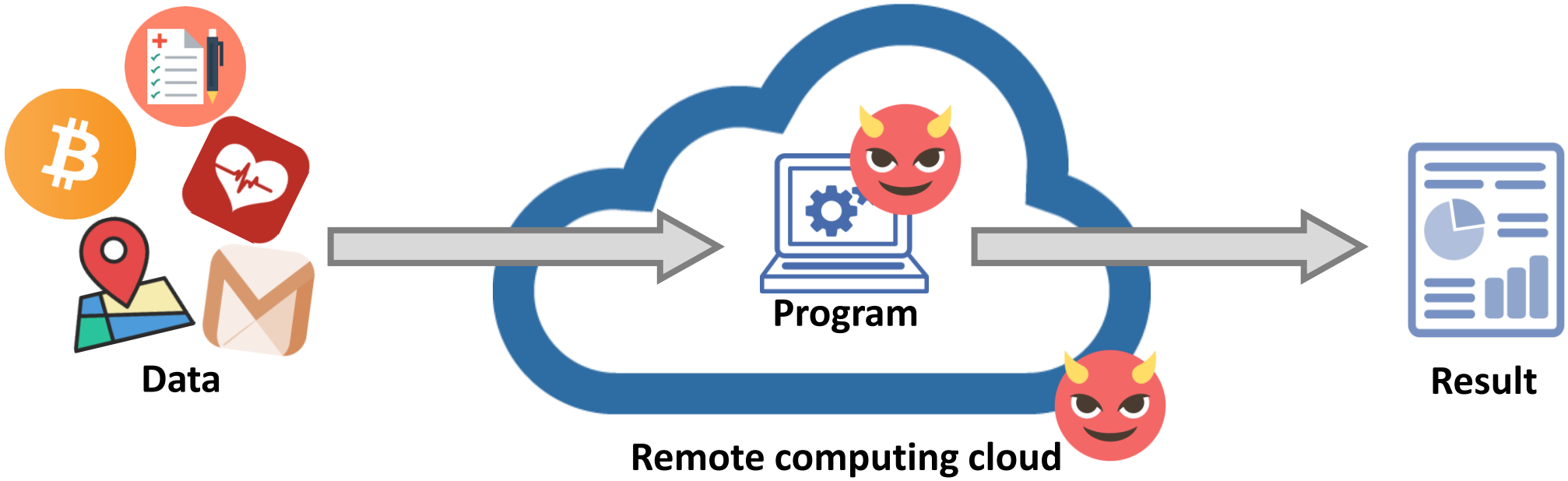
Patient medical records



Remote computing cloud

New drug discovery

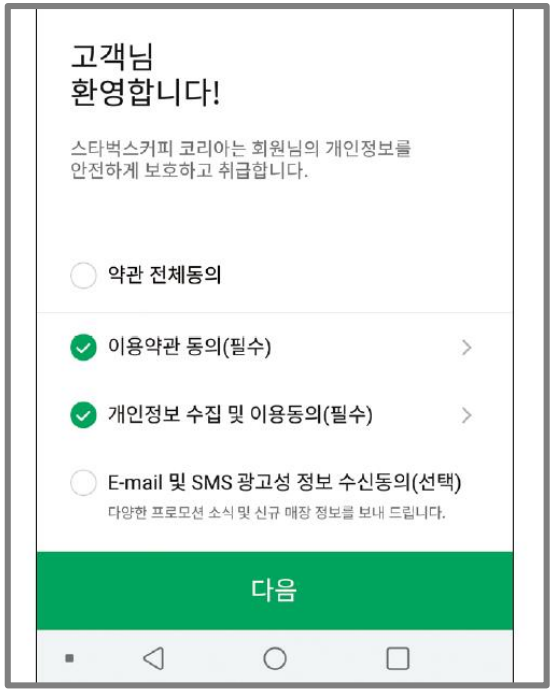
Security and Privacy Threats

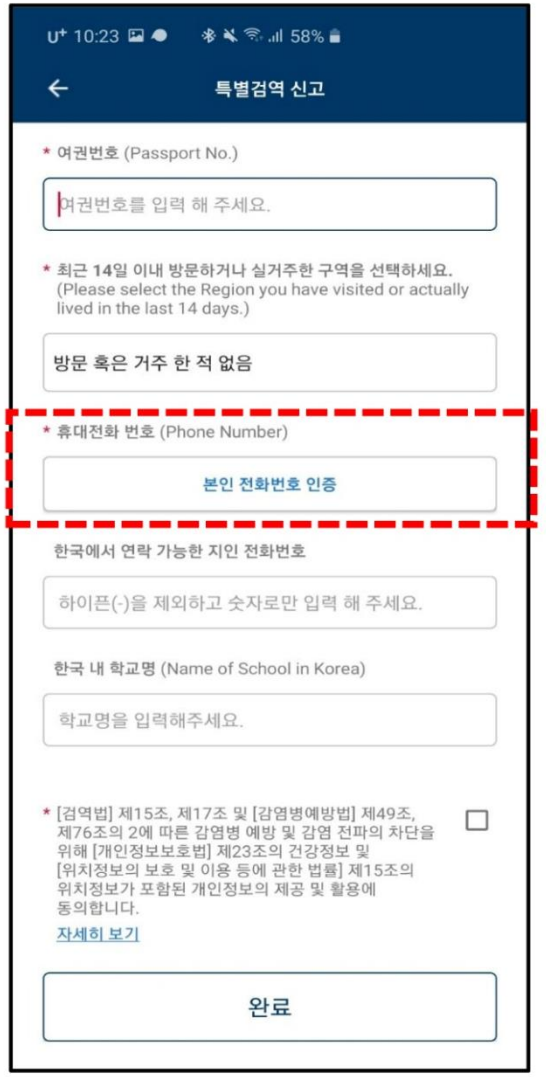


Data anarchy: Users have no control over their data

Challenges: Too strong attack models

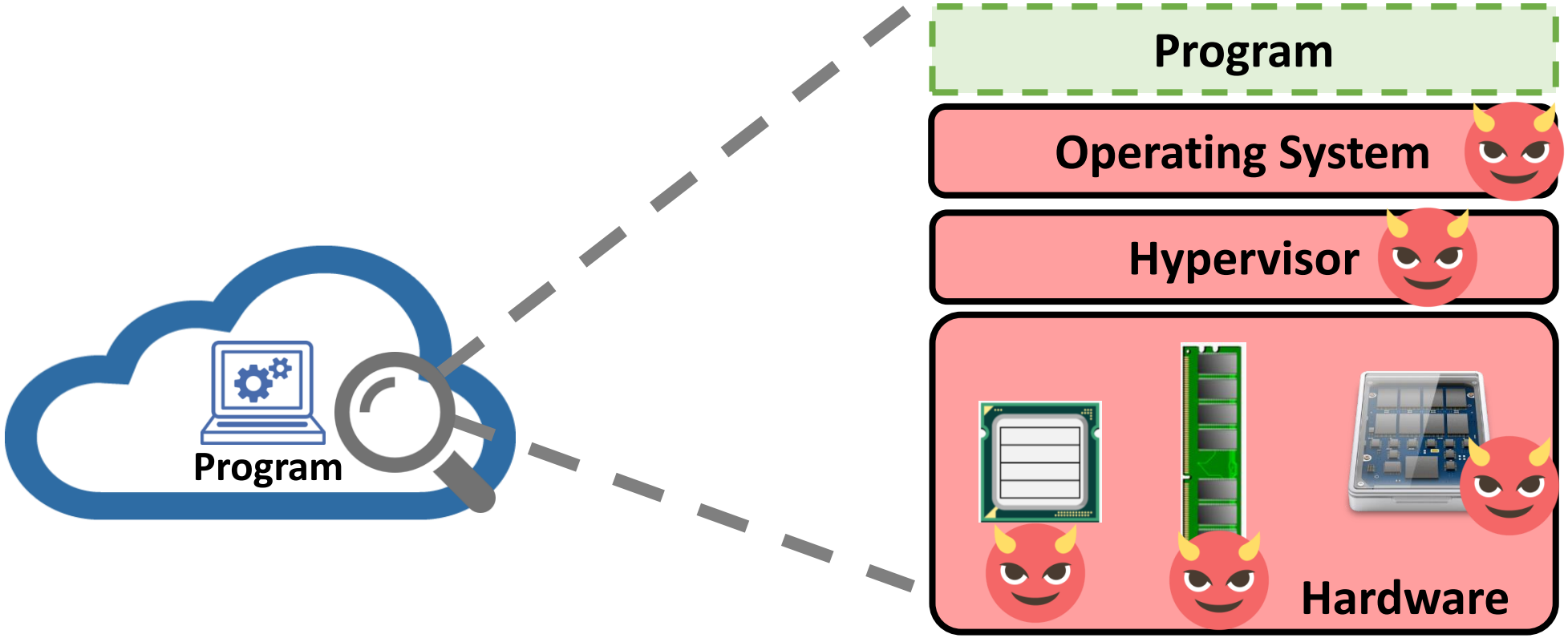
- A program (or program owners) can be malicious
  - A program may promise it would not abuse the data, but there's no technical enforcement





Challenges: Too strong attack models

- Cloud infrastructures can be malicious
  - Clouds include entire computing infrastructure to run a program
  - If any of those is malicious, user's data can be leaked

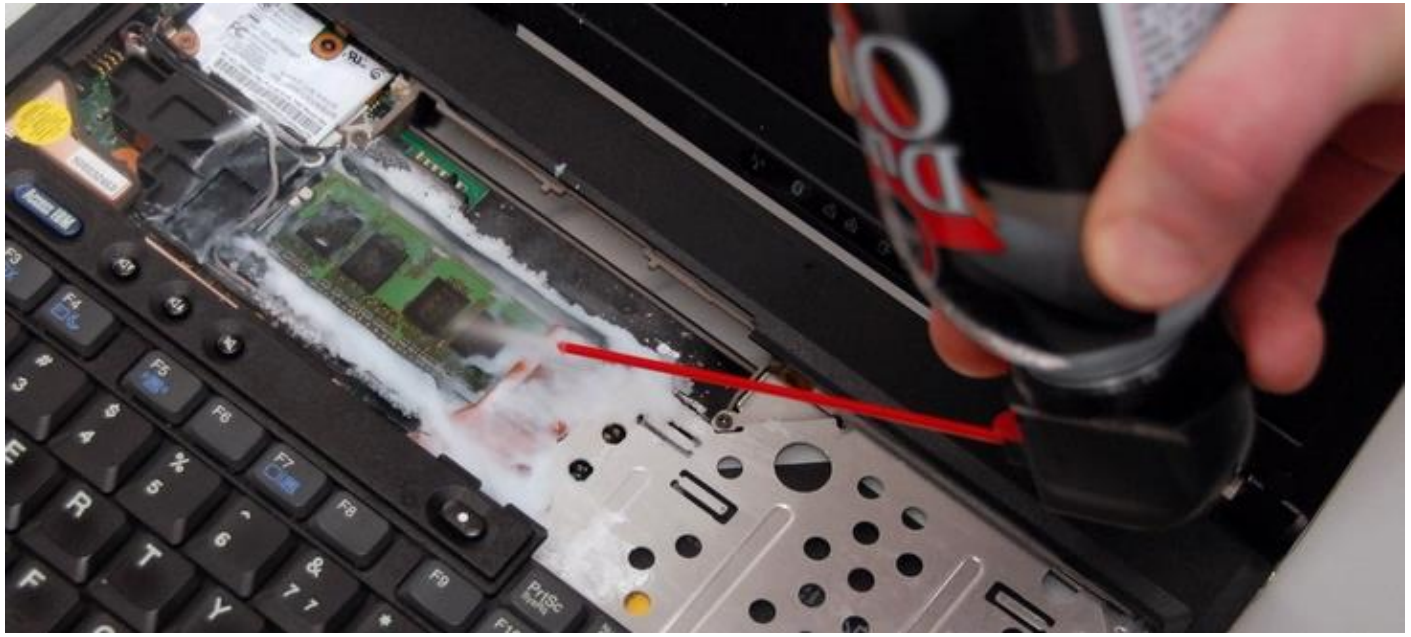


Challenges: Too strong attack models

- Clouds can be malicious
  - Physical attacks make this problem even more challenging
  - System admins can easily pull out the disk to read the data

Challenges: Too strong attack models

- Clouds can be malicious
  - Cold-boot attack: Even DRAM's data can be stolen



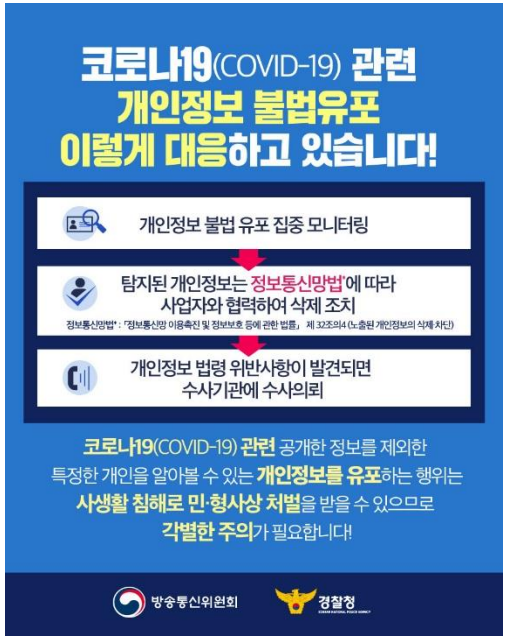
-50˚C: less than 0.2% decay after 1 minute

"Lest We Remember: Cold Boot Attacks on Encryption Keys [USENIX Security 08]"

Fundamental Issue: Data Utility vs. Data Privacy

- **Data utility**
  - Data is the key to truly enable AI/ML/DL services
- **Data privacy**
  - Data contains critical privacy information of users

- How to satisfy both **data utility** and **data privacy**?



Potential Solutions for Data Security

- Data anonymization (데이터 비식별화)
- Homomorphic Encryption (동형암호)
- **Hardware-Assisted Trusted Computing (신뢰계산)**

Data Anonymization (데이터 비식별화)

- Remove personally identifiable information from data
  - While maintaining the data utilization
- k-anonymity
  - Blend each data item with k-1 items having identical column information

microdata

id	Zipcode	Sex	National.	Disease
1	13053	28	Russian	Heart Disease
2	13068	29	American	Heart Disease
3	13068	21	Japanese	Viral Infection
4	13053	23	American	Viral Infection
5	14853	50	Indian	Cancer
6	14853	55	Russian	Heart Disease
7	14850	47	American	Viral Infection
8	14850	49	American	Viral Infection
9	13053	31	American	Cancer
10	13053	37	Indian	Cancer
11	13068	36	Japanese	Cancer
12	13068	35	American	Cancer

4-anonymous data

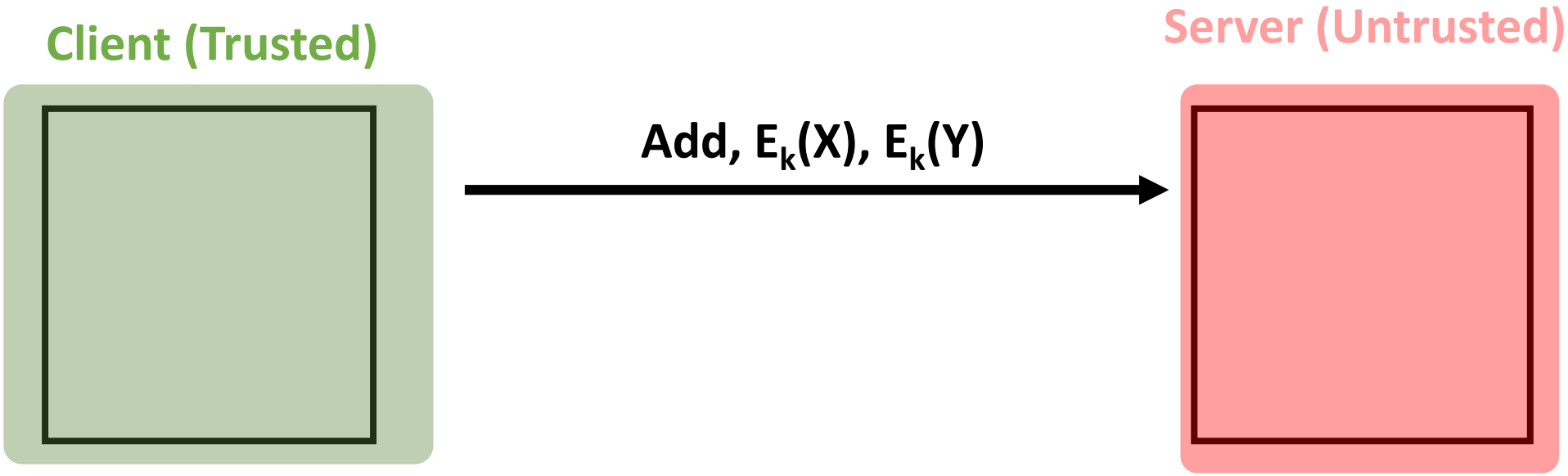
id	Zipcode	Sex	National.	Disease
1	130**	<30	*	Heart Disease
2	130**	<30	*	Heart Disease
3	130**	<30	*	Viral Infection
4	130**	<30	*	Viral Infection
5	1485*	≥40	*	Cancer
6	1485*	≥40	*	Heart Disease
7	1485*	≥40	*	Viral Infection
8	1485*	≥40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

Homomorphic Encryption (동형암호)

- Computation over encrypted data
  - Example: Client wants to offload the computation, X+Y

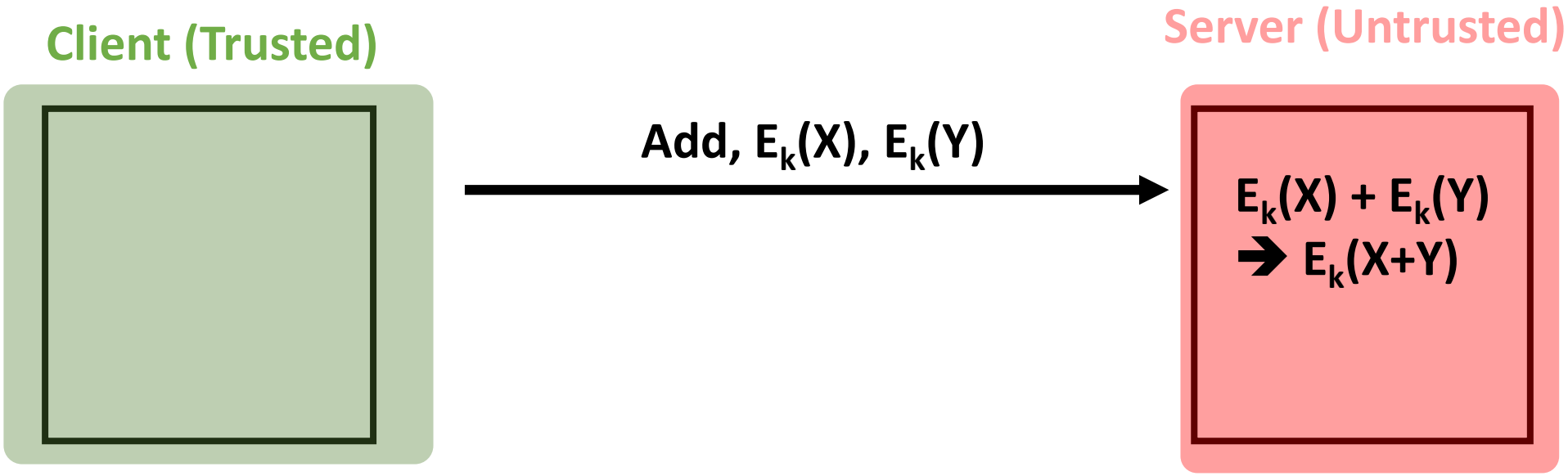
Homomorphic Encryption (동형암호)

- Computation over encrypted data
  - Example: Client wants to offload the computation, X+Y



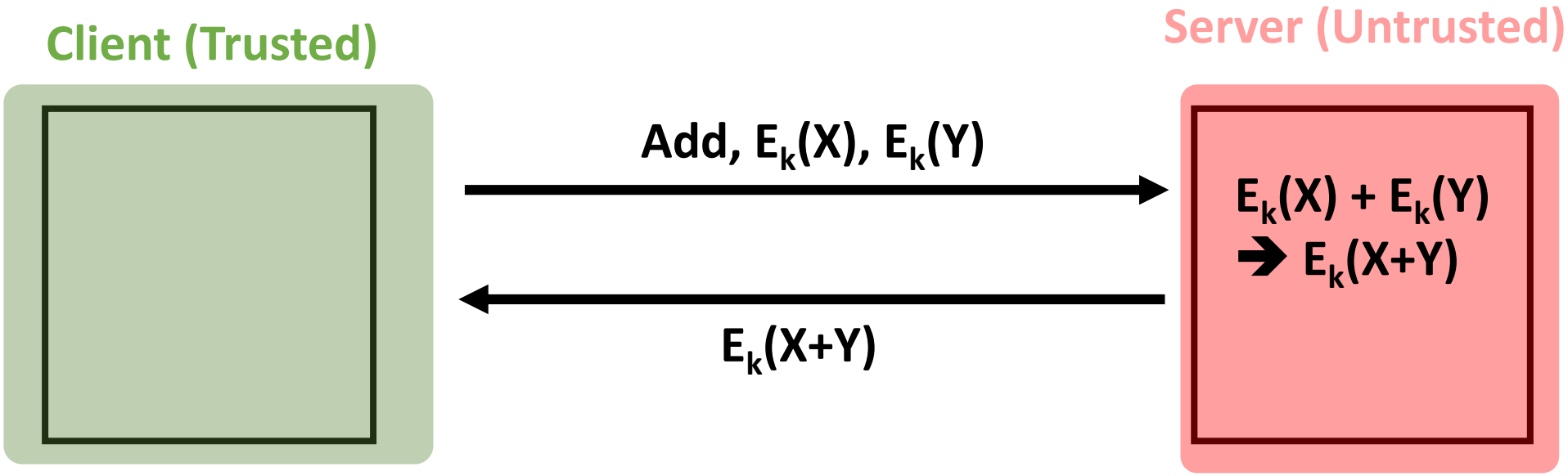
Homomorphic Encryption (동형암호)

- Computation over encrypted data
  - Example: Client wants to offload the computation, X+Y



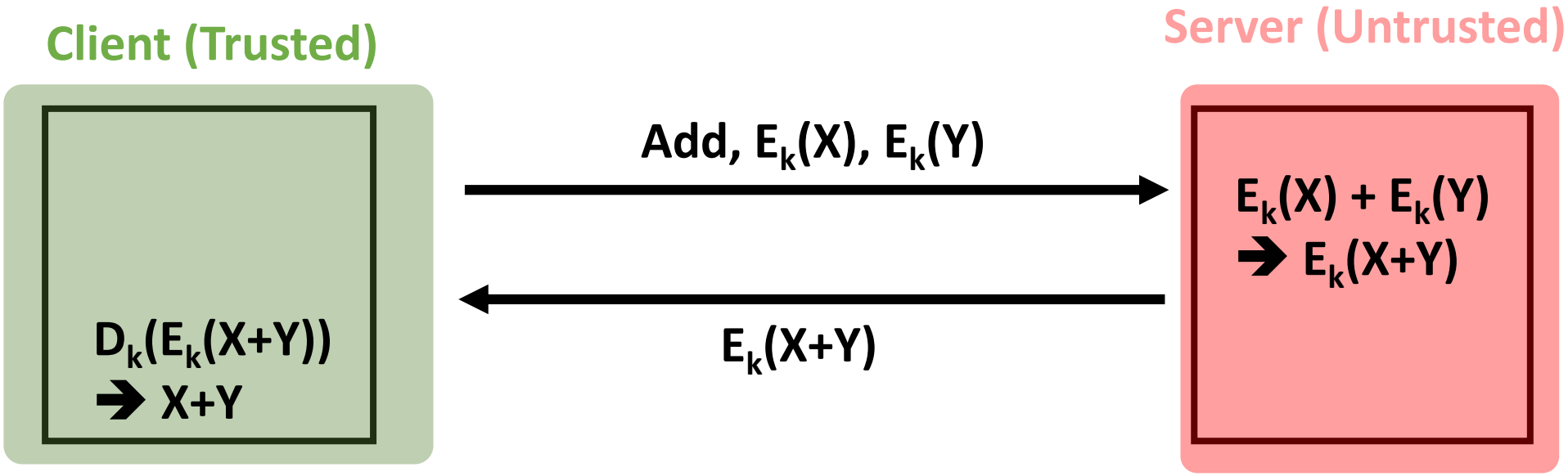
Homomorphic Encryption (동형암호)

- Computation over encrypted data
  - Example: Client wants to offload the computation, X+Y



Homomorphic Encryption (동형암호)

- Computation over encrypted data
  - Example: Client wants to offload the computation, X+Y

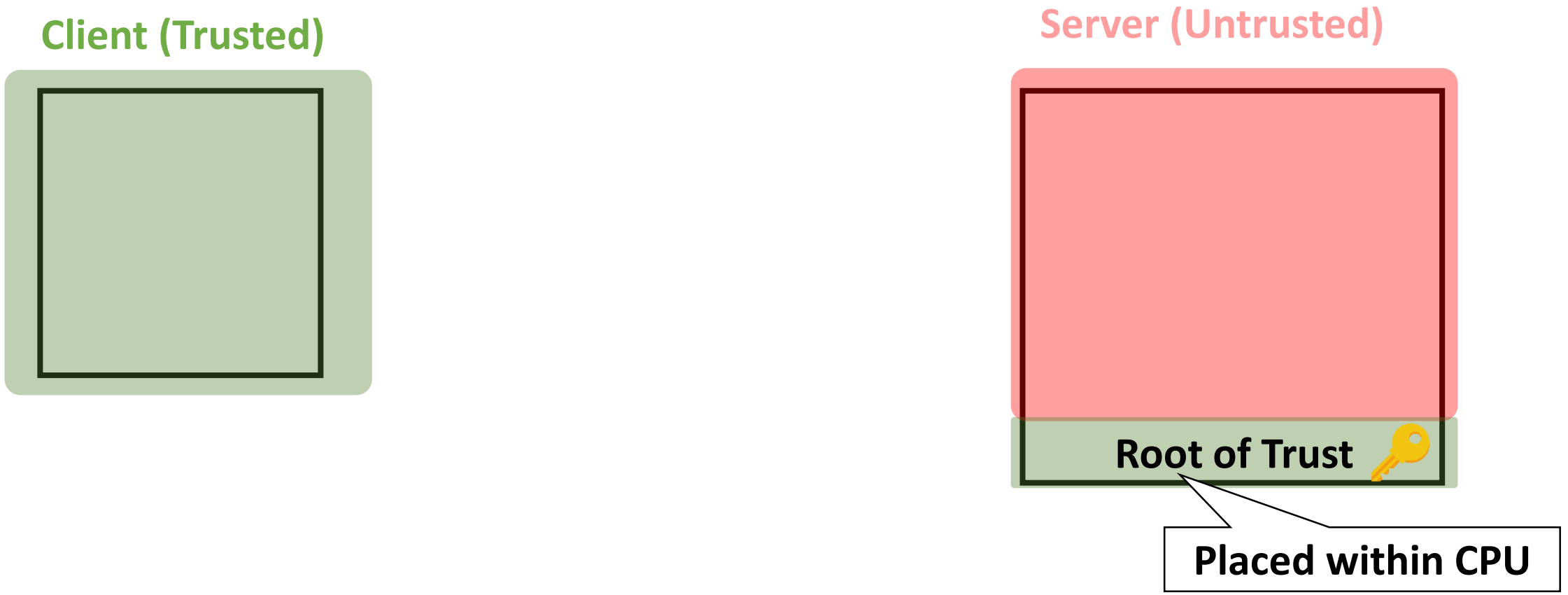


Hardware-Assisted Trusted Computing (신뢰계산)

- Trusted computation by placing a small root of trust in hardware

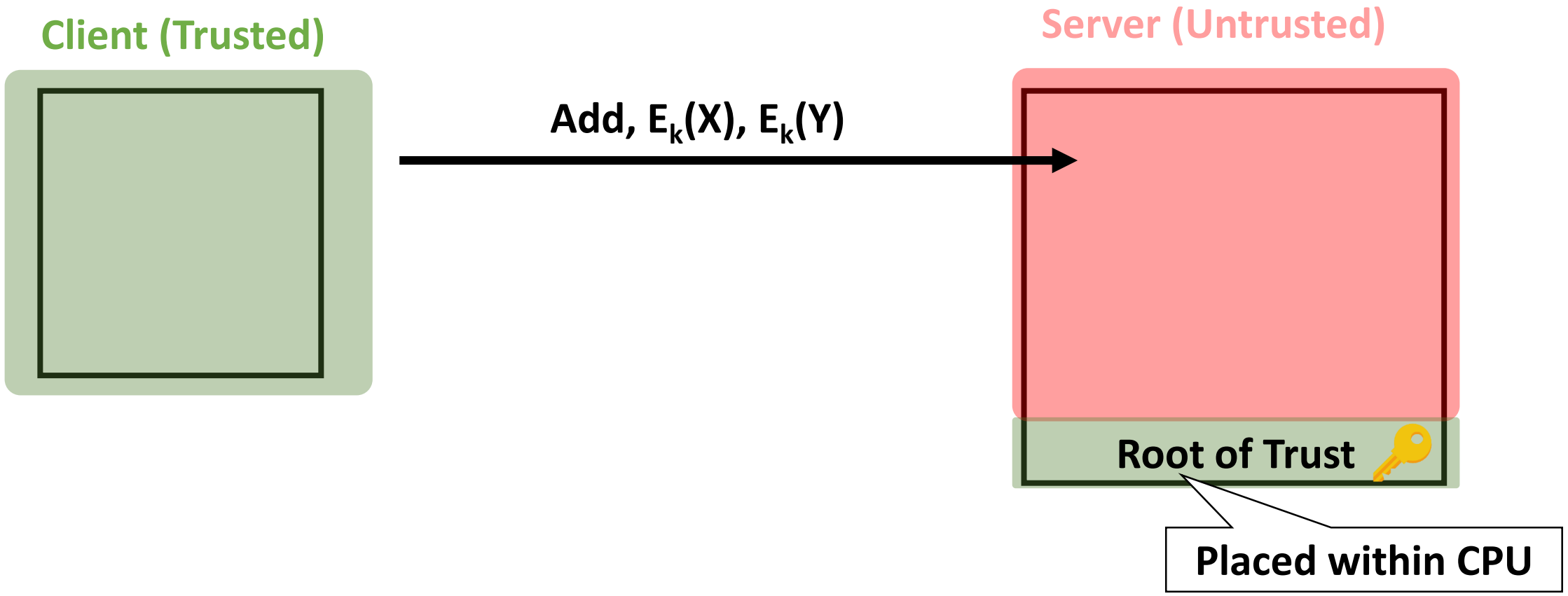
Hardware-Assisted Trusted Computing (신뢰계산)

- Trusted computation by placing a small root of trust in hardware



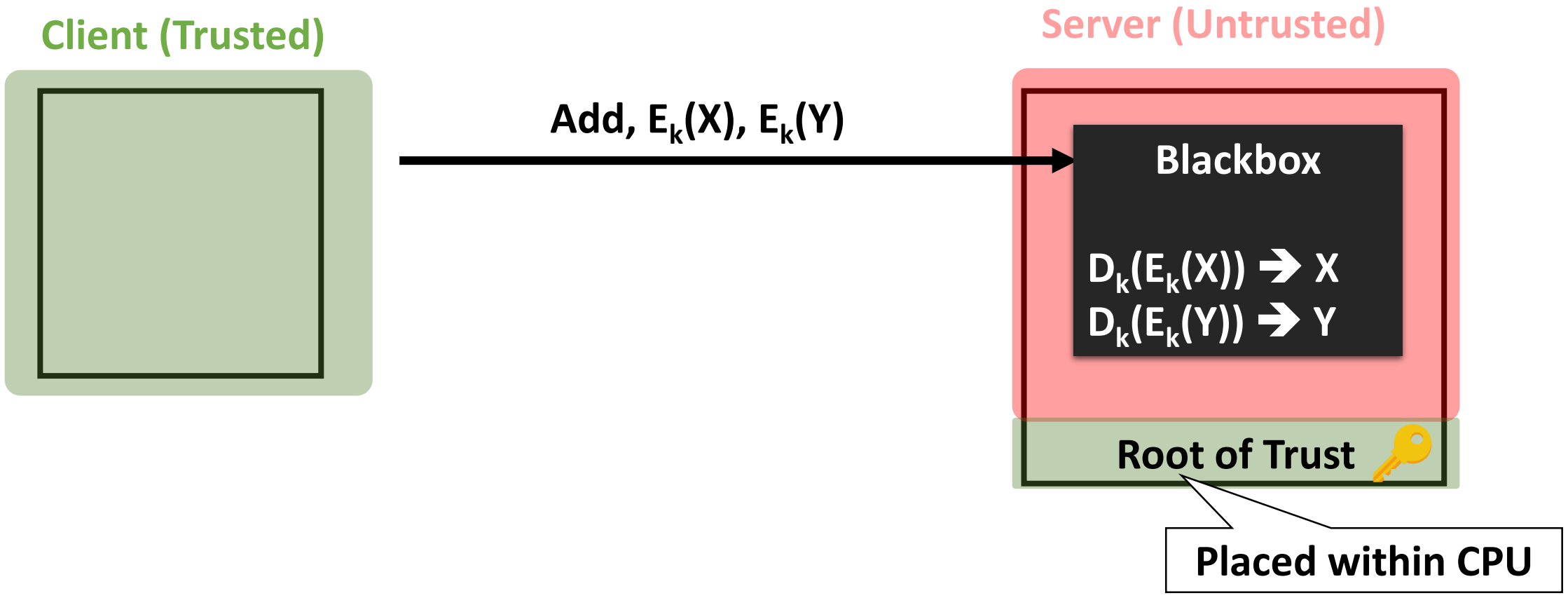
Hardware-Assisted Trusted Computing (신뢰계산)

- Trusted computation by placing a small root of trust in hardware



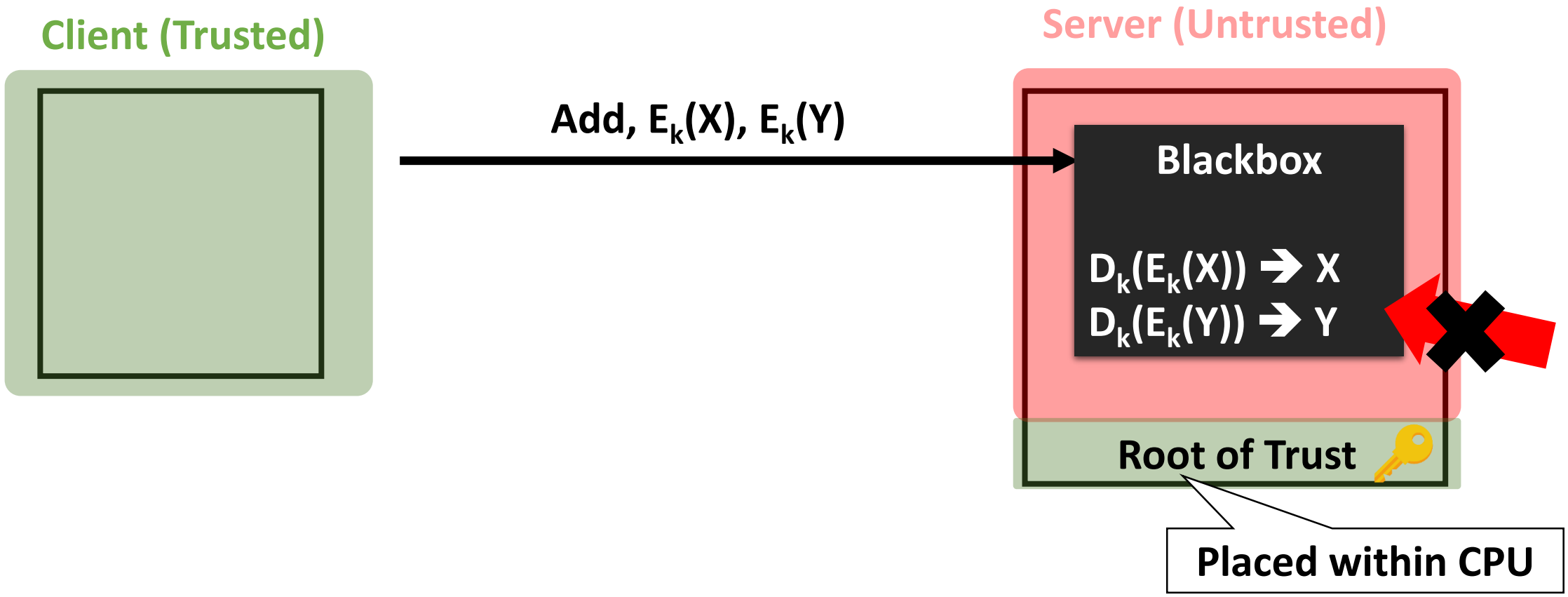
Hardware-Assisted Trusted Computing (신뢰계산)

- Trusted computation by placing a small root of trust in hardware



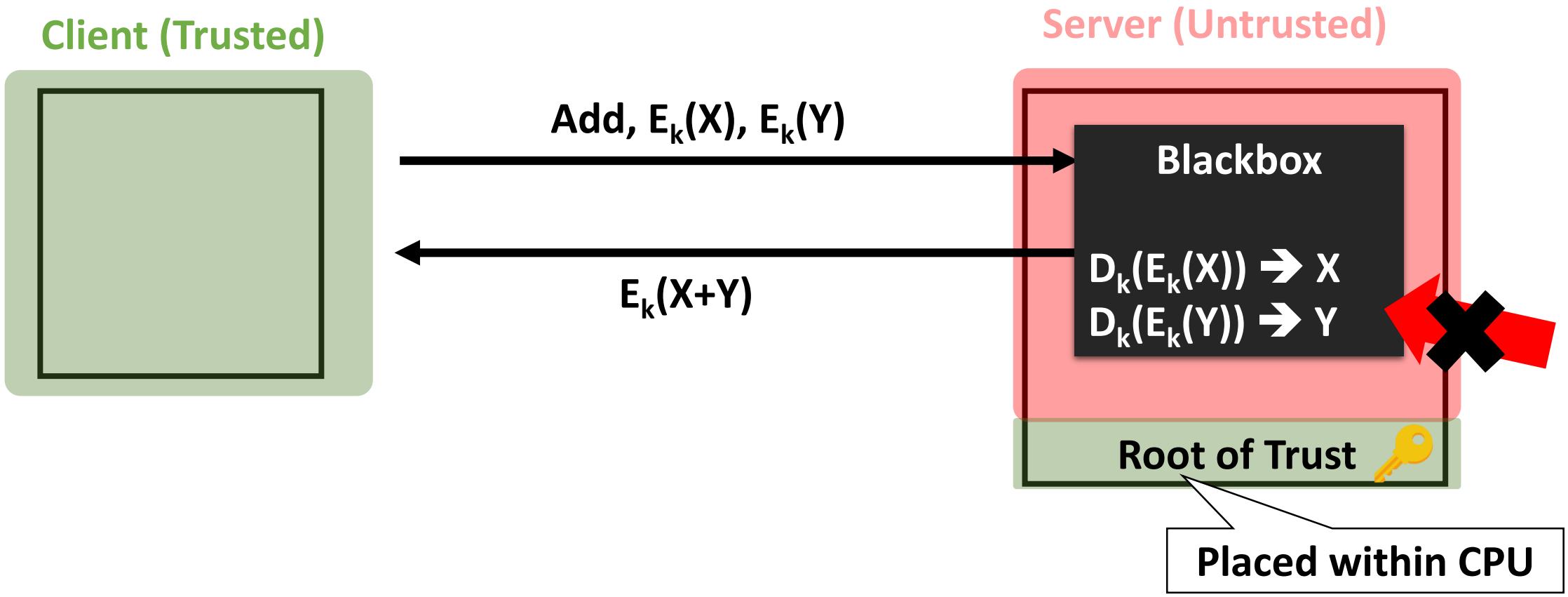
Hardware-Assisted Trusted Computing (신뢰계산)

- Trusted computation by placing a small root of trust in hardware



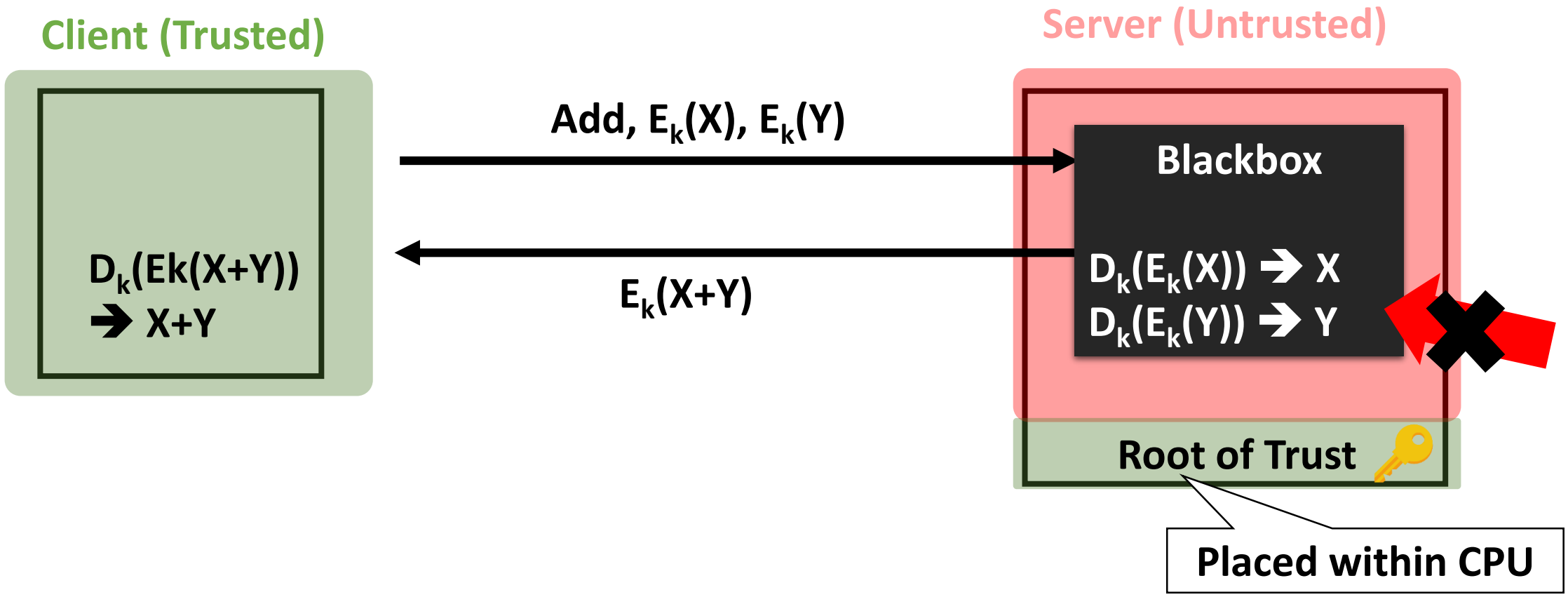
Hardware-Assisted Trusted Computing (신뢰계산)

- Trusted computation by placing a small root of trust in hardware

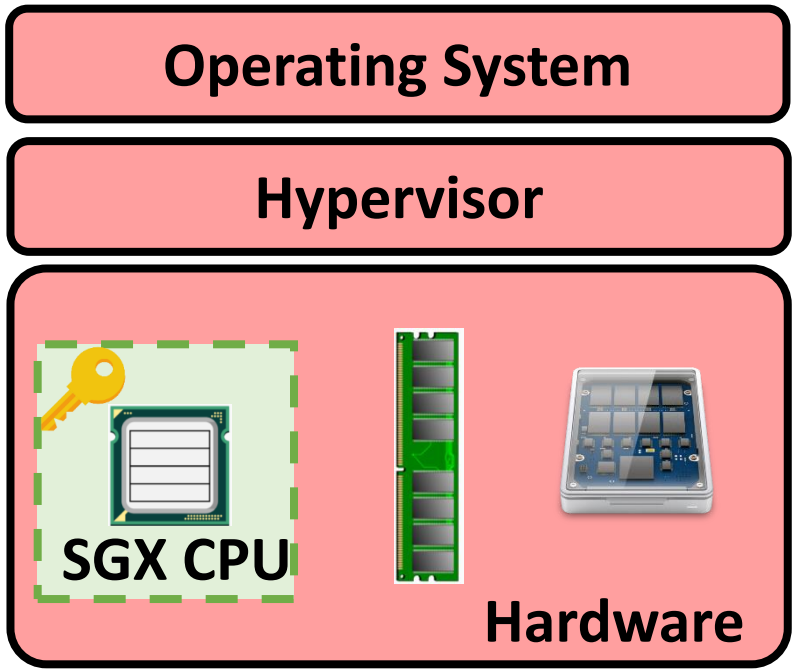


Hardware-Assisted Trusted Computing (신뢰계산)

- Trusted computation by placing a small root of trust in hardware

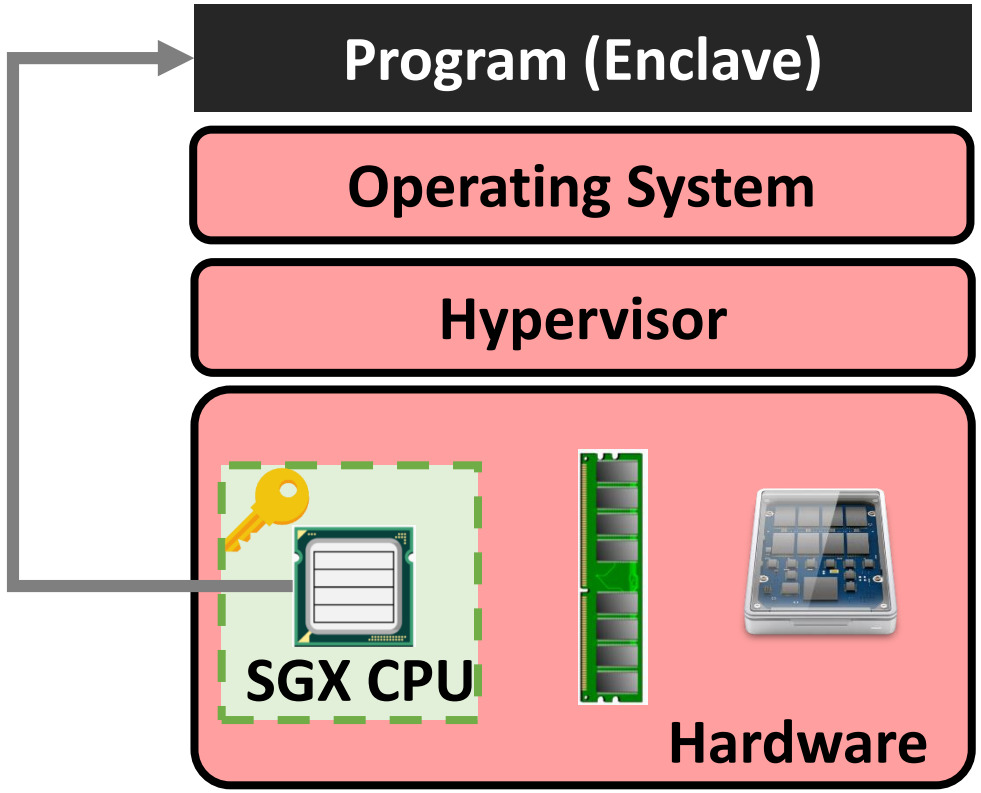


Intel SGX: Data Security Feature for the Future



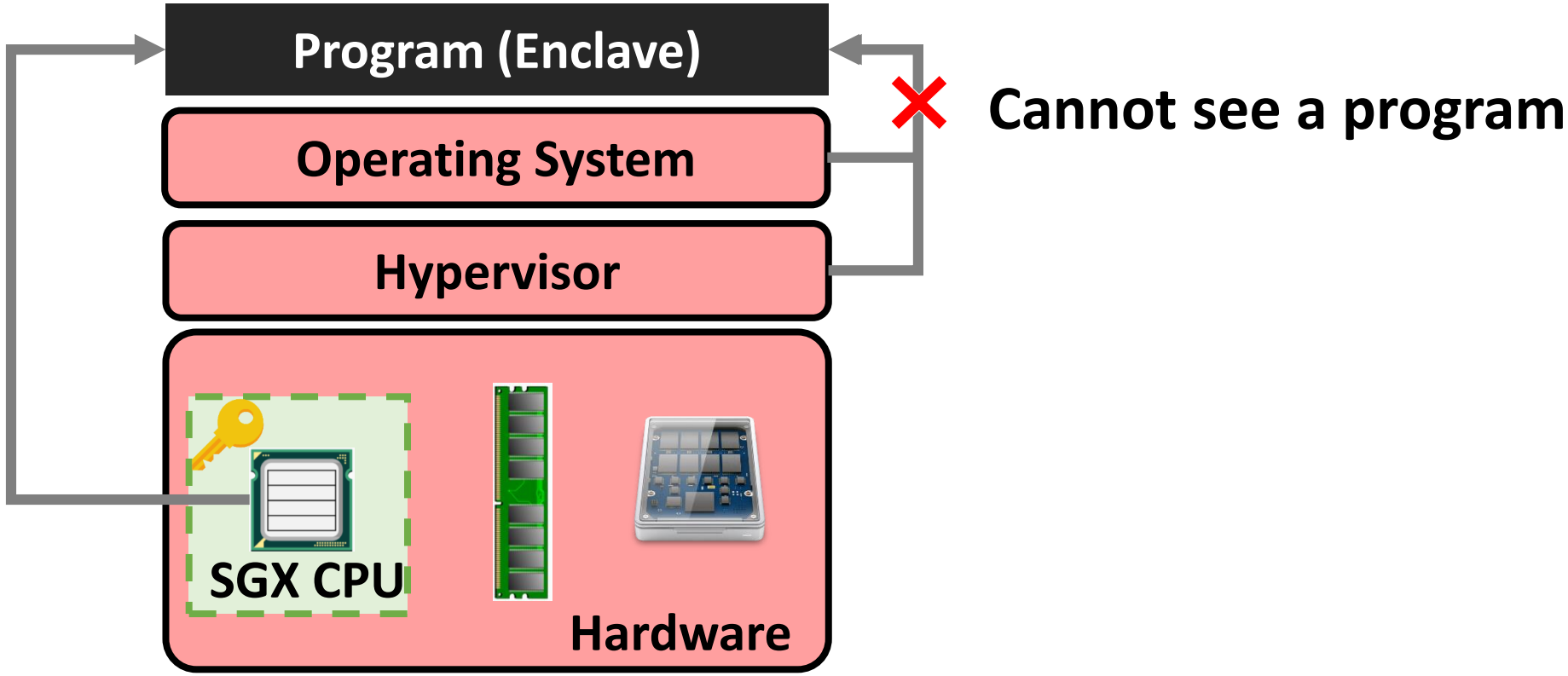
Intel SGX: Data Security Feature for the Future

Hardware-protected execution region



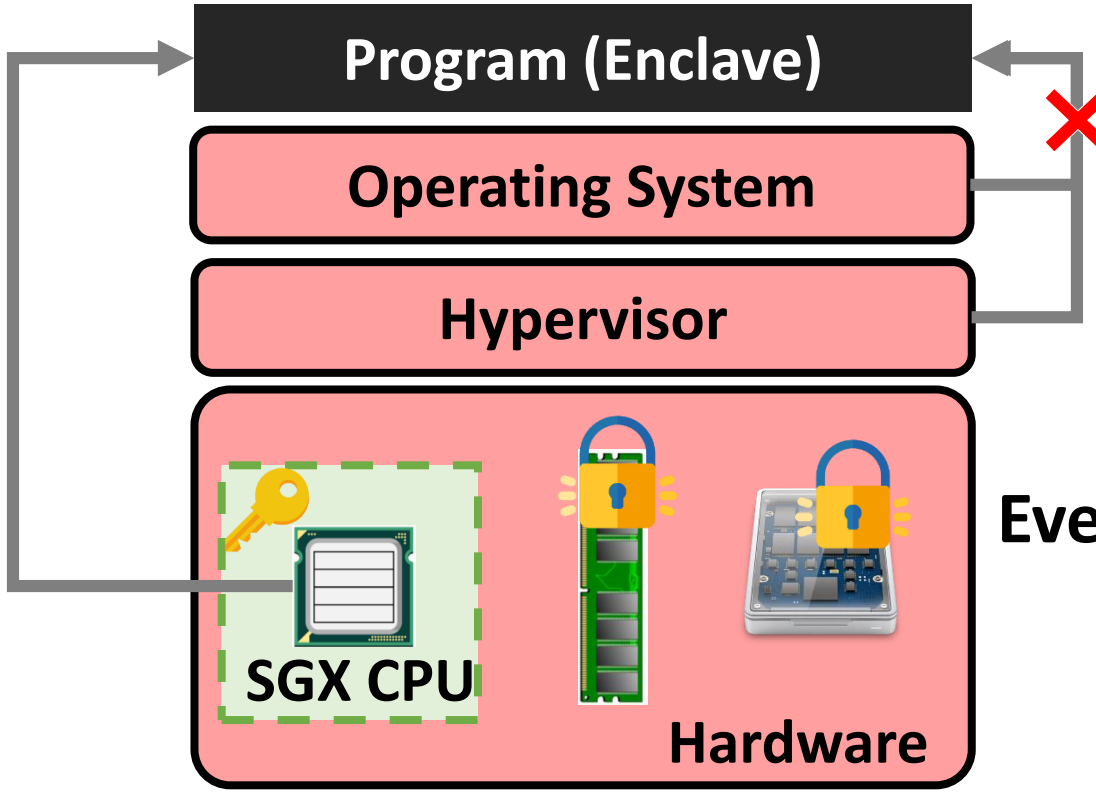
Intel SGX: Data Security Feature for the Future

Hardware-protected execution region



Intel SGX: Data Security Feature for the Future

Hardware-protected execution region

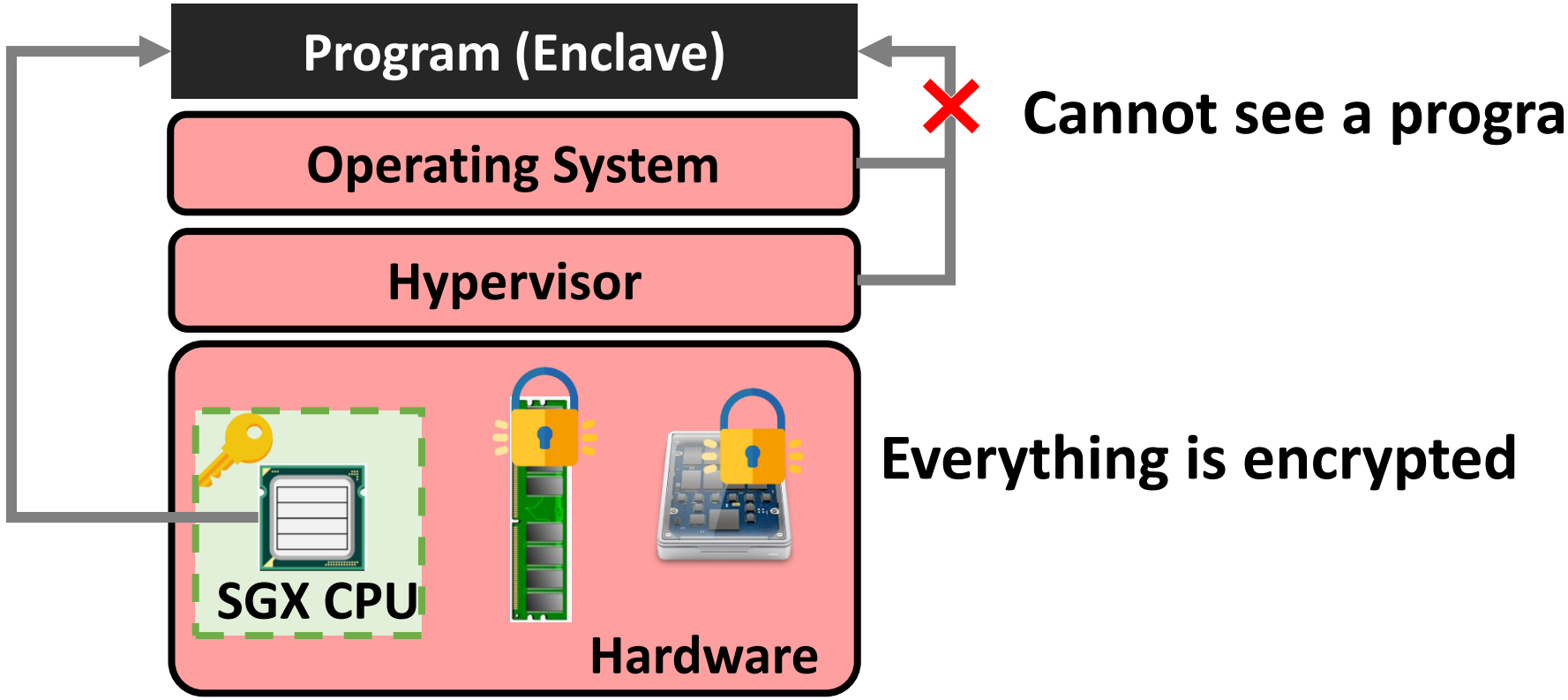


Cannot see a program

Everything is encrypted

Intel SGX: Data Security Feature for the Future

Hardware-protected execution region

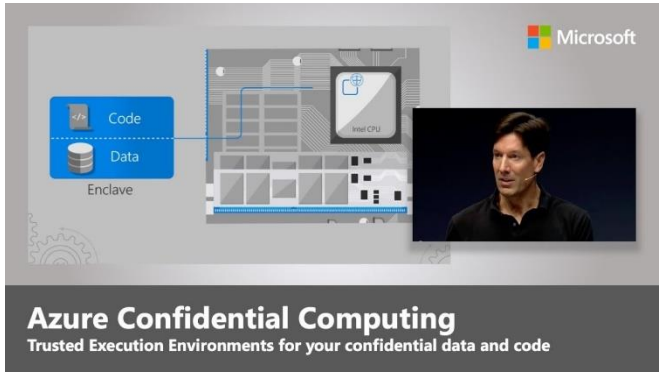


Most Intel CPUs today are shipped with SGX support.

Intel SGX: already market available

- Most of consumer-grade Intel CPUs are shipped with SGX support

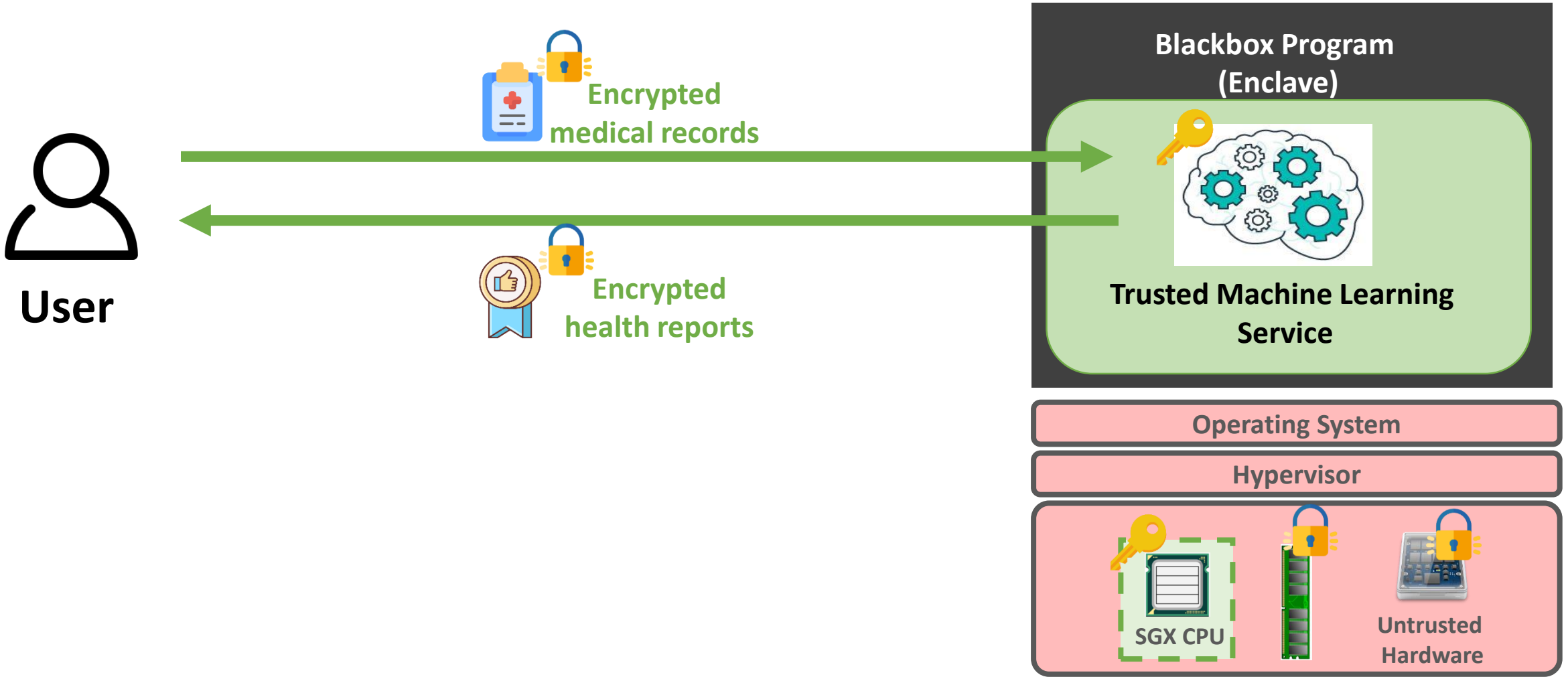
- Strong demands on SGX features from cloud providers
  - Growing security needs for trusted computing
    - Observing EU GDPR and any (expected) national regulation
  - Azure Confidential Computing is already available (since 2020 May)
    - SGX-based secure cloud services



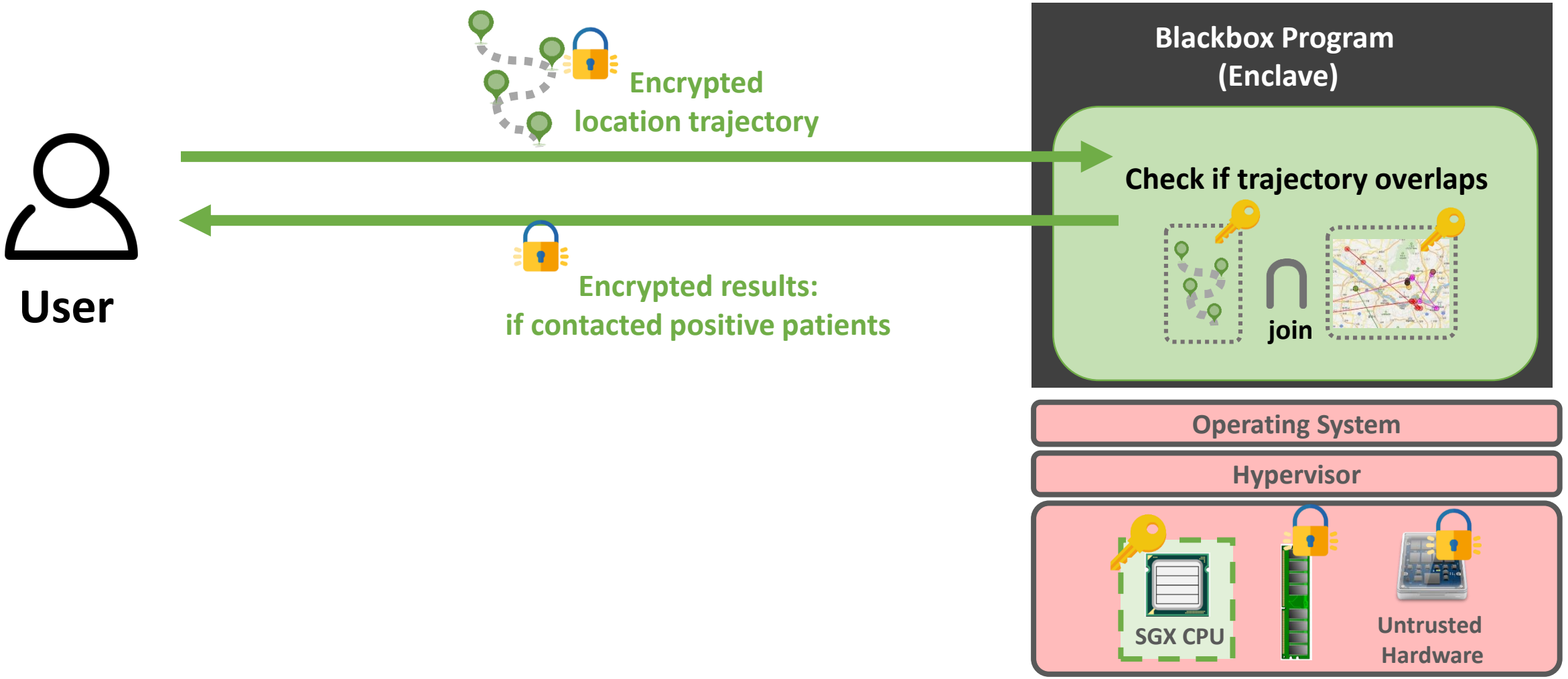
Expected Future Security Applications

- Trusted Machine Learning
  - 예제: 안전한 AI 기반 건강관리 서비스
- Trusted Private Join
  - 예제: 개인정보를 보호하는 코로나바이러스 환자 동선 확인
- Trusted Network Middleware/Server
  - 예제: 안전한 화상회의 아키텍쳐 (Zoom, Google Meet)
- Trusted Coin Mining for Blockchain Network
  - 예제: Scalable blockchain network

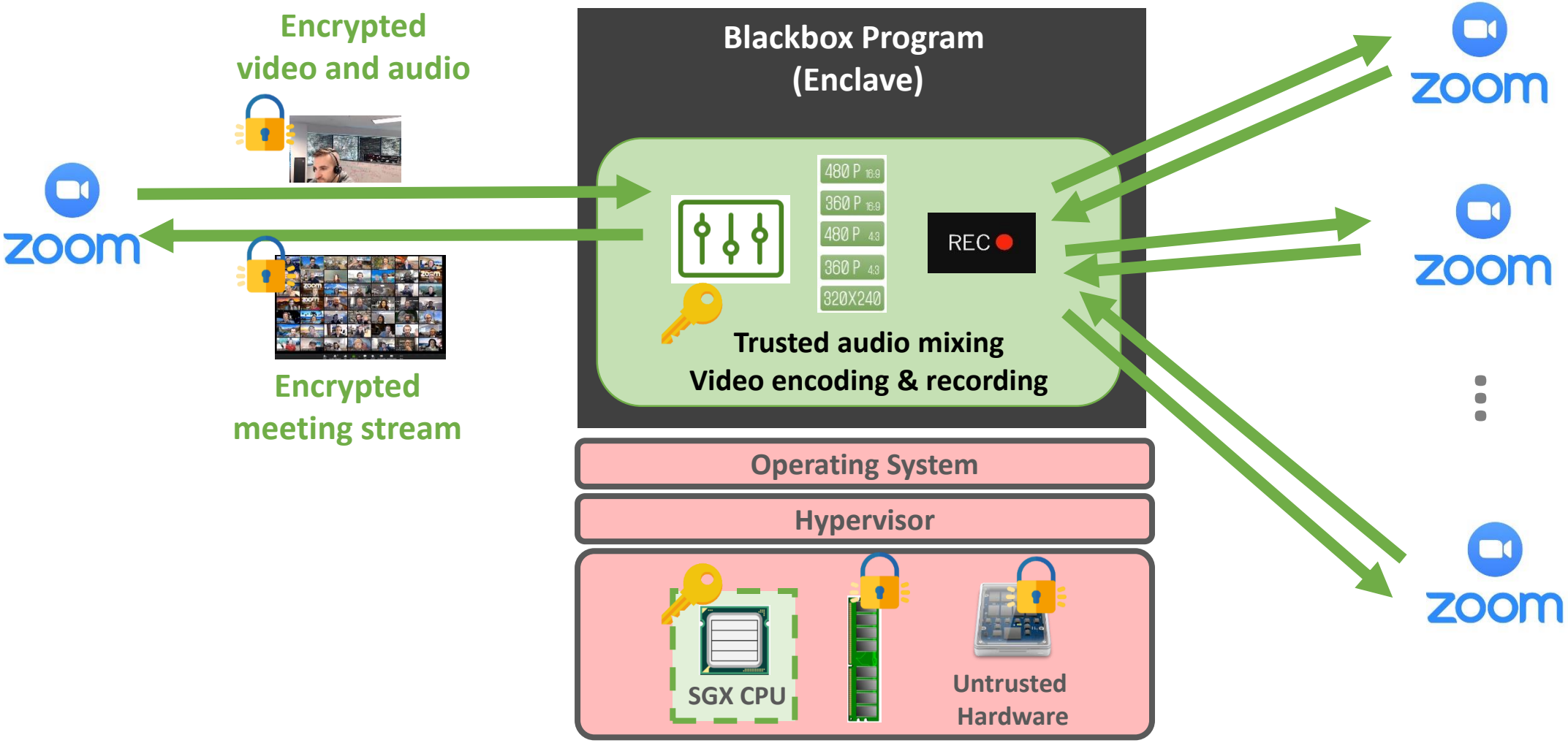
Trusted Machine Learning: Health Prediction



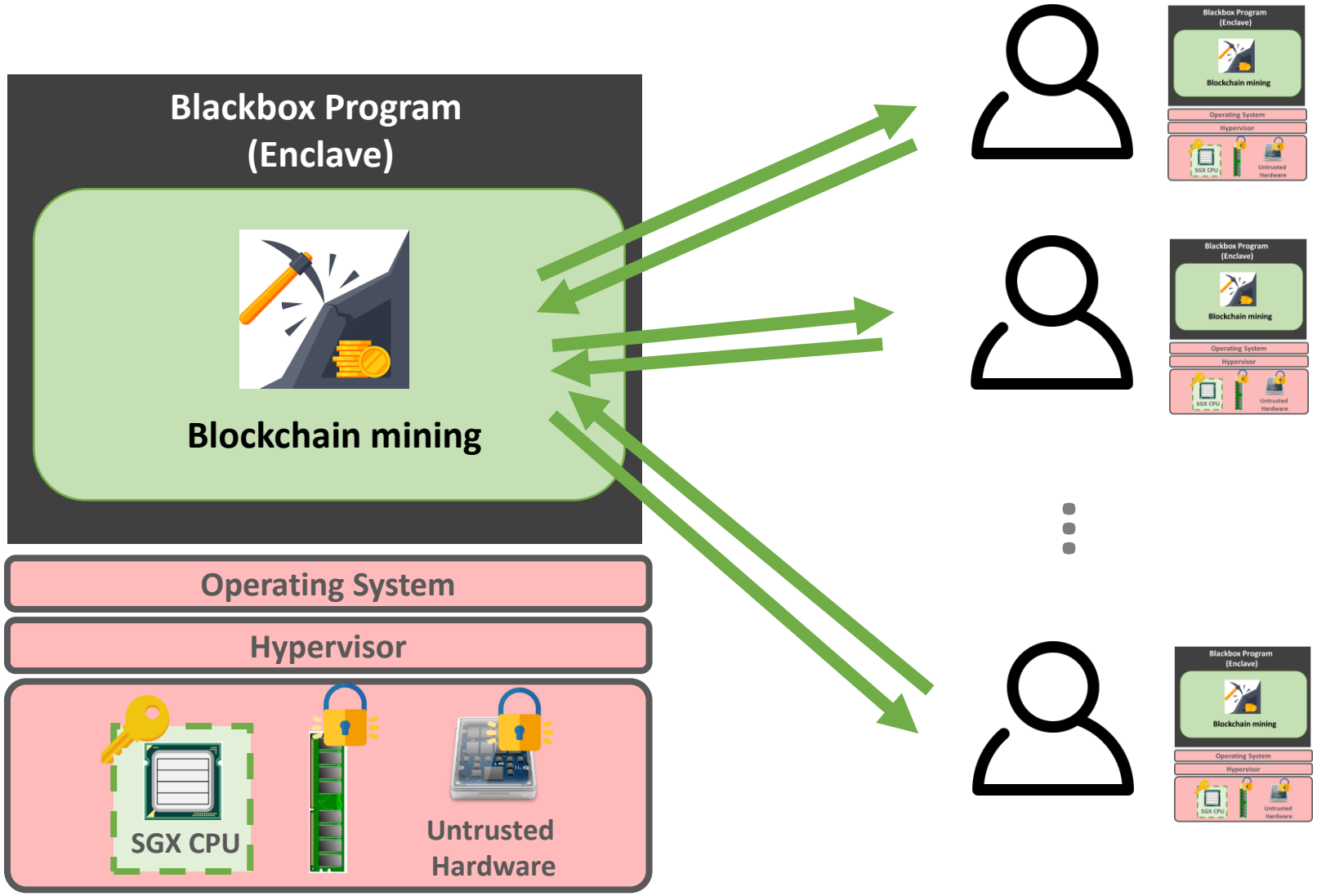
Trusted Private Join: Covid-19 Proximity Check



Trusted Network Server: Trusted Online Meeting



Trusted Coin Mining for Blockchain



Conclusion

- **Hacking**
  - No single solution
- **Data Security**
  - Encrypt everything if possible
  - Minimize the attack surface

감사합니다

서울대학교
공과대학 전기정보공학부
이병영
byoungyoung@snu.ac.kr