CAFE: A Virtualization-Based Approach to Protecting Sensitive Cloud Application Logic Confidentiality

Chung Hwan Kim
Dongyan Xu

Purdue University

Sungjin Park
Jong-jin Won

The Attached Institute of ETRI

Junghwan Rhee

NEC Laboratories America

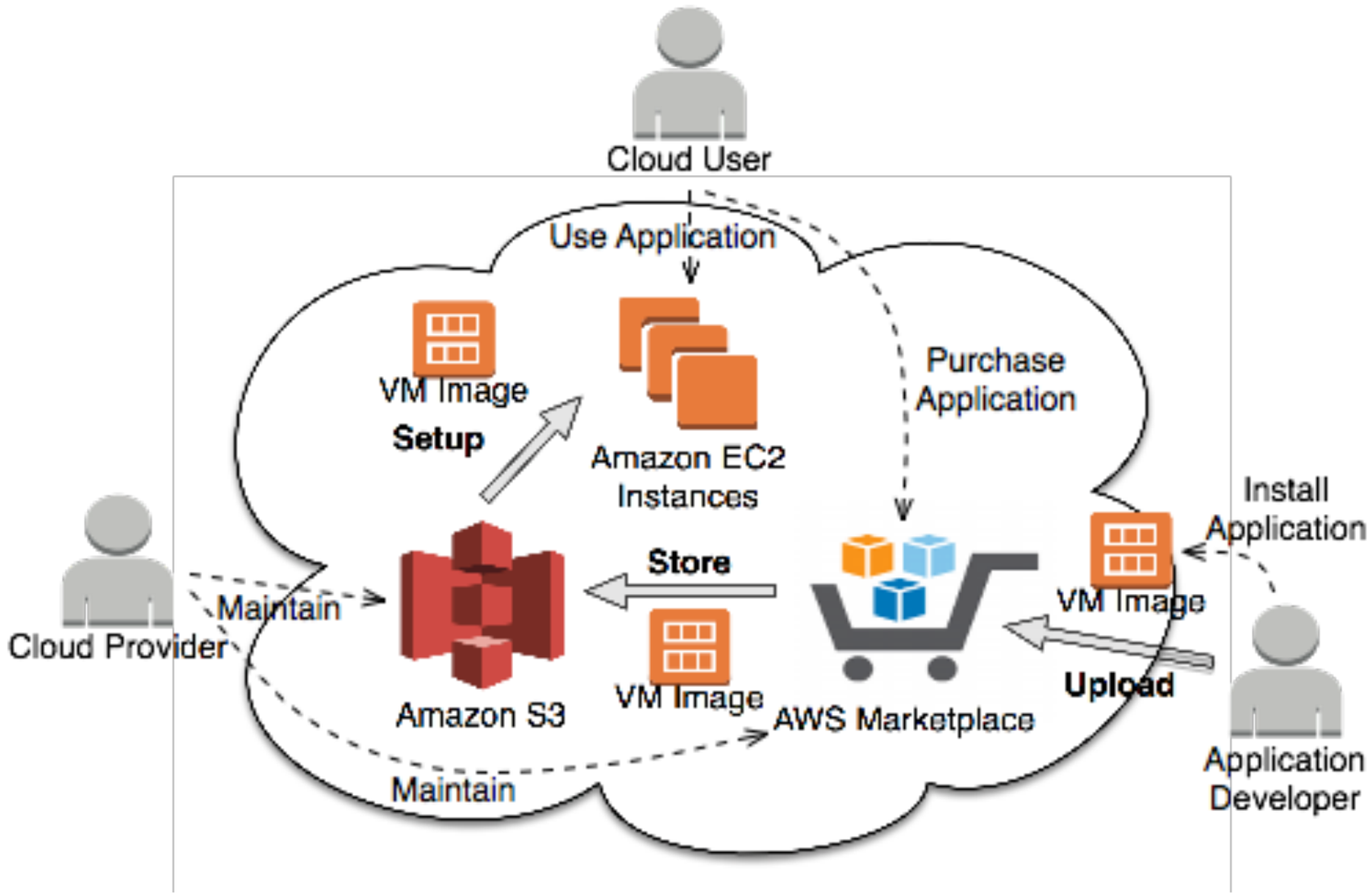
Taisook Han

KAIST

Outline

- Background and Challenges
- CAFE Framework
- Evaluation
- Related Work
- Conclusion

Background: Cloud Marketplace



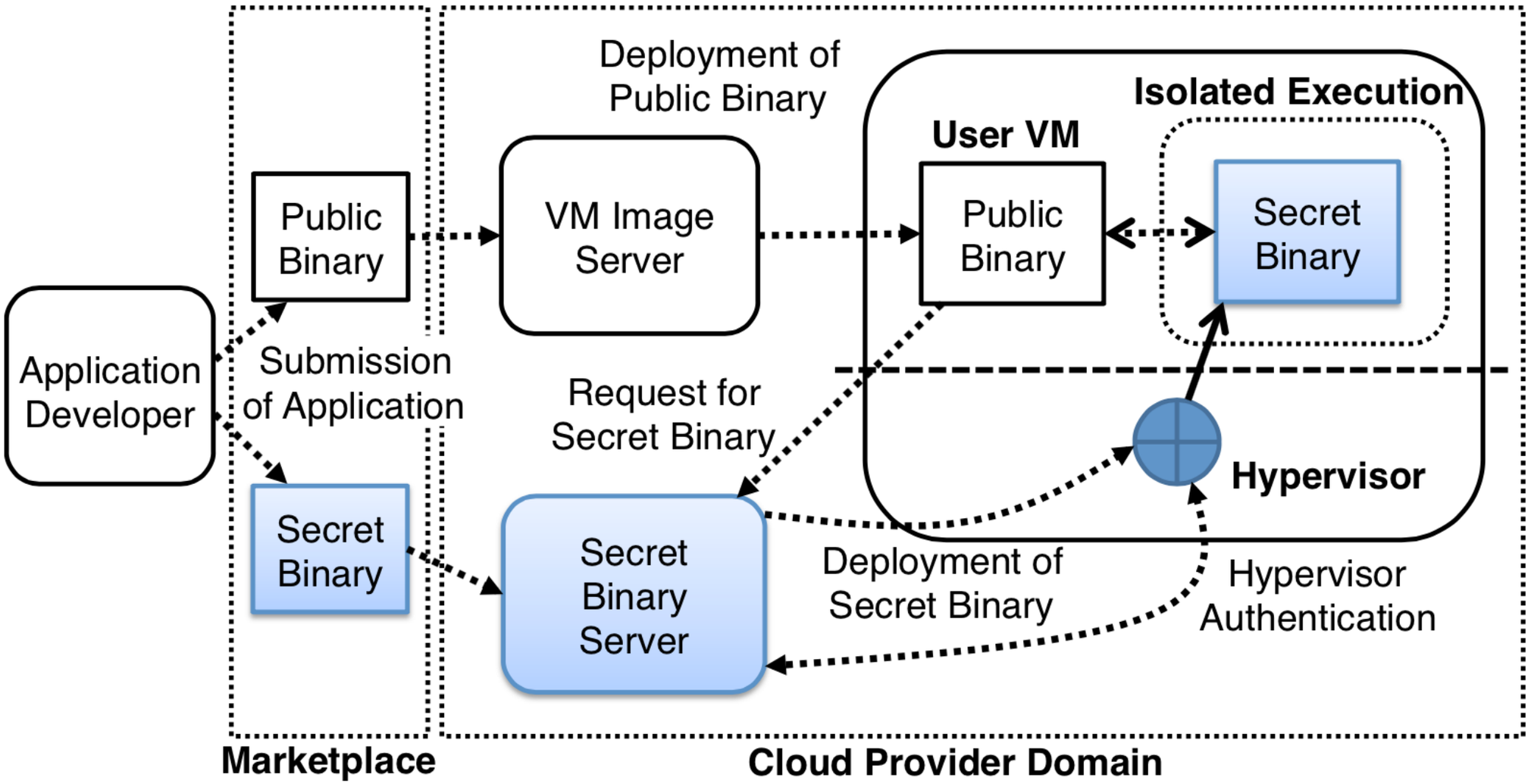
Challenges

- The deployed software faces the risk of piracy and reverse engineering.
- Cloud tenants can easily access the binary of software deployed in the guest VMs.
  - E.g., file access, debugger, memory dump, etc.
  - They can deploy the same applications without the marketplace.

Goals

- Secure execution of sensitive application logic confidential to user VMs
  - A cloud user with admin privilege cannot obtain the sensitive application logic.
- Scalable and practical distribution of secret binaries for cloud marketplaces
  - Content of the binary remains confidential end-to-end from its submission to its execution.

Design of CAFE



Creation of Secret Binary

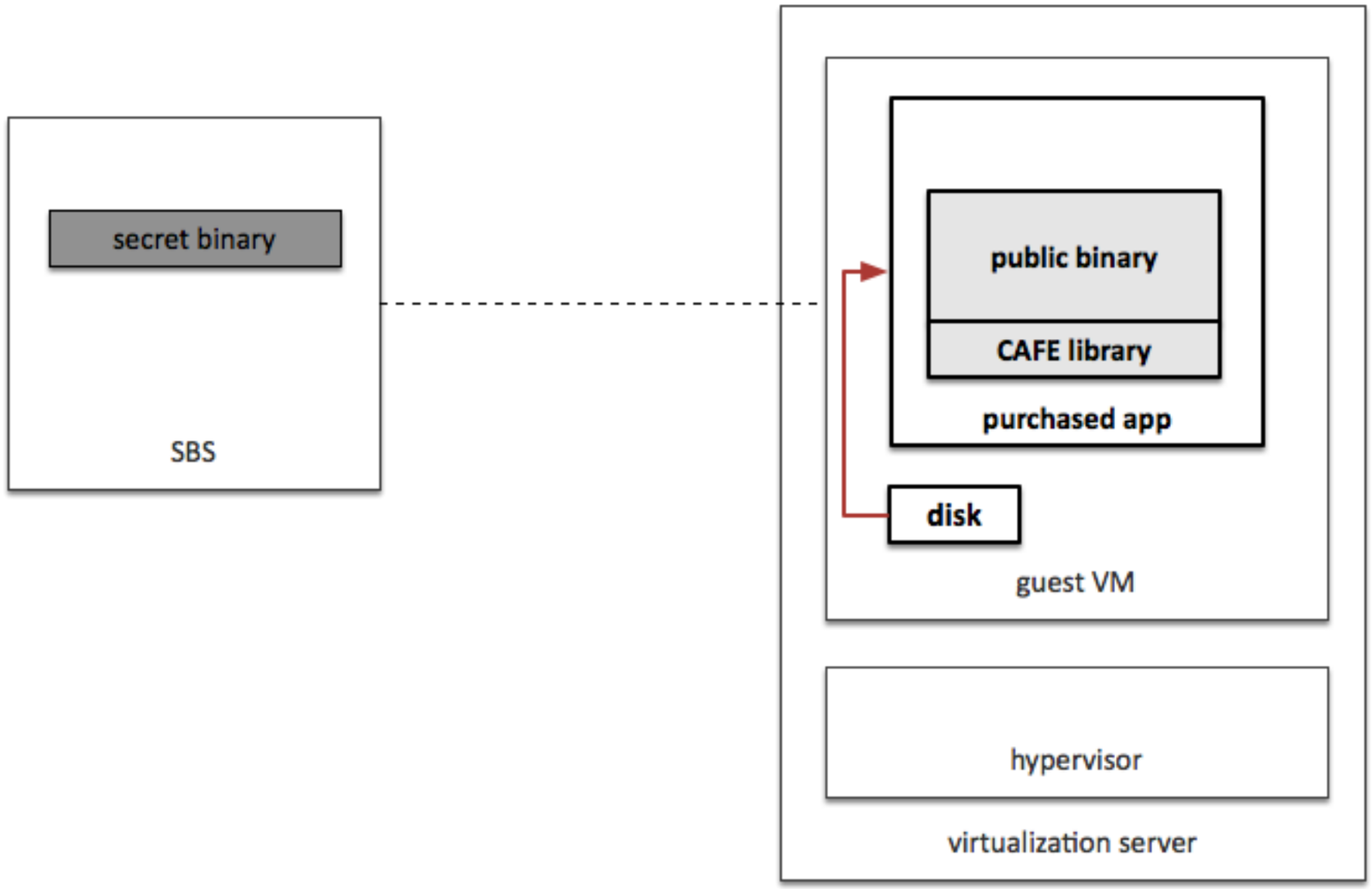
- Process
  - Application developers determine which part of application logic needs confidentiality.
  - Implement public and secret functions separately.
    - The secret functions are placed in the secret code section of the shared library.
    - Public binary objects are linked with the CAFE library.
- CAFE library features
  - Transmission layer for the hypervisor
  - Hypercall interfaces

Submission of Cloud Application

- Application developers build their program into two separate groups and submit them to the marketplace.

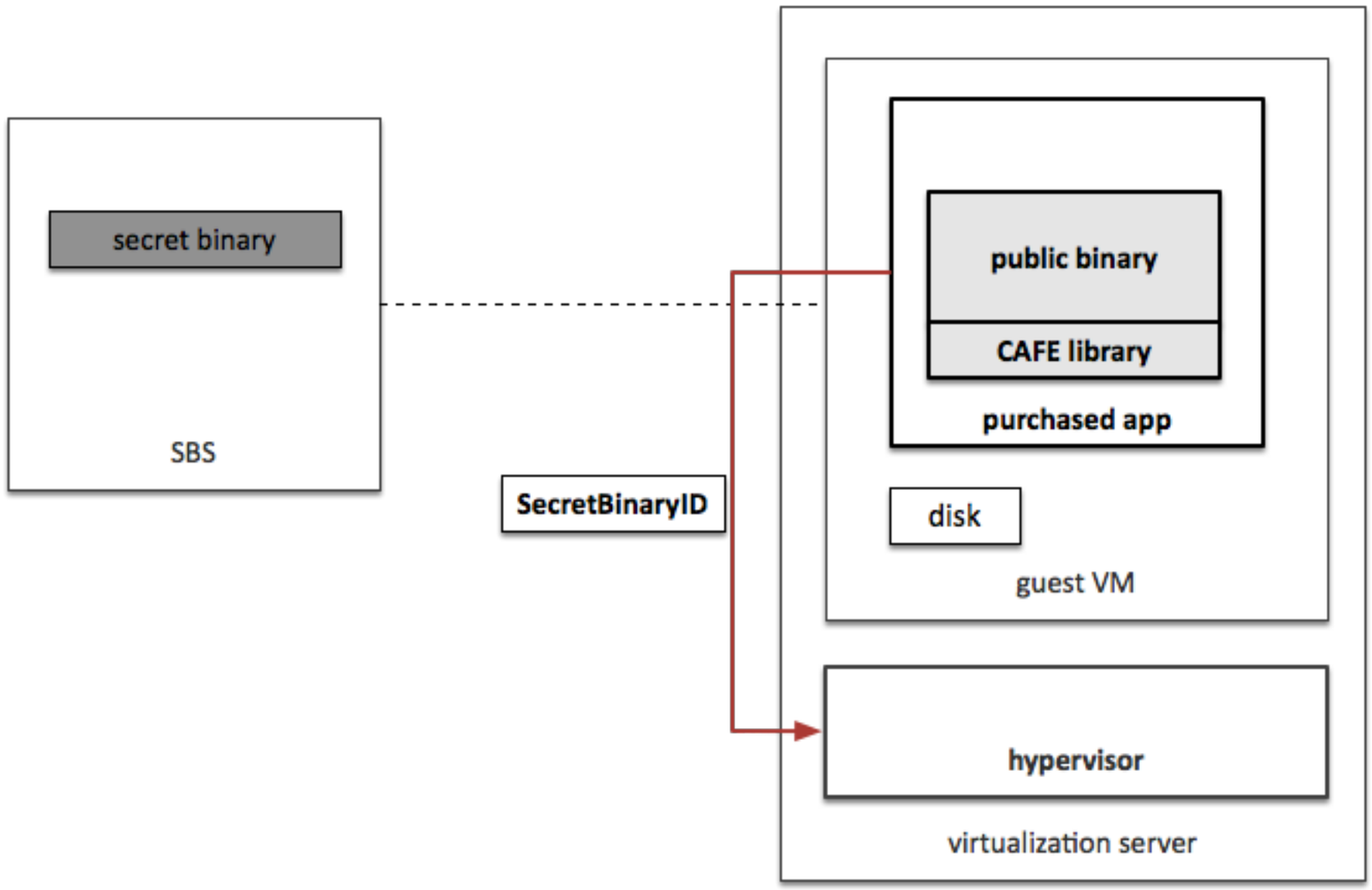
- Public binaries
  - Packaged in a VM image along with other binaries
- Secret binaries
  - Stored in the Secret Binary Server (SBS) in the cloud provider domain

Secret Binary Deployment Protocol



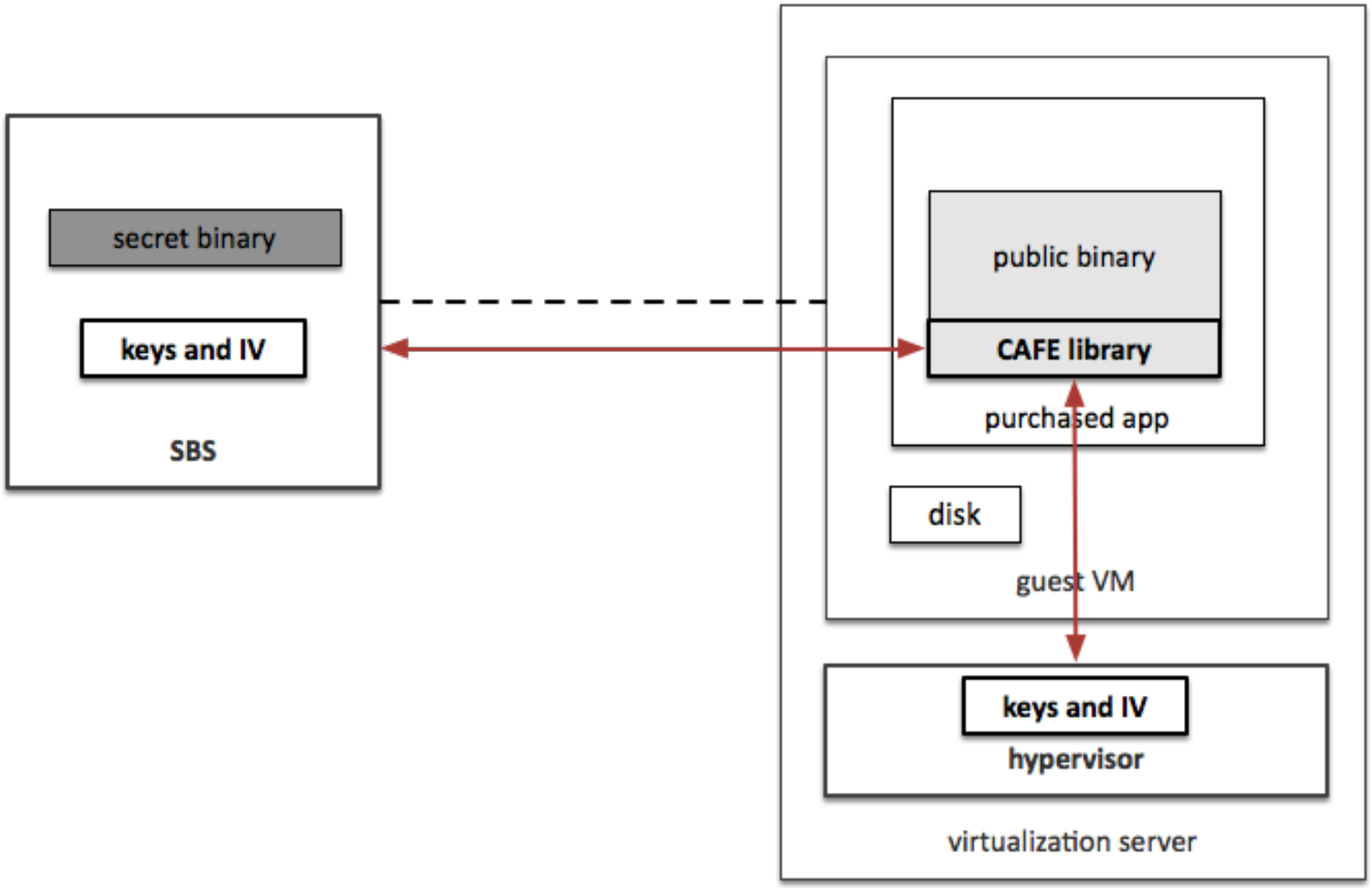
1. A cloud user executes a purchased cloud app.

Secret Binary Deployment Protocol



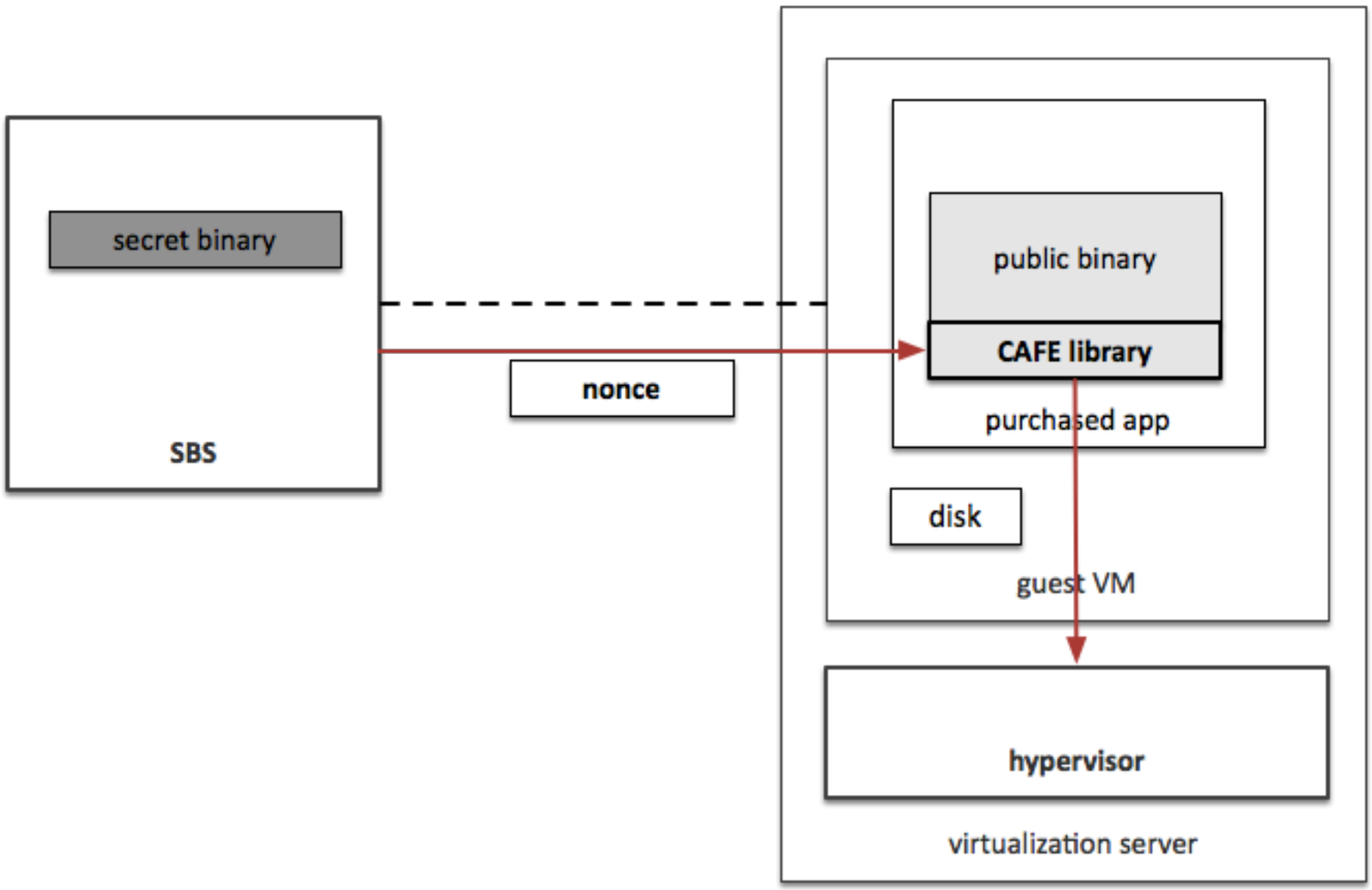
2. Pass a SecretBinaryID to the hypervisor in order to match the purchased app with a proper secret binary

Secret Binary Deployment Protocol



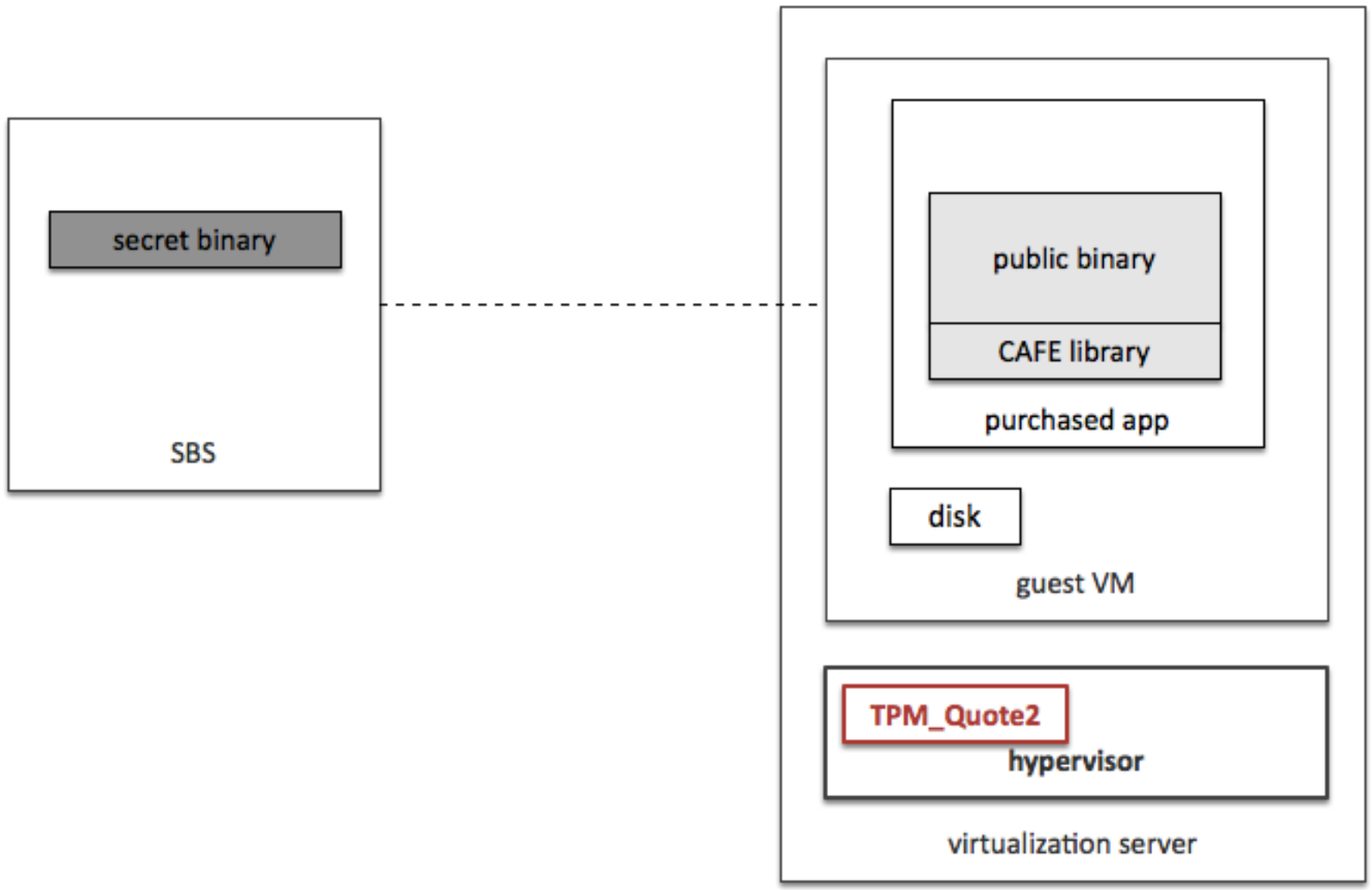
3. The SBS and the hypervisor establish a secure channel using a variant of the TLS protocol and share secrets for the secret binary encryption.

Secret Binary Deployment Protocol



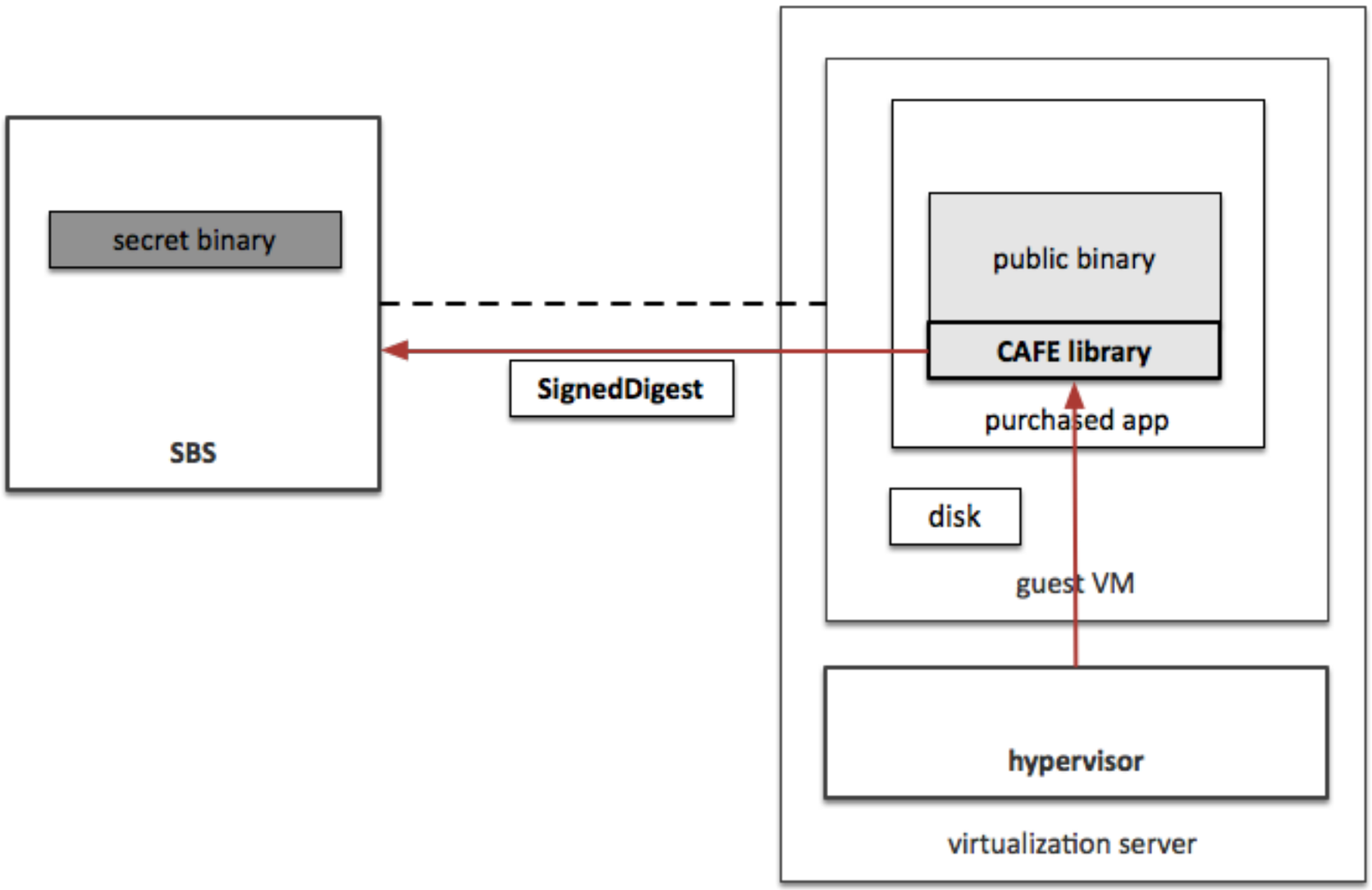
4. The SBS generates and sends a nonce to the hypervisor for the freshness of the attestation evidence.

Secret Binary Deployment Protocol



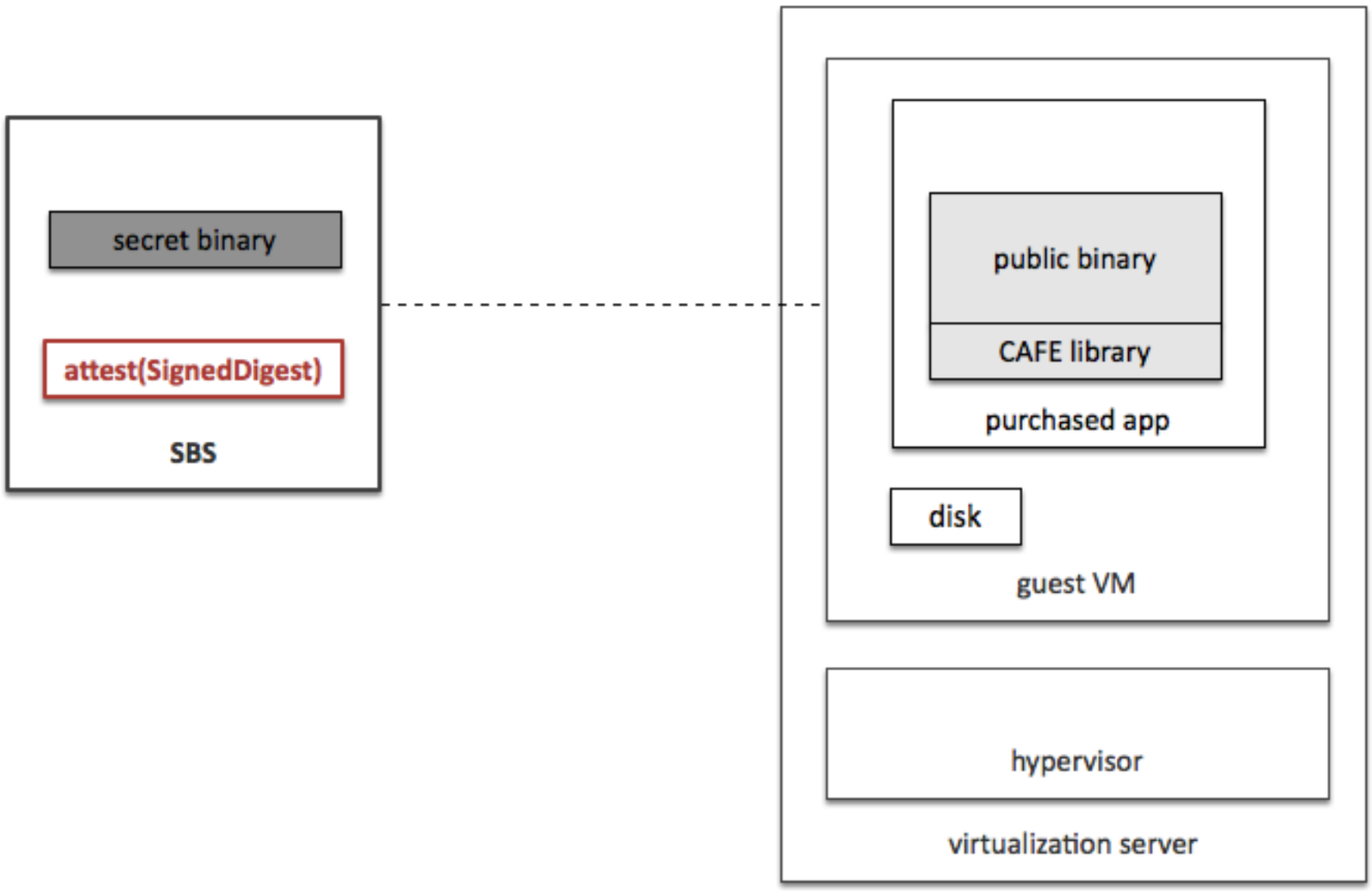
5. The hypervisor performs the TPM_Quote2 operation to attest the integrity of the hypervisor, the trusted computing base of CAFE.

Secret Binary Deployment Protocol



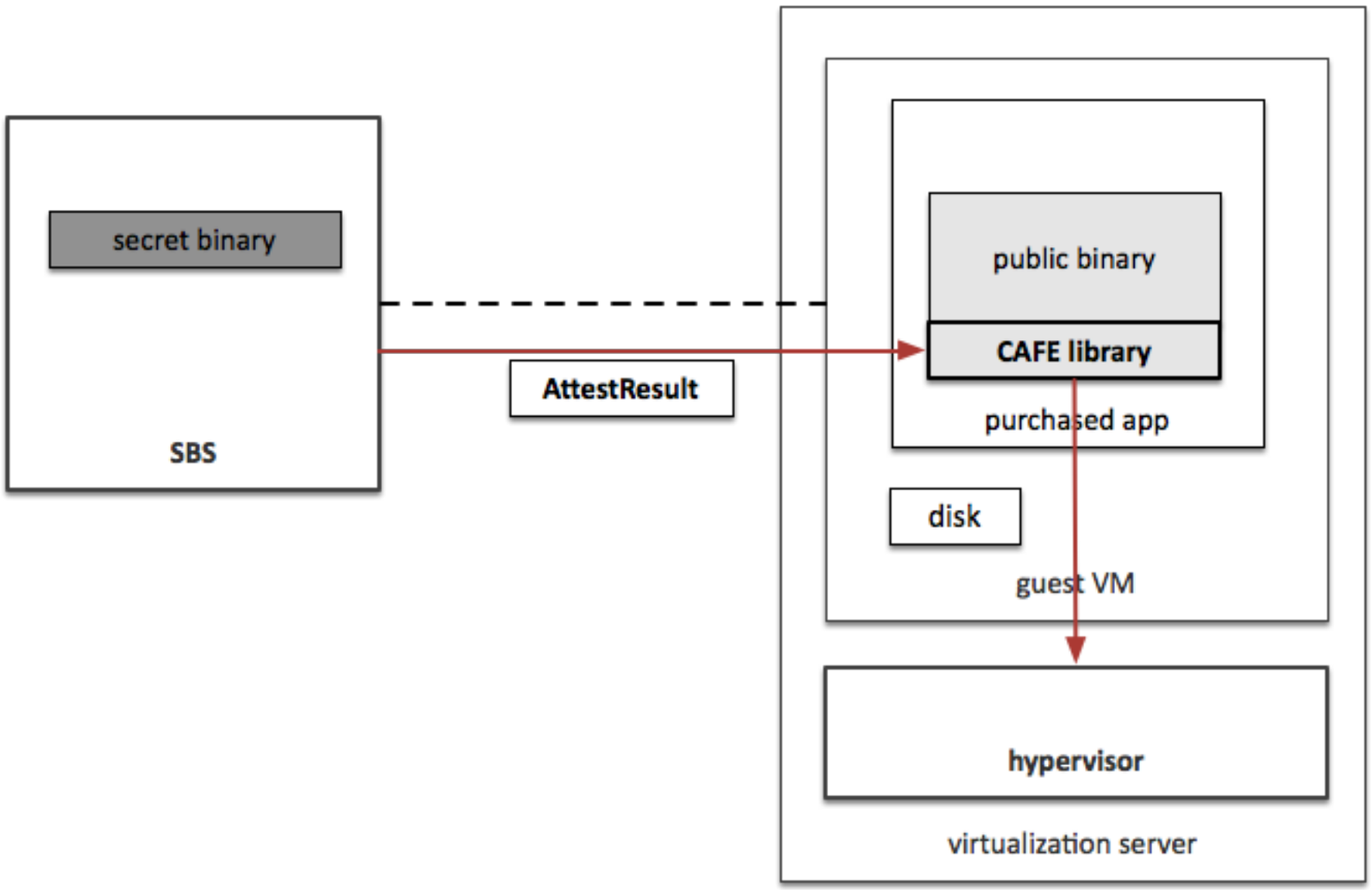
6. The hypervisor transmits a SignedDigest, the result of the TPM_Quote2 operation, to the SBS

Secret Binary Deployment Protocol



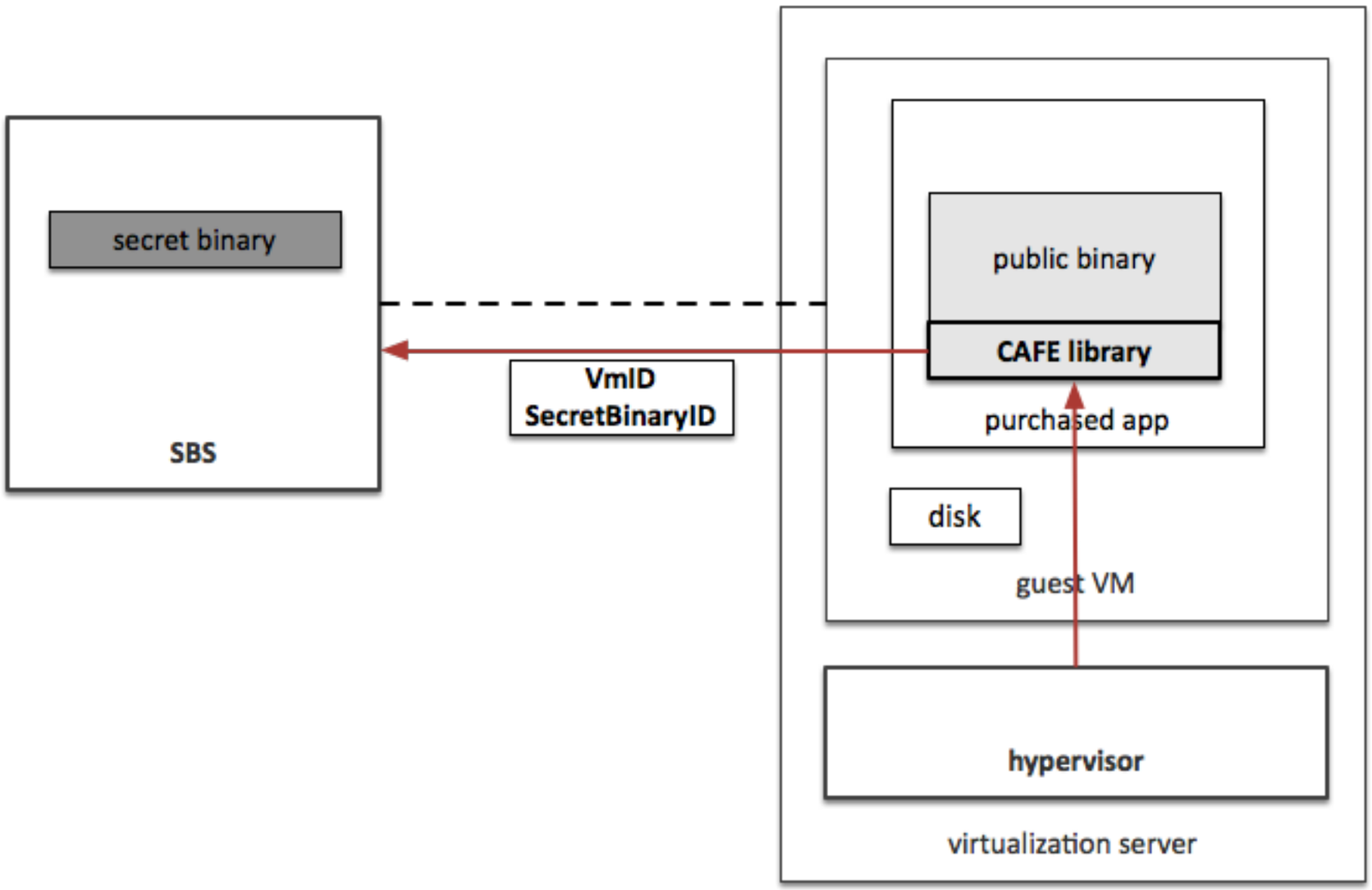
7. The SBS attests the integrity of the hypervisor with the VM server's PU_{AIK}. The success of the attestation means the TCB is not compromised.

Secret Binary Deployment Protocol



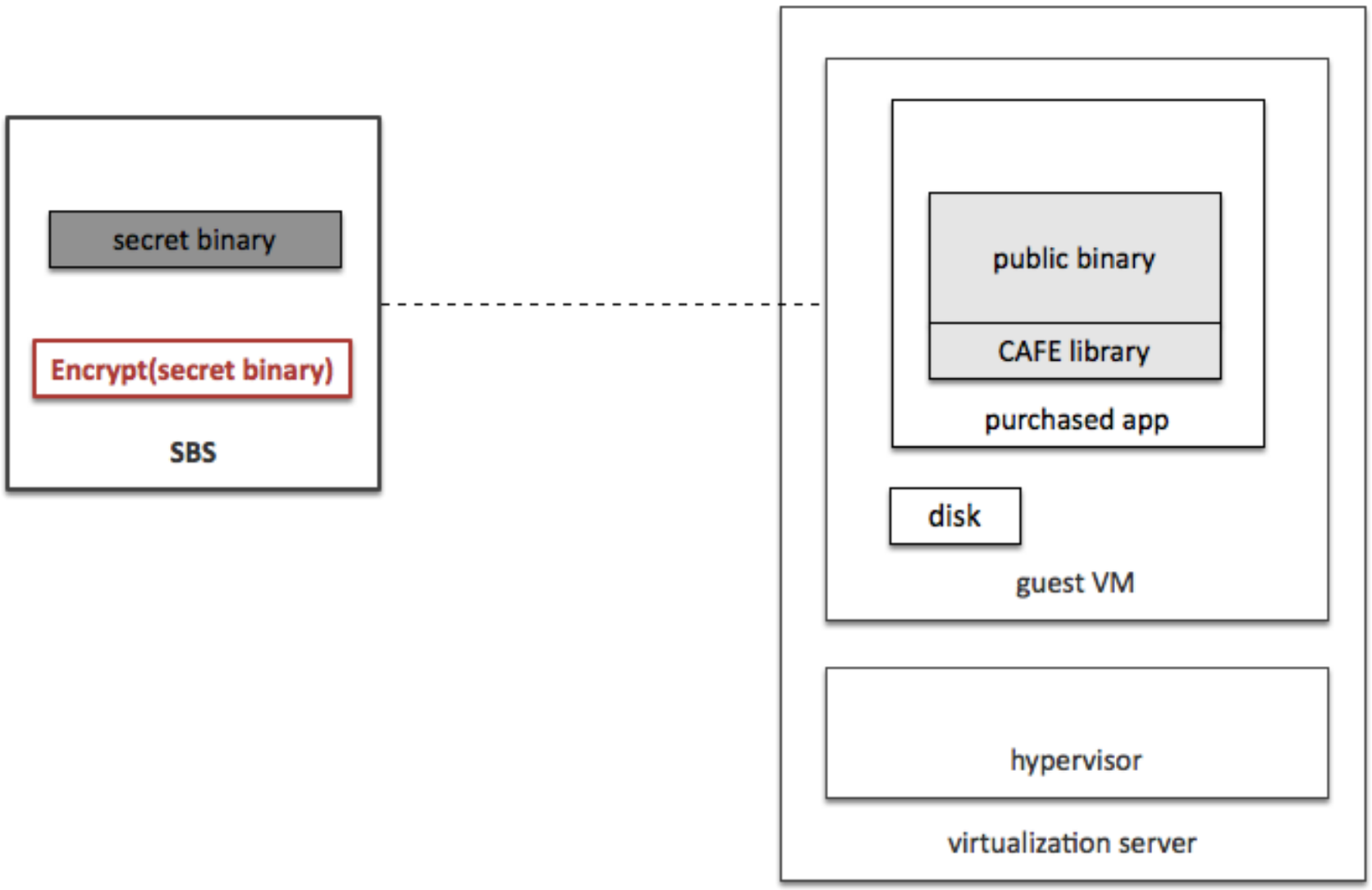
8. The SBS sends the AttestResult to the hypervisor.

Secret Binary Deployment Protocol



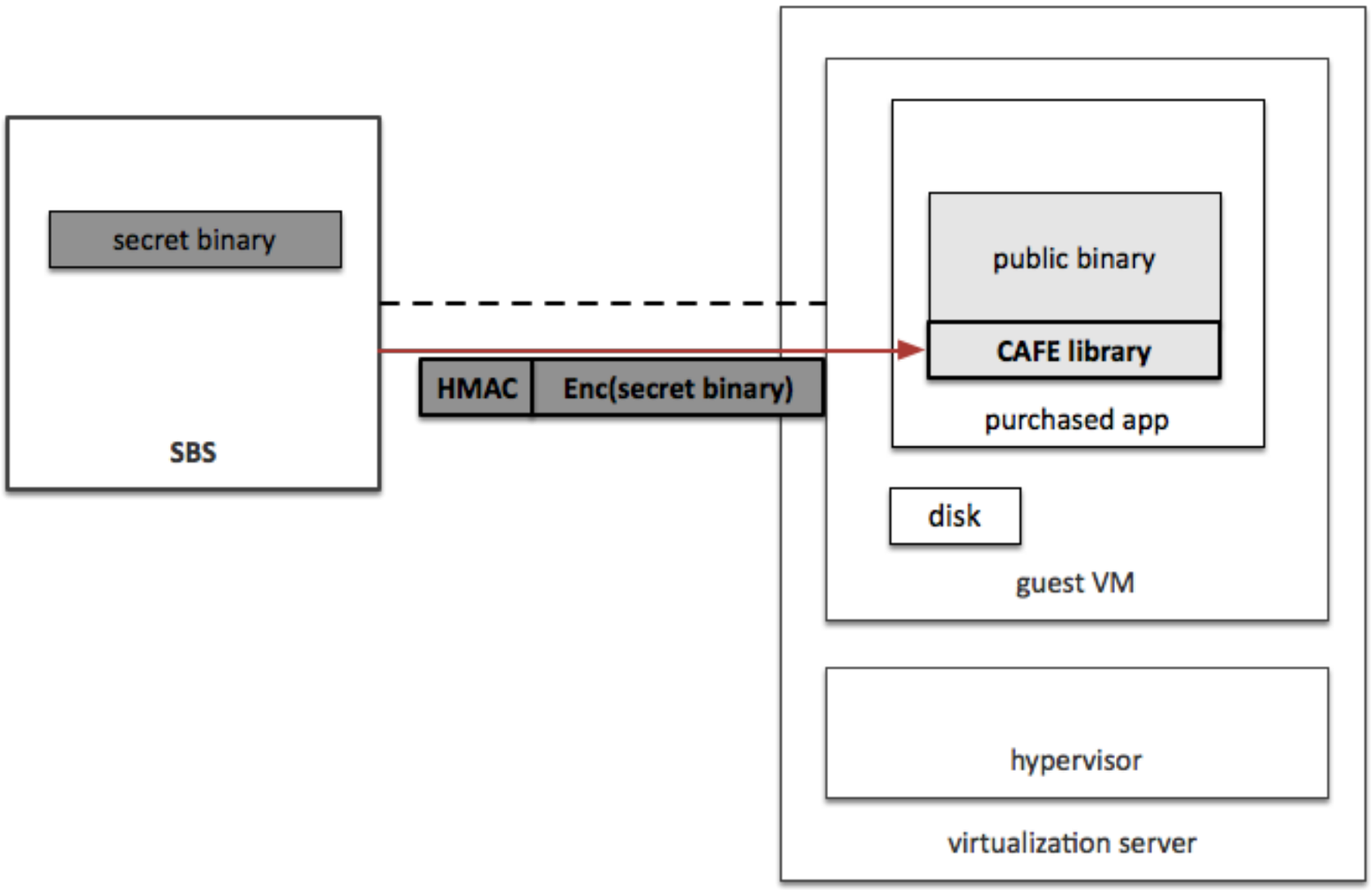
9. If the AttestResult is SUCCESS, the hypervisor sends a Virtual Machine ID and the SecretBinaryID to the SBS.

Secret Binary Deployment Protocol



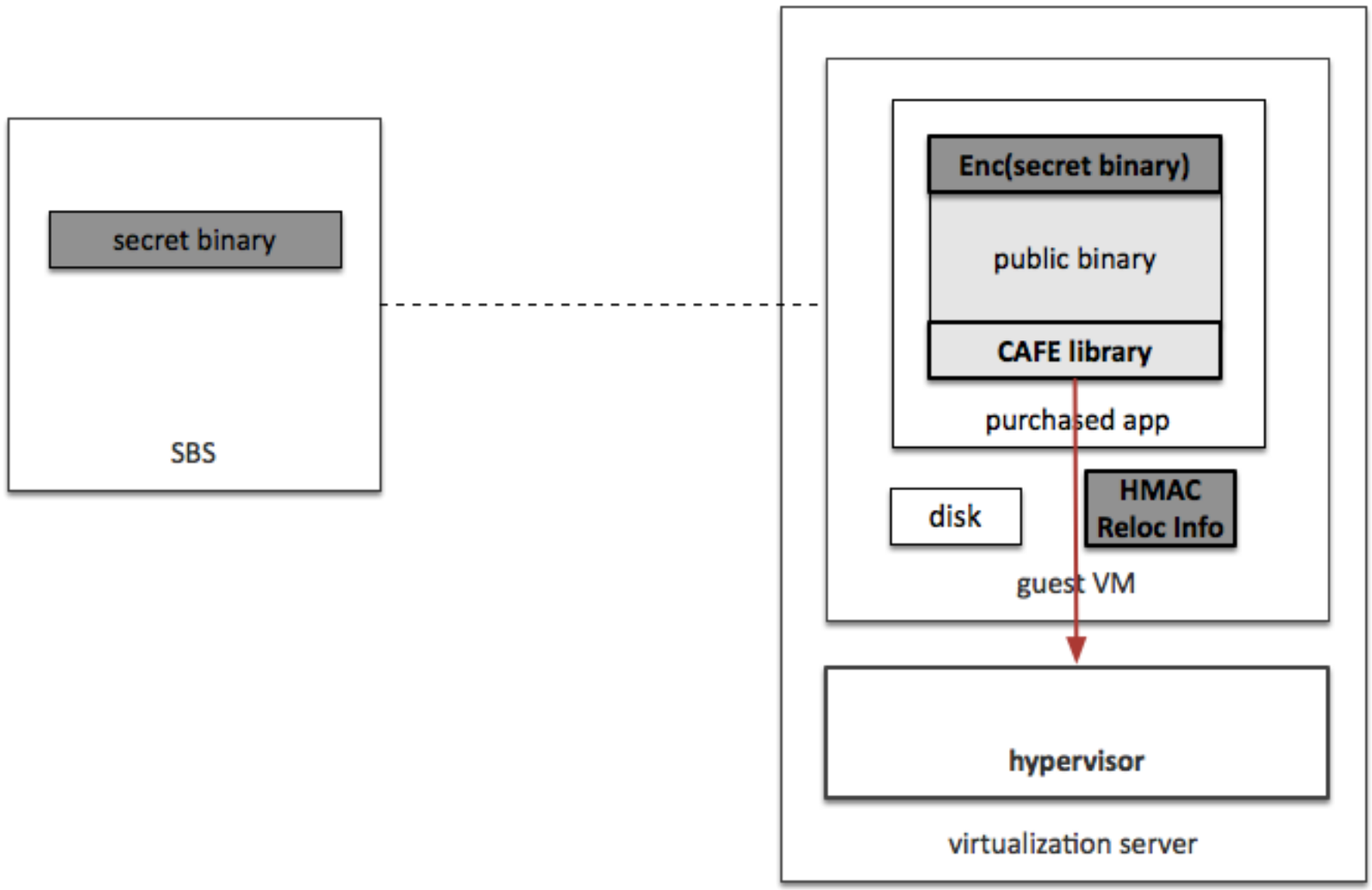
10. After checking the license, the SBS encrypts the secret code section of a proper secret binary with the pre-shared secrets in the Step 3.

Secret Binary Deployment Protocol



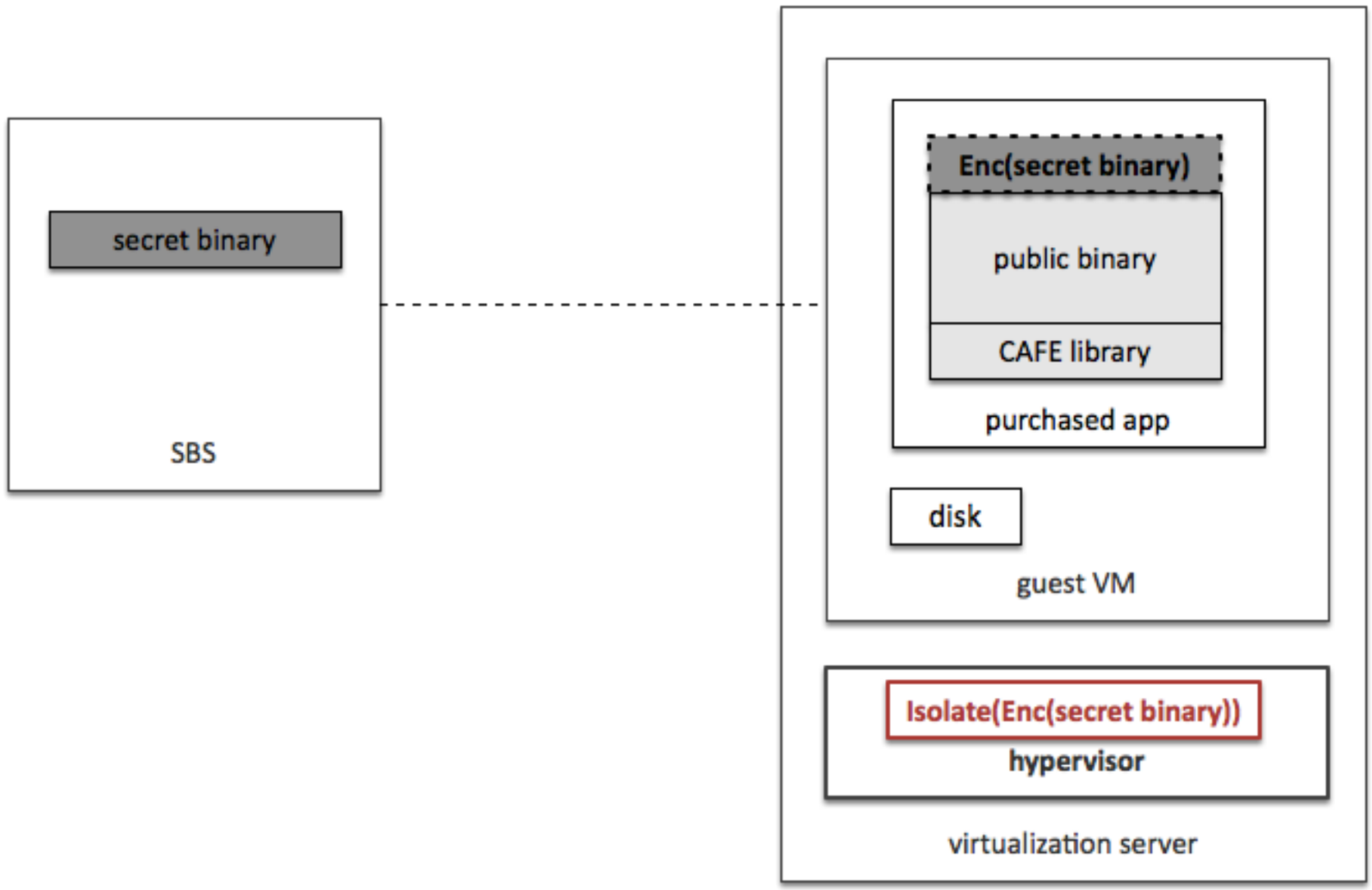
11. The SBS sends an encrypted secret binary and a HMAC value of the encrypted secret binary to the CAFE library

Secure Loading of Secret Binary



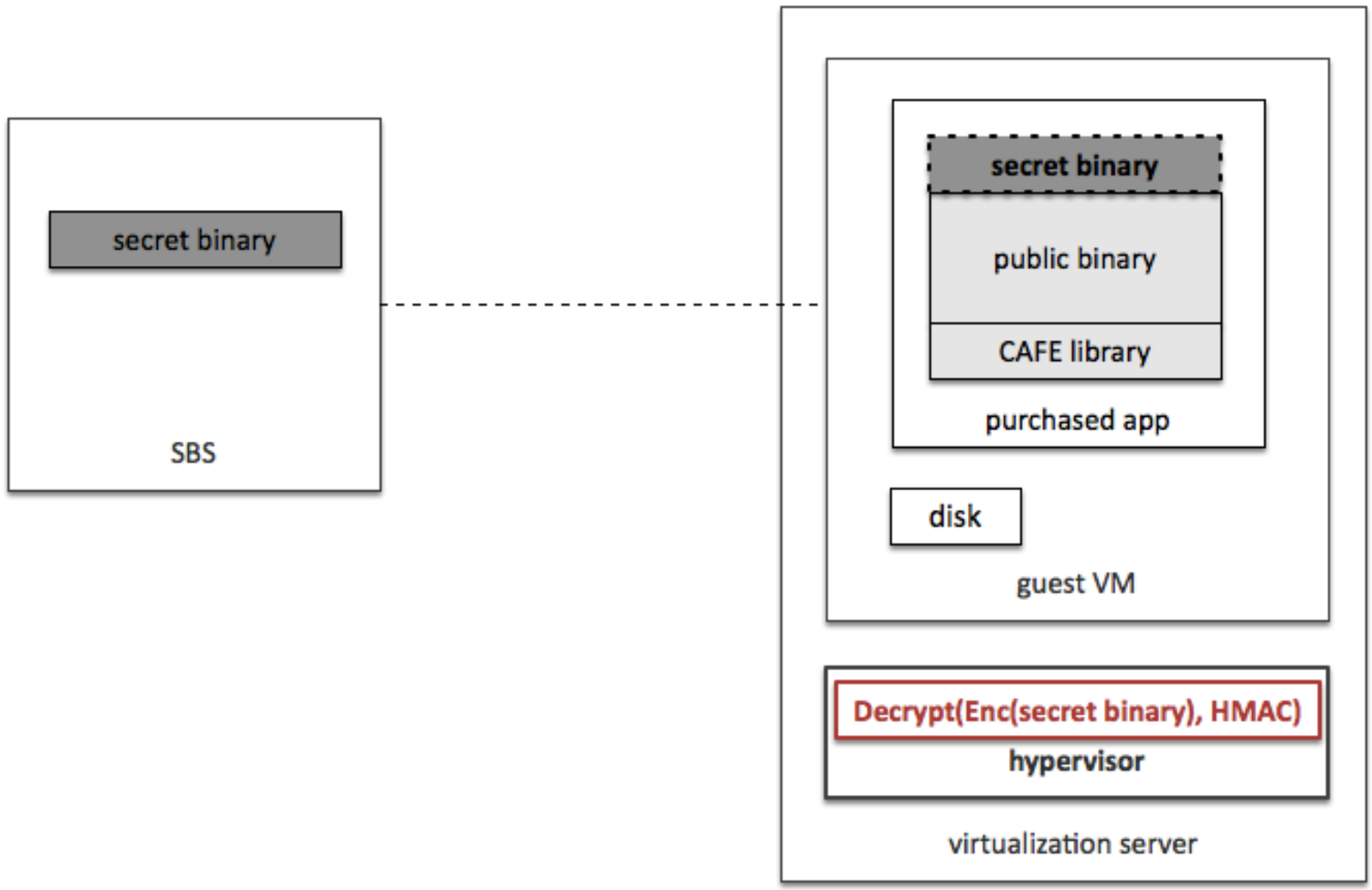
1. The CAFE library loads the encrypted secret binary and requests the hypervisor to securely load it.

Secure Loading of Secret Binary



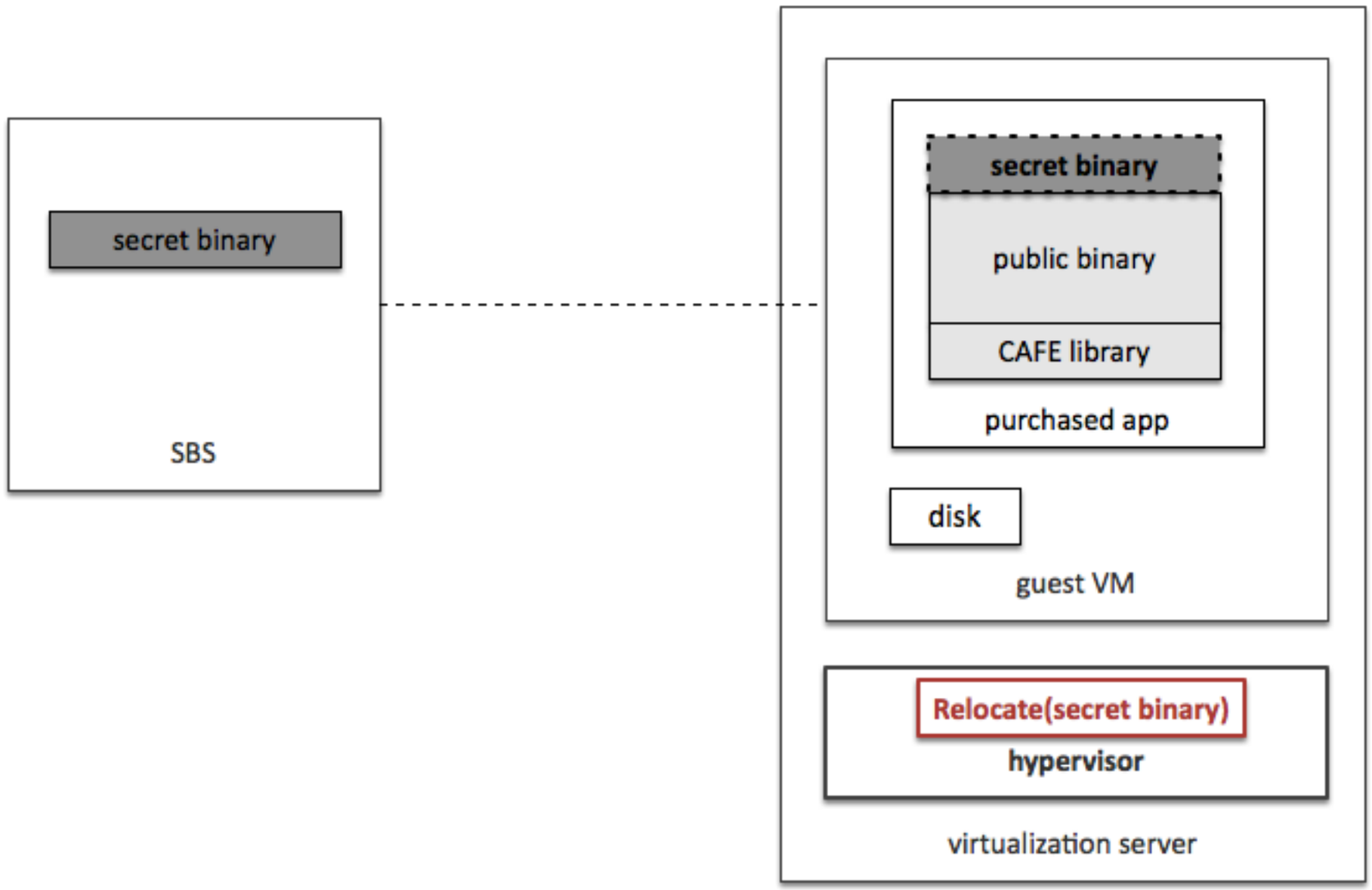
2. Prior to decryption, the hypervisor isolates the memory area of the encrypted secret binary from the guest VMs.

Secure Loading of Secret Binary



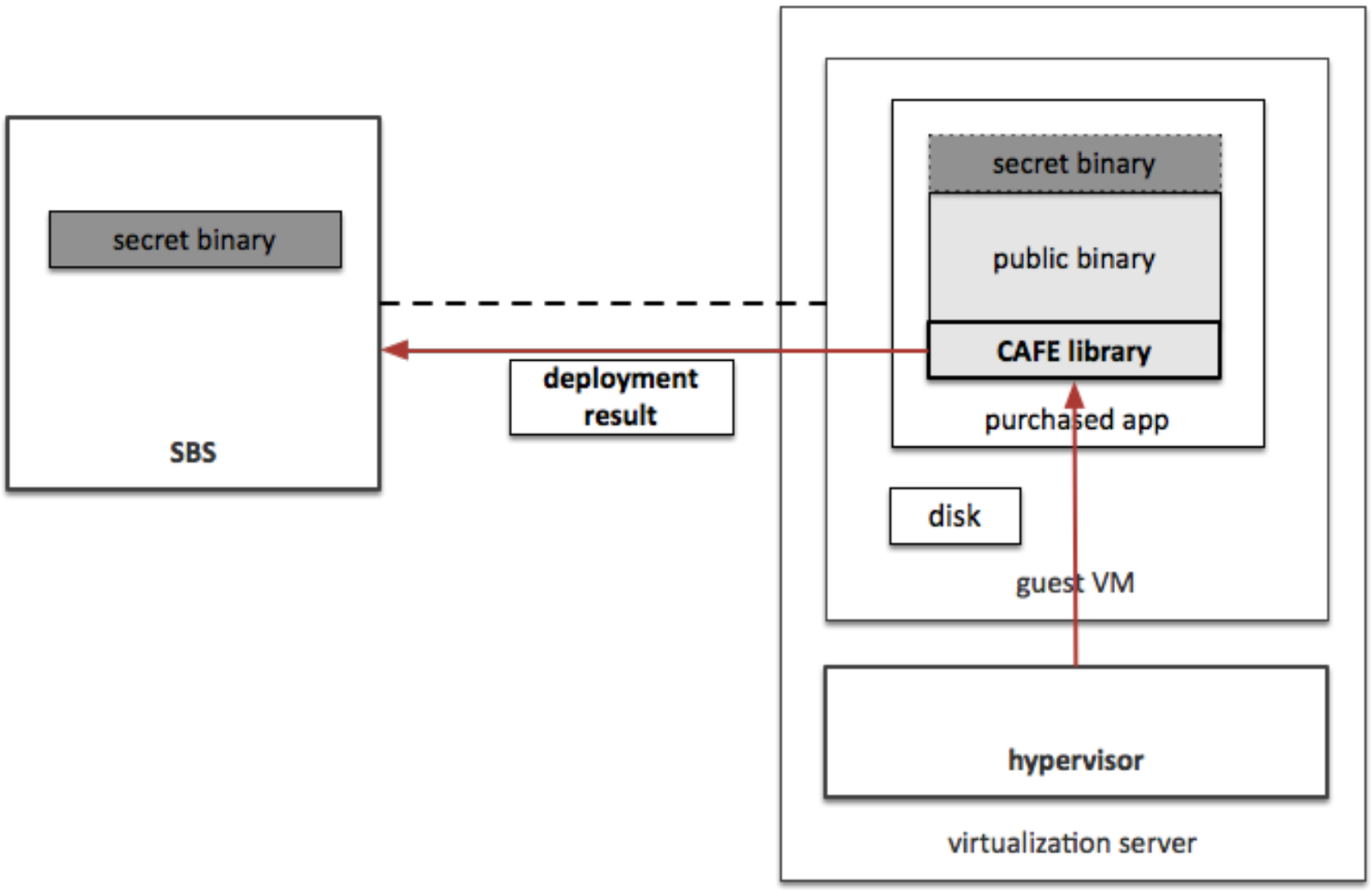
3. The hypervisor verifies the HMAC value and decrypts the encrypted secret binary.

Secure Loading of Secret Binary



4. The hypervisor relocates the decrypted secret binary with the relocation information from the CAFE library.

Secure Loading of Secret Binary



5. The hypervisor returns the result of the deployment process.

Execution of Secret Binary

- Call secret functions
  - Any function call to the code inside the secret binary will trap to the hypervisor.
- Pass input parameters to secret binaries
  - Marshal input parameters into the secret function's stack
- Pass return values to public binaries
  - Unmarshal outputs into the public binary function's stack

Implementation

- Hypervisor
  - Implemented on top of the eXtensible and Modular Hypervisor Framework (XMHF) [Oakland'13]
- Machine
  - Processor: AMD Turion II P520 2.3 GHz
  - Memory: 4GB
  - Storage: 256 GB SSD
  - Guest OS: 32-bit version of Ubuntu 12.04

Performance Overhead

Application Category	Program Name	Program Info	Overhead
Decision-making logic	NGINX	Access module	1.90%
	Sendmail	Mail filter (Milter)	2.81%
Cryptographic operations	Google Authenticator	One-time passcode generation	2.52%
	EncFS	ARIA block encryption / decryption	**900.13%**
Data processing workload	MapReduce	k-means clustering	8.04%
	Hadoop	Word counting	5.82%

- EncFS is a heavy I/O stress test. Other typical usages of security sensitive operations have low overhead.

Related Work

- Overshadow [ASPLOS'08]
  - Provides cloaking for general purpose legacy unmodified applications and untrusted kernel.
  - CAFE provides stronger code confidentiality.
    - Tightly verified and sanitized input and output
    - A constrained scope of sensitive code
- TrustVisor [Oakland'10]
  - Provides an infrastructure for executing security-sensitive code in isolated memory
  - CAFE provides the confidentiality of the binaries in an end-to-end manner for the entire lifetime of the deployed software.

Conclusion

- CAFE provides the confidential distribution and execution of cloud applications.
- We show the effectiveness and practicality of CAFE.
  - Reasonable performance overhead
  - Evaluation on six applications commonly used in cloud marketplaces