Blockhub: Blockchain-based Secure Cross-domain Software Development and Sharing System

Denis Ulybyshev¹, Bharat Bhargava¹, Miguel Villarreal-Vasquez³, Aala Alsalem¹, Donald Steiner², Leon Li², Jason Kobes⁵, Harry Halpin⁴, Rohit Ranchal³, Leszek Lilien³

¹Computer Science and CERIAS, Purdue University; ²Northrop Grumman; ³IBM; ⁴MIT

This research is supported by Northrop Grumman. We collaborated with Donald Steiner, Leon Li, Jason Kobes, Steve Seaberg, Peter Melyn, Paul Conoval.

ACKNOWLEDGEMENT: This research is supported by Northrop Grumman. We collaborated with Donald Steiner, Leon Li, Jason Kobes, Steve Seaberg, Peter Melyn, Paul Conoval.

SOFTWARE SPILLAGE DETECTION

- By CM, web crawler (in public directories), automatic updates
- CM checks policies: whether SM is supposed to be at Y
- If SM is leaked in decrypted form: digital watermark will be checked
- Data request transferred to EHR, access control policies evaluated

DATA ANALYTICS FOR SOFTWARE SPILLAGE DETECTION

- CM checks policies: whether SM is supposed to be at Y
- If SM is leaked in decrypted form: digital watermark will be checked
- Data request transferred to EHR, access control policies evaluated

FUTURE WORK

- Integrate blockchain-based technology in WAXEDPRUNE prototype
- Comprehensive performance and scalability evaluation
- Support isolated SB execution (Linux Docker Container)
- Implement Data Analytics framework

WAXEDPRUNE ARCHITECTURE

- Client requests data from EHR
- Service (authenticated client) requests trust and leakage verification from CM
- CM responds with trust level of requesting service and leakage check result
- Data request sent to EHR

DATA PUBLICATIONS AND PROTOTYPE

2. NG WAXEDPRUNE Prototype https://github.com/DenisUlybyshev/WAXEDPRUNE

SERVICE SPILLAGE DETECTION

- SB contains Enc [ Software (S) ] = {Encₐₙ⁽SMₐ⁾, ... , Encₐₙ⁽SMₙ⁾} and Access Control Policies (P) = {P₁, ..., Pₙ}
- X is authorized to extract and decrypt SM₁ from SB
- X leaks Enc(SM₁) or SM₁ to unauthorized service Y
- When Y tries to decrypt SM₁ CM checks policies: whether SM₁ is supposed to be at Y
- If SM₁ is leaked in decrypted form: digital watermark will be checked by CM, web crawler (in public directories), automatic updates

SOFTWARE SHARING SYSTEM

- Collaborator X
  - External Storage: Software Bundles
    - SB₁, SB₂, SB₃
  - Blockchain
    - Registration of software attributes and ID information
    - Access Authorization
    - Process Automation

- Collaborator Y
  - External Storage: Software Bundles
    - SB₃, SB₄
  - Blockchain
    - Registration of software attributes and ID information
    - Access Authorization
    - Process Automation

FEATURES

- Self-integrity check
- Policy evaluation and enforcement
- Selective dissemination of software modules

OBJECTIVES

- Provide secure software sharing and software access auditing
- Provide integrity of provenance data
- Detect software spillage

AUTHENTICATED CLIENT

- Browser's Crypto Level: High
  - Auth. Method: Fingerprint
  - Client's device: Desktop
  - Source network: Corp. Intranet

SOFTWARE SPILLAGE DETECTION

- SB contains Enc [ Software (S) ] = {Encₐₙ⁽SMₐ⁾, ... , Encₐₙ⁽SMₙ⁾} and Access Control Policies (P) = {P₁, ..., Pₙ}
- X is authorized to extract and decrypt SM₁ from SB
- X leaks Enc(SM₁) or SM₁ to unauthorized service Y
- When Y tries to decrypt SM₁ CM checks policies: whether SM₁ is supposed to be at Y
- If SM₁ is leaked in decrypted form: digital watermark will be checked by CM, web crawler (in public directories), automatic updates

SOFTWIRARE BUNDLE (SB)

- SB contains Enc[f(S)_i] = {Encₐₙ⁽SMₐ⁾, ... , Encₐₙ⁽SMₙ⁾} and Access Control Policies (P) = {P₁, ..., Pₙ}
- X is authorized to extract and decrypt SM₁ from SB
- X leaks Enc(SM₁) or SM₁ to unauthorized service Y
- When Y tries to decrypt SM₁ CM checks policies: whether SM₁ is supposed to be at Y
- If SM₁ is leaked in decrypted form: digital watermark will be checked by CM, web crawler (in public directories), automatic updates

AUTHENTICATED CLIENT

- Browser's Crypto Level: Low
  - Auth. Method: Password
  - Client's device: Mobile
  - Source network: Unknown

PROPOSED SOLUTION

- SB contains Enc [ Software (S) ] = {Encₐₙ⁽SMₐ⁾, ... , Encₐₙ⁽SMₙ⁾} and Access Control Policies (P) = {P₁, ..., Pₙ}
- X is authorized to extract and decrypt SM₁ from SB
- X leaks Enc(SM₁) or SM₁ to unauthorized service Y
- When Y tries to decrypt SM₁ CM checks policies: whether SM₁ is supposed to be at Y
- If SM₁ is leaked in decrypted form: digital watermark will be checked by CM, web crawler (in public directories), automatic updates

SOFTWARE SPILLAGE DETECTION

- SB contains Enc [ Software (S) ] = {Encₐₙ⁽SMₐ⁾, ... , Encₐₙ⁽SMₙ⁾} and Access Control Policies (P) = {P₁, ..., Pₙ}
- X is authorized to extract and decrypt SM₁ from SB
- X leaks Enc(SM₁) or SM₁ to unauthorized service Y
- When Y tries to decrypt SM₁ CM checks policies: whether SM₁ is supposed to be at Y
- If SM₁ is leaked in decrypted form: digital watermark will be checked by CM, web crawler (in public directories), automatic updates

SOFTWIRARE SPILLAGE DETECTION

- By CM, web crawler (in public directories), automatic updates
- CM checks policies: whether SM₁ is supposed to be at Y
- If SM₁ is leaked in decrypted form: digital watermark will be checked by CM, web crawler (in public directories), automatic updates

SOFTWARE SPILLAGE DETECTION

- SB contains Enc [ Software (S) ] = {Encₐₙ⁽SMₐ⁾, ... , Encₐₙ⁽SMₙ⁾} and Access Control Policies (P) = {P₁, ..., Pₙ}
- X is authorized to extract and decrypt SM₁ from SB
- X leaks Enc(SM₁) or SM₁ to unauthorized service Y
- When Y tries to decrypt SM₁ CM checks policies: whether SM₁ is supposed to be at Y
- If SM₁ is leaked in decrypted form: digital watermark will be checked by CM, web crawler (in public directories), automatic updates

SOFTWARE SPILLAGE DETECTION

- By CM, web crawler (in public directories), automatic updates
- CM checks policies: whether SM₁ is supposed to be at Y
- If SM₁ is leaked in decrypted form: digital watermark will be checked by CM, web crawler (in public directories), automatic updates

SOFTWARE SPILLAGE DETECTION

- SB contains Enc [ Software (S) ] = {Encₐₙ⁽SMₐ⁾, ... , Encₐₙ⁽SMₙ⁾} and Access Control Policies (P) = {P₁, ..., Pₙ}
- X is authorized to extract and decrypt SM₁ from SB
- X leaks Enc(SM₁) or SM₁ to unauthorized service Y
- When Y tries to decrypt SM₁ CM checks policies: whether SM₁ is supposed to be at Y
- If SM₁ is leaked in decrypted form: digital watermark will be checked by CM, web crawler (in public directories), automatic updates