CryptoGuard

SaTC:TTP:Medium:Collaborative: Deployment-quality and Accessible Solutions for Cryptography Code Development

Danfeng (Daphne) Yao and Na Meng (Virginia Tech)
Barton Miller (University of Wisconsin-Madison)

https://yaogroup.cs.vt.edu/index.html#current-research

Main Project Goal: Real-world deployment of static program analysis based cryptographic code screening to secure massive codebases in practice

This is a **transition to practice (TTP)** project, focusing on our **CryptoGuard** technology. Our deployment environment is Oracle Labs. CryptoGuard detection and alert refinement approaches have been integrated into Oracle Parfait framework and used to secure crypto code in production-level Java projects in practice.

Dr. Cristina Cifuentes
(Collaborator at Oracle)

Who wouldn't want to write secure code?

Overhead

False positives

Limited resources

Lack of knowledge

Key Challenges:

- **Scalability:** continuous deployment and continuous integration (CD/CI)
- **Precision:** meaningful alerts and fixes

Scientific Impact:

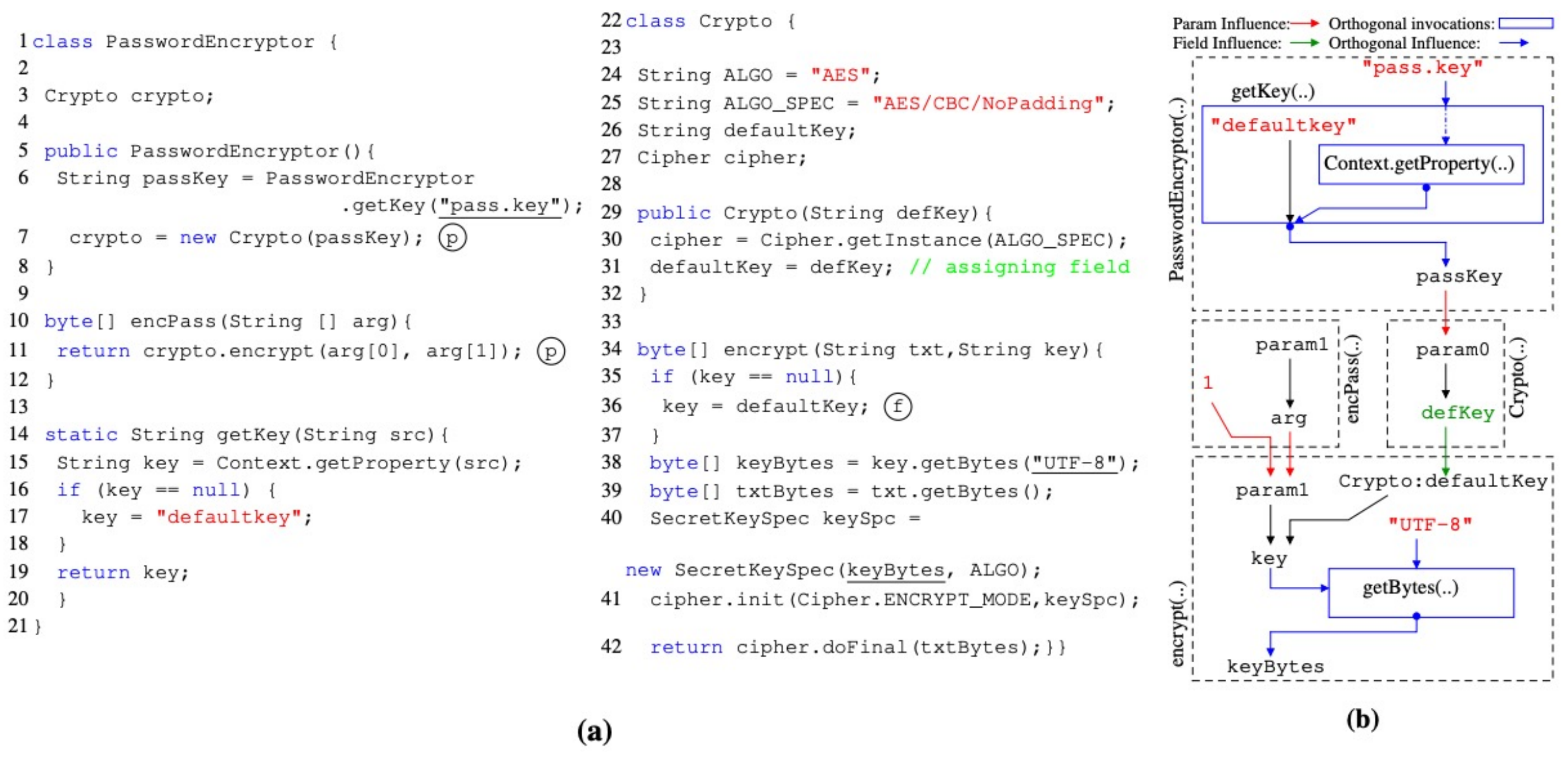
Domain-specific static analysis is deployable!

A common misconception: static analysis-based detection is not practical, due to false alarms.

The key to high precision (low false positives) is domain-specific alert refinement.

Our Work

- Methods for mapping abstract crypto to concrete program analysis algorithms



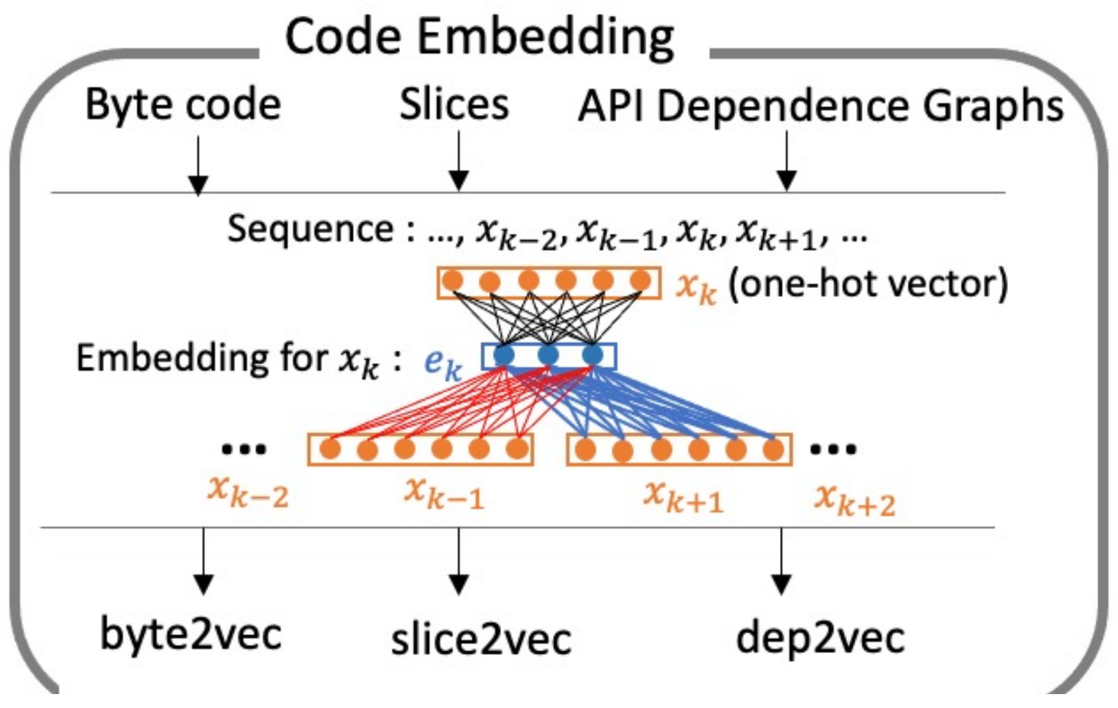
```
class PasswordEncryptor {

 Crypto crypto;

 public PasswordEncryptor(){
  String passKey = PasswordEncryptor
                        .getKey("pass.key");
   crypto = new Crypto(passKey); (p)
 }

 byte[] encPass(String [] arg){
  return crypto.encrypt(arg[0], arg[1]); (p)
 }

 static String getKey(String src){
  String key = Context.getProperty(src);
  if (key == null) {
    key = "defaultkey";
  }
  return key;
  }
}
```

```
class Crypto {

 String ALGO = "AES";
 String ALGO_SPEC = "AES/CBC/NoPadding";
 String defaultKey;
 Cipher cipher;

 public Crypto(String defKey){
  cipher = Cipher.getInstance(ALGO_SPEC);
  defaultKey = defKey; // assigning field
 }

 byte[] encrypt(String txt,String key){
  if (key == null){
   key = defaultKey; (f)
  }
  byte[] keyBytes = key.getBytes("UTF-8");
  byte[] txtBytes = txt.getBytes();
  SecretKeySpec keySpc =
 new SecretKeySpec(keyBytes, ALGO);
  cipher.init(Cipher.ENCRYPT_MODE,keySpc);

  return cipher.doFinal(txtBytes);}}
```

(a)

(b)

- AI code repair, analysis-guided learning
- Systematic benchmarking (IEEE TSE'22)



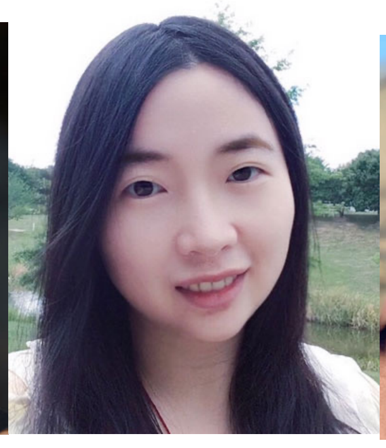
Being the Developers' Friend: Our Experience Developing a High Precision Tool for Secure Coding. Yao et al. *IEEE S&P.* To appear.

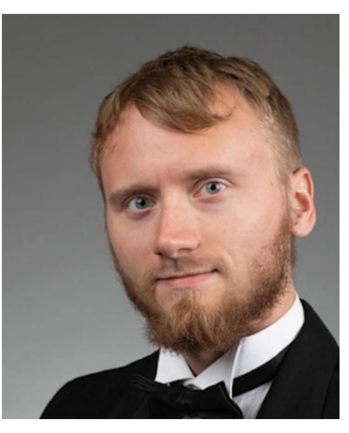
Broader Impact:

Democratizing secure coding

Many 90-minute tutorials on secure coding (ESORICS, IEEE SecDev, Supercomputing)

Engagement with software developing community



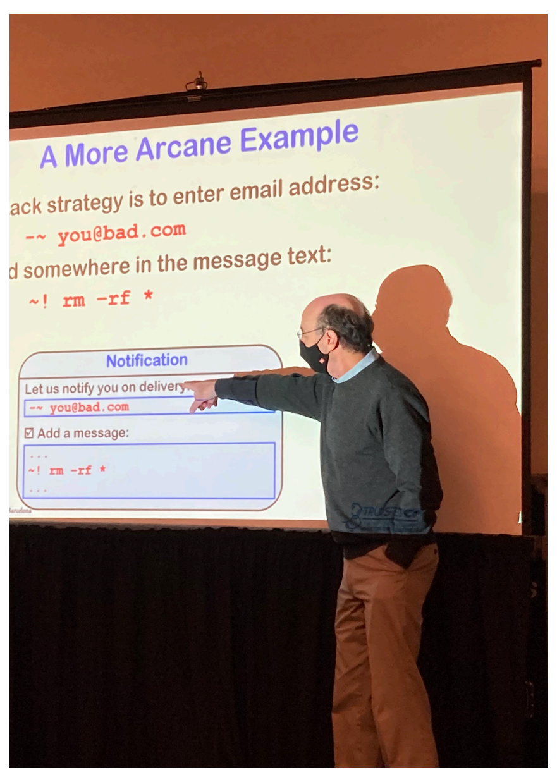


Broader Impact:

Videos and training docs

UW-Madison online software security course

Heymann and Miller Software Security for the People: Free and Open Resources for Software Security Training, *IEEE S&P.* March/April 2022



Broader Impact:

The project involves:

- 7 graduate students (across VT & UW-Wisconsin)
- 5+ undergraduate student researchers
- 5 industrial collaborators (Oracle and DST)
- Training 500+ professionals annually

The 5th NSF Secure and Trustworthy Cyberspace Principal Investigator Meeting (2022 SaTC PI Meeting)
June 1-2, 2022 | Arlington, Virginia

Award ID# 1929701