
SECURE INTEGRATION OF CRYPTOGRAPHIC SOFTWARE

Speaker: Stefan Krüger



HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN



TECHNISCHE
UNIVERSITÄT
DARMSTADT

When a Developer Uses a Crypto API

```
KeyGenerator keyGen = KeyGenerator.getInstance("AES");  
keyGen.init(128);  
SecretKey key = keyGen.generateKey();
```

```
Cipher cipher = Cipher.getInstance("AES");  
cipher.init(Cipher.ENCRYPT_MODE, key, iv);  
cipher.doFinal(data);
```

- `getInstance(String transformation) : Cipher - Cipher`
- `getInstance(String transformation, Provider provider) : Cipher - Cipher`
- `getInstance(String transformation, String provider) : Cipher - Cipher`

Uses Electronic
Codebook (ECB)

The Average Developer is no Crypto Expert



of Android apps contain at least one misuse



Popular vendors misuse TLS libraries



of crypto-related vulnerabilities result from API misuse

What shall we do about it?



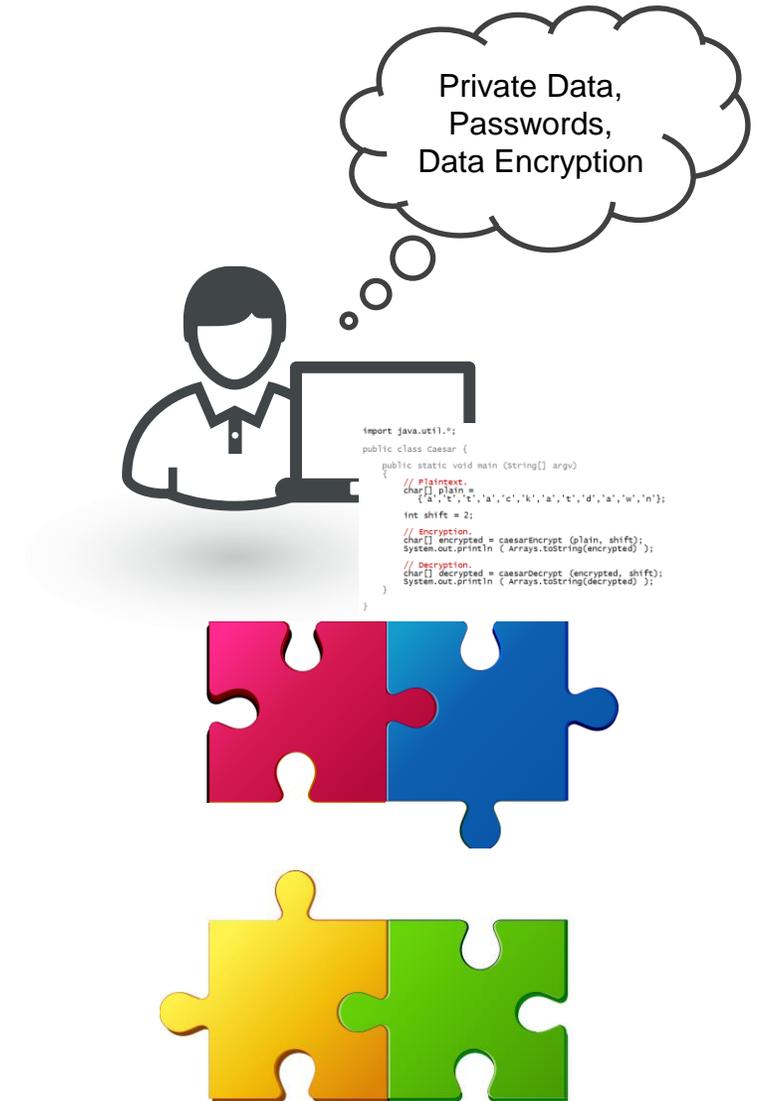
What shall we do about it?



What shall we do about it?



What shall we do about it?



What shall we do about it?



What shall we do about it?

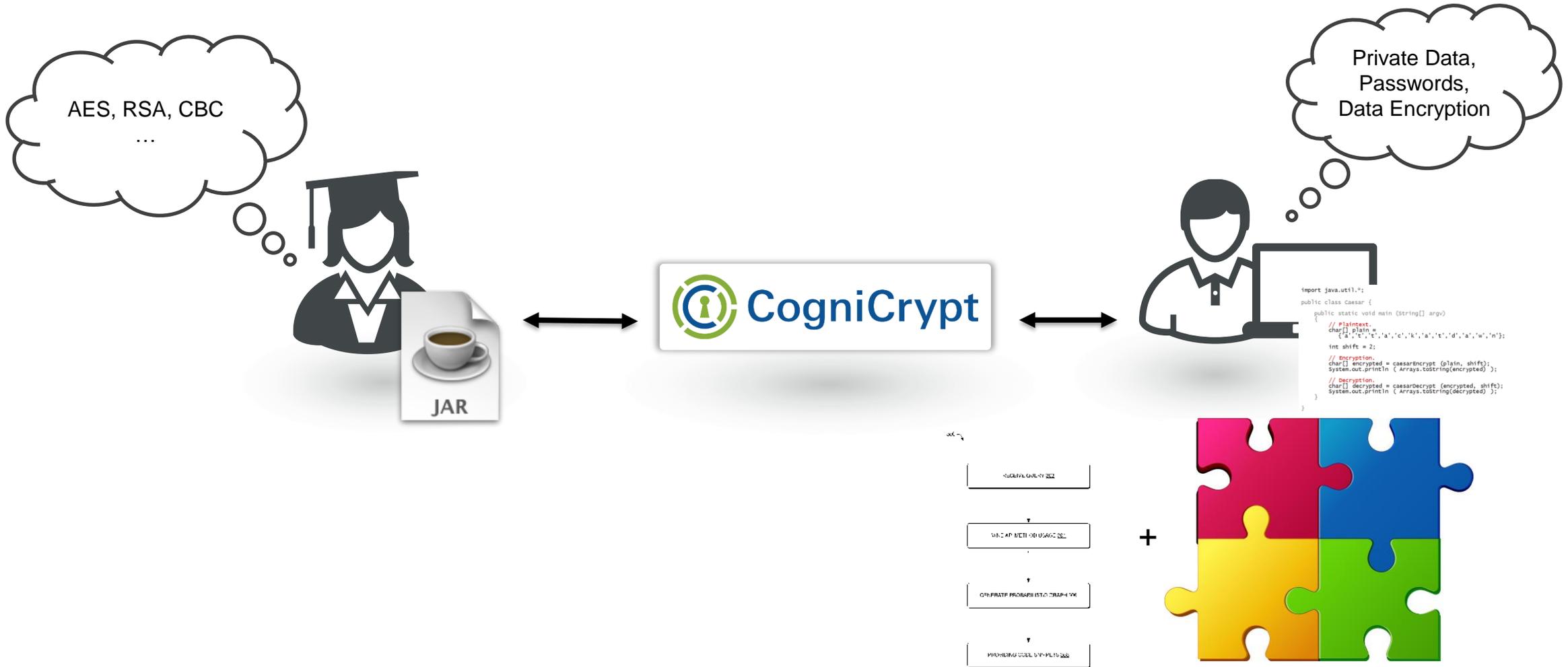


FIG. 3

What shall we do about it?

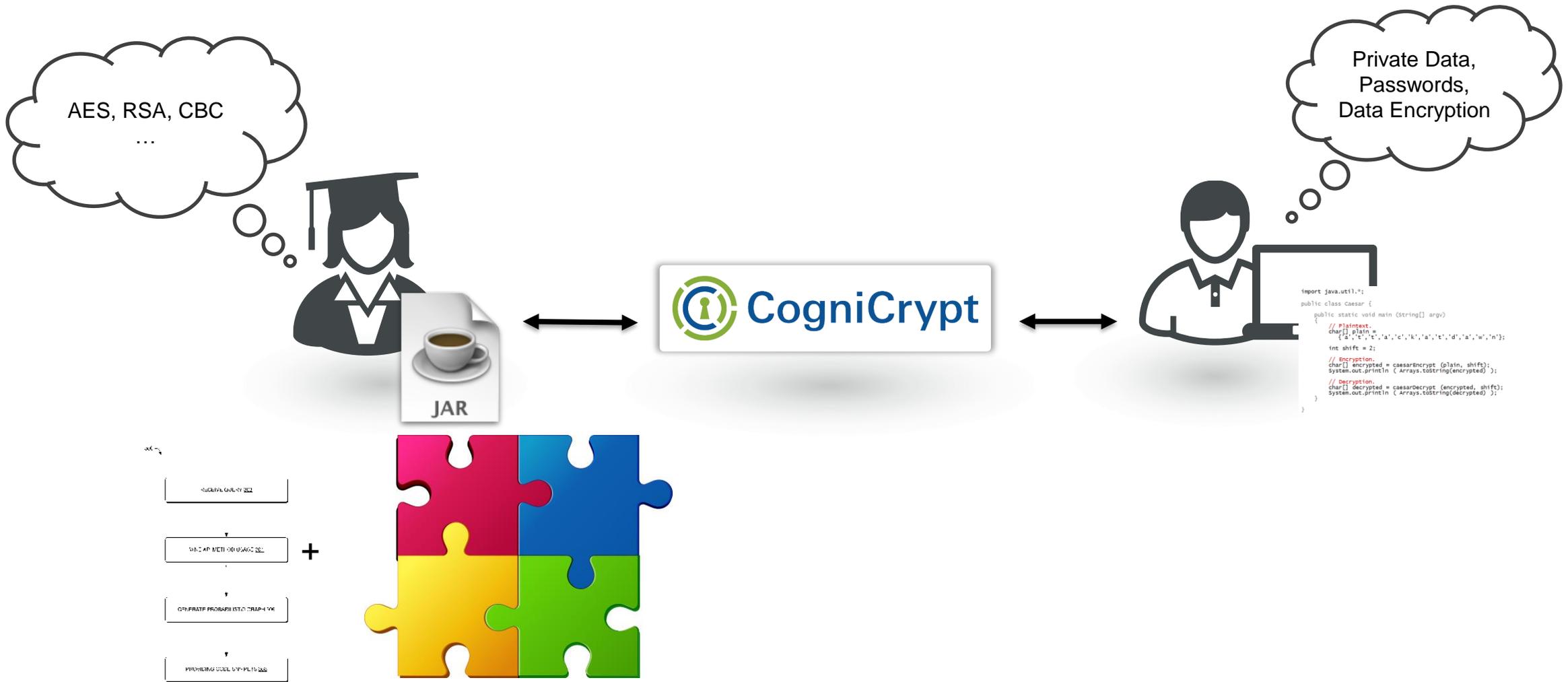
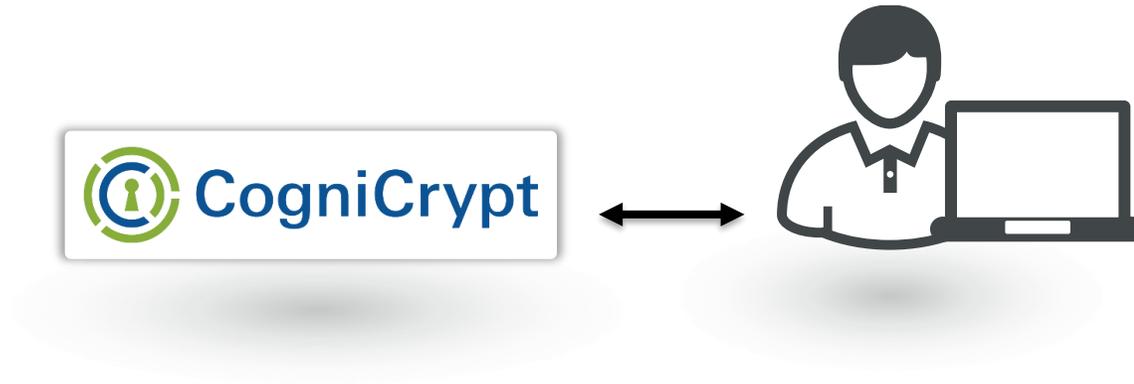


FIG. 3

CogniCrypt supports Developers as an Eclipse Plugin

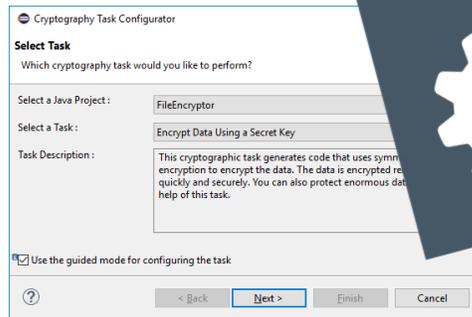


CogniCrypt supports Developers as an Eclipse Plugin

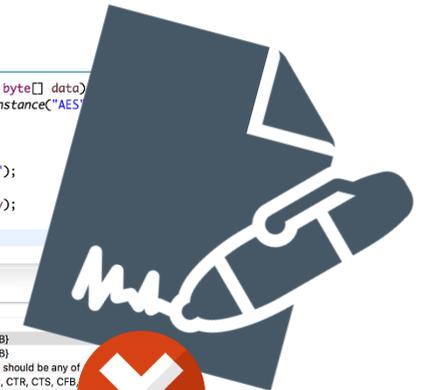
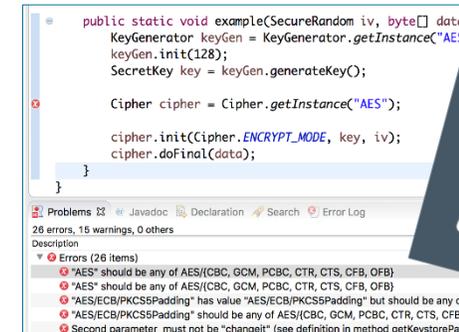


CogniCrypt's Components

CogniCrypt_{GEN}

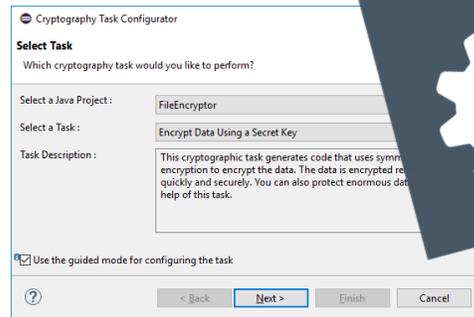


CogniCrypt_{SAST}

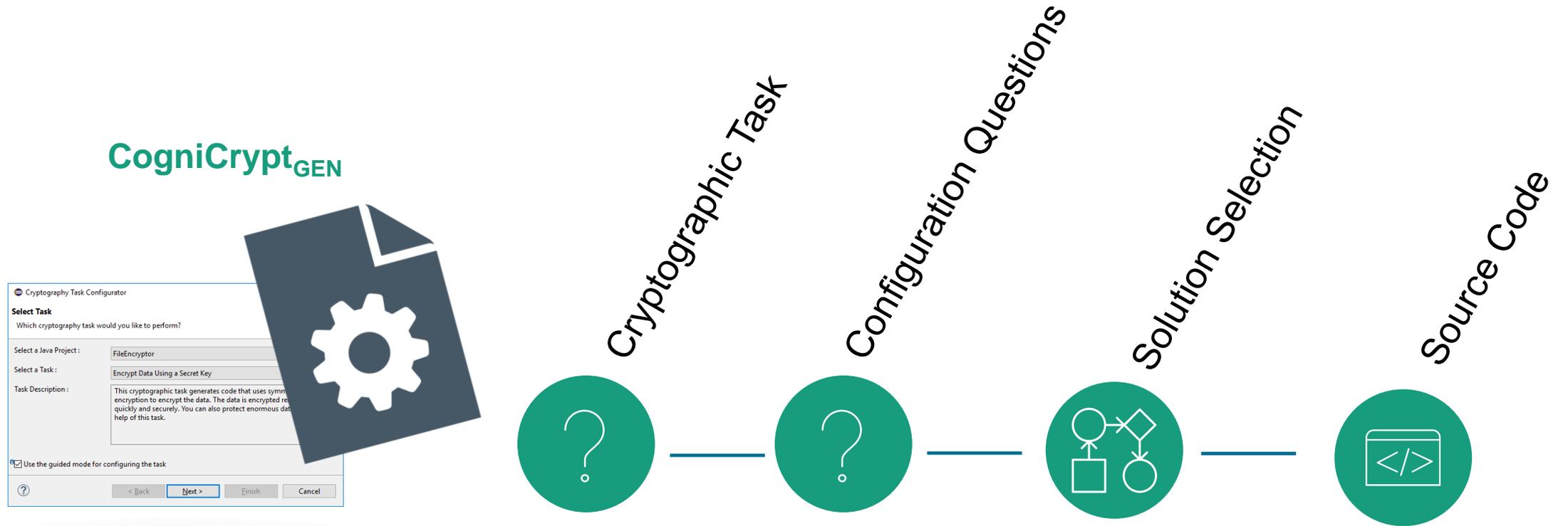


CogniCrypt_{GEN} – Code Generation For Crypto APIs

CogniCrypt_{GEN}



CogniCrypt_{GEN} – Code Generation For Crypto APIs



[CogniCrypt: Supporting Developers in using Cryptography. Stefan Krüger, Sarah Nadi, Michael Reif, Karim Ali, Eric Bodden, Mira Mezini, ASE Tool Track 2017.]

CogniCrypt_{GEN} – Code Generation For Crypto APIs



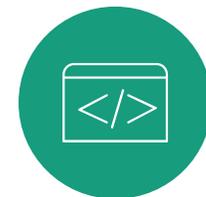
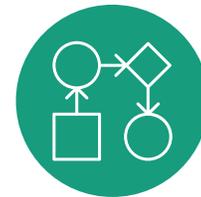
Task Descriptions

Clafer

Algorithm Model



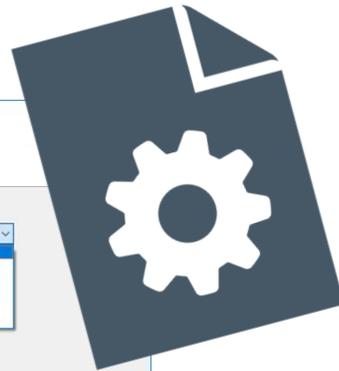
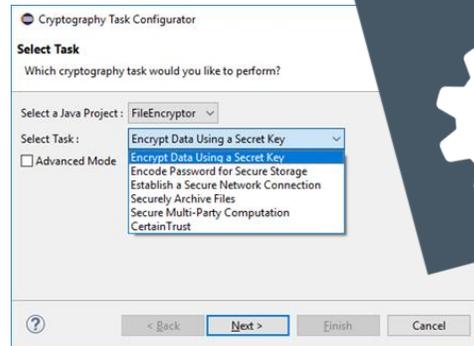
Code Template



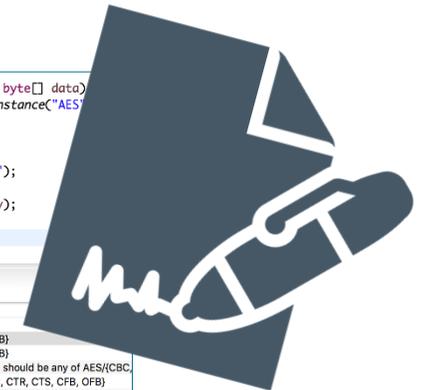
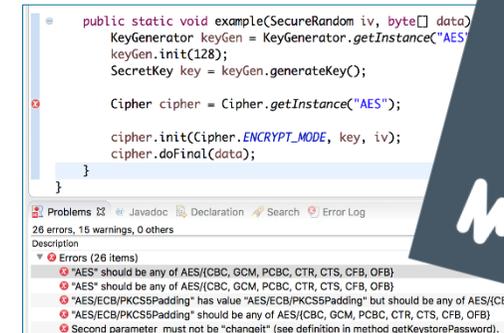
[CogniCrypt: Supporting Developers in using Cryptography. Stefan Krüger, Sarah Nadi, Michael Reif, Karim Ali, Eric Bodden, Mira Mezini, ASE Tool Track 2017.]

But... I have to Change My Code!

CogniCrypt_{GEN}



CogniCrypt_{SAST}



CogniCrypt_{SAST} – Detection of Crypto-API Misuses

CogniCrypt_{SAST}

```
public static void example(SecureRandom iv, byte[] data) {
    KeyGenerator keyGen = KeyGenerator.getInstance("AES");
    keyGen.init(128);
    SecretKey key = keyGen.generateKey();

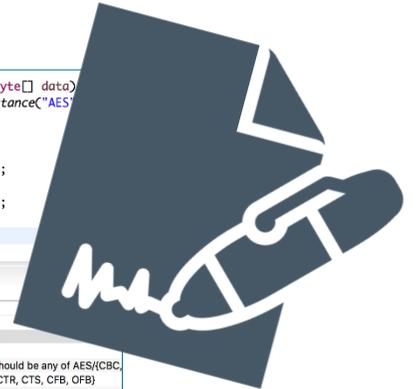
    Cipher cipher = Cipher.getInstance("AES");

    cipher.init(Cipher.ENCRYPT_MODE, key, iv);
    cipher.doFinal(data);
}
```

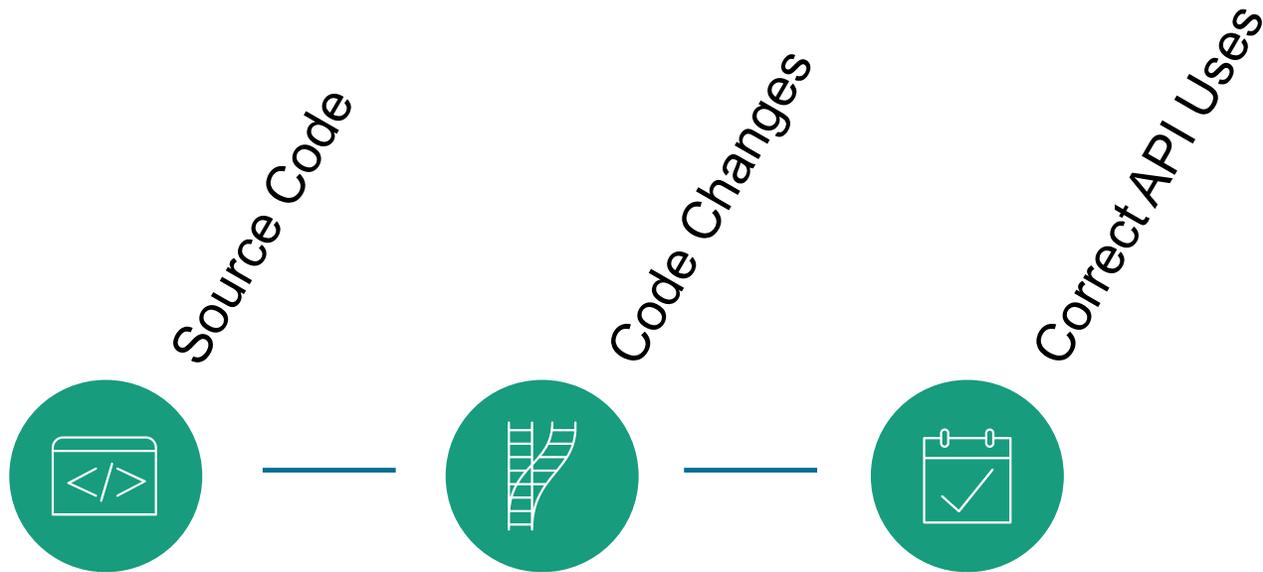
Problems 26 errors, 15 warnings, 0 others

Errors (26 items)

- ⊗ "AES" should be any of AES/CBC, GCM, PCBC, CTR, CTS, CFB, OFB
- ⊗ "AES" should be any of AES/CBC, GCM, PCBC, CTR, CTS, CFB, OFB
- ⊗ "AES/ECB/PKCS5Padding" has value "AES/ECB/PKCS5Padding" but should be any of AES/CBC, GCM, PCBC, CTR, CTS, CFB, OFB
- ⊗ "AES/ECB/PKCS5Padding" should be any of AES/CBC, GCM, PCBC, CTR, CTS, CFB, OFB
- ⊗ Second parameter must not be "changeit" (see definition in method getKeyStorePassword).



CogniCrypt_{SAST} – Detection of Crypto-API Misuses



CogniCrypt_{SAST}

```
public static void example(SecureRandom iv, byte[] data) {
    KeyGenerator keyGen = KeyGenerator.getInstance("AES");
    keyGen.init(128);
    SecretKey key = keyGen.generateKey();

    Cipher cipher = Cipher.getInstance("AES");

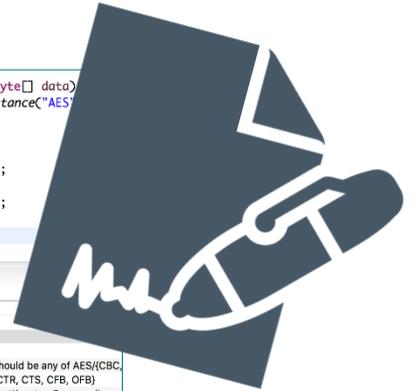
    cipher.init(Cipher.ENCRYPT_MODE, key, iv);
    cipher.doFinal(data);
}
```

Problems 26 errors, 15 warnings, 0 others

Description

▼ Errors (26 items)

- ⊗ "AES" should be any of AES/CBC, GCM, PCBC, CTR, CTS, CFB, OFB)
- ⊗ "AES" should be any of AES/CBC, GCM, PCBC, CTR, CTS, CFB, OFB)
- ⊗ "AES/ECB/PKCS5Padding" has value "AES/ECB/PKCS5Padding" but should be any of AES/CBC, GCM, PCBC, CTR, CTS, CFB, OFB)
- ⊗ "AES/ECB/PKCS5Padding" should be any of AES/CBC, GCM, PCBC, CTR, CTS, CFB, OFB)
- ⊗ Second parameter must not be "changeIt" (see definition in method getKeyStorePassword).



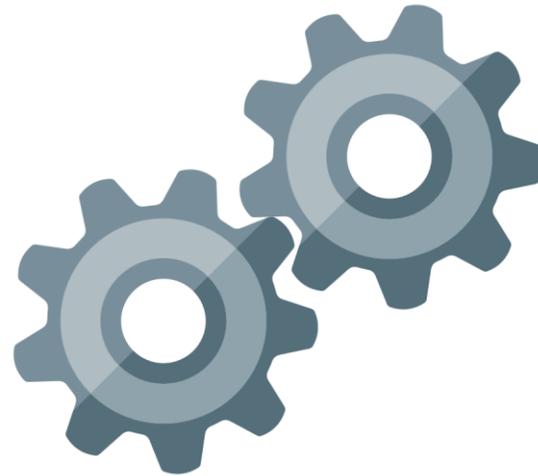
[CrySL: An Extensible Approach to Validating the Correct Usage of Cryptographic APIs. Stefan Krüger, Johannes Spaeth, Karim Ali, Eric Bodden, Mira Mezini. ECOOP 2018.]

...within Eclipse...!

Save



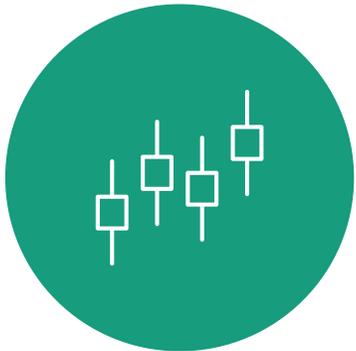
Static Analysis



Error Markers



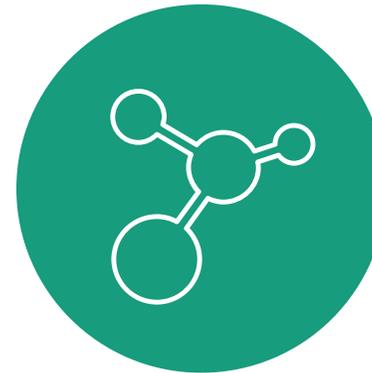
Main Error Types of CogniCrypt_{SAST}



Insecure Parameters



Incorrect Usage Pattern



Insecure Class Composition

I.E. Error Type: Insecure Parameters

```
KeyGenerator keyGen = KeyGenerator.getInstance("AES");  
keyGen.init(128);  
SecretKey key = keyGen.generateKey();
```

```
Cipher cipher = Cipher.getInstance("AES");  
cipher.init(Cipher.ENCRYPT_MODE, key, iv);  
cipher.doFinal(data);
```



„AES“ should rather be „AES/{CBC/GCM}“

[CrySL: An Extensible Approach to Validating the Correct Usage of Cryptographic APIs.
Stefan Krüger, Johannes Spaeth, Karim Ali, Eric Bodden, Mira Mezini. ECOOP 2018.]

CrySL – Specifying The Use of Crypto APIs (1/2)

SPEC `javax.crypto.KeyGenerator`

OBJECTS

```
int keySize;  
javax.crypto.SecretKey key;  
java.lang.String alg;
```

EVENTS

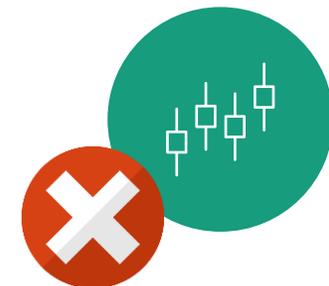
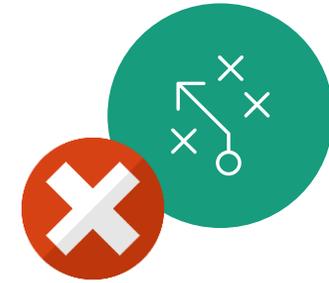
```
g1: getInstance(alg);  
g2: getInstance(alg, _);  
Gets := g1 | g2;  
i1: init(keySize);  
i2: init(keySize, _);  
i3: init(_);  
i4: init(_, _);  
Inits := i1 | i2 | i3 | i4;  
gk: key = generateKey();
```

ORDER

Gets, Inits?, gk

CONSTRAINTS

```
alg in {"AES"} => keySize in {128, 192, 256};  
alg in {"DES"} => keySize in {56};  
alg in {"Blowfish"} => keySize in {40, 44, 48,  
52, 56, ..., 436, 440};
```



[CrySL: An Extensible Approach to Validating the Correct Usage of Cryptographic APIs. Stefan Krüger, Johannes Spaeth, Karim Ali, Eric Bodden, Mira Mezini. ECOOP 2018.]

CrySL – Specifying The Use of Crypto APIs (2/2)

ENSURES

```
generatedKey(key, alg);
```

SPEC `javax.crypto.Cipher`

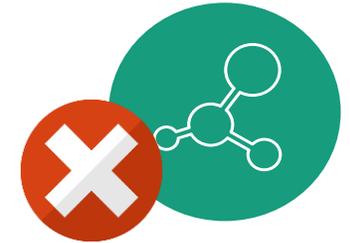
...

REQUIRES

```
generatedKey(key, alg(transformation));
```

ENSURES

```
encrypted(plainText, cipherText);
```



[CrySL: An Extensible Approach to Validating the Correct Usage of Cryptographic APIs. Stefan Krüger, Johannes Spaeth, Karim Ali, Eric Bodden, Mira Mezini. ECOOP 2018.]

CrySL Rules for the Java Cryptographic Architecture (JCA)



Encryption
&
Decryption



Signing
&
Verification



Key
Generation



```
SPEC javax.crypto.KeyGenerator
OBJECTS
java.lang.String algorithm;
int keySize;
javax.crypto.SecretKey key;
EVENTS
g1: getInstance(algorithm);
g2: getInstance(algorithm, _);
GetInstance := g1 | g2;
i1: init(keySize);
i2: init(keySize, _);
i3: init(_);
i4: init(_, _);
Init := i1 | i2 | i3 | i4;
GenKey: key = generateKey();
```

23 Classes & Interfaces



Randomness

Hashing
and
MACs



Persisting
of Key Material

[CrySL: An Extensible Approach to Validating the Correct Usage of Cryptographic APIs. Stefan Krüger, Johannes Spaeth, Karim Ali, Eric Bodden, Mira Mezini. ECOOP 2018.]

Evaluation - Android



10k

Android Apps

96%

of these Apps
are insecure

[CrySL: An Extensible Approach to Validating the Correct Usage of Cryptographic APIs.
Stefan Krüger, Johannes Spaeth, Karim Ali, Eric Bodden, Mira Mezini. ECOOP 2018.]

Evaluation - Maven

Maven™

☰ The Central Repository

2.7+

Million
Software Artefacts

73%

of these Artefacts
are insecure

When a Developer Uses a Crypto API

```
KeyGenerator keyGen = KeyGenerator.getInstance("AES");  
keyGen.init(128);  
SecretKey key = keyGen.generateKey();
```

```
Cipher cipher = Cipher.getInstance("AES");  
cipher.init(Cipher.ENCRYPT_MODE, key, iv);  
cipher.doFinal(data);
```


• **getInstance**(String transformation) : Cipher - Cipher
• **getInstance**(String transformation, Provider provider) : Cipher - Cipher
• **getInstance**(String transformation, String provider) : Cipher - Cipher

When a De

What shall we do about it?

KeyGenera
keyGen.in
SecretKey

Cipher ci
cipher.in
cipher.do

```
@getInstance()  
@getInstance()  
@getInstance()
```

© CRC 1119 CROSSING, TU Darm
Folie 2



```
import java.util.*;  
public class Caesar {  
    // plaintext  
    char[] plain;  
    char[] cipher; // encrypted  
    int shift = 2;  
    // encryption  
    char[] encrypted = caesarEncrypt(plain, shift);  
    System.out.println ("Array.toString(encrypted)");  
    // decryption  
    char[] decrypted = caesarDecrypt(encrypted, shift);  
    System.out.println ("Array.toString(decrypted)");  
}
```

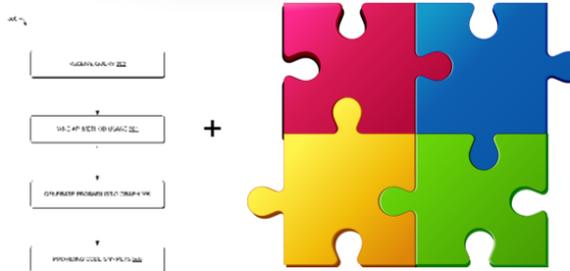


FIG. 3

When a D

CogniCrypt's Components

KeyGener
keyGen.i
SecretKe

Cipher <
cipher.i
cipher.c

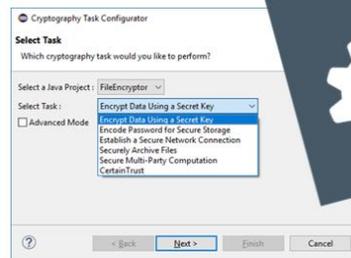


© CRC 1119 CROSSING, TU I
Folie 2

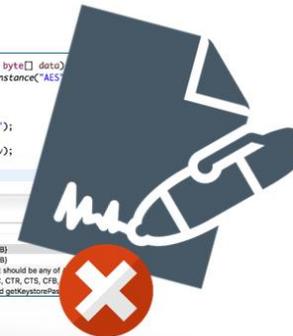
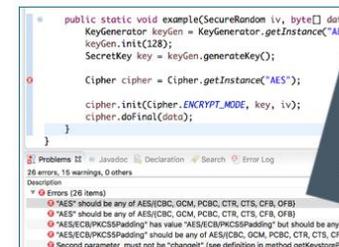
a Data,
words,
ryption

UT
CROSSING

CogniCrypt_{GEN}



CogniCrypt_{SAST}



© CRC 1119 CROSSING, TU Darmstadt/Heinz Nixdorf Institut
Folie 6

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN



When a D

Evaluation

KeyGener
keyGen.i
SecretKe

Cipher <
cipher.i
cipher.c



© CRC 1119 CROSSING, TU I
Folie 2

CogniCry



© CRC 1119 CROSSING, T
Folie 8

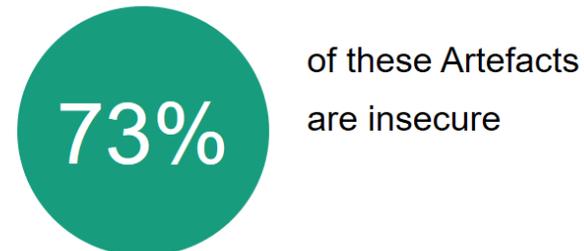
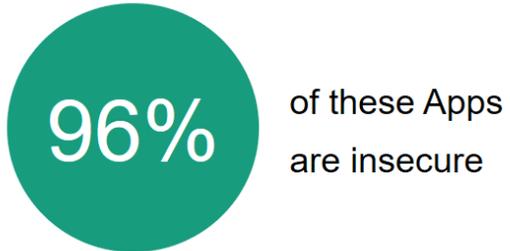
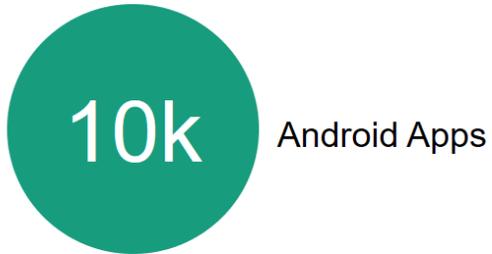


Maven™

The Central Repository



Small text in a cloud icon



© CRC 1119 CROSSING, TU Darmstadt/Heinz Nixdorf Institut
Folie 20

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN



When a Developer Uses a Crypto API

```
KeyGenerator keyGen = KeyGenerator.getInstance("AES");
keyGen.init(128);
SecretKey key = keyGen.generateKey();
```

```
Cipher cipher = Cipher.getInstance("AES");
cipher.init(Cipher.ENCRYPT_MODE, key, iv);
cipher.doFinal(data);
```

```
getinstance(String transformation) : Cipher - Cipher
getinstance(String transformation, Provider provider) : Cipher - Cipher
getinstance(String transformation, String provider) : Cipher - Cipher
```

What shall we do about it?

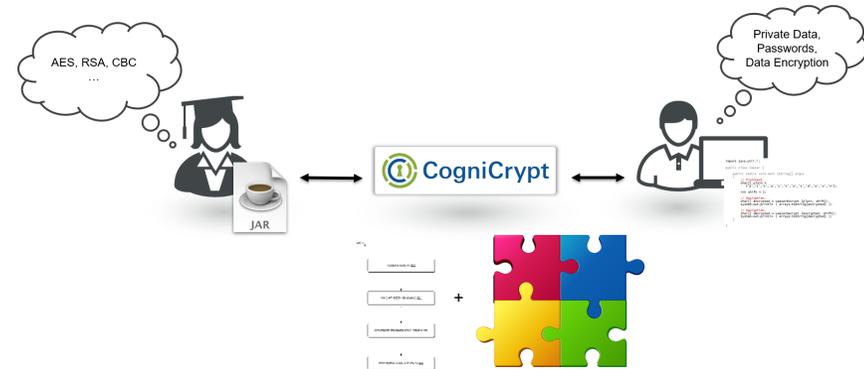


FIG. 3

CogniCrypt's Components



Evaluation



10k Android Apps

2.7+ Million Software Artefacts

96% of these Apps are insecure

73% of these Artefacts are insecure