SECURE INTEGRATION OF CRYPTOGRAPHIC SOFTWARE

Speaker: Stefan Krüger

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

CROSSING

TECHNISCHE
UNIVERSITÄT
DARMSTADT

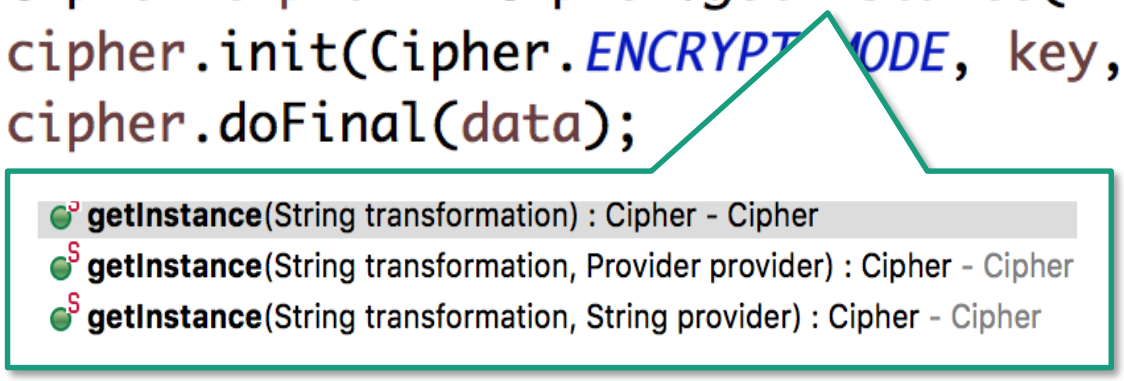
HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

CROSSING

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);
```



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 vulnerabilites
result from API misuse

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

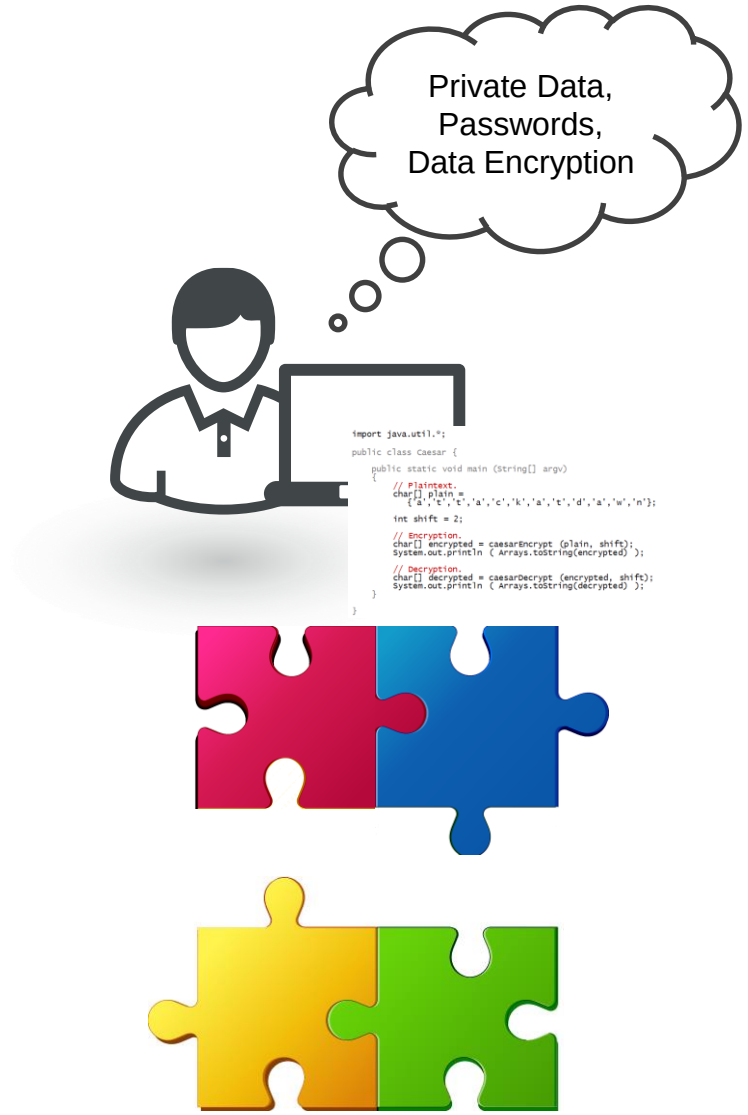
CROSSING

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?

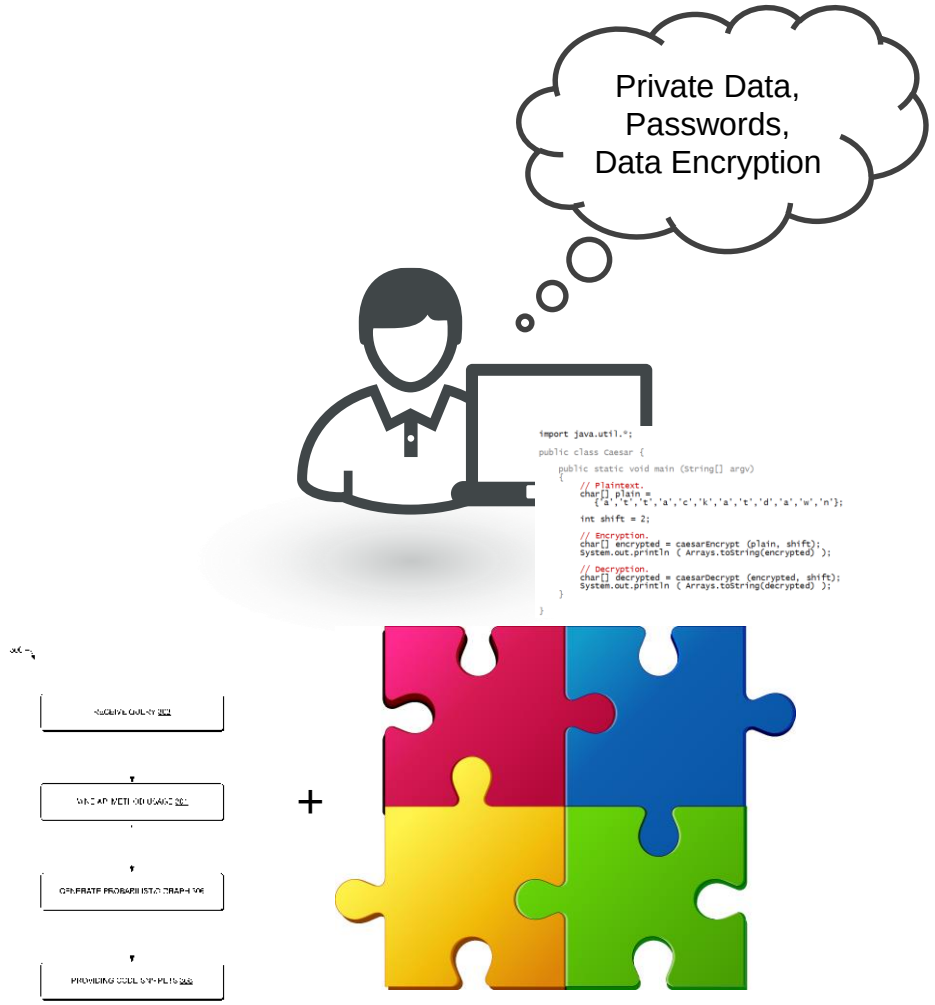


FIG. 3

What shall we do about it?

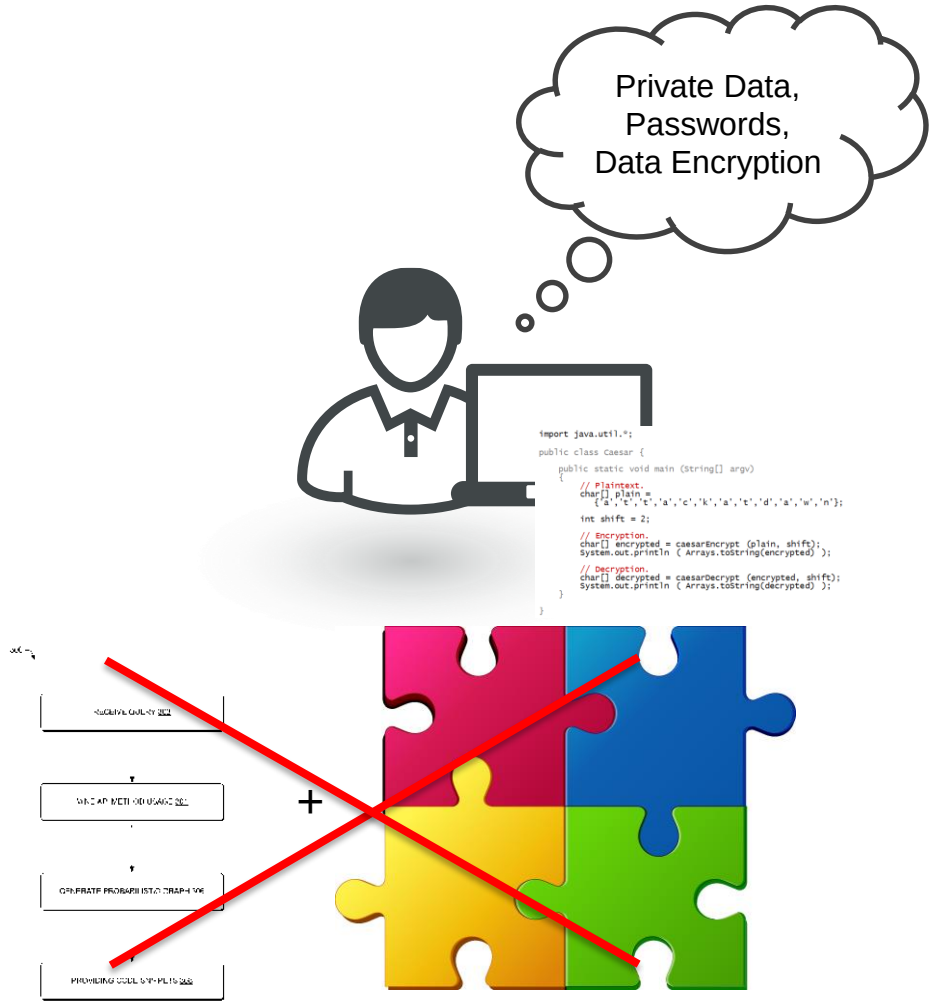


FIG. 3

What shall we do about it?

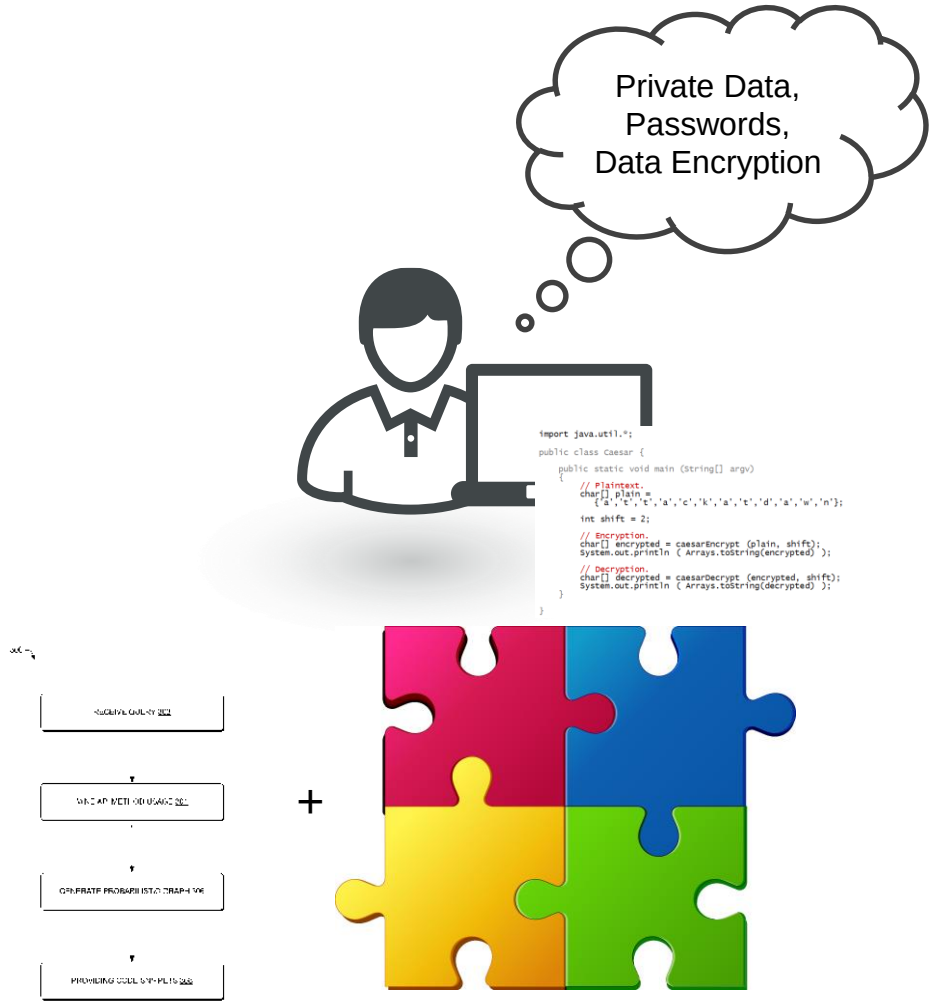


FIG. 3

What shall we do about it?

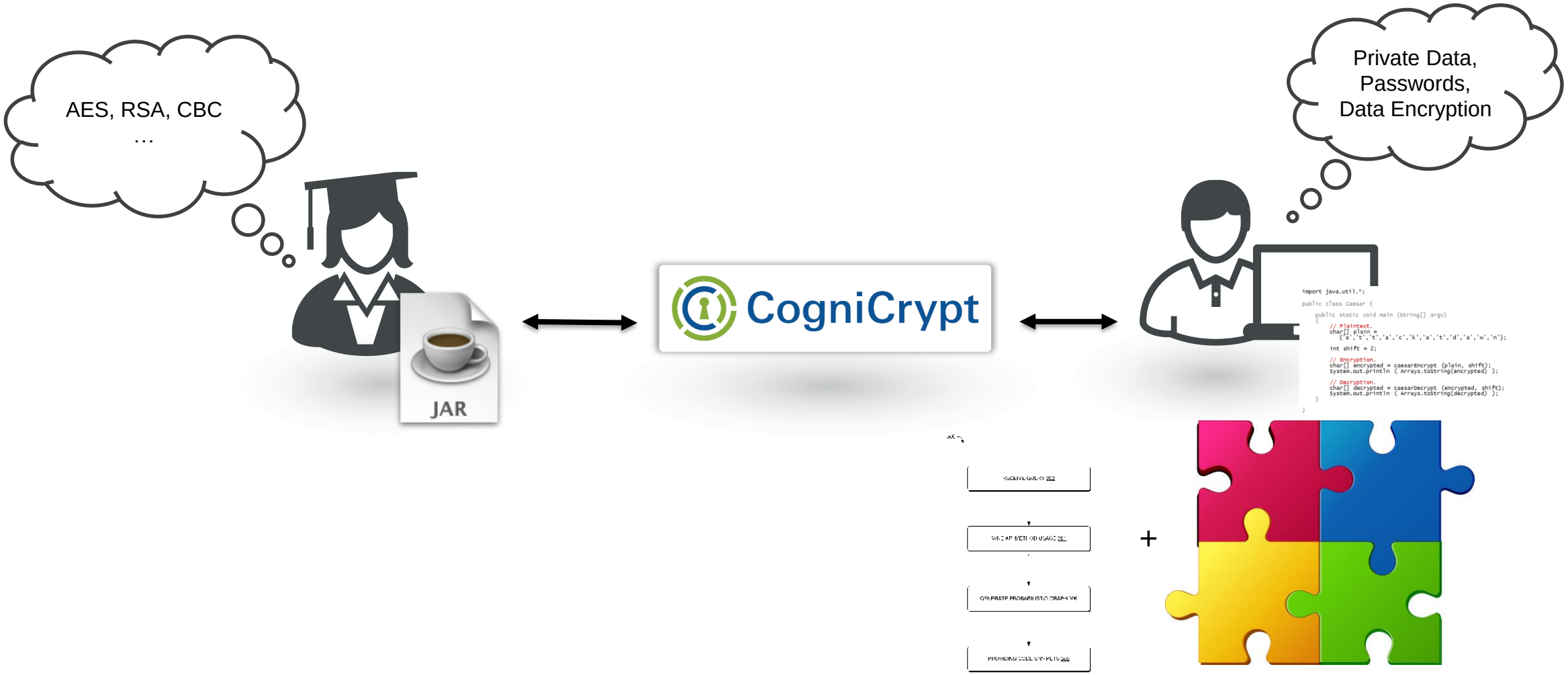


FIG. 3

What shall we do about it?

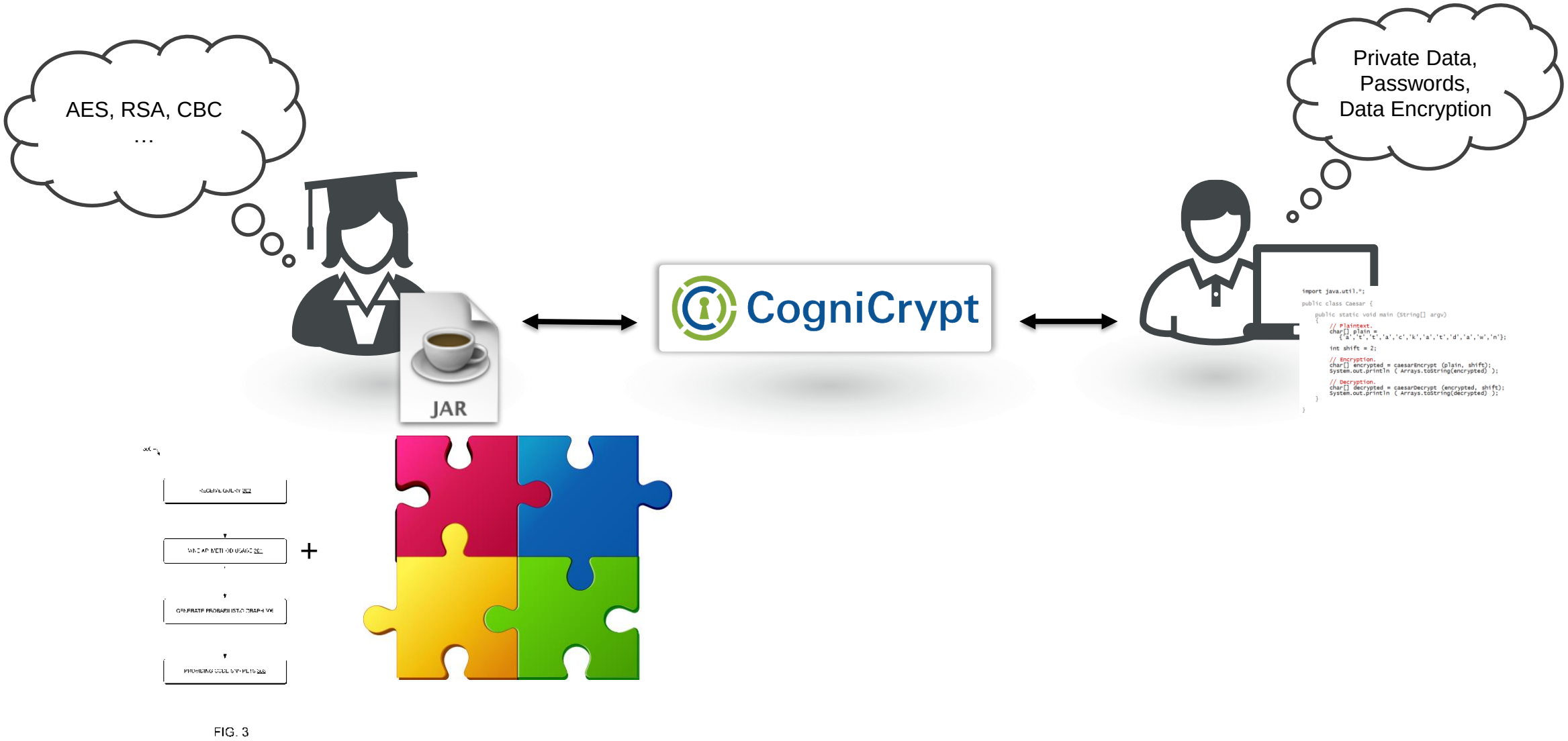
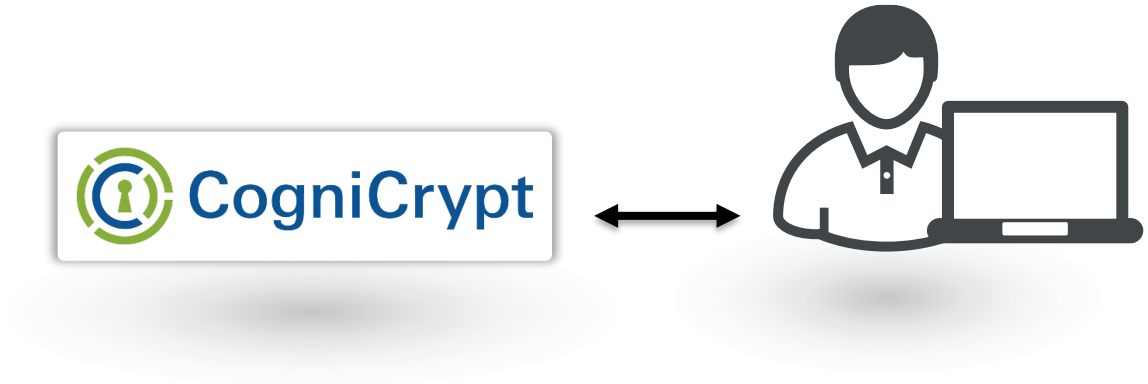


FIG. 3

CogniCrypt supports Developers as an Eclipse Plugin



CogniCrypt supports Developers as an Eclipse Plugin

CogniCrypt‘s Components

CogniCrypt$_{GEN}$

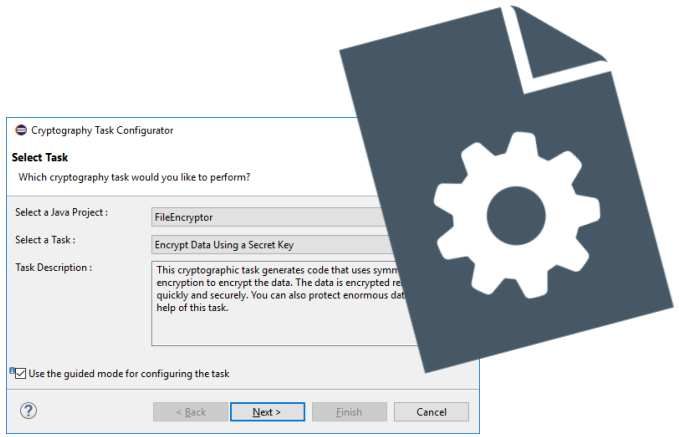
CogniCrypt$_{SAST}$

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

CROSSING

CogniCrypt$_{GEN}$ – Code Generation For Crypto APIs

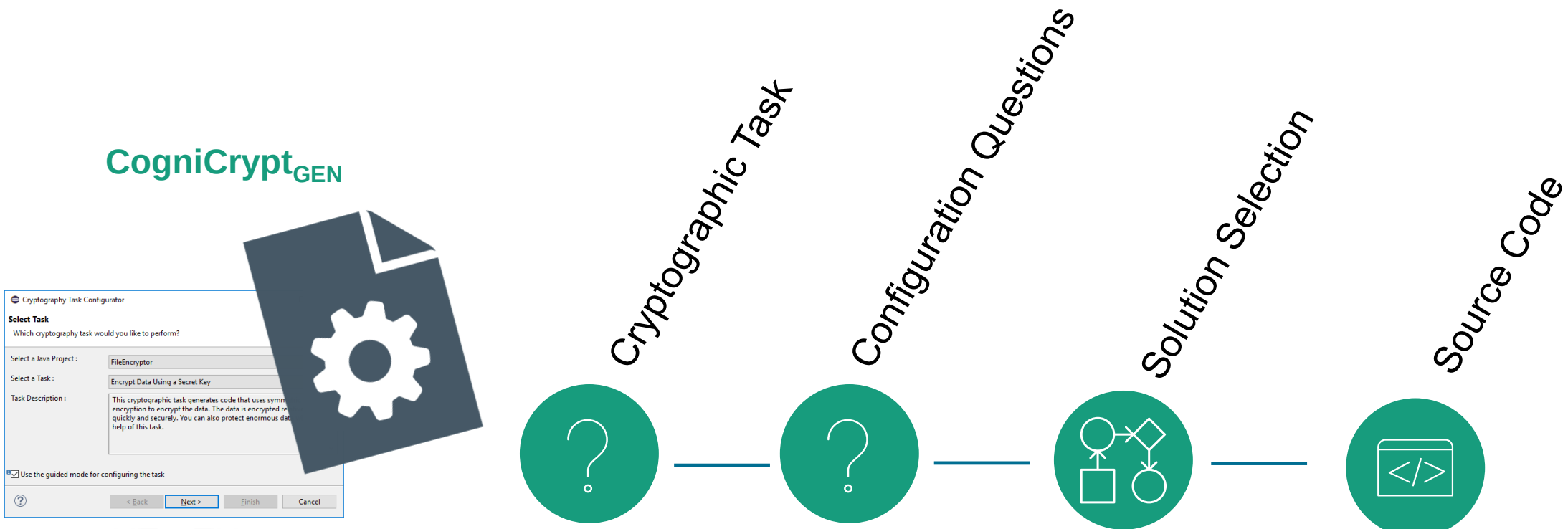
CogniCrypt$_{GEN}$



HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

CROSSING

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.]

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

CROSSING

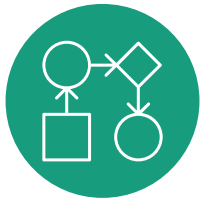
CogniCrypt$_{GEN}$ – Code Generation For Crypto APIs

Task Descriptions

Clafer

Algorithm Model

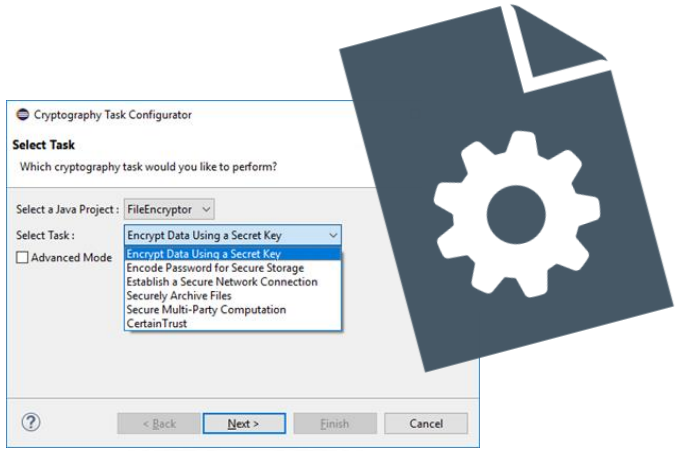
Code Template



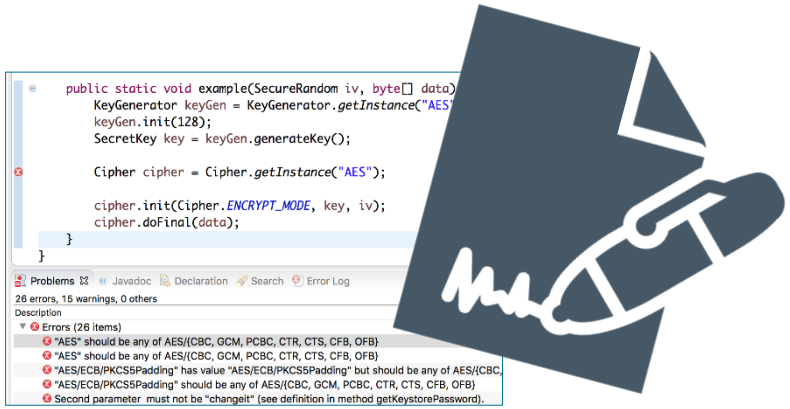
[CogniCrypt: Supporting Developers in using Cryptography. Stefan Krüger, Sarah Nadi, Michael Reif, Karim Ali, Eric Bodden, Mira Mezini, … . ASE Tool Track 2017.]

But… I have to Change My Code!

CogniCrypt$_{GEN}$

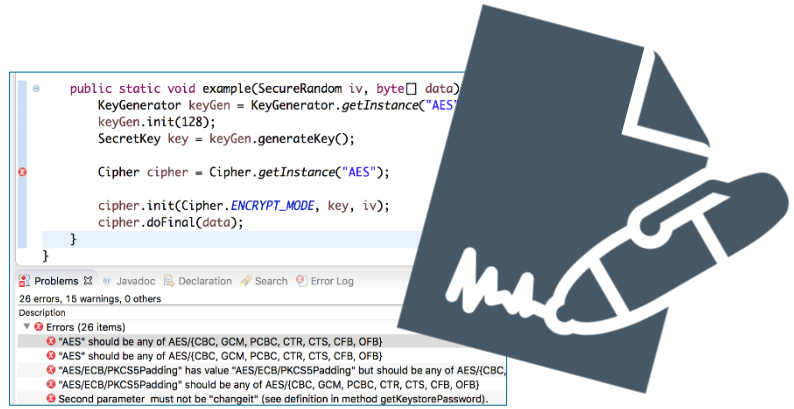


CogniCrypt$_{SAST}$

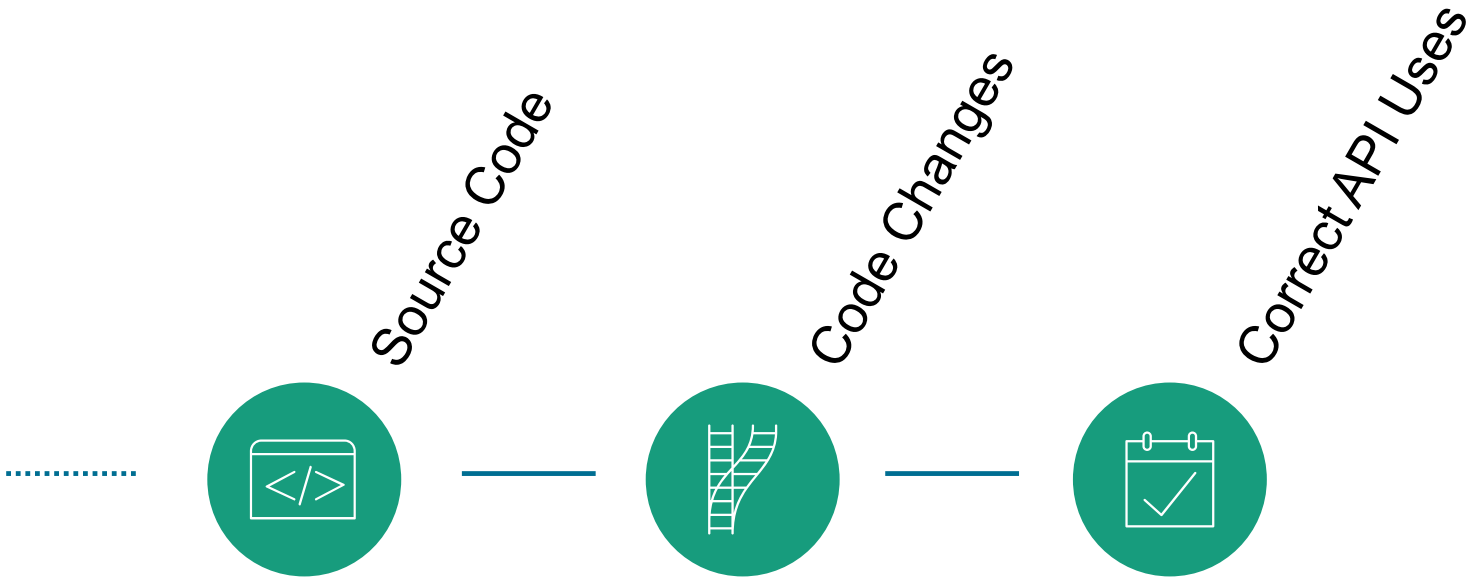


CogniCrypt$_{SAST}$ – Detection of Crypto-API Misuses

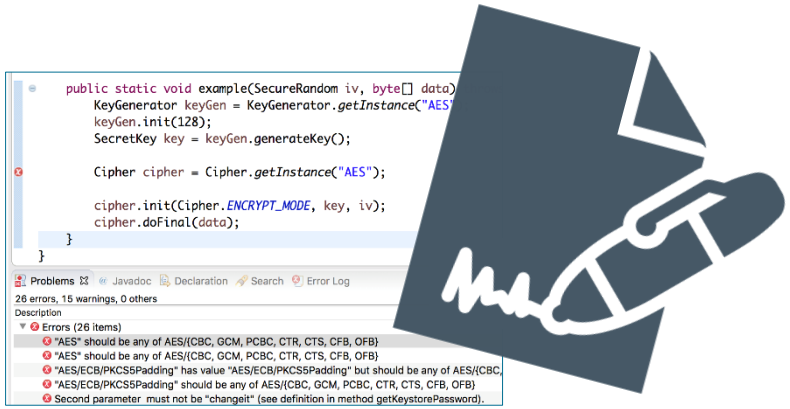
CogniCrypt$_{SAST}$



CogniCrypt$_{SAST}$ – Detection of Crypto-API Misuses



CogniCrypt$_{SAST}$



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

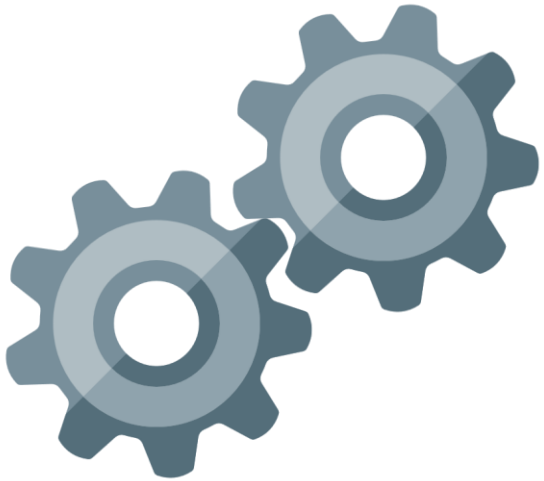
HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

CROSSING

…within Eclipse…!

Save

Static Analysis



Error Markers

Main Error Types of CogniCrypt$_{SAST}$

Incorrect Usage Pattern

Insecure Parameters

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);
```

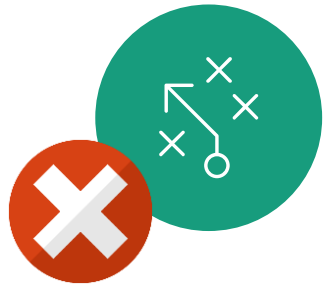
„AES“ should rather be „AES/{CBC/GCM}“

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

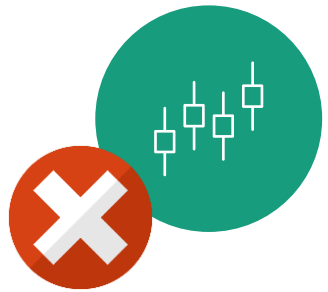
CrySL – Specifiying The Use of Crypto APIs (1/2)

```
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
```



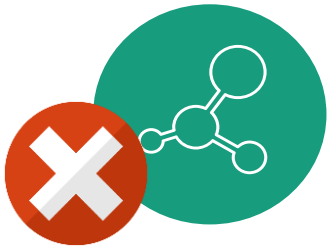
```
CONSTRAINTS
 alg in {"AES"} => keySize in {128, 192, 256};
 alg in {"DES"} => keySize in {56};
 alg in {"Blowfish"} => keySize in {40, 44, 48,
52, 56, ..., 436, 440};
```



[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 – Specifiying 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

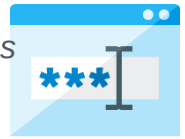
Signing & Verification

Key Generation

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

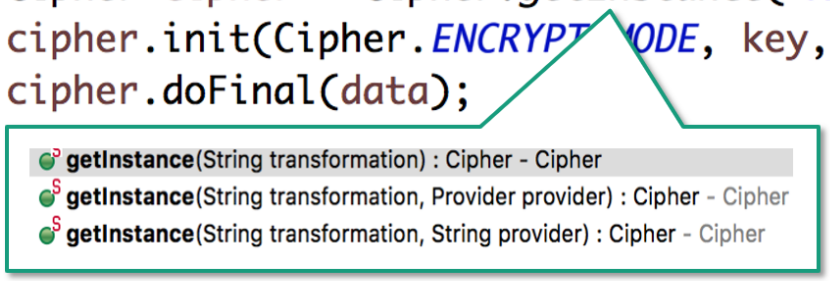
2.7+ Million Software Artefacts

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);
```



© CRC 1119 CROSSING, TU Darmstadt/Heinz Nixdorf Institut

When a De

KeyGenera
keyGen.in
SecretKey

Cipher ci
cipher.in
cipher.do

getInstance(
getInstance(
getInstance(

What shall we do about it?

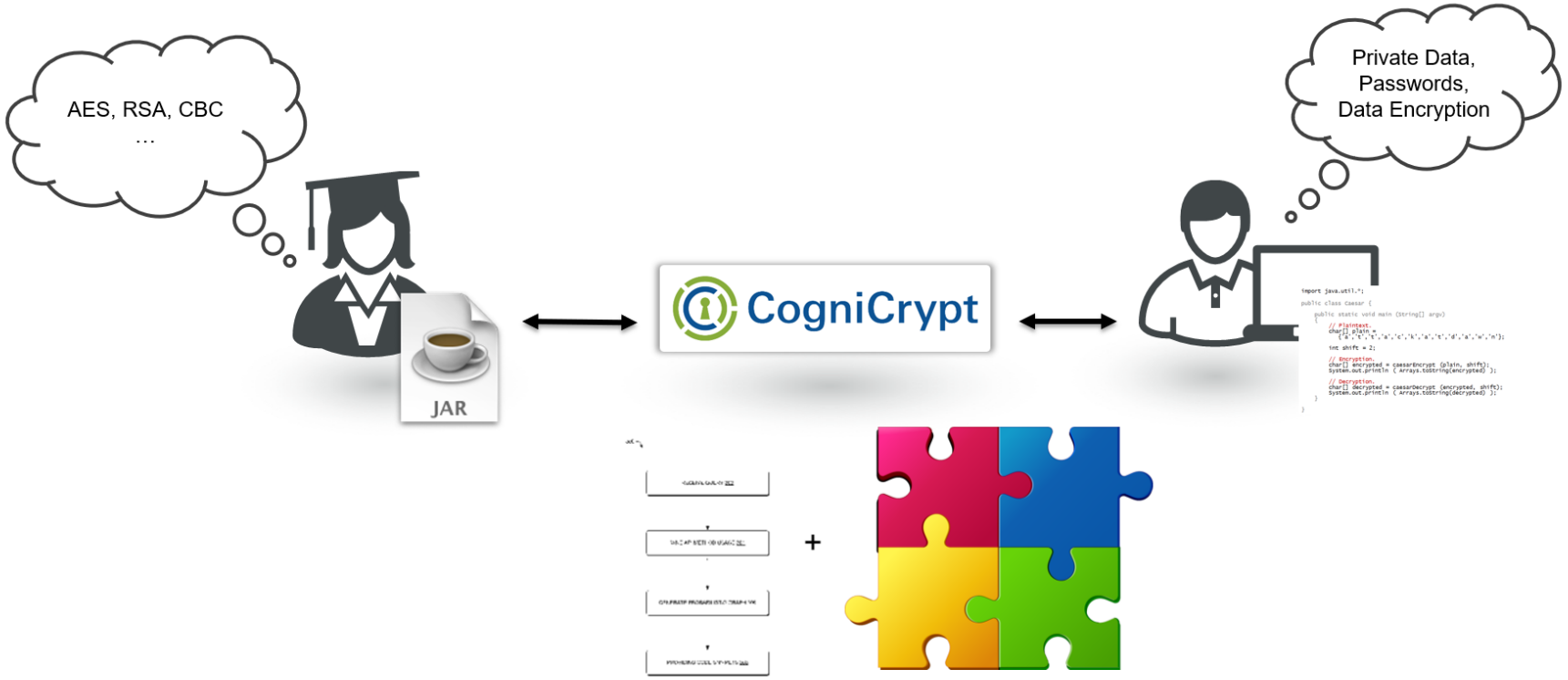


FIG. 3

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

CROSSING

© CRC 1119 CROSSING, TU Darmstadt/Heinz Nixdorf Institut

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

CROSSING

When a D

CogniCrypt‘s Components

KeyGener
keyGen.i
SecretKe

Cipher c
cipher.i
cipher.d

getInstan
getInstan
getInstan

e Data,
words,
cryption

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN
CROSSING

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN
CROSSING

Evaluation

10k Android Apps

2.7+ Million Software Artefacts

96% of these Apps are insecure

73% of these Artefacts are insecure

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

CROSSING

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

CROSSING

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?

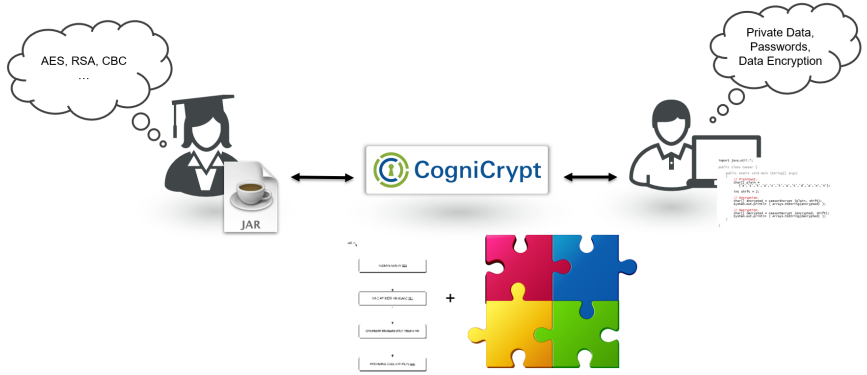


FIG. 3

CogniCrypt‘s Components

Evaluation

ANDROID

Maven™

The Central Repository

10k	Android Apps
96%	of these Apps are insecure
2.7+	Million Software Artefacts
73%	of these Artefacts are insecure