Functional Encryption & Homomorphic Encryption for RIOT

Experiences with Benchmarking
Advanced Cryptography in RIOT

Max Pengrin
max.pengrin@hft-stuttgart.de
HFT Stuttgart, Germany

RIOT Summit 2020
Bio

- Max Pengrin
- B. Sc. Computer Science
- Prospective M. Sc. IT-Security
- Academic Associate at HFT Stuttgart, SensAR

Sensory data and augmented reality, a research project at HFT Stuttgart sponsored by the Carl-Zeiss-Foundation (https://www.hft-stuttgart.com/research/projects/current/sensar)
Overview of this Talk

• Objectives of our Work
• Introduction to selected Crypto Schemes and their Applicability to IoT-Scenarios
• Such special Crypto Schemes in RIOT
• Benchmarking Challenges and initial results
• Lessons Learned
• Outlook & Future Work
Advances in the Crypto-Community

• Many specific crypto schemes have been developed beyond traditional symmetric/asymmetric crypto (AES, RSA)
  • E.g. Secure multiparty computation, proxy re-encryption, functional encryption, homomorphic encryption, quantum-proof encryption, …
  • Each of these offer useful properties/features for many IoT scenarios
  • Mostly theoretic work by the formal crypto community, but some prototype implementations exist

→ There is still a lack of **systematic exploration** and a thorough evaluation of the actual scalability and performance of these approaches, in particular on **restricted IoT-nodes**
Objectives

• Our goals …
  • Closing the gap between the theoretical cryptography community (i.e. mathematical proofs) and the network security community (i.e. applied cryptography in actual real-world systems)
  • Working out ready-to-use applied cryptography solutions that make special crypto usable in RIOT and providing performance estimations to the IoT community

• Current Work and Focus
  • Functional and Homomorphic encryption
  • Enabling comparable benchmarking for various such schemes under RIOT
Functional Encryption

Identity-based Encryption (IBE)
- Allows anyone to generate a public key for some entity based on an ASCII string
  - No need to retrieve public keys / certificates for encryption
  - Receiving entity receives corresponding private key for decryption from special trusted third party
➢ IoT nodes can encrypt different data for various receivers without the need to retrieve / store multiple public keys

Attribute-based Encryption (ABE)
- Allows to encrypt data with a policy attached
  - Example policy: (level = manager OR department = C)
- Only entities whose private key matches the policy can decrypt the data
➢ Very useful for IoT as a means for enabling Data-Centric Security (Content Object Security)
  - Access policy is included in the data itself
  - Access policy can be specified at data production time in an abstract way without knowing the exact identities of consumers
IoT Example: Attribute-based Encryption (ABE)
Homomorphic Encryption

- Applying some mathematical function on encrypted data
- Crypto community is working hard on enabling fully-homomorphic encryption (FHE)
  - I.e. using any mathematical function
  - Not realistic for IoT devices anytime soon from a performance perspective …
- But: **partially homomorphic** encryption may be feasible on IoT nodes
  - E.g., homomorphic addition $e(x) + e(y) = e(x + y)$
  - E.g., homomorphic multiplication $e(x) \times e(y) = e(x \times y)$
- Example IoT Use Case
  - Sensor node A can forward encrypted data to next sensor node B, which can add its encrypted data without reading sensor A’s value; sink node can decrypt the sum (and calculate the average)
  - Zero-Knowledge Computation - out-sourcing of computational task, etc. more use cases imaginable
Integrating Functional and Homomorphic Encryption in RIOT?

• Several open source libraries exist, but:
  • Some outdated / not maintained
    • E.g. last commit over 2 years old
  • Some not suitable for RIOT
    • E.g. using non-supported programming language
    • E.g. using dynamic memory allocation

• What had been done?
  • Relic toolkit
  • Palisade (FHE, ABE, C++)
  • Paring-based Library Stanford (just one example w\ IBE)
  • SEAL Microsoft (>2GB RAM)
  • TFHE: Fast Fully Homomorphic Encryption (requires x86_64)
  • Many more in development (ibm, google, helib)

• What to do?
Integrating Functional and Homomorphic Encryption in RIOT

• What's done in RIOT?
  • Relic toolkit
    • Developed with embedded in mind
    • Supports many interesting special crypto schemes that have not been investigated in depth in RIOT
      • "Cryptographic protocols (RSA, Rabin, ECDSA, ECMQV, ECSS (Schnorr), ECIES, Sakai-Ohgishi-Kasahara ID-based authenticated key agreement, Boneh-Lynn-Schacham and Boneh-Boyen short signatures, Paillier and Benaloh homomorphic encryption systems)"
    • RELIC already used/integrated in RIOT - Available as a plugin

Repository and Image: https://github.com/relic-toolkit/relic
Some Initial Results

<table>
<thead>
<tr>
<th>label</th>
<th>algorithm</th>
<th>type</th>
<th>runtime</th>
<th>std-dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>cp_rsa_enc</td>
<td>RSA Encryption</td>
<td>traditional</td>
<td>526</td>
<td>46,56</td>
</tr>
<tr>
<td>cp_rsa_dec</td>
<td>RSA Decryption</td>
<td>traditional</td>
<td>1625</td>
<td>281,62</td>
</tr>
<tr>
<td>cp_phpe_enc</td>
<td>Paillier Encryption</td>
<td>homomorphic</td>
<td>4772</td>
<td>700,81</td>
</tr>
<tr>
<td>cp_phpe_dec</td>
<td>Paillier Decryption</td>
<td>homomorphic</td>
<td>2410</td>
<td>312,09</td>
</tr>
<tr>
<td>cp_ecdh_key</td>
<td>Elliptic-Curve Diffie-Hellman Key Derivation</td>
<td>traditional</td>
<td>6217</td>
<td>1187,12</td>
</tr>
<tr>
<td>cp_ibe_enc</td>
<td>Boneh-Franklin IBE Encryption</td>
<td>functional</td>
<td>55592</td>
<td>2871,34</td>
</tr>
<tr>
<td>cp_ibe_dec</td>
<td>Boneh-Franklin IBE Decryption</td>
<td>functional</td>
<td>27184</td>
<td>1446,86</td>
</tr>
<tr>
<td>cp_bgn_enc1</td>
<td>Boneh-Goh-Nissim Encryption</td>
<td>homomorphic</td>
<td>6988</td>
<td>1137,37</td>
</tr>
<tr>
<td>cp_bgn_dec1</td>
<td>Boneh-Goh-Nissim Decryption</td>
<td>homomorphic</td>
<td>7008</td>
<td>764,36</td>
</tr>
<tr>
<td>cp_bgn_mul</td>
<td>Boneh-Goh-Nissim Homomorphic Multiplication</td>
<td>homomorphic</td>
<td>115179</td>
<td>8645,88</td>
</tr>
<tr>
<td>cp_bgn_add</td>
<td>Boneh-Goh-Nissim Homomorphic Addition</td>
<td>homomorphic</td>
<td>608</td>
<td>62,35</td>
</tr>
</tbody>
</table>

Runtime of various crypto schemes under RIOT
(RIOT-native on i7 CPU, results in µs)
Benchmarking Challenges

• Relic build-in benchmarks not supported in RIOT
  • partly hardcoded parameters
  • Me = Noob - no development background

• Ongoing work on harmonizing keylengths etc. to enable comparable benchmarking of different algorithms
  • two types of keys at the same time elliptic curve- & prime-based
  • Key length ≠ Key length; Securitybits?

• Repository: https://github.com/maksim-ka/RIOT_Projects
Lessons Learned

• Memory is crucial, also Power usage (but not necessarily in our usecases), Architecture, Library jungle, Buildsystems are hard ...etc. pp.

• RIOT-OS is nice, because...
  • Process managing
  • Drivers
  • Plugins

• More mainstream like/newbie-friendly access would be appreciated
  • Vscode extension (out of the box debugging)
  • Arduino-Code/Libs, PlatformIO support

• Thanks to the friendly 😊 RIOT community for help in getting started
  • Special thanks to Peter Kietzmann for lots of initial assistance!
Outlook and Future Work

Ongoing Work

• Harmonizing benchmarking among various crypto schemes
• Running large-scale experiments on FIT-IoT-Lab

Future Work

• Have students run experiments on IoT hardware
  • at HFT Stuttgart security lab
• Integrate more advanced crypto into RIOT
  • Find suitable crypto-libs / open-source implementations
  • With interesting crypto-schemes (e.g. quantumproof-crypto)