# Entwicklung und Validierung von Kryptoalgorithmen mit der vom BSI evaluierten Kryptobibliothek Botan

HTV Halbleiter-Test & Vertriebs-GmbH, Bensheim, Deutschland

## Kurzfassung

Die moderne Kommunikationstechnik benötigt sichere kryptographische Abläufe, um die Schutzziele der Informationssicherheit (Vertraulichkeit, Integrität, Authentizität und Verfügbarkeit) zu realisieren. Damit eine sichere Kommunikation in Deutschland und weltweit möglich ist, begleitet und optimiert das BSI (Bundesamt für Sicherheit in der Informationstechnik) die Kryptobibliothek Botan. HTV nutzt Botan für die Entwicklung und Validierung von hardwarebasierten Kryptoalgorithmen, die in FPGA-Chips zum Einsatz kommen. FPGAs (Field Programmable Gate Arrays) sind integrierte Schaltkreise der Digitaltechnik und kommen in vielen Gebieten der digitalen Elektronik zum Einsatz. Aufgrund ihrer hohen Flexibilität und Parallelität werden sie gerne als Co-Prozessoren für CPUs eingesetzt, um deren Performance zu beschleunigen. In Rechenzentren dienen FPGAs z. B. zur Berechnung kryptographischer Algorithmen.

## 1 Die Kryptobibliothek Botan

Im Jahr 2017 wählte das BSI aus 18 öffentlich verfügbaren Kryptobibliotheken Botan als Favoriten für eine Weiterentwicklung zu einer sicheren Kryptobibliothek aus [1]. Botan ist quelloffen und gilt als sicher, übersichtlich, kontrollierbar und gut dokumentiert. Es ist für viele Einsatzszenarien geeignet. Das BSI fügte in einem eigenen Entwicklungszweig von Botan auf GitHub der Bibliothek weitere Testfälle und Anpassungen hinzu, damit es auch in Anwendungen mit erhöhtem Sicherheitsbedarf (z. B. VS-NfD) eingesetzt werden kann [2]. Dieser BSI-Entwicklungszweig von Botan wird in den nächsten Jahren kontinuierlich weiter gepflegt.

## 2 Modifizierbarkeit von Botan

Ein besonderer Vorteil von Botan ist es, dass es modular aufgebaut ist. So können bei Botan über ein Konfigurationsscript (configure.py) einzelne Module bzw. Algorithmen ausgewählt und anschließend zu einer Bibliothek in Linux kompiliert werden. Mit dem folgenden Befehl kann z. B. unter Beachtung von BSI-Vorgaben (module-policy) das TLS-Modul für die Kompilierung vorbereitet werden.

```
python ./configure.py --module-policy=bsi
--enable-
modules=tls,pkcs11,filters,codec_filt
```

Noch interessanter für die Validierung eines gewünschten Kryptoalgorithmus ist die Tatsache, dass mit Botan separate C++ Quellcodedateien zum jeweiligen Algorithmus erzeugt werden können. Diese Dateien lassen sich dann als ein separates C++ Projekt aufsetzten , welches bei einem anschließenden Debugging schrittweise durchlaufen werden kann. Hierbei können die einzelnen Zwischenergebnisse des kryptographischen Algorithmus ausgegeben werden. Die Erzeugung der einzelnen Dateien ist durch den folgenden Zusatz des obigen Befehls möglich:

```
--amalgamation --single-amalgamation-file
```

## 3 MPSoC-FPGA-Hardware

Bei heutigen MPSoC-FPGAs (Multi-processor System on Chip) befinden sich im selben Gehäuse neben der reinen programmierbaren FPGA-Logik (PL) noch weitere Chips mit mehreren Prozessoren (PS, processing system). Diese kombinierte Hardware (z. B. die Zynq UltraScale+ Bausteine von Xilinx) erlaubt es die Vorteile beider Technologien zu nutzen und stellt ein eigenständiges Computersystem dar (vgl. Abbildung 1).
Während in der FPGA-Hardware einzelne Algorithmen oder Funktionen hardwarenah und zeitoptimiert ablaufen, kann auf den Prozessoren parallel ein Linux-Betriebssystem ausgeführt werden, das mit der FPGA-Hardware kommuniziert.
Im Linux-System wird dann für den FPGA-Hardware-Test eine passende Testumgebung mit Botan aufgebaut.

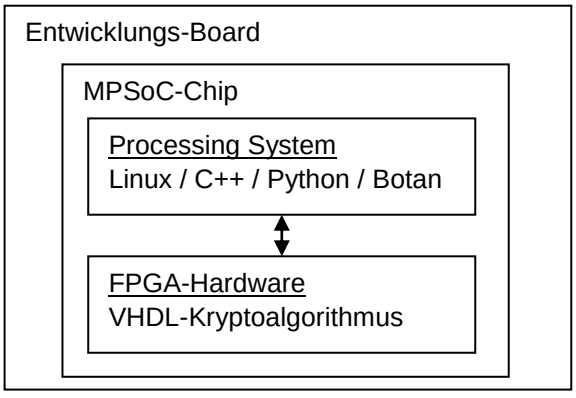


**Abbildung 1: Gesamtsystem**

# 4 Entwicklung und Evaluation eines Kryptoalgorithmus

Damit ein Kryptoalgorithmus in der FPGA-Logik ablaufen kann, muss dieser zunächst in der hardwarenahen Programmiersprache VHDL (Very High Speed Integrated Circuit Hardware Description Language) umgesetzt werden. Bereits in dieser frühen Entwicklungsphase kann Botan unterschiedliche Hilfestellungen bei der Entwicklung geben.

**Erzeugung fehlerfreier Testvektoren:**

- Für die Entwicklung kryptographischer Algorithmen ist es essentiell auf fehlerfreie Testvektoren (Eingangs- und Ausgangsdaten eines Algorithmus) zugreifen zu können. Nur so kann die korrekte Arbeitsweise eines Algorithmus final nachgewiesen werden.
- Botan liefert genau solche Testvektoren zu allen implementierten Algorithmen und Betriebsmodi, um die eigene korrekte Arbeitsweise sicherzustellen (vgl. Abbildung 2). Diese Testvektoren können direkt mit einer VHDL-Testumgebung (Testbench) eingelesen und für die Validierung desAlgorithmus verwendet werden.

```
[AES-128]
Key = 000102030405060708090A0B0C0D0E0F
In = 00112233445566778899AABBCCDDEEFF
Out = 69C4E0D86A7B0430D8CDB78070B4C55A
```

**Abbildung 2: Beispiel eines Testvektors aus Botan für den AES (Advanced Encryption Standard) mit 128 Bit Schlüssel.**

**Erzeugung von Zwischenergebnissen:**

- Während der Algorithmusentwicklung werden zusätzlich auch Zwischenergebnisse benötigt, um überprüfen zu können, dass ein Algorithmus intern fehlerfrei arbeitet. Auch diese Zwischenergebnisse können mit Botan, wie im Abschnitt „Modifizierbarkeit von Botan“ beschrieben, erzeugt werden.

**Validierung eines Algorithmus in Hardware:**

- Sobald eine vollständige VHDL-Beschreibung eines Algorithmus vorliegt, kann diese für den ausgewählten MPSoC-FPGA-Chip durch die Schritte Synthese und Implementation in eine Konfigurationsdatei, den sogenannten Bitstrom, umgewandelt werden.
- Mit diesem Bitstrom wird der FPGA-Chip anschließend konfiguriert, wodurch er die gewünschte Funktionalität erhält.
- Im Processing System kann jetzt im Linux-Betriebssystem eine Entwicklungsumgebung aufgesetzt werden, die zunächst mit Botan passende Eingangsvektor erzeugt und dann das Ergebnis zu einem ausgewählten Kryptoalgorithmus berechnet.
- Anschließend werden die erzeugten Eingangsdaten an den FPGA-Algorithmus übertragen.
- Sobald das Ergebnis der FPGA-Logik zur Verfügung steht, wird dieses final mit dem zuvor berechneten Ergebnis aus Botan überprüft.

**Validierung weitere Sicherheitsaspekte:**

- Sobald die fehlerfreie Arbeitsweise eines Kryptoalgorithmus sowohl in Software als auch in Hardware nachgewiesen werden konnte, erfolgen weitere Untersuchungen ob es über sogenannte Seitenkanäle möglich ist, an geheime Informationen zu gelangen.
  Sollte dies der Fall sein müssen weitere Gegenmaßnahmen in der Hardware implementiert werden [3].

# 5 Fazit

Die Komplexität aktueller kryptographischer Algorithmen erfordert umfangreiche Unterstützung und Beratung bzgl. Einsatz, Entwicklung und Evaluation welche von HTV bedarfsspezifische geliefert werden kann.
Auf Grundlage der Kryptobibliothek Botan können Testumgebungen in C, C++ und Python aufgebaut und Testvektoren erzeugt werden.
Sowohl auf der Software-Ebene (in C, C++, und Python) als auch auf der Hardware-Ebene (in VHDL) können kryptographische Algorithmen, z. B. für (MPSoC)-FPGAs, zur Verfügung gestellt werden.

# 6 Literatur


[1] Bundesamt für Sicherheit in der Informationstechnik: Sichere Implementierung einer allgemeinen Kryptobibliothek – Projektzusammenfassung, Version 1.0.0, Projekt 197, 27.03.2017

[2] Bundesamt für Sicherheit in der Informationstechnik: BSI-Projekt: Entwicklung einer sicheren Kryptobibliothek, https://www.bsi.bund.de/DE/Themen/Kryptografie_Kryptotechnologie/Kryptografie/Kryptobibliothek/kryptobibliothek_node.html, 05.02.2020

[3] Thomas Kuhn: FPGAs in Anwendungen mit hoher IT-Sicherheit. ELEKTRONIKPRAXIS Embedded System Development + IoT. September 2020