Decimated in Linear Time; Single Power Trace, Full Key Recovery Attack on Toeplitz Hash Privacy Amplification

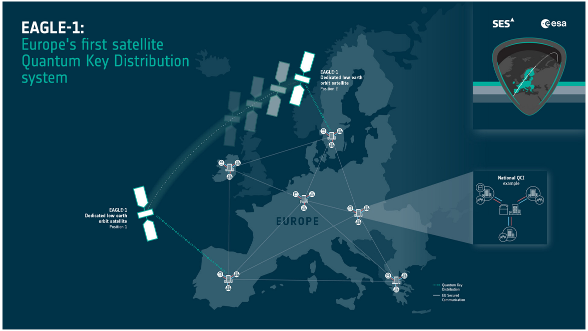
Niall Canavan, Tuan Hoang, Ayesha Khalid, Máire O'Neill
Queen's University Belfast, Centre for Secure Information Technologies (CSIT)

Security for Space Systems, 5 November 2025

Contents

A Basic Overview of QKD

- Quantum computers threaten modern classical cryptography [1].
- Q-day, could be right around the corner
- Thankfully, Quantum Key Distribution (QKD) offers a solution!
- Eagle-1 [2] and SAGA [3] projects both aim to further QKD development

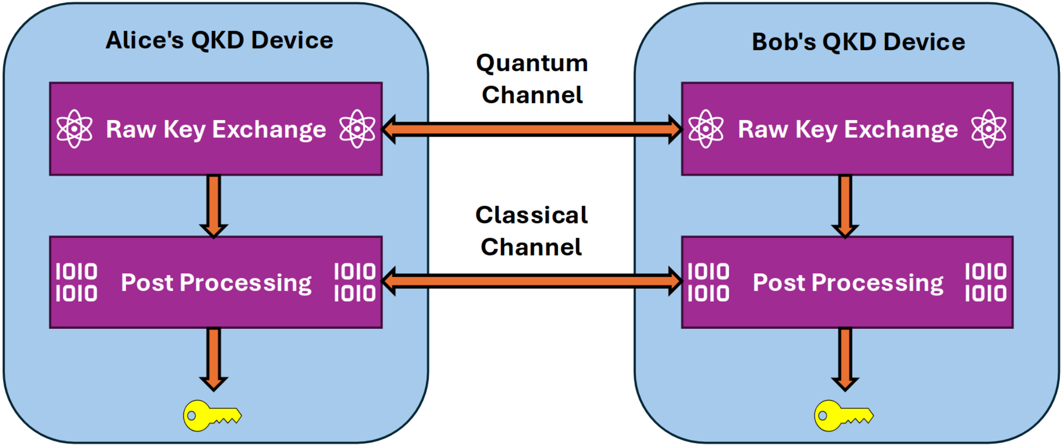


[1] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM Journal on Computing*, vol. 26, no. 5, Oct. 1997. DOI: 10.1137/S0097539795293172.

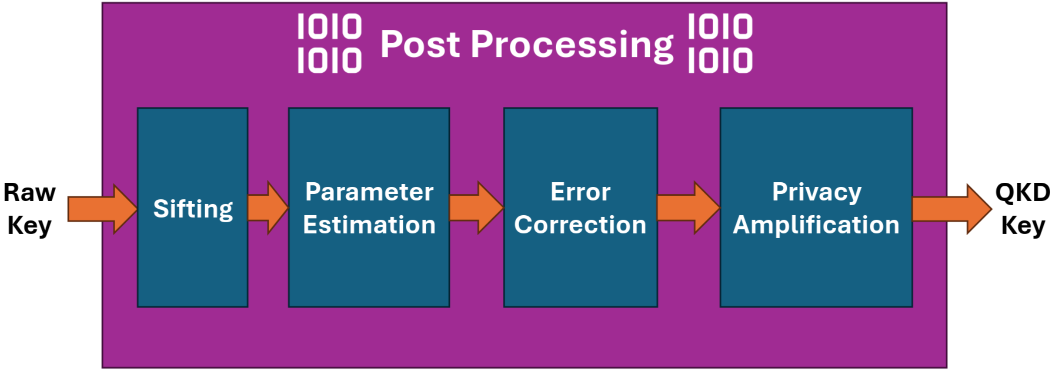
[2] ESA, "*Eagle-1 Mission*", [Online]. Available: https://www.esa.int/Applications/Connectivity_and_Secure_Communications/Eagle-1.

[3] ESA, "*SAGA Mission*", [Online]. Available: https://connectivity.esa.int/ultrasecure-communications-saga.

A Basic Overview of QKD



A Basic Overview of QKD

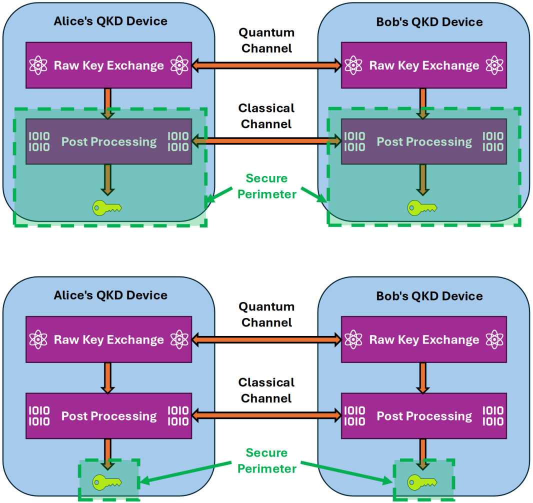


Contents

Challenging the Common Security Assumption

- QKD assumes the classical device is contained in a *secure perimeter*
- Research is active on quantum attacks [4]
- A more robust threat model considers classical attacks also
- PQC schemes consider a similar threat model

[4] V. Zapatero, Á. Navarrete, et al., "Implementation Security in Quantum Key Distribution," en, *Advanced Quantum Technologies*, Oct. 2023. DOI: 10.1002/qute.202300380.



Side Channel Analysis

- Power Side Channel Analysis (SCA) can reveal sensitive information [5]
- A SCA attack aims to recover the QKD key
- QKD classical protocols have received little consideration against such attacks

[5] S. Mangard and E. Oswald, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*, English. Springer, 2007.

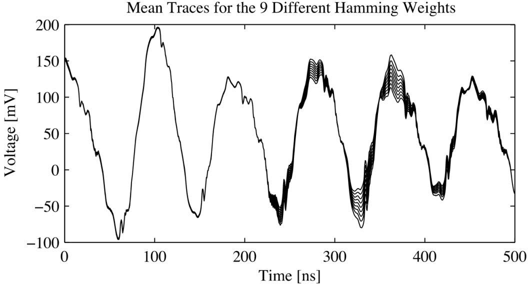
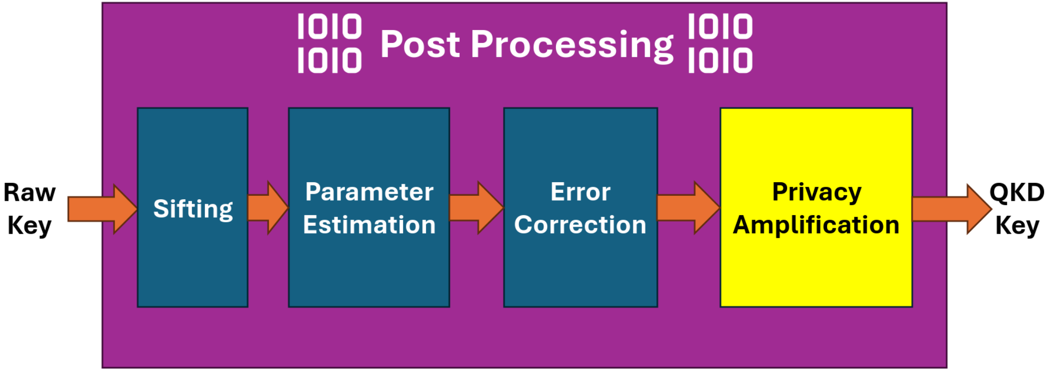


Figure from [5]

Contents

Privacy Amplification



Toeplitz Hashing

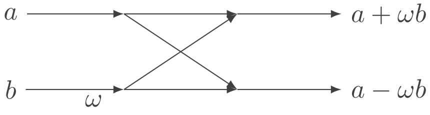
- Privacy Amplification (PA) removes any leaked information from the error corrected key, $\mathbf{K}_{EC}$.
- Toeplitz hashing based PA is the most popular scheme
- A Toeplitz matrix is shared between genuine QKD parties, defined by random binary seed, $\mathbf{S} \in \{0,1\}^n$.
- Naive matrix vector multiplication in $O(n^2)$ is not good enough...

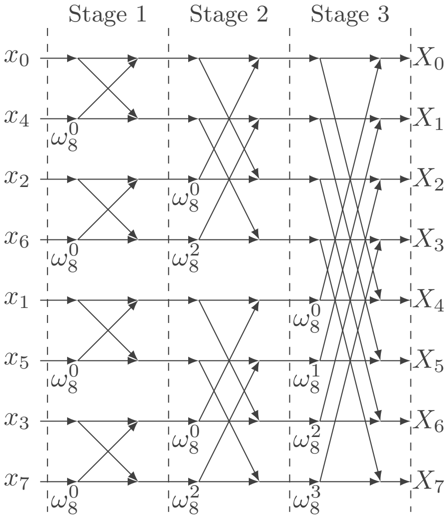
$$\mathbf{T}(\mathbf{S}) = \begin{pmatrix} s_r & s_{r+1} & s_{r+2} & \cdots & s_{r+n} \\ s_{r-1} & s_r & s_{r+1} & \cdots & s_{r+n-1} \\ s_{r-2} & s_{r-1} & s_r & \cdots & s_{r+n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s_3 & s_4 & s_5 & \cdots & s_{n+2} \\ s_2 & s_3 & s_4 & \cdots & s_{n+1} \\ s_1 & s_2 & s_3 & \cdots & s_n \end{pmatrix}$$

$$\mathbf{K}_{PA} = \mathbf{T}(\mathbf{S})\mathbf{K}_{EC}$$

DIT-FFT Optimisation

- High performance Toeplitz hashing schemes use the Decimation in Time Fast Fourier Transform (DIT-FFT), scaling in $O(nlogn)$
- Matrix and vector, $\boldsymbol{x}$, are projected into FFT domain, $\mathbf{X}$, for pointwise multiplication and back into original domain after.
- The butterfly operation is at the core of DIT-FFT





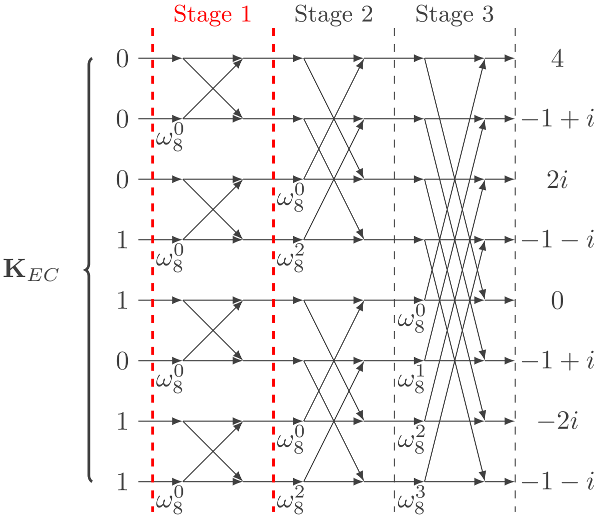
Contents

Attack Area

Algorithm 1: FFT Toeplitz Hash

Input: $\mathbf{K}_{EC}$, $\mathbf{S}$, n, r
Output: $\mathbf{K}_{PA}$
$\boldsymbol{v} \leftarrow FFT(\mathbf{S})$
$\boldsymbol{y} \leftarrow FFT(\mathbf{K}_{EC})$
$\boldsymbol{u} \leftarrow \boldsymbol{v} \circ \boldsymbol{y}$
$\mathbf{K}_{PA} \leftarrow IFFT(\boldsymbol{u}) \bmod 2$

- Considering $FFT(\mathbf{K}_{EC})$, the inputs, (a, b), to each butterfly are always binary at stage 1



Hypothetical Information Leakage

- Distinct Hamming Weight (HW) or Hamming Distance (HD) for each input may lead to leakage in the power trace
- We examine HW and HD for stage 1 butterfly operations
- Each of the four possible inputs have distinct HW/HD

IEEE 754 Representation

a	b	$B(a,b,\omega)$	HW In	HW Out	HD
0	0	(0,0)	0	0	0
0	1	(1,-1)	7	15	8
1	0	(1,1)	7	14	7
1	1	(2,0)	14	1	13

Q8.8 Fixed-Point Representation

a	b	$B(a,b,\omega)$	HW In	HW Out	HD
0	0	(0,0)	0	0	0
0	1	(1,-1)	2	10	8
1	0	(1,1)	2	4	5
1	1	(2,0)	4	1	3

Proposed Attack Strategy

- Only a single target power trace is available in a realistic PA setting
- Template attacks allow characterising each stage 1 butterfly operation
- Templates are generated for each input on an identical device controlled by an adversary
- The best template match to target trace is recorded as the value of $\mathbf{K}_{EC}$

Algorithm 2: DIT-FFT Template Attack

Input: $\mathbf{t}$, n, L

Output: $\mathbf{K}'_{EC}$

$\{\mathbf{t}_0, \mathbf{t}_1, \ldots, \mathbf{t}_{n/2}\} \leftarrow \mathbf{t}$

$\mathbf{K}'_{EC} \leftarrow \{0\}^n$

$G_1 \leftarrow (0,0), G_2 \leftarrow (0,1), G_3 \leftarrow (1,0), G_4 \leftarrow (1,1)$

for i *in* $n/2$ **do**

    for j *in* 4 **do**

        $\mathbf{K}'_{EC} \leftarrow setGuess(\mathbf{K}'_{EC}, G_j, i)$

        $\mathbf{T}_j \leftarrow genTemplate(\mathbf{K}'_{EC}, L, i)$

        $s_j \leftarrow compare(\mathbf{T}_j, \mathbf{t}^i)$

    $G_{best} \leftarrow best((G_1, s_1), (G_2, s_2), (G_3, s_3), (G_4, s_4))$

    $\mathbf{K}'_{EC} \leftarrow setGuess(\mathbf{K}'_{EC}, G_{best}, i)$

Contents

Experimental Setup

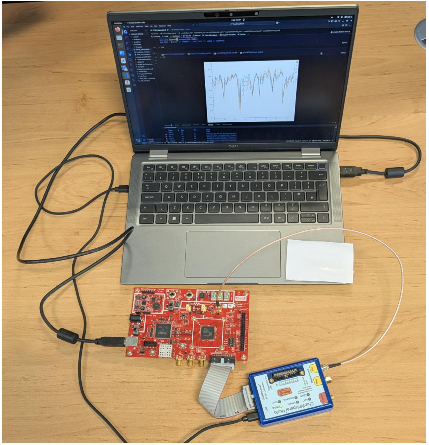
Hardware

- ChipWhisperer Husky oscilliscope [6]
- CW312 target board (Arm Cortex-M4 MCU)
- CW305 target board (Artix-7 FPGA)

Software

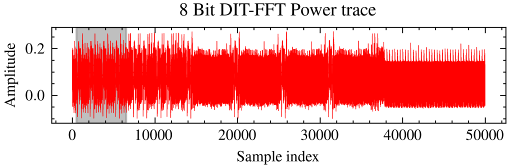
- Custom unprotected implementations

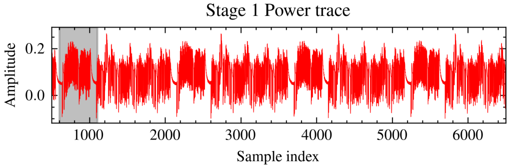
[6] C. O'Flynn and Z. Chen, "ChipWhisperer: An Open-Source Platform for Hardware Embedded Security Research," en, in E. Prouff, Ed., vol. 8622, Cham: Springer International Publishing, 2014. DOI: 10.1007/978-3-319-10175-0_17.

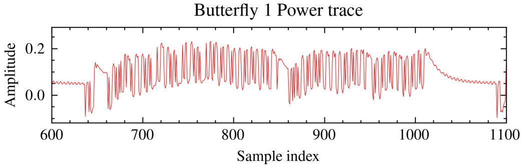


Capturing Traces

- Traces are recorded for each variation of input on target butterfly
- Each trace is aligned in time
- First layer is isolated and further separated into butterfly operations
- Average of each input on target butterfly is the template

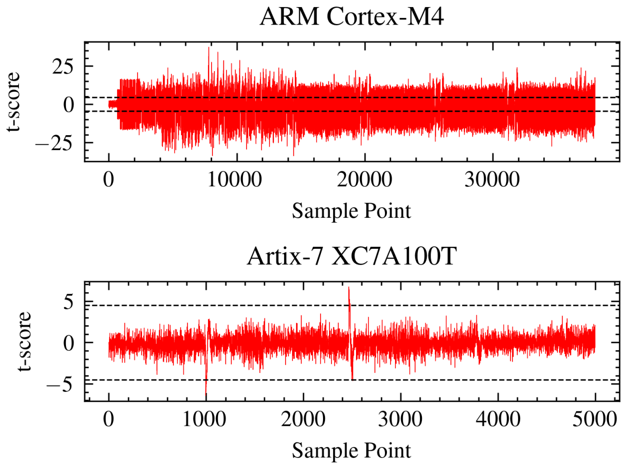






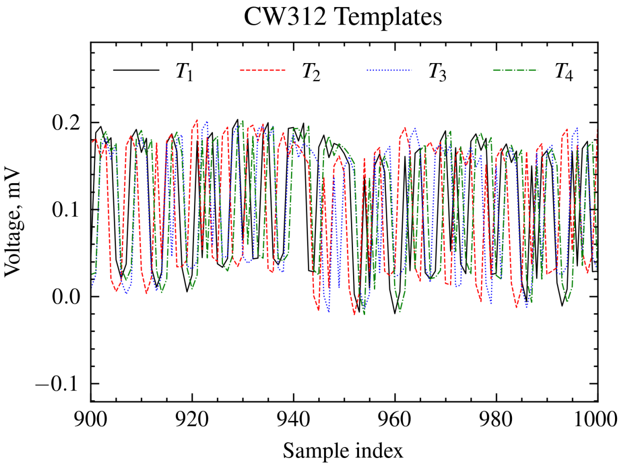
TVLA Test

- A TVLA test measures the statistical difference between a fixed input and random inputs
- Each sample point has a t-score
- Input data information leaks at a sample point if $|t| \geq 4.5$
- Both platforms leak, MCU more so



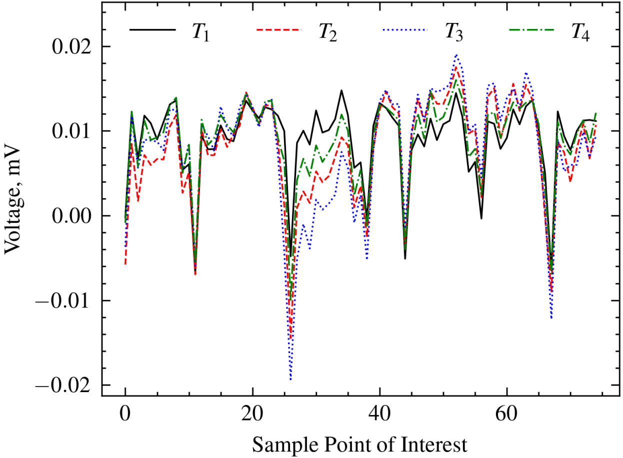
MCU Templates

- Traces for each template, T_i, are distinct
- Signal to noise ratio is low
- Target traces can be reliably matched to a template with a single trace



FPGA Templates

- Traces for each template, T_i, are distinct
- Signal to noise ratio is high
- Target traces cannot be reliably matched to a template with a single trace



Attack Performance

- Individual butterfly input recovery is reliably possible with single trace on MCU
- Full key attack time scales linearly with key size n
- $\sim 1s$ per butterfly, $\sim 4s$ for 8 bit key
- Attack cannot currently reliably recover key on FPGA under current experimental setup

```
Segment 0
Guess 0. MSE: 0.026930091417854338,
      PC: 0.4978433287673936
Guess 1. MSE: 0.03299193977045687,
      PC: 0.3856896788187195
Guess 2. MSE: 0.04104327327827073,
      PC: 0.2338513997655182
Guess 3. MSE: 0.00018392346275788736,
      PC: 0.9965736472717602


Segment 1
...

...

[1 0 1 1 1 1 1 0] : Eve's Guess
[1 0 1 1 1 1 1 0] : Correct Key
Successfull Key Recovery
Time Taken: 4.05157208442688s
```

Contents

Conclusions

- QKD is coming closer to real world use, practical attacks on devices should be considered
- Side channel leakage of QKD post-processing algorithms remain largely unknown
- DIT-FFT optimised Toeplitz hashing based PA is leaky on unprotected implementations
- Full QKD can be recovered on Cortex ARM-M4 using a single target trace
- Future work could refine the FPGA attack to recover enough useable information from a single trace

Thank You!

Scan the link for my paper and contact details!

... or email me, ncanavan815@qub.ac.uk

This PhD project receives funding from the Cyber AI Hub Doctoral Training Program at CSIT, Queen's University Belfast, supported by the UK Government as part of the New Deal for Northern Ireland, administered through Innovate UK/UKRI.