1. Abstract

This work investigates the efficiency of implementing the isogeny-based post-quantum key exchange protocol on ARM-powered embedded platforms. The proposed implementation is constant-time and resistant against the best known quantum attacks. We analyze the recent projective isogeny formulas presented by Microsoft research. Results are the fastest implementation of SIDH protocol on ARM-powered devices found in the literature.

2. Background

- Isogenies of Elliptic Curves: Suppose $E_1$ and $E_2$ are elliptic curves defined over a finite field $\mathbb{F}_q$. An isogeny $\phi: E_1 \rightarrow E_2$ is a non-constant rational map defined over $\mathbb{F}_q$ such that $\phi$ is a group homomorphism from $E_1(P_0)$ to $E_2(Q_0)$. If $E_1$ and $E_2$ are isogenous, their $j$-invariant values are the same.

- SIDH Key Exchange:

Figure 1: SIDH Key Exchange Protocol by Jao and De Feo [2011]

\[
\begin{align*}
\text{Alice} &: \text{Input: } X, \beta, \text{Public Data } A, \text{uID } \mu_A, X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7 \\
\phi_A(P_B) &\rightarrow \phi_A(Q_B) \\
\phi_A(R_B) &\rightarrow \phi(B) \\
\phi_B(R_A) &\rightarrow \phi(A) \\
\end{align*}
\]

Output: $\phi(B), \phi(A)$

3. Finite Field Arithmetic Implementation

The required finite field arithmetic are implemented using ARMv7 NEON capabilities to boost the performance:

- **Multiplication:** Comba-based multiplication using NEON
- **Reduction:** Comba-based Montgomery reduction using NEON
- **Multiplication over $F_p$:**

![Diagram of finite field multiplication using ARMv7 NEON](image)

4. Target Platforms

![ARMv7 (Ation TK1) and ARMv8 (Nexus 6P) Target Platforms in the work](image)

5. Results

<table>
<thead>
<tr>
<th>Work</th>
<th>Field</th>
<th>Device</th>
<th>vendor</th>
<th>Computing Time [ns]</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Costello et al. [2016]</td>
<td>ARMv7</td>
<td>657</td>
<td>Proj.</td>
<td>1.284</td>
<td>1.289</td>
</tr>
<tr>
<td>Azarderakhsh et al. [2014]</td>
<td>ARMv7</td>
<td>503</td>
<td>Proj.</td>
<td>1.289</td>
<td>1.302</td>
</tr>
<tr>
<td>This work</td>
<td>ARMv8</td>
<td>657</td>
<td>Proj.</td>
<td>1.394</td>
<td>1.397</td>
</tr>
</tbody>
</table>

6. Conclusions

- In this work, we proved that SIDH can be implemented efficiently on emerging ARM embedded devices and represent a new alternative to classical cryptosystems.
- We conclude that affine isogeny formulas are comparative in terms of performance with its projective counterpart on ARM devices.
- In terms of security, projective SIDH performs all the required computations in constant-time, while affine SIDH suffers from non-constant time field inversion computations.
- Different optimized libraries for different quantum security levels are proposed.
- SIDH smaller key size compared to other PQ cryptography makes this scheme suitable for the applications where the communication bandwidth is a concern.

References


Efficient Implementation of Supersingular Isogeny Diffie-Hellman Key Exchange on ARM Processors

Amir Jalali, Reza Azaderakhsh, Mehran Mozaffari-Kermani

Abstract

We investigate the efficiency of implementing the Jao and De Feo isogeny-based post-quantum key exchange protocol (from PQCrypto2011) on ARM-powered embedded platforms. In this work, we propose new primes to speed up constant-time finite field arithmetic and perform isogenies quickly. Montgomery multiplication and reduction are employed to produce a speedup of 3 over the GNU Multiprecision Library. We analyze the recent projective isogeny formulas presented in Costello et al. (Crypto 2016) and conclude that affine isogeny formulas, in terms of performance, are comparable with projective SIDH on ARM devices. However, projective SIDH implementation can be implemented in constant-time for all the key exchange and key generation computations; therefore provides resistance against timing and cache attacks. We provide fast affine and projective SIDH libraries over different post-quantum security levels, taking advantage of Single Instruction Multiple Data (SIMD) capabilities of ARMv7 processors along with vectorization. Our assembly-optimized field arithmetic cuts the computation time for the protocol by 50% in comparison to our portable C implementation and performs approximately 3 times faster than the only other ARM-powered results found in the literature. The goal of this work is to show that isogeny-based cryptosystems can be implemented further and be used as an alternative to classical cryptosystems on embedded devices.