Security Analysis and Efficient Implementation of Code-based Signature Schemes

Partha Sarathi Roy1, Kirill Morozov2, Kazuhide Fukushima1, Shinsaku Kiyomoto1, and Tsuyoshi Takagi3

1Information Security Laboratory, KDDI Research, Inc., 2-1-15 Ohara, Fujimino-shi, Saitama 356-8502, Japan
2Department of Computer Science and Engineering, University of North Texas, Denton, TX 76207, U.S.A.
3Department of Mathematical Informatics, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

Keywords: Post-quantum Cryptography, Code-based Cryptography, Signature Scheme, Fiat-Shamir Transformation, Identification Scheme.

Abstract: In this paper, we derive code-based signature schemes using Fiat-Shamir transformation on code-based zero-knowledge identification schemes, namely the Stern scheme, the Jain-Krenn-Pietrzak-Tentes scheme, and the Cayrel-Veron-El Yousfi scheme. We analyze the security of these code-based signature schemes and derive the security parameters to achieve the 128-bit level security. Furthermore, we implement these signature schemes and compare their performance on a PC.

1 INTRODUCTION

Digital signatures are essential components of IT-security solutions. Security of the schemes, already used in practice, relies on the number-theoretic hardness assumptions. Unfortunately, these problems are known to be solvable in polynomial time on quantum computers using Shor’s algorithm (Shor, 1994). Hence, quantum computers would be able to break popular cryptosystems such as RSA or ElGamal (including its elliptic-curve variant) in polynomial time. Given these circumstances, it is important to consider the transition to post-quantum digital signature schemes. In this work, we focus on the practical use of code-based signature scheme as one of the post-quantum candidates.

CFS (Courtois et al., 2001), KKS (Kabatianskii et al., 1997), and their variants are the most celebrated code-based signature schemes. Unfortunately, there are various drawbacks in respect of security, computation time, key or signature size. The CFS scheme was proven existentially unforgeable under chosen message attack (EUF-CMA) by Dallot (Dallot, 2008) under the hardness of the Goppa-Parametrized Bounded Decoding problem and the Goppa-Code Distinguishing (GD) problem. Even with the existence of a distinguisher (Faugère et al., 2011) for the Goppa codes of high rate, a simple modification can provide Strong EUF-CMA security for the CFS signature (Morozov et al., 2018). However, from the practical perspective, signing time of CFS signature is somewhat high due to the difficulty of finding decodable syndromes. Debris-Alazard et al. (Debris-Alazard et al., 2017) exposed a problem with the recently introduced SURF signature scheme. The main drawback of the KKS-like signature schemes is their security as shown by Otmani et al. (Otmani and Tillich, 2011). Therefore, we will explore the time-tested approach to signature schemes via the Fiat-Shamir transform over zero-knowledge identification schemes.

Our Contribution. In this paper, we study code-based signature schemes via Fiat-Shamir transformation on zero-knowledge (ZK) identification schemes. The Fiat-Shamir transform seems to be the promising avenue to have both efficient and secure code-based signature schemes. We chose to study and compare zero-knowledge identification schemes by Stern (Stern, 1994), Jain et al. (Jain et al., 2012) and Cayrel et al. (Cayrel et al., 2010). Note that Jain et al. (Jain et al., 2012) pointed out a flaw in the proof of zero-knowledge property of Veron’s code-based identification scheme (Véron, 1997), and so Jain et al. provided an alternative scheme which is indeed ZK. As security assumption, Jain et al. used the so-called Exact-LPN (xLPN) problem (Jain et al.,...
2012), which is, in fact, identical to the general decoding problem considered both in Veron’s paper, and in this work.¹ There are some existing studies on the efficient implementation of Stern’s scheme by Gaborit et al. (Gaborit and Girault, 2007), and Cayrel et al. (Cayrel et al., 2008). El Yousfi et al. (Alaoui et al., 2013) have studied the implementations of Stern (Stern, 1994), Veron (Véron, 1997) and Cayrel et al. (Cayrel et al., 2010). However, these works show system parameters for identification schemes that achieve 80-bit security. Due to the growing power of modern computing systems, it seems preferable to investigate the performance of the signature schemes for 128-bit security benchmark. We implement in C the digital signatures based on Stern’s (Stern, 1994), Jain et al.’s (Jain et al., 2012) and Cayrel et al.’s (Cayrel et al., 2010) identification schemes, and compare their performance on a PC.

2 PRELIMINARIES

2.1 Notations

We will use the following notations throughout the article.

- \(\{0,1\}^*\): bit string of arbitrary length.
- \(\mathbb{F}_q\): Galois field of \(q\) elements.
- \(\mathbb{F}_q^n\): Vector with \(n\) elements over \(\mathbb{F}_q\).
- \(\mathbb{F}_{q}^{m \times n}\): Matrix with \(m\) rows and \(n\) columns whose elements are in \(\mathbb{F}_q\).
- \(\text{wt}(c)\): Hamming weight of string \(c\), i.e., the number of non-zero positions of the string.
- \(\|\|\): Concatenation of strings. We see a string as a column vector.

2.2 Zero-knowledge Identification Scheme

An identification scheme consists of three probabilistic, polynomial-time algorithms \((G,P,V)\) such that:

- The randomized key generation algorithm \(G\) takes as input the security parameter \(1^k\). It outputs a pair of keys \((pk,sk)\), where \(pk\) is called the public key and \(sk\) is called the private key. We assume the security parameter is implicit in both \(pk\) and \(sk\).
- \(P\) and \(V\) are interactive protocols. The prover algorithm \(P\) takes as input a private key \(sk\) and the verification algorithm \(V\) takes as input a public key \(pk\). At the conclusion of the protocol, \(V\) outputs 1 or \(\bot\).

It holds the following properties:

Completeness: \(\Pr[[P(pk),V(pk)]=1]=1\).

Soundness: \(\Pr[[P^*(V(pk)]=1)]\approx \text{negl}(\lambda)\).

Zero-knowledge: \(\Pr[[P(sk),V^*(pk)]=1]\approx \text{negl}(\lambda)\).

2.3 Digital Signature

A digital signature scheme \(\Sigma = (\text{Key Generation, Signature Generation, Signature Verification})\) is consists of three algorithms.

- Key Generation: The key generation algorithm takes a security parameter \(1^k\) and outputs a pair of keys \((pk,sk)\).
- Signature Generation: The signature generation algorithm takes a message \(m\) and a private key \(sk\) as inputs and outputs a signature \(\sigma\) on the message \(m\).
- Signature Verification: The signature verification algorithm takes as input a public key \(pk\), a message \(m\) and a signature \(\sigma\), and outputs a bit denoting accept or reject, respectively.

The standard security notion for a signature scheme is existential unforgeability against chosen message attack (EUF-CMA): The forger gets a public key from a challenger who generates a key pair \((sk, pk)\). The forger can query a signing oracle on polynomially many messages \(m_i\) hereby obtaining signatures \(\sigma_i\). The forger can also issue a hash query and obtains its hash value. We say that the forger wins the EUF-CMA game, if the forger successfully outputs a pair \((m^*, \sigma^*)\), where \(\sigma^*\) is a valid signature of a message \(m^*\) under the private key with the restriction that \(m^*\) has never appeared in the query phase.

2.4 Security Assumptions

An \([n,k]\) Linear Code \(C\) is a subspace of dimension \(k\) of the vector space \(\mathbb{F}_q^n\). A linear code can be described by its Parity-check matrix \(H\). The parity-check matrix describes the code as follows:

\[x \in C \iff Hx = 0 \quad (\forall x \in \mathbb{F}_q^n)\.

¹Jain et al. presented the ZK identification scheme based on the standard LPN problem as well, but it has higher soundness error as compared to the XL-LPN-based one. Hence it would result in signatures of larger size, and hence it is out of the scope of this work.
The product \( Hx \) is known as the syndrome of the vector \( x \).

**Definition 1. Gilbert-Varshamov Bound:**

Let \( C \) be an \([n,k]\) linear code over \( \mathbb{F}_q \). The Gilbert-Varshamov (GV) Distance is the largest integer \( \omega \) such that

\[
\frac{k}{n} = 1 - H_q\left(\frac{\omega}{n}\right),
\]

where \( H_q \) is the q-ary entropy function.

Now, we describe the main hard problems on which the security of code-based signature schemes, presented in the paper, relies.

**Definition 2. Syndrome Decoding Problem (SDP) (Augot et al., 2003):**

- **Input:** a matrix \( H \in \mathbb{F}_q^{(n-k)\times n} \), a positive integer \( \omega \), and a vector \( s \in \mathbb{F}_q^{n-k} \).
- **Output:** a codeword \( x \) such that \( \text{wt}(x) = \omega' \) where \( 0 < \omega' \leq \omega \) and \( Hx = s \).

This problem is NP-complete (Berlekamp et al., 1978), which means that at least some instances of the problem are difficult. However, it is a common belief that they should be difficult on average (for well-chosen parameter ranges), which means that random instances are difficult. It is also proved that there exists a unique solution to SDP if the weight \( \omega \) is below the GV Bound.

**Definition 3. General Decoding Problem (GDP):**

- **Input:** a matrix \( G \in \mathbb{F}_q^{(n-k)\times n} \), a non negative integer \( \omega \), and a vector \( y \in \mathbb{F}_q^n \).
- **Output:** a pair \((m,e) \in \mathbb{F}_q^{n} \times \mathbb{F}_q^n \) such that \( \text{wt}(e) = \omega' \) where \( 0 < \omega' \leq \omega \) and \( mG \oplus e = y \).

An extension of SDP over arbitrary finite field is as follows:

**Definition 4. q-ary Syndrome Decoding Problem (q-SDP) (Augot et al., 2003):**

- **Input:** a matrix \( H \in \mathbb{F}_q^{(n-k)\times n} \), a positive integer \( \omega \), and a vector \( s \in \mathbb{F}_q^{n-k} \).
- **Output:** a codeword \( x \) such that \( \text{wt}(x) = \omega' \) where \( 0 < \omega' \leq \omega \) and \( \text{wt}(Hx) = s \).

q-SDP is proven to be NP-hard by S.Barg (Barg, 1994).

**Definition 5. Exact Learning Parity with Noise (sLPN) (Jain et al., 2012):**

The decisional-sLPN problem is \((n,t,\varepsilon)\)-hard if for every distinguisher \( D \) running in time \( t \)

\[
\Pr_{s,A,e \leftarrow \mathbb{F}_q^n} \left[ D(A,As \oplus e) = 1 \right] - \Pr_{A,r \leftarrow \mathbb{F}_q^n} \left[ D(A,r) = 1 \right] \leq \varepsilon
\]

where \( s \leftarrow \mathbb{F}_q^n \), \( e \leftarrow \{0,1\}^n \) such that \( \text{wt}(e) = \omega \), \( r \leftarrow \mathbb{F}_q^n \), and \( A \leftarrow \mathbb{F}_q^{n}\times n \).

## 3 SIGNATURE SCHEME

In this section, we will derive the signature scheme from code-based zero-knowledge identification schemes. Using Fiat-Shamir transformation (Pointcheval and Stern, 2000), we can derive signature schemes from the Stern (Stern, 1994) and Jain et al. (Jain et al., 2012) identification schemes. We need to use the extended version of Fiat-Shamir transformation (Alaoui et al., 2012) to derive a signature scheme from Cayrel et al.’s scheme (Cayrel et al., 2010).

### 3.1 Stern Signature Scheme.

**System Parameters.** The Stern signature scheme uses the following system parameters:

- Positive integer \( n \) (length of codeword),
- Positive integer \( k \) such that \( k < n \) (dimension of the code),
- Positive integer \( \omega \) (minimum distance of the code),
- Matrix \( H \in \mathbb{F}_q^{(n-k)\times n} \) sampled randomly,
- random oracle: \( h : \{0,1\}^* \rightarrow \{0,1\}^* \)
- random oracle: \( O : \{0,1\}^* \rightarrow \{0,1,2\} \)

**Key Generation.** The key generation algorithm outputs the pair of the private key \( s \) and public key \( y \).

1. Sample a vector \( s \in \mathbb{F}_q^n \) such that \( \text{wt}(s) = \omega \).
2. Calculate a vector \( y \in \mathbb{F}_q^n \) such as \( y = Hs \).
3. Output private key \( s \) and public key \( y \).

**Signature Generation.** Takes as input private key \( s \) and a message \( m \), and output \( \text{Sig} \). The detailed algorithm is described in Algorithm 3.

**Signature Verification.** Takes as input public key \( y \), message \( m \), and \( \text{Sig} \). Compute \( b_i \leftarrow O(m || c_i) \) and output 1 if the following respective equation is valid for all \( 1 \leq i \leq \delta \):

\[
\begin{align*}
\text{Check } c_{i,0} &= h(\sigma_i || H_{ui}) \text{ and } c_{i,1} = \sigma_i(u_i). \quad (b_0 = 0) \\
\text{Check } c_{i,0} &= h(\sigma_i || H_{(ui \oplus s)} \oplus y) \text{ and } c_{i,2} = h(\sigma_i(u_i \oplus s)). \quad (b_1 = 1) \\
\text{Check } c_{i,1} &= \sigma_i(u_i), \text{ } c_{i,2} = h(\sigma_i(u_i) \oplus \sigma_i(s)). \quad (b_2 = 2)
\end{align*}
\]

or \( \perp \) otherwise.
Algorithm 1: Signature Generation in Stern signature scheme.

**Input:** Private key \( s \), Message \( m \), and System parameters

**Output:** Signature \( Sig \)

for \( i \leftarrow 0 \) to \( \delta - 1 \) do

1. \( u_i \leftarrow s^{F_2^n} \);
2. \( \sigma_i \leftarrow S_n \);
3. \( c_{i,0} \leftarrow h(\sigma_i || H u_i) \);
4. \( c_{i,1} \leftarrow \sigma_i(u_i) \);
5. \( c_{i,2} \leftarrow h(\sigma_i(u_i \oplus s)) \);
6. \( c_i = (c_{i,0} || c_{i,1} || c_{i,2}) \);
7. \( b_i = O(m || c_i) \);
8. \( \text{rsp}_i \leftarrow \begin{cases} \sigma_i || u_i & (b_i = 0) \\ \sigma_i(u_i \oplus s) & (b_i = 1) \\ \sigma_i(u_i) || \sigma_i(s) & (b_i = 2) \end{cases} \)
9. \( \text{sig}_i = c_i || \text{rsp}_i \);
end

\( \text{Sig} \leftarrow \text{sig}_0 || \text{sig}_1 || \ldots || \text{sig}_{\delta-1} \);
return \( \text{Sig} \);

Algorithm 2: Signature Generation in Jain et al. signature scheme.

**Input:** Private key \( e \), Message \( m \), and System parameters

**Output:** Signature \( Sig \)

for \( i \leftarrow 0 \) to \( \delta - 1 \) do

1. \( u_i \leftarrow s^{F_2^n} \);
2. \( v_i \leftarrow s^{F_2^n} \);
3. \( \sigma_i \leftarrow S_n \);
4. \( y_{i,0} \leftarrow v_i A \oplus u_i \);
5. \( c_{i,0} \leftarrow h(\sigma_i || y_{i,0}) \);
6. \( y_{i,1} \leftarrow \sigma_i(u_i) \);
7. \( c_{i,1} \leftarrow h(y_{i,1}) \);
8. \( y_{i,2} \leftarrow \sigma_i(u_i \oplus e) \);
9. \( c_{i,2} \leftarrow h(y_{i,2}) \);
10. \( c_i = (c_{i,0} || c_{i,1} || c_{i,2}) \);
11. \( b_i = O(m || c_i) \);
12. \( \text{rsp}_i \leftarrow \begin{cases} \sigma_i || y_{i,0} || y_{i,1} & (b_i = 0) \\ \sigma_i || y_{i,0} || y_{i,2} & (b_i = 1) \\ y_{i,1} || y_{i,2} & (b_i = 2) \end{cases} \)
13. \( \text{sig}_i = c_i || \text{rsp}_i \);
end

\( \text{Sig} \leftarrow \text{sig}_0 || \text{sig}_1 || \ldots || \text{sig}_{\delta-1} \);
return \( \text{Sig} \);

\[ \text{Check } c_{i,0} = h(\sigma_i || y_{i,0}) \]
\[ c_{i,1} \leftarrow h(y_{i,1}) \]
\[ c_{i,2} \leftarrow h(y_{i,2}) \]
\[ b_i = O(m || c_i) \]
\[ \text{rsp}_i \leftarrow \begin{cases} \sigma_i || y_{i,0} || y_{i,1} & (b_i = 0) \\ \sigma_i || y_{i,0} || y_{i,2} & (b_i = 1) \\ y_{i,1} || y_{i,2} & (b_i = 2) \end{cases} \]
\[ \text{sig}_i = c_i || \text{rsp}_i \]
end

or \( \bot \) otherwise.

3.2 Jain et al. Signature Scheme

In this section, we will derive the signature scheme from the identification scheme of (Jain et al., 2012). We will modify the design a little, without any security breach, for the sake of fast implementation.

System Parameters. The signature scheme uses the system parameters as in Stern signature scheme apart from random sampling of \( A \in F_2^{k \times n} \) instead of \( H \in F_2^{(n-k)\times n} \).

Key Generation. The key generation algorithm outputs the pair of the private key \( e \) and public key \( y \).

1. Sample \( (s, e) \leftarrow s^{F_2^n} \times F_2^n \) such that \( \text{wt}(e) = \omega \).
2. Calculate \( y = sA \oplus e \).
3. Output private key \( e \) and public key \( y \).

Signature Generation. The signature generation algorithm takes private key \( e \) and a message \( m \), output \( Sig \), and system parameters as inputs. Algorithm 2 describes the detailed algorithm.

Signature Verification. The signature verification algorithm takes as input public key \( y \), message \( m \), \( Sig \), and system parameters. It computes \( b_i \leftarrow O(m || c_i) \) and outputs \( 1 \) if the following respective equation is valid for all \( 0 \leq i \leq \delta - 1 \):

3.3 Cayrel et al. Signature Scheme

To present the Cayrel et al., signature scheme, we first introduce a special transformation that will be used in the scheme.

Definition 6. Let \( \sigma \in S_n \) and \( \gamma = (\gamma_1, \ldots, \gamma_n) \in (F_q^n)^n \) such that \( \gamma_i \neq 0 \) for all \( i \). The transformation \( \Pi_{\gamma, \sigma} \) is defined as follows:

\[ \Pi_{\gamma, \sigma} : F_q^n \rightarrow F_q^n \]
\[ v \mapsto \left( \gamma_{0,0} \sigma_{0,0}, \gamma_{0,1} \sigma_{0,1}, \ldots, \gamma_{n-1,0} \sigma_{n-1,0}, \gamma_{n-1,1} \sigma_{n-1,1}, \ldots, \gamma_{n-1,n-1} \sigma_{n-1,n-1} \right) \]

Notice that this transformation is linear transformation, and satisfies \( \Pi_{\gamma, \sigma}(v + w) = \Pi_{\gamma, \sigma}(v) + \Pi_{\gamma, \sigma}(w) \)
and \( \Pi_{\gamma}(\alpha v) = \alpha \Pi_{\gamma}(v) \) for all \( v, w, \alpha \in \mathbb{F}_q \). Furthermore, the transformation preserves the Hamming weight of the vector.

Now, we are in the state to present the signature scheme:

**System Parameters.** The signature scheme uses the following system parameters:
- Positive integer \( n \) (length of codeword),
- Positive integer \( k \) such that \( k < n \) (dimension of the code),
- Positive integer \( \omega \) (minimum distance of the code),
- Matrix \( H \in \mathbb{F}_q^{(n-k) \times n} \) sampled randomly,
- Random oracle \( h : \{0,1\}^* \rightarrow \{0,1\}^* \),
- Random oracle \( O_1 : \{0,1\}^* \rightarrow \mathbb{F}_q^n \),
- Random oracle \( O_2 : \{0,1\}^* \rightarrow \{0,1\} \).

**Key Generation.** The key generation algorithm outputs the pair of the private key \( s \) and public key \( y \).

1. Sample a vector \( s \in \mathbb{F}_q^n \) such that \( \omega \cdot s = \omega \).
2. Calculate a vector \( y \in \mathbb{F}_q^{n-k} \) as \( y = Hs \).
3. Output private key \( s \) and public key \( y \).

**Signature Generation.** The signature generation algorithm takes as input private key \( s \) and a message \( m \), and outputs a signature \( \text{Sig} \). The detailed algorithm is described in Algorithm 3.

**Signature Verification.** The signature verification algorithm takes public key \( y \), message \( m \), and signature \( \text{Sig} \) as inputs. It computes \( \alpha_i \leftarrow O_1(m||c_i) \) and \( b_i \leftarrow O_2(m||c_i)(\alpha_i||\beta_i) \); then, it outputs 1 if the following respective equation is valid for all \( 0 \leq i \leq \delta - 1 \):

\[
\begin{align*}
\text{Check } c_{i,0} &= h(\alpha_i||\gamma_i||\Pi_{\gamma,\sigma_i}^{-1}(\beta_i) - \alpha_i \gamma) \quad (b_i = 0) \\
\text{Check } c_{i,1} &= h(\beta_i - \alpha_i \Pi_{\gamma,\sigma_i}(u_i)||\Pi_{\gamma,\sigma_i}(s)) \quad (b_i = 1),
\end{align*}
\]

and \( \omega = \text{wt}(\Pi_{\gamma}(s)) \) otherwise.

### 4 SECURITY ANALYSIS AND PARAMETER SELECTION

**Best Known Attack:** It is required to consider structural attack and key-recovery attack to measure the security of code-based signature schemes. Due to the use of random code only, consideration of structural attack abolish. Signature schemes, constructed by using Fiat-Shamir transformation and it is extended version on zero knowledge identification schemes, are EUF-CMA secure (Pointcheval and Stern, 2000; Aloui et al., 2012). Moreover, EUF-CMA security includes the security against key-recovery attack. Therefore, to choose secure parameters, it is required to measure the hardness of the problem, where the proof of EUF-CMA security is reduced.

The security of the Stern and Cayrel et al. signatures are reduced to the hardness of SDP and qSDP, respectively. The most efficient known algorithm to attack SDP is the Information Set Decoding (ISD) algorithm by Stern (Stern, 1988). Further, there are few many improvements over (Stern, 1988). One of the intermediate notable improvement is by Finiasz et al. (Finiasz and Sendrier, 2009). Further improvements over (Finiasz and Sendrier, 2009) are asymptotic (May et al., 2011; Becker et al., 2012). So, we have used the measurement of (Finiasz and Sendrier, 2009) to measure the hardness of SDP. The hardness of q-ary SDP is measured by the formulation of (Niebuhr et al., 2017), which is the extension of (Finiasz and Sendrier, 2009) from binary to an arbitrary finite field.

Security of Jain et al. signature scheme is reduced to the hardness of XLPN. However, in actuality, the number of sample of the instance is small, and we cannot apply the BKW algorithm to solve the XLPN problem. So, the hardness is turned down to the hard-
Table 1: System parameters and data sizes for Stern signature scheme.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>80-bit Security</th>
<th>128-bit Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>620</td>
<td>1,024</td>
</tr>
<tr>
<td>δ</td>
<td>137</td>
<td>219</td>
</tr>
<tr>
<td>k</td>
<td>310</td>
<td>512</td>
</tr>
<tr>
<td>ω</td>
<td>68</td>
<td>112</td>
</tr>
</tbody>
</table>

Independent parameters:

Derived parameters:

Data size:

Table 2: System parameters and data sizes of Jain et al. signature scheme.

<table>
<thead>
<tr>
<th>Name</th>
<th>80-bit Security</th>
<th>128-bit Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>620</td>
<td>1,024</td>
</tr>
<tr>
<td>δ</td>
<td>137</td>
<td>219</td>
</tr>
<tr>
<td>k</td>
<td>310</td>
<td>512</td>
</tr>
<tr>
<td>ω</td>
<td>68</td>
<td>112</td>
</tr>
</tbody>
</table>

Independent parameters:

Derived parameters:

Data size:

4.3 Cayrel et al. Signature Scheme

We select parameters $n$, $k$, and $\omega$ so that they lie on the GV bound i.e.,

$$k/n = 1 - H_2\left(\frac{\omega}{n}\right),$$

(1)

to maximize the security against attacks using the ISD algorithm. We select $k = n/2$ and $\omega$ is around 0.110$n$. The complexity of the ISD (Finiasz and Sendrier, 2009) is

$$WF_{ISD}(n,k,\omega) = \min_{p} \frac{2l \min (\binom{n}{m}, 2^r)}{(1 - e^{-1})^{(r-l)}(l/p)^{1/2}}$$

(2)

where

$$l = \log_2 \left( \frac{2\omega}{k}\sqrt{\frac{k}{p}} \right)$$

for the parameters on the GV bound, and $n$ should satisfy $WF_{ISD}(n,k,\omega) > 2^k$.

The number of rounds $\delta$ depends on a soundness error of the underlying identification scheme. The soundness error of the Stern identification scheme (Stern, 1994) is 2/3. So, $\delta$ should satisfy $(2/3)^\delta < 2^{-\lambda}$. System parameters and data sizes are presented in Table 1.

4.2 Jain et al. Signature Scheme

We select parameters $n$, $k$, and $\omega$ according to the equation 1 & 2. The number of rounds $\delta$ depends on the Soundness error of the underlying identification scheme. Soundness error of the Jain et al. identification scheme (Jain et al., 2012) is 2/3. So, $\delta$ should satisfy $(2/3)^\delta < 2^{-\lambda}$. System parameters and data sizes are presented in Table 2.

5 IMPLEMENTATIONS

We implement the Stern, Jain et al. and Cayrel et al. signature schemes with 128-bit level security in C language. Execution time and data sizes are presented in
Table 3: System parameters and data sizes for Cayrel et al. signature scheme over $\mathbb{F}_{256}$.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>80-bit Security</th>
<th>128-bit Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>144</td>
<td>230</td>
</tr>
<tr>
<td>$\delta$</td>
<td>80</td>
<td>128</td>
</tr>
<tr>
<td>Derived parameters:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k$</td>
<td>72</td>
<td>115</td>
</tr>
<tr>
<td>$\omega$</td>
<td>54</td>
<td>87</td>
</tr>
<tr>
<td>Data size:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$sk$</td>
<td>1.132 bits</td>
<td>1.840 bits</td>
</tr>
<tr>
<td>$pk$</td>
<td>576 bits</td>
<td>920 bits</td>
</tr>
<tr>
<td>Signature</td>
<td>89.6 kB</td>
<td>229 kB</td>
</tr>
<tr>
<td>System param.</td>
<td>10.4 kB</td>
<td>26.5 kB</td>
</tr>
</tbody>
</table>

Table 4: Execution time and data sizes of the three signature schemes for 128-bit security level.

<table>
<thead>
<tr>
<th></th>
<th>Stern</th>
<th>Jain et al.</th>
<th>Cayrel et al.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key-gen</td>
<td>0.017 ms</td>
<td>0.0201 ms</td>
<td>0.339 ms</td>
</tr>
<tr>
<td>Sign</td>
<td>31.5 ms</td>
<td>16.5 ms</td>
<td>24.3 ms</td>
</tr>
<tr>
<td>Verify</td>
<td>2.27 ms</td>
<td>135 ms</td>
<td>9.81 ms</td>
</tr>
<tr>
<td>$sk$</td>
<td>1024 bit</td>
<td>1356 bit</td>
<td>1840 bit</td>
</tr>
<tr>
<td>$pk$</td>
<td>512 bit</td>
<td>512 bit</td>
<td>920 bit</td>
</tr>
<tr>
<td>Signature</td>
<td>245 kB</td>
<td>263 kB</td>
<td>229 kB</td>
</tr>
</tbody>
</table>

* System parameters

Table 4. Eight variables of $\mathbb{F}_2$ in the Stern and Jain et al. signature schemes, and a variable of $\mathbb{F}_{256}$ in the Cayrel signature scheme are stored in an eight-bit int8 variable. Our implementation uses a pre-computation table for multiplication between $\mathbb{F}_{256}$ elements. We used SHA3-256 to implement random oracles used in the signature schemes. For example, a random oracle $h : \{0, 1\}^* \rightarrow \{0, 1\}^{1024}$ is implemented as $v \mapsto SHA3-256(0x00||v)||SHA3-256(0x01||v)||SHA3-256(0x02||v)||SHA3-256(0x03||v)$. One byte prefix $0x00, 0x01, ... 0x03$ are auxiliary inputs to achieve independent hash functions. Durstenfeld Shuffle (Durstenfeld, 1964) that outputs a random permutation within $O(n)$ computational complexity, is used in the signature schemes.

The signature size in the Cayrel et al. signature scheme is smaller and the execution time of the signature generation algorithm is smaller comparing to the Stern signature scheme. Conversely, the signature verification algorithm in the Stern signature scheme is faster than that in the Cayrel et al. signature scheme since it consists only of hash calculations, permutations, exclusive-or operations, and Hamming weight checks. Table 4 show the execution time of the signature schemes on a PC with a 3.5 GHz CPU and 16 GB of RAM and the size of the secret key, public key, and a signature. The size of the input messages is 32 B.

Alaoui et al. (Alaoui et al., 2013) implemented the signature schemes of Stern, Veron and Cayrel et al. for 80 bits of security. Further, security of Veron’s scheme is fixed by Jain et al. We have implemented the signature schemes of the Stern, Jain et al. and Cayrel et al. for 128 bits of security. In table 5, we have provided a comparison with the Stern and Cayrel et al. schemes.

6 CONCLUSION

We derived code-based signature schemes using Fiat-Shamir transformation on code-based zero-knowledge identification schemes: the Stern scheme, the Jain-Krenn-Pietrzak-Tentes scheme, and the Cayrel-Veron-Alaoui scheme. We then analyzed the security of the three signature schemes and derived the security parameters to achieve the 128-bit level of security. Furthermore, we implemented these signature schemes and compared their performance on a PC. Our security analysis and implementation show the Stern signature scheme is the most efficient regarding the size of the secret and public key and the execution time of key generation and signature verification, and the Cayrel et al. scheme is the most superior in terms of signature size. The signature generation of the Jain et al. scheme is the fastest, but the signature verification of the scheme is slowest. In our future work, we will optimize the implementation for IoT devices and study techniques for reducing the signature size.

REFERENCES


