Fault assisted Man-In-The-Middle Attack on MAYO

Siddhesh Shinde[1] and Sayandeep Saha[1]

Indian Institute of Technology Bombay, Mumbai, India,
siddheshh@cse.iitb.ac.in,sayandeepsaha@cse.iitb.ac.in

Abstract.
Multivariate quadratic (MQ) signature schemes such as MAYO are among the leading candidates for post-quantum digital signatures, with security relying on the hardness of solving systems of multivariate quadratic equations. In this paper, we present a fault-assisted man-in-the-middle attack targeting the key-generation procedure of MAYO. Specifically, we target the computation of the public key, which is represented as a system of m quadratic polynomials. Following the MAYO specification, each polynomial is associated with a matrix representation composed of the components $P_i^{(1)}, P_i^{(2)}$, and $P_i^{(3)}$. Our attack injects a single fault during the computation of the $P_i^{(3)}$ component of the public key, causing the resulting public key to satisfy a simplified linear relation involving the secret oil matrix. From the faulty public key, we derive an overdetermined linear system over F_{16} that enables complete recovery of the secret oil matrix.
Unlike previous fault attacks that primarily target the signing algorithm, our attack targets the key generation procedure to recover the oil secret and exploit the recovered oil secret to reconstruct the legitimate public key corresponding to the victim's secret key i.e correcting the public key. This allows the adversary to transparently participate in the communication as a man-in-the-middle, producing valid signatures on behalf of the victim.
We evaluate the attack on a fault simulated MAYO implementation and demonstrate successful recovery of the oil matrix from a single faulty key-generation execution followed by public key correction.

Keywords: Man-In-The-Middle Attack · Fault Attack · MAYO · Post-quantum Multivariate cryptography · Key recovery

1 Introduction

The National Institute of Standards and Technology (NIST) recently completed the first phase of its Post-Quantum Cryptography (PQC) standardization process, selecting cryptographic schemes designed to resist attacks from quantum computers. To further diversify the landscape of post-quantum digital signatures, NIST launched a second call for signature schemes based on alternative security assumptions and performance trade-offs. Among the candidates advancing in this process is MAYO [Beu21], a multivariate signature scheme whose security relies on the hardness of solving systems of multivariate quadratic equations over finite fields.

While MAYO offers strong theoretical security guarantees, practical deployments must also withstand implementation-level threats. Fault injection attacks have repeatedly demonstrated that physical adversaries can exploit transient computation errors to reveal secret information, creating a gap between mathematical security and real-world robustness. Several fault attacks against MAYO, have been proposed in recent years, primarily targeting

the signing procedure and exploiting faulty signatures to recover secret information, which also includes signature correction attacks such as [ABCR25, Mar26].

In this work, we present a fault-assisted man-in-the-middle attack against the MAYO key-generation procedure. We consider an adversary capable of injecting a fault during the computation of the $P^{(3)}$ component of the public key. The induced fault removes the quadratic contribution involving the secret oil matrix, resulting in a faulty public key that leaks linear information about the secret oil matrix. Using this leakage, the attacker efficiently reconstructs the complete oil matrix by solving an overdetermined linear system, followed by correction of the faulty public key.

More specifically, the injected fault causes the public-key component $P_i^{(3)}$ to be generated from a reduced expression in which the quadratic term involving the secret oil matrix and $P_i^{(1)}$ is omitted. Since the same oil appears across every public polynomial, combining the resulting equations on $P^{(3)}$ yields an overdetermined linear system on $P_i^{(2)}$ and $P_i^{(3)}$ public matrices and O secret oil matrix with a full rank whose unique solution reveals all entries of O.

Once the oil matrix has been recovered, the attacker recomputes the missing quadratic contribution and reconstructs the legitimate public key corresponding to the victim's secret key. This reconstructed public key can then replace the faulty public key during transmission, enabling an active man-in-the-middle attack while remaining indistinguishable from an honestly generated public key.

1.1 Contributions

- To the best of our knowledge, this is the first work that demonstrates a single-fault man-in-the-middle attack against the MAYO key-generation procedure leading to recovery of the full secret oil-space and public key correction.
- We propose a fault model targeting the computation of $P^{(3)}$ during MAYO key generation, resulting in a faulty public key that leaks linear information about the secret oil matrix.
- We derive an explicit system of linear equations relating the public matrices $P^{(2)}$ and $P^{(3)}$ to the secret oil matrix O and demonstrate that combining the equations obtained from all public polynomials yields a full-rank system, enabling complete secret recovery.
- We evaluate the feasibility of the attack on MAYO reference implementation from the pqm4 repository using fault simulation on CW308-STM32F415 target board.

2 Preliminaries

2.1 Multivariate Quadratic Maps

A multivariate quadratic map $\mathcal{P}$ over the finite field $\mathbb{F}_q$ consists of m quadratic polynomials in n variables. Each component

$$p_i(\mathbf{x}), \qquad \mathbf{x} = (x_0, \ldots, x_{n-1}),$$

is a multivariate polynomial of total degree at most two with coefficients in $\mathbb{F}_q$. Together, these components define the map

$$\mathcal{P}(\mathbf{x}) = (p_0, \ldots, p_{m-1}) : \mathbb{F}_q^n \to \mathbb{F}_q^m.$$

The differential of $\mathcal{P}$ is the bilinear map

$$\mathcal{P}'(\mathbf{x}, \mathbf{y}) = \mathcal{P}(\mathbf{x} + \mathbf{y}) - \mathcal{P}(\mathbf{x}) - \mathcal{P}(\mathbf{y}). \tag{1}$$

When $\mathcal{P}$ has degree two, $\mathcal{P}'$ is linear in either variable when the other is fixed.

The Unbalanced Oil and Vinegar (UOV) trapdoor function is a multivariate quadratic map that vanishes on a secret linear subspace $\mathcal{O} \subset \mathbb{F}_q^n$ of dimension $\dim(\mathcal{O}) = m$, i.e.,

$$\mathcal{P}(\mathbf{o}) = 0, \qquad \forall\ \mathbf{o} \in \mathcal{O}.$$

The subspace $\mathcal{O}$ is called the *oil space* and forms the trapdoor information. Let

$$O \in \mathbb{F}_q^{(n-m)\times m}$$

be the matrix whose columns form a basis of $\mathcal{O}$. The basis is chosen in systematic form,

$$\mathcal{O} = \begin{pmatrix} \bar{O} \\ I_m \end{pmatrix},$$

where I_m is the $m \times m$ identity matrix and $\bar{O} \in \mathbb{F}_q^{(n-m)\times m}$.

Given the trapdoor information, one can efficiently compute a preimage of a target digest $\mathbf{t} \in \mathbb{F}_q^m$. A random vinegar vector $\mathbf{v} \in \mathbb{F}_q^n$ is first selected, after which an oil vector $\mathbf{o} \in \mathcal{O}$ satisfying

$$\mathcal{P}(\mathbf{v} + \mathbf{o}) = \mathbf{t}$$

is determined. Using the differential, we obtain

$$\mathbf{t} = \mathcal{P}(\mathbf{v} + \mathbf{o}) = \mathcal{P}'(\mathbf{v}, \mathbf{o}) + \mathcal{P}(\mathbf{o}) + \mathcal{P}(\mathbf{v}). \tag{2}$$

Since $\mathcal{P}(\mathbf{o}) = 0$ by construction and $\mathcal{P}'(\mathbf{v}, \mathbf{o})$ is linear in $\mathbf{o}$, signing reduces to solving a linear system.

2.2 The MAYO Signature Scheme

MAYO is a post-quantum digital signature scheme built upon the UOV framework. Unlike classical UOV, MAYO significantly reduces the dimension of the secret oil space while preserving efficient signing through the whipping transformation. Its security relies on the hardness of solving systems of multivariate quadratic equations over finite fields.

Key Generation. The secret key space $\mathcal{O}$ is spanned by row vectors of $(O\ I_o)$, where $O \in \mathbb{F}_q^{(n-o)\times o}$. Matrix O can be deterministically expanded from a uniformly random seed using SHAKE256. Once O is obtained, the trapdoored multivariate quadratic map $\mathcal{P}$ is constructed such that all m multivariate quadratic polynomials vanish on $\mathcal{O}$. As every multivariate quadratic polynomial $p_i(\mathbf{x}) = \mathbf{x}^\top P_i \mathbf{x}$ is represented with an upper triangular matrix P_i, we partition

$$P_i = \begin{pmatrix} P_i^{(1)} & P_i^{(2)} \\ 0 & P_i^{(3)} \end{pmatrix}.$$

Then, for every $i \in [m]$, the following equation holds:

$$(O\ I_o) P_i (O\ I_o)^\top = O^\top P_i^{(1)} O + O^\top P_i^{(2)} + P_i^{(3)} = 0.$$

Submatrices $P_i^{(1)} \in \mathbb{F}_q^{(n-o)\times(n-o)}$ (upper triangular) and $P_i^{(2)} \in \mathbb{F}_q^{(n-o)\times o}$ are generated pseudorandomly from a uniformly random seed (seed_{pk}) using SHAKE256. $P_i^{(3)} \in \mathbb{F}_q^{o\times o}$ is derived directly from

$$P_i^{(3)} = \text{Upper}\Big(-O^\top P_i^{(1)} O - O^\top P_i^{(2)}\Big),$$

Upper$(\cdot)$ transforms a square matrix into its upper-triangular form. The Upper$(\cdot)$ does not influence the quadratic polynomial represented by P_i

Signing. To sign a message M, the signer computes the digest $\mathbf{t} = H(M\|salt)$, samples a random vinegar vector $\mathbf{v}$, and solves for an oil vector $\mathbf{o}$ satisfying $\mathcal{P}'(\mathbf{v}, \mathbf{o}) = \mathbf{t} - \mathcal{P}(\mathbf{v})$. Writing $\mathbf{o} = (O^\top \; I_m)^\top \mathbf{x}$, the system becomes $L\mathbf{x} = \mathbf{t} - \mathcal{P}(\mathbf{v})$, where

$$L[i,:] = \mathbf{v}^\top S_i, \qquad S_i = \Big(P_i^{(1)} + P_i^{(1)\top}\Big) O + P_i^{(2)}.$$

If the linear system is nonsingular, the solution uniquely determines the signature. Otherwise, a new vinegar vector is sampled.

Verification. The verifier receives message M, salt $salt$, signature s, and locally computes $\mathcal{P}(s)$ and $t' = H(M\|salt)$. The verifier accepts s as the valid signature if $\mathcal{P}(s) = t'$. Otherwise, the signature is rejected.

3 Proposed Fault Attack and Key Correction

Our attack targets the key-generation procedure of MAYO, specifically the computation of the public-key component $P_i^{(3)}$. Unlike previous fault attacks that exploit faulty signatures, our attack injects a fault during key generation, producing a faulty public key that leaks information about the secret oil matrix.

The attack assumes that an adversary is capable of inducing a transient fault during the computation of $P_i^{(3)}$. More precisely, we target the computation

$$P_i^{(3)} = \text{Upper}\left(-O^\top P_i^{(1)} O - O^\top P_i^{(2)}\right),$$

and skip the computation of the quadratic term $O^\top P_i^{(1)} O$. Consequently, the generated public key satisfies

$$P_i^{(3)} = \text{Upper}\left(-O^\top P_i^{(2)}\right),$$

which introduces linear relations between the public matrices and the secret oil matrix. Our attack consists of two main steps.

1. **Fault Injection and Oil Recovery.** A fault is injected during the computation of the $P_i^{(3)}$ matrices in the key-generation algorithm. The resulting faulty public key satisfies a simplified linear equation involving the unknown oil matrix. By collecting the equations corresponding to all public polynomials, we obtain an overdetermined linear system over $\mathbb{F}_{16}$ whose unique solution reveals the complete secret oil matrix.
2. **Correction of the Faulty Public Key.** Once the oil matrix has been recovered, the attacker recomputes the omitted quadratic contribution $O^\top P_i^{(1)} O$ and reconstructs the correct $P_i^{(3)}$ matrices. This allows reconstruction of the legitimate verifiable public key corresponding to the victim's secret key. During public-key distribution, an adversary can replace the faulty public key with the reconstructed one, thereby enabling a man-in-the-middle attack.

The following subsections describe these two phases in detail. We begin by describing the implementation of the MAYO key-generation procedure relevant to our attack. We then identify the fault target and show how a single skipped computation produces a faulty public key that satisfies a simplified linear relation in the unknown oil matrix.

3.1 Fault Injection and Oil Recovery

The first phase of our attack exploits an implementation detail of the MAYO key-generation algorithm. Although the MAYO specification [BCC+23] defines the computation of $P_i^{(3)}$ as a single matrix expression, the reference implementation computes it incrementally through an intermediate update of $P_i^{(2)}$. By targeting this intermediate computation, a single skipped operation eliminates the quadratic contribution involving the secret oil matrix, resulting in a faulty public key that leaks sufficient information to recover the complete oil matrix.

As per the implementation in pqm4 repository, the $P_i^{(3)}$ component is computed as follows:

It first updates the randomly generated matrix $P_i^{(2)}$ as

$$P_i^{(2)} \leftarrow P_i^{(2)} + OP_i^{(1)},$$

after which the computation proceeds as

$$P_i^{(3)} = O^\top P_i^{(2)}.$$

Substituting the updated value of $P_i^{(2)}$ gives

$$\begin{aligned} P_i^{(3)} &= O^\top \left(P_i^{(2)} + OP_i^{(1)} \right) \\ &= O^\top P_i^{(2)} + O^\top OP_i^{(1)}, \end{aligned}$$

which is equivalent to the specification before the upper-triangular transformation.

Fault Model. Our fault model targets the intermediate update

$$P_i^{(2)} \leftarrow P_i^{(2)} + OP_i^{(1)}.$$

By injecting an instruction-skip fault, the entire computation responsible for this update is omitted. Consequently, the subsequent multiplication operates on the original pseudorandom matrix $P_i^{(2)}$, producing the faulty public-key component

$$P_i^{(3)} = \mathrm{Upper}(O^\top P_i^{(2)}).$$

Since $P_i^{(2)}$ and the faulty $P_i^{(3)}$ are both publicly available after expansion of the compact public key, the attacker obtains a linear relation in the unknown entries of the oil matrix. Reconstructing the symmetric matrix from its compressed upper-triangular $\mathrm{Upper}(\cdot)$ representation yields

$$P_i^{(3)} = O^\top P_i^{(2)} + P_i^{(2)\top} O,$$

where the second term arises from symmetrizing the upper-triangular representation of $P_i^{(3)}$. Writing the above equation entry-wise, for every polynomial $\ell \in [m]$ and every upper-triangular entry (i, j),

$$P_\ell^{(3)}(i,j) = \sum_{k=0}^{v-1} O_{k,i} P_\ell^{(2)}(k,j) + \sum_{k=0}^{v-1} O_{k,j} P_\ell^{(2)}(k,i). \tag{1}$$

Each such equation is linear in the unknown entries of O. Collecting the equations corresponding to every polynomial and every upper-triangular matrix entry yields an overdetermined linear system $A\mathbf{x} = \mathbf{b}$, where $\mathbf{x}$ contains the $(v \times o)$ unknown entries of the oil matrix. The system consists of $(m \times \frac{o(o+1)}{2})$ linear equations over $\mathbb{F}_{16}$. The resulting coefficient matrix is full rank system for all MAYO parameter sets, allowing unique recovery of the entire secret oil matrix using Gaussian elimination.

Algorithm 1 summarizes the key recovery procedure.

Algorithm 1 Recovery of the Oil Matrix from a Faulty Public Key

Input: Faulty public key cpk
Output: Oil matrix O
// Compute $P^{(2)}$ and $P^{(3)}$ from cpk
$P \leftarrow$ AES-128-CTR($seed_{pk}$, $P1_bytes + P2_bytes$) ▷ $P_i^{(2)} \in \mathbb{F}_q^{(n-o)\times o}$
$\{P_i^{(2)}\}_{i\in[m]} \leftarrow \text{Decode}_{P^{(2)}}(P[P1_bytes : P1_bytes + P2_bytes])$
$\{P_i^{(3)}\}_{i\in[m]} \leftarrow \text{Decode}_{P^{(3)}}(cpk[pk_seed_bytes :])$

//Build equation system
$u \leftarrow v * o, \qquad e \leftarrow m * \dfrac{o(o+1)}{2}$
for $\ell = 0, \ldots, m-1$ **do**
  for $i = 0, \ldots, o-1$ **do**
    for $j = i, \ldots, o-1$ **do**
      $b_{eq} \leftarrow P_\ell^{(3)}(i,j)$ ▷ $\mathbf{b} \in \mathbb{F}_{16}^{e}$
      for $k = 0, \ldots, v-1$ **do**
        $A_{eq,\, iv+k} \leftarrow P_\ell^{(2)}(k,j)$ ▷ $A \in \mathbb{F}_{16}^{e\times u}$
        if $i \neq j$ **then**
          $A_{eq,\, jv+k} \leftarrow P_\ell^{(2)}(k,i)$

//Solve $A\mathbf{x} = \mathbf{b}$ over $\mathbb{F}_{16}$ using Gaussian elimination.
$\mathbf{x} \leftarrow \text{SOLVE}(A, \mathbf{b})$ ▷ $\mathbf{x} \in \mathbb{F}_{16}^{u}$

// Reshape $\mathbf{x} \to O \in \mathbb{F}_{16}^{(n-o)\times o}$.
for $i = 0, \ldots, o-1$ **do**
  for $k = 0, \ldots, v-1$ **do**
    $O_{k,i} \leftarrow x_{iv+k}$
return O

3.2 Correction of the Faulty Public Key

Since the public-key seed $seed_{pk}$ is included as part of the compact public key cpk, the matrices $P_i^{(1)}$ and $P_i^{(2)}$ can be deterministically regenerated by following the standard MAYO key-generation procedure. The recovered oil matrix is then used to recompute the missing quadratic contribution, producing the correct $P_i^{(3)}$ matrices.

More precisely, the attacker first derives the pseudorandom matrices $P_i^{(1)}$ and $P_i^{(2)}$ from $seed_{pk}$ using the same AES-128-CTR based expansion. Subsequently, for every polynomial i, the attacker re-computes

$$P_i^{(3)} = O^T P_i^{(2)} + O^T P_i^{(1)} O,$$

where O is the recovered oil matrix. Refer Algorithm 2 for the correction procedure.

Algorithm 2 Correction of the Faulty Public Key

Input: $seed_{pk}$, recovered oil matrix O
Output: Correct public key cpk
$P \leftarrow \text{AES-128-CTR}(seed_{pk}, P1_bytes + P2_bytes)$ ▷ $P_i^{(2)} \in \mathbb{F}_q^{(n-o)\times o}$
$\{P_i^{(1)}\}_{i\in[m]} \leftarrow \text{Decode}_{P^{(1)}}(P[0 : P1_bytes])$
$\{P_i^{(2)}\}_{i\in[m]} \leftarrow \text{Decode}_{P^{(2)}}(P[P1_bytes : P1_bytes + P2_bytes])$
for $i = 0, \ldots, m-1$ **do**
    $P_i^{(3)} \leftarrow \text{Upper}\left(-O^\top P_i^{(1)} O - O^\top P_i^{(2)}\right)$
$cpk \leftarrow seed_{pk} \parallel \text{Encode}_{P^{(3)}}(\{P_i^{(3)}\}_{i\in[m]})$
return cpk

Listing 1 C code snippet of the MAYO key generation focusing the computation of $P_i^{(3)}$. The proposed fault skips the call to `P1_times_O()`.

```
static inline
void compute_P3(const mayo_params_t* p, const uint64_t* P1, uint64_t *P2,
                const unsigned char *O, uint64_t *P3)
{
    const int m_vec_limbs = PARAM_m_vec_limbs(p);
    const int param_v = PARAM_v(p);
    const int param_o = PARAM_o(p);

    // compute P1*O + P2
    P1_times_O(p, P1, O, P2);

    // compute P3 = O^t * (P1*O + P2)
    mul_add_mat_trans_x_m_mat(m_vec_limbs, O, P2, P3, param_v,
                              param_o, param_o);
}
```

Finally, the corrected public key is obtained by concatenating $seed_{pk}$ with the encoded representation of the recomputed matrices $P_i^{(3)}{}_{i\in[m]}$. Since the reconstructed public key is identical to the one produced by an uncompromised execution of the MAYO key-generation algorithm, it is accepted by any verifier.

4 Experimental Results

We evaluate the proposed fault-assisted man-in-the-middle attack on the MAYO reference implementation from the pqm4 repository[1]. The attack targets the computation of the public-key component $P_i^{(3)}$ during key generation. Listing 1 shows the relevant excerpt from the reference implementation. The function `P1_times_O()` performs the update $P_i^{(2)} \leftarrow P_i^{(2)} + OP_i^{(1)}$, thereby incorporating the quadratic contribution of the secret oil matrix. The updated matrix is subsequently used to compute

$$P_i^{(3)} = O^\top (P_i^{(2)} + OP_i^{(1)}).$$

[1] https://github.com/PQCMayo/pqm4/

Listing 2 Assembly code snippet corresponding to the fault target in `compute_P3()`.

```
...
87d8:    mov     r1, r3
87da:    mov     r2, r4
87dc:    mov     r0, r7
87de:    str     r3, [sp, #4]

87e0:    bl      P1_times_O.isra.0

87e4:    mov     r2, r4
87e6:    ldr     r1, [sp, #4]
87e8:    add     r0, sp, #2400
87ec:    bl      mul_add_mat_trans_x_m_mat_m4f_V_O_O_asm
...
```

Table 1: Evaluation of the linear equation system for key recovery

Scheme	Variables	Equations	Rank	Oil Recovery	Verification
MAYO-1	624	2808	624	✓	Pass
MAYO-2	1088	9792	1088	✓	Pass
MAYO-3	1080	5940	1080	✓	Pass
MAYO-5	1704	11076	1704	✓	Pass

4.1 Fault Simulation

Our fault model targets this intermediate computation. By skipping the call to `P1_times_O()` (Refer Listing 1), the update to $P_i^{(2)}$ is omitted and the subsequent multiplication operates on the original pseudorandom matrix $P_i^{(2)}$, eliminating the quadratic term $O^\top P_i^{(1)} O$ and producing the faulty public key required by the proposed attack. To demonstrate the feasibility of the proposed attack, we evaluate a fault-simulated MAYO implementation on a CW308-STM32F415 target mounted on a CW308 UFO Board by bypassing the call to `P1_times_O()` while leaving the remainder of the key-generation procedure unmodified. This fault model emulates the effect of a single instruction-skip fault, producing the same program state as if the function call were skipped during execution.

The target computes the faulty public key which is then used in the key recovery algorithm, which successfully constructs the linear system, recovers the complete secret oil matrix, and reconstructs the legitimate public key as described in Section 3. The corrected public key successfully verifies signatures generated using the corresponding secret key, demonstrating the practicality of the proposed fault-assisted man-in-the-middle attack. We have evaluated the proposed fault attack across every MAYO parameter set (Table 1), and have successfully recovered the Oil secret, corrected the public key and verified the corrected key for a given signature.

5 Conclusion

In this paper, we presented a fault-assisted man-in-the-middle attack against the key-generation procedure of the MAYO signature scheme. Our attack exploits an implementation detail in the computation of the public-key component $P^{(3)}$, where the reference implementation incrementally updates the intermediate matrix $P^{(2)}$ before computing $P^{(3)}$. By skipping this intermediate computation, the generated public key loses its quadratic dependence on the secret oil matrix, yielding a faulty public key that satisfies a linear

relation in the unknown secret.

We showed that the resulting leakage can be formulated as an overdetermined system of linear equations over oil variables. Solving this system enables complete recovery of the secret oil matrix from a single faulty key-generation execution. Using the recovered oil matrix together with the public seed, the attacker can reconstruct the legitimate public key corresponding to the victim's secret key, thereby enabling a fault-assisted man-in-the-middle attack.

We evaluate the proposed attack on a fault simulated MAYO implementation by emulating an instruction-skip fault during the computation of $P^{(3)}$. Our experimental results demonstrate successful recovery of the oil matrix and reconstruction of a valid public key across all MAYO parameter sets. These findings identify key generation as an important attack surface for multivariate post-quantum signature schemes

References

[ABCR25] Mohamed Abdelmonem, Lejla Batina, Durba Chatterjee, and Håvard Raddum. Correction-based fault attack against randomized mayo. *Cryptology ePrint Archive*, 2025.

[BCC+23] Ward Beullens, Fabio Campos, Sofía Celi, Basil Hess, and Matthias Kannwischer. Mayo-algorithm specifications. *MAYO team*, 2023.

[Beu21] Ward Beullens. Mayo: practical post-quantum signatures from oil-and-vinegar maps. In *International Conference on Selected Areas in Cryptography*, pages 355–376. Springer, 2021.

[Mar26] Alberto Marcos. Secret-subspace recovery in mayo via linearization of errors from a single fault. *Cryptology ePrint Archive*, 2026.