IDEMIA SECURE TRANSACTIONS

UVSQ université PARIS-SACLAY

Differential Fault Attacks on MQOM

Breaking the Heart of Multivariate Evaluation

Vladimir Sarde, Nicolas Debande

March 31, 2026

Outline

Introduction

- MPCitH was introduced in 2007
- NIST signature standardization: 6 out of 14 candidates are MPCitH based
- MQOM combines efficiency and conservative security

Introduction

- MPCitH was introduced in 2007
- NIST signature standardization: 6 out of 14 candidates are MPCitH based
- MQOM combines efficiency and conservative security

This work

Fault attacks on MQOM targeting the MPCitH paradigm.

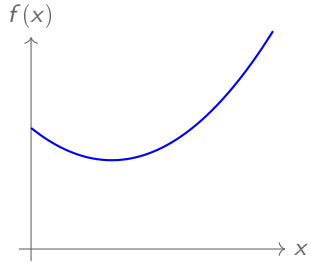
1 Some Definitions

Shamir Secret Sharing (SSS) Modified

Shamir Secret Sharing (SSS) Modified

› Choose a random polynomial of degree t:

$$f(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1} + s x^t$$



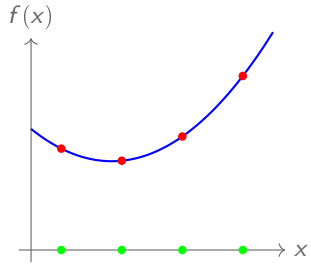
Example: $t = 3$, $n = 4$

Shamir Secret Sharing (SSS) Modified

- Choose a random polynomial of degree t:

$$f(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1} + s x^t$$

- Distribute $f(i)$ to each participant.



Example: $t = 3$, $n = 4$

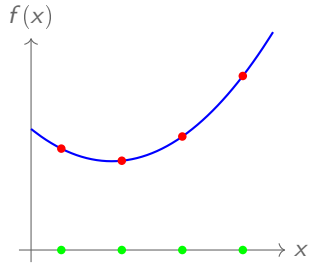
Shamir Secret Sharing (SSS) Modified

- Choose a random polynomial of degree t:

$$f(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1} + s x^t$$

- Distribute $f(i)$ to each participant.

- Coalition of:
  $\geqslant t + 1$ participants recover s via interpolation.



Example: $t = 3$, $n = 4$

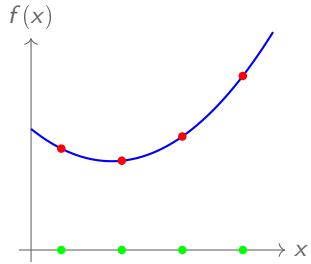
Shamir Secret Sharing (SSS) Modified

- Choose a random polynomial of degree t:

$$f(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1} + s x^t$$

- Distribute $f(i)$ to each participant.

- Coalition of:
  - $\geqslant t+1$ participants recover s via interpolation.
  - $< t+1$ participants have no information.



Example: $t = 3$, $n = 4$

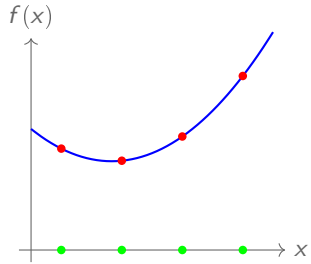
Shamir Secret Sharing (SSS) Modified

- Choose a random polynomial of degree t:

$$f(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1} + s x^t$$

- Distribute $f(i)$ to each participant.

- Coalition of:
  - $\geqslant t+1$ participants recover s via interpolation.
  - $< t+1$ participants have no information.



Example: $t = 3$, $n = 4$

Vector $\mathbf{x} \in (\mathbb{F}_q)^n$ are shared coordinate-wise.

MQ Problem

Definition

Over $\mathbb{F}_q$, the MQ problem is to solve:

$$\begin{cases} \sum_{j,k} a_{jk}^{(1)} x_j x_k + \sum_j b_j^{(1)} x_j = y_1 & \iff \mathbf{x}^T \mathbf{A}_1 \mathbf{x} + \mathbf{b}_1 \mathbf{x} = y_1 \\ \vdots \qquad\qquad\qquad \vdots & \iff \qquad\qquad \vdots \\ \sum_{j,k} a_{jk}^{(m)} x_j x_k + \sum_j b_j^{(m)} x_j = y_m & \iff \mathbf{x}^T \mathbf{A}_m \mathbf{x} + \mathbf{b}_m \mathbf{x} = y_m \end{cases}$$

MQ Problem

Definition

Over $\mathbb{F}_q$, the MQ problem is to solve:

$$\begin{cases} \sum_{j,k} a_{jk}^{(1)} x_j x_k + \sum_j b_j^{(1)} x_j = y_1 & \iff \mathbf{x}^T \mathbf{A}_1 \mathbf{x} + \mathbf{b}_1 \mathbf{x} = y_1 \\ \vdots \qquad\qquad\qquad \vdots & \iff \qquad\qquad \vdots \\ \sum_{j,k} a_{jk}^{(m)} x_j x_k + \sum_j b_j^{(m)} x_j = y_m & \iff \mathbf{x}^T \mathbf{A}_m \mathbf{x} + \mathbf{b}_m \mathbf{x} = y_m \end{cases}$$

MQOM → Public Key: $\mathbf{A}_i, \mathbf{b}_i, \mathbf{y}$
→ Secret Key: $\mathbf{x}$

MQ Problem

Definition

Over $\mathbb{F}_q$, the MQ problem is to solve:

$$\begin{cases} \sum_{j,k} a_{jk}^{(1)} x_j x_k + \sum_j b_j^{(1)} x_j = y_1 & \iff \mathbf{x}^T \mathbf{A}_1 \mathbf{x} + \mathbf{b}_1 \mathbf{x} = y_1 \\ \vdots \qquad\qquad \vdots & \iff \qquad \vdots \\ \sum_{j,k} a_{jk}^{(m)} x_j x_k + \sum_j b_j^{(m)} x_j = y_m & \iff \mathbf{x}^T \mathbf{A}_m \mathbf{x} + \mathbf{b}_m \mathbf{x} = y_m \end{cases}$$

MQOM $\rightarrow$ Public Key: $\mathbf{A}_i, \mathbf{b}_i, \mathbf{y}$
$\rightarrow$ Secret Key: $\mathbf{x}$

Let:

$$F : \mathbb{F}^n \rightarrow \mathbb{F}^m$$
$$(F(\mathbf{x}))_i = \mathbf{x}^\top \mathbf{A}_i \mathbf{x} + \mathbf{b}_i \mathbf{x}$$

MQ Problem

Definition

Over $\mathbb{F}_q$, the MQ problem is to solve:

$$\begin{cases} \sum_{j,k} a_{jk}^{(1)} x_j x_k + \sum_j b_j^{(1)} x_j = y_1 & \iff \mathbf{x}^T \mathbf{A}_1 \mathbf{x} + \mathbf{b}_1 \mathbf{x} = y_1 \\ \vdots \qquad\qquad \vdots & \iff \qquad \vdots \\ \sum_{j,k} a_{jk}^{(m)} x_j x_k + \sum_j b_j^{(m)} x_j = y_m & \iff \mathbf{x}^T \mathbf{A}_m \mathbf{x} + \mathbf{b}_m \mathbf{x} = y_m \end{cases}$$

MQOM $\rightarrow$ Public Key: $\mathbf{A}_i, \mathbf{b}_i, \mathbf{y}$
$\rightarrow$ Secret Key: $\mathbf{x}$

Let:

$$F : \mathbb{F}^n \rightarrow \mathbb{F}^m$$

$$(F(\mathbf{x}))_i = \mathbf{x}^\top \mathbf{A}_i \mathbf{x} + \mathbf{b}_i \mathbf{x}$$

Therefore:

$$F(\mathbf{x}) = \mathbf{y}$$

MPCitH 2

Zero Knowledge Proof

Prover

Verifier

Zero Knowledge Proof

Prover

- Generate a SSS $\mathbf{P}_x$ and $\mathbf{P}_u$ s.t. $d(\mathbf{P}_x) = d(\mathbf{P}_u) = 1$

Verifier

Zero Knowledge Proof

Prover

- Generate a SSS $\mathbf{P}_x$ and $\mathbf{P}_u$ s.t. $d(\mathbf{P}_x) = d(\mathbf{P}_u) = 1$

Oracle

$[\mathbf{P}_x, \mathbf{P}_u]$

Verifier

Zero Knowledge Proof

Prover

- Generate a SSS $\mathbf{P}_x$ and $\mathbf{P}_u$ s.t. $d(\mathbf{P}_x) = d(\mathbf{P}_u) = 1$
- $\mathbf{P}_z = F(\mathbf{P}_x) - \mathbf{y}X^2$,
  $\mathbf{P}_z = F(\mathbf{x})X^2 - \mathbf{y}X^2 + ...X + ...$
  Therefore $d(\mathbf{P}_z) = 1$ as $F(\mathbf{x}) = \mathbf{y}$.

Oracle

$[\mathbf{P}_x, \mathbf{P}_u]$

Verifier

Zero Knowledge Proof

Prover

- Generate a SSS $\mathbf{P}_x$ and $\mathbf{P}_u$ s.t. $d(\mathbf{P}_x) = d(\mathbf{P}_u) = 1$
- $\mathbf{P}_z = F(\mathbf{P}_x) - \mathbf{y}X^2$,
  $\mathbf{P}_z = F(\mathbf{x})X^2 - \mathbf{y}X^2 + ...X + ...$
  Therefore $d(\mathbf{P}_z) = 1$ as $F(\mathbf{x}) = \mathbf{y}$.
- $\mathbf{P}_\alpha = \mathbf{P}_u + \Gamma \cdot \Phi(\mathbf{P}_z)$.

Oracle

$[\mathbf{P}_x, \mathbf{P}_u]$

Verifier

Zero Knowledge Proof

Prover

- Generate a SSS $\mathbf{P}_x$ and $\mathbf{P}_u$ s.t. $d(\mathbf{P}_x) = d(\mathbf{P}_u) = 1$
- $\mathbf{P}_z = F(\mathbf{P}_x) - \mathbf{y}X^2$,
  $\mathbf{P}_z = F(\mathbf{x})X^2 - \mathbf{y}X^2 + ...X + ...$
  Therefore $d(\mathbf{P}_z) = 1$ as $F(\mathbf{x}) = \mathbf{y}$.
- $\mathbf{P}_\alpha = \mathbf{P}_u + \Gamma \cdot \Phi(\mathbf{P}_z)$.

Oracle
$[\mathbf{P}_x, \mathbf{P}_u]$

$\mathbf{P}_\alpha$
$\deg(\mathbf{P}_\alpha) = 1$

Verifier

Zero Knowledge Proof

Prover

- Generate a SSS $\mathbf{P}_x$ and $\mathbf{P}_u$ s.t. $d(\mathbf{P}_x) = d(\mathbf{P}_u) = 1$
- $\mathbf{P}_z = F(\mathbf{P}_x) - \mathbf{y}X^2$,
  $\mathbf{P}_z = F(\mathbf{x})X^2 - \mathbf{y}X^2 + ...X + ...$
  Therefore $d(\mathbf{P}_z) = 1$ as $F(\mathbf{x}) = \mathbf{y}$.
- $\mathbf{P}_\alpha = \mathbf{P}_u + \Gamma \cdot \Phi(\mathbf{P}_z)$.

$$\xrightarrow[[\mathbf{P}_x, \mathbf{P}_u]]{\text{Oracle}}$$

$$\xrightarrow[\deg(\mathbf{P}_\alpha) = 1]{\mathbf{P}_\alpha}$$

Verifier

- Choose a random point r.
- Query $\mathbf{P}_x(r)$, $\mathbf{P}_u(r)$ from oracle.
- Compute $\mathbf{P}_{\tilde{z}}(r) = F(\mathbf{P}_x(r)) - \mathbf{y}X^2$.
- Check $\mathbf{P}_\alpha(r) \stackrel{?}{=} \mathbf{P}_u(r) + \Gamma \cdot \Phi(\mathbf{P}_{\tilde{z}}(r))$.

Our New Fault Attacks

3

On Public Values

On Secret Values

Fault Target

Prover

- Generate a SSS $\mathbf{P}_x$ and $\mathbf{P}_u$ s.t. $d(\mathbf{P}_x) = d(\mathbf{P}_u) = 1$
- $\mathbf{P}_z = F(\mathbf{P}_x) - \mathbf{y}X^2$,
  $\mathbf{P}_z = F(\mathbf{x})X^2 - \mathbf{y}X^2 + ...X + ...$
  Therefore $d(\mathbf{P}_z) = 1$ as $F(\mathbf{x}) = \mathbf{y}$.
- $\mathbf{P}_\alpha = \mathbf{P}_u + \Gamma \cdot \Phi(\mathbf{P}_z)$.

$$\xrightarrow[]{\text{Oracle}} \quad [\mathbf{P}_x, \mathbf{P}_u]$$

$$\xrightarrow[]{\mathbf{P}_\alpha} \quad \deg(\mathbf{P}_\alpha) = 1$$

Verifier

- Choose a random point r.
- Query $\mathbf{P}_x(r)$, $\mathbf{P}_u(r)$ from oracle.
- Compute $\mathbf{P}_{\tilde{z}}(r) = F(\mathbf{P}_x(r)) - \mathbf{y}X^2$.
- Check $\mathbf{P}_\alpha(r) \stackrel{?}{=} \mathbf{P}_u(r) + \Gamma \cdot \Phi(\mathbf{P}_{\tilde{z}}(r))$.

Strategy

Idea

Fault prover's computation:

$$\mathbf{P}_z = F(\mathbf{P}_x) - \mathbf{y}X^2$$

Strategy

Idea

Fault prover's computation:

$$\mathbf{P}_z = F(\mathbf{P}_x) - \mathbf{y}X^2$$

The verifier receives:

- A **faulty evaluation** $\widetilde{\mathbf{P}_\alpha}(r)$
- The **correct oracle evaluation** $\mathbf{P}_\alpha(r)$

Strategy

Idea

Fault prover's computation:

$$\mathbf{P}_z = F(\mathbf{P}_x) - \mathbf{y}X^2$$

The verifier receives:

- A **faulty evaluation** $\widetilde{\mathbf{P}_\alpha}(r)$
- The **correct oracle evaluation** $\mathbf{P}_\alpha(r)$
- Compare
- Recover information

Our New Fault Attacks

3

- On Public Values
- On Secret Values

Fault on Public Values

Fault model: **stuck-at**

$$F(\mathbf{P}_x)_i = \mathbf{P}_x^\top \mathbf{A}_i \mathbf{P}_x + \mathbf{b}_i^\top \mathbf{P}_x \cdot X - y_i \cdot X^2 \quad \text{for all } i \in \{1, \ldots, m\}$$

- Fault on $\mathbf{b}_i$

Fault on Public Values

Fault model: **stuck-at**

$$F(\mathbf{P}_x)_i = \mathbf{P}_x^\top \mathbf{A}_i \mathbf{P}_x + \mathbf{b}_i^\top \mathbf{P}_x \cdot X - y_i \cdot X^2 \quad \text{for all } i \in \{1, \ldots, m\}$$

- Fault on $\mathbf{b}_i$

Take the difference $\widetilde{\mathbf{P}_\alpha}(r) - \mathbf{P}_\alpha(r) = -(\mathbf{b}_i)_j (x_0)_j \cdot r$

Fault on Public Values

Fault model: **stuck-at**

$$F(\mathbf{P}_x)_i = \mathbf{P}_x^\top \mathbf{A}_i \mathbf{P}_x + \mathbf{b}_i^\top \mathbf{P}_x \cdot X - y_i \cdot X^2 \quad \text{for all } i \in \{1, \ldots, m\}$$

- Fault on $\mathbf{b}_i$

Take the difference $\widetilde{\mathbf{P}_\alpha}(r) - \mathbf{P}_\alpha(r) = -(\mathbf{b}_i)_j (x_0)_j \cdot r$

$\rightarrow$ 48 to 160 stuck at faults required.

3 Our New Fault Attacks

- On Public Values
- On Secret Values

Fault on Secret Values

Fault model: **random**

$$F(\mathbf{P}_x)_i = \mathbf{P}_x^\top \mathbf{A}_i \mathbf{P}_x + \mathbf{b}_i^\top \mathbf{P}_x \cdot X - y_i \cdot X^2 \quad \text{for all } i \in \{1, \ldots, m\}$$
$$\text{with } \mathbf{P}_x = \mathbf{x}_0 + \mathbf{x} \cdot X$$

Fault on Secret Values

Fault model: **random**

$$F(\mathbf{P}_x)_i = \mathbf{P}_x^\top \mathbf{A}_i \mathbf{P}_x + \mathbf{b}_i^\top \mathbf{P}_x \cdot X - y_i \cdot X^2 \quad \text{for all } i \in \{1, \ldots, m\}$$
$$\text{with } \mathbf{P}_x = \mathbf{x}_0 + \mathbf{x} \cdot X$$

- Fault on $\mathbf{x}_0$: not exploitable because of SSS.

Fault on Secret Values

Fault model: **random**

$$F(\mathbf{P}_x)_i = \mathbf{P}_x^\top \mathbf{A}_i \mathbf{P}_x + \mathbf{b}_i^\top \mathbf{P}_x \cdot X - y_i \cdot X^2 \quad \text{for all } i \in \{1, \ldots, m\}$$
$$\text{with } \mathbf{P}_x = \mathbf{x}_0 + \mathbf{x} \cdot X$$

- Fault on $\mathbf{x}_0$: not exploitable because of SSS.
- Fault on $\mathbf{x}$: $\widetilde{\mathbf{P}_\alpha}(r) - \mathbf{P}_\alpha(r)$ gives quadratic equations.

Fault on Secret Values

$$\widetilde{\mathbf{P}_\alpha}(r) - \mathbf{P}_\alpha(r) = F(x+\delta) - F(x)$$

Fault on Secret Values

$$\widetilde{\mathbf{P}_\alpha}(r) - \mathbf{P}_\alpha(r) = F(x+\delta) - F(x)$$

› A fault gives m equations linear in $\mathbf{x}$ and quadratic in the fault δ.

Fault on Secret Values

$$\widetilde{\mathbf{P}_\alpha}(r) - \mathbf{P}_\alpha(r) = F(x+\delta) - F(x)$$

- A fault gives m equations linear in $\mathbf{x}$ and quadratic in the fault δ.
- Perform an exhaustive search over δ.

Fault on Secret Values

$$\widetilde{\mathbf{P}_\alpha}(r) - \mathbf{P}_\alpha(r) = F(x+\delta) - F(x)$$

- A fault gives m equations linear in $\mathbf{x}$ and quadratic in the fault δ.
- Perform an exhaustive search over δ.

The rank of the system depends on Φ and Γ.

- 1 to 20 faults require.
- Complexity: $(2^{|\delta|})^1$ to $(2^{|\delta|})^{20}$.

Fault on Secret Values

$$\widetilde{\mathbf{P}_\alpha}(r) - \mathbf{P}_\alpha(r) = F(x+\delta) - F(x)$$

- A fault gives m equations linear in $\mathbf{x}$ and quadratic in the fault δ.
- Perform an exhaustive search over δ.

The rank of the system depends on Φ and Γ.

- 1 to 20 faults require.
- Complexity: $(2^{|\delta|})^1$ to $(2^{|\delta|})^{20}$.

Improvement

Equations lie in $\mathbb{K}$, while the secret lies in $\mathbb{F}$.

Fault on Secret Values

One or two fault are enough.

Variant	n	1 Fault	2 Faults
MQOM - Fast	48	47	48

Table: Rank of the system

Variant	$\lvert\delta\rvert$ on 1 bit	$\lvert\delta\rvert$ on 16 bits	$\lvert\delta\rvert$ on 32 bits
MQOM - Fast	2^{16}	2^{16}	2^{32}

Table: Search space

Experimentally 2^{32} system resolutions take 22 hours on a PC.

Countermeasures

First idea:

- Performing a verification: effective but expensive

Countermeasures

First idea:

- Performing a verification: effective but expensive

Possible protections:

- Partial verification, checksums on sensitive coefficients
- Shuffling

Conclusion
4

Conclusion – Takeaways

- A large attack surface
- Two key recovery attacks:
  - Few faults on unprotected values
  - Single random fault on secret values
- Practical Countermeasures
- Extends to other MPCitH schemes