Identification and Signature based On MQ Problem

Manish Kumar (21044)

March 14, 2023

Abstract

Multivariate Quadratic Identification protocol (MQ-IDS) and Digital Signature (MQ-DSS) scheme are part of Multivariate public key cryptography (MPKC). MPKC is considered a prospective candidate for post quantum cryptography.[1] Even MQ-DSS was a second round candidate in NIST post quantum cryptography standarisation project. [2] A secure public key crpytosystem requires a secure trapdoor function. In this scribe, we will discuss a trapdoor based on Multivariate quadratics. We will do security analysis of a 3-pass IDS (Sakumoto et.al) in terms of Zero Knowledge and Proof of Knowledge. This IDS can be extended to create MQ-DSS. Finally we do brief analysis of 5-pass IDS in signature construction which was floated as an exercise.

Knowing The Trapdoor Function: The Multivariate Quadratic Problem

Consider the below m-systems of Multivariate polynomial in n-variables over a finite field $\mathbb{F}_q$:

$$f^1(x_1, ..., x_n) = \sum_{i=1}^{n}\sum_{i=1}^{n} \alpha_{ij}^{(1)} x_i x_j + \sum_{i=1}^{n} \beta_i^{(1)} x_i + \gamma^{(1)}$$
$$\cdots$$
$$f^m(x_1, ..., x_n) = \sum_{i=1}^{n}\sum_{i=1}^{n} \alpha_{ij}^{(m)} x_i x_j + \sum_{i=1}^{n} \beta_i^{(m)} x_i + \gamma^{(m)}$$

$MQ\ problem$ is a task to compute $\hat{x} = (x_1, ..., x_n)$ from $y = (y_1, ..y_m)$ such that $f^k(\hat{x}) = y_k, \forall\ k \in \{1, ..., m\}$, where all coefficients and variables belongs to finite field $\mathbb{F}_q$. The problem is originally from computational algebraic geometry. It has been shown that the decision version of $MQ\ problem$ is NP-hard. Based on current known state of art classical and quantum algorithms, it is conjectured that $MQ\ problem$ has no efficient solution. The problem would get much harder if we increase the order of polynomial to cubic, quartic or above. But public key (pk) size would also increase accordingly. Since any polynomial is defined uniquely by its coefficients. Hence, the set of coefficients and $\mathbb{F}_q$ is part of pk. In case of a d-degree polynomial with n-variables, number of all the coefficients is $\binom{n+d}{d}$. Hence, for m-sets of d-degree polynomials, size of pk roughly varies $\sim O(mn^d)$. The best balance between desired toughness of the problem and manageable key size is achieved for $d = 2$, i.e, Quadratic. The case of $d = 1$ is called system of linear equations. It has an efficient classical as well as quantum solution. [3]

3-pass IDS of Sakumoto, Shirai and Hiwatari (2011) [4]

It is a challenge-response based interactive identification protocol. We will discuss its three components namely Key-Gen, Identification and Verification.

Key-Gen: It takes a security parameter κ (say) and generate $\mathbf{F} \in_R \mathbf{MQ}(m, n, \mathbb{F}_q)$, which is m-tuple of random multivariate quadratic polynomials. Then a random vector $s \in_R \mathbb{F}_q^n$ is used to generate v such that $v = \mathbf{F}(s)$. Finally the output of Key-Gen as $(pk,\ sk) = (v,\ s)$.

[1] Bernstein, Buchmann and Dahmen, Post Quantum Cryptography (Springer), Chapter-8
[2] https://csrc.nist.gov/CSRC/media/Presentations/mqdss-round-2-presentation/images-media/mqdss-hulsing.pdf
[3] https://arxiv.org/abs/0811.3171
[4] https://www.iacr.org/archive/crypto2011/68410703/68410703.pdf

Identification: The prover $\mathcal{P}$ has to prove she has secret s without revealing it to verifier $\mathcal{V}$. Since interactive proofs employs cut-and-choose protocol where a prover first divides her secret into shares and then proves the correctness of some shares depending on the choice of a verifier without revealing the secret itself. The dividing technique employed here is based on bilinearity of polar form of MQ function. The polar form $\mathbf{G}$ and MQ function $\mathbf{F}$ satisfy the relation $\mathbf{G}(x_1, x_2) = \mathbf{F}(x_1 + x_2) - \mathbf{F}(x_1) - \mathbf{F}(x_2)$.

• The interaction starts when $\mathcal{P}$ makes a commitment ct. For ct, she pick $\mathbf{a}_0, \mathbf{b}_0 \in_R \mathbb{F}_q^n$ and $\mathbf{c}_0 \in_R \mathbb{F}_q^m$. Now she embed secret s as $\mathbf{a}_1 = \mathbf{s} - \mathbf{a}_0$ and $\mathbf{b}_1 = \mathbf{a}_0 - \mathbf{b}_0$. These parameters are used to create $\mathbf{c}_1 = \mathbf{F}(\mathbf{a}_0) - \mathbf{c}_0$, $\mathbf{ct}_0 = H(\mathbf{a}_1 \parallel \mathbf{G}(\mathbf{b}_0, \mathbf{a}_1) + \mathbf{c}_0)$, $\mathbf{ct}_1 = H(\mathbf{b}_0 \parallel \mathbf{c}_0)$ and $\mathbf{ct}_2 = H(\mathbf{b}_1 \parallel \mathbf{c}_1)$. Finally, $\mathbf{ct} = (ct_0, ct_1, ct_2)$ is send to $\mathcal{V}$ as commitment. Here, H is a collision resistant hash function (CRHF). In principle, we can use any function instead of H. But that function should have $statistically\ hiding$ as well as $computationally\ binding$ property.

• After $\mathcal{V}$ receives **ct**, he sends a random challenge **ch** $\in_R \{0, 1, 2\}$.

• Based on received **ch**, $\mathcal{P}$ sends response rs. If $ch = 0$, $rs = (\mathbf{a}_0, \mathbf{b}_1, \mathbf{c}_1)$. If $ch = 1$, $rs = (\mathbf{a}_1, \mathbf{b}_1, \mathbf{c}_1)$. If $ch = 2$, $rs = (\mathbf{a}_1, \mathbf{b}_0, \mathbf{c}_0)$.

Verification: Now, $\mathcal{V}$ checks the response rs as per below criteria, and then accordingly declare if verification is successful or not. Failure at any single stage is taken as failure of the verification process.

• If ch =0, parse rs as $(\mathbf{a}_0, \mathbf{b}_1, \mathbf{c}_1)$ and check, if $ct_1 \stackrel{?}{=} H(\mathbf{a}_0 - \mathbf{b}_1 \parallel \mathbf{F}(\mathbf{a}_0) - \mathbf{c}_0)$, and $ct_2 = H(\mathbf{b}_1 \parallel \mathbf{c}_1)$.

• If ch =1, parse rs as $(\mathbf{a}_1, \mathbf{b}_1, \mathbf{c}_1)$ and check, if $ct_0 \stackrel{?}{=} H(\mathbf{a}_1 \parallel \mathbf{v} - \mathbf{F}(\mathbf{a}_1) - \mathbf{G}(\mathbf{b}_1, \mathbf{a}_1) - \mathbf{c}_0)$, and $ct_2 = H(\mathbf{b}_1 \parallel \mathbf{c}_1)$.

• If ch =2, parse rs as $(\mathbf{a}_1, \mathbf{b}_0, \mathbf{c}_0)$ and check, if $ct_0 \stackrel{?}{=} H(\mathbf{a} \parallel \mathbf{G}(\mathbf{b}_0), \mathbf{a}_1) + \mathbf{c}_0$, and $ct_1 = H(\mathbf{b}_0 \parallel \mathbf{c}_0)$.

Security analysis: We expect $zero\ knowledge$ and $proof\ of\ knowledge$ from a secure Identification scheme. MQ-IDS meets both of the requirement as analysed below.

$\star$ $\underline{Honest\ verifier\ zero\ knowledge}$ (HVZK): The proof strategy is based on the logic that if two data-set have similar distribution, then the statistical insight we get from one data-set can be inferred from the other data-set as well. Hence, owing one of the data-set provides no more advantage over the other data-set. To be more precise, Let the data harvested by an adversary $\mathcal{A}$ by participating in the identification process be
$\mathcal{A} := \{(pk, sk, \pi) : (pk, sk) \leftarrow Key - Gen; \pi \leftarrow Trans(\langle P(pk, sk), V(pk)\rangle)\}$
And, a simulator generated, $\mathcal{S} := \{(pk, sk, \pi) : (pk, sk) \leftarrow Key - Gen; \pi \leftarrow Simulate(pk)\}$
If $\mathcal{A}$ and $\mathcal{S}$ are indistinguishable distribution, then we can say no statistically significant knowledge can be gained by participating in the protocol. Now we will see such $\mathcal{S}$ is possible to construct as follow:
If we look closely at the challenge-response protocol, we can make three interesting statistical observation. First, the challenge $\mathbf{a}_0, \mathbf{b}_0, \mathbf{c}_0, \mathbf{a}_1$ and $\mathbf{b}_1$ are random. Only $\mathbf{a}_1$ is influenced by secret key $\mathbf{s}$. Second, challenge ch is picked and send randomly from $\{0, 1, 2\}$, and the three possible response rs collectively have uniform distribution of its components $\mathbf{a}_i, \mathbf{b}_i$ and $\mathbf{c}_i$. Third, the H function acts like a random function with hiding property. The simulator generates a proof as:

• It chooses a $ch \in_R \{0, 1, 2\}$ and a fake secret **s'** $\in \mathbb{F}_q^n$.

• if $ch \in \{0, 2\}$ then $ct = (ct_0, ct_1, ct_2)$ and response rs are generated exactly like the protocol employing **s'**.

• If $ch = 1$, then rs and ct_2 are generated as in protocol but ct_0 is computed as done during verification as $ct_0 = H(\mathbf{a}_1 \parallel \mathbf{v} - \mathbf{F}(\mathbf{a}_1) - \mathbf{G}(\mathbf{b}_1, \mathbf{a}_1) - \mathbf{c}_0)$.

Here, H acting as a random function is making sure that ct doesn't convey any information of **s'** and corresponding response rs. It implies there is no efficient way to distinguish the original proof and the one simulated by the above method. Hence, $zero\ knowledge$ is guaranteed.

$\star$ $\underline{Proof\ of\ knowledge}$ (via 3-special Soundness): The idea stems from soundness of a proof system in Logic. For an IDS = (Key-Gen, P, V), it is said to be sound with knowledge error k if all probabilistic polynomial time (PPT) adversary $\mathcal{A}$ we have: $P[b = 1 : (pk, sk) \leftarrow Key - Gen; b \leftarrow \langle \mathcal{A}(pk, \perp); V(pk)\rangle] \leq k + \epsilon\ ;\ \epsilon \to 0$. In other words, if a person possess the secret key then the verifier always output success ($b = 1$), otherwise the maximum probability of success without knowing s is upper bounded by $k + \epsilon$.

• 3-pass MQ-IDS satisfies 3-special soundness property. It means if we somehow harvest three $special$ valid transcript then we can build an extractor that can efficiently compute witness sk. Three special transcripts are: $\pi = (ct, ch, rs)$, $\pi^{'} = (ct, ch', rs')$ and $\pi^{"} = (ct, ch", rs")$ with $ch \neq ch' \neq ch"$.

• The recovery of witness sk from valid transcripts: Start with transcripts $(ct, 0, rs_0)$; $(ct, 1, rs_1)$; and $(ct, 2, rs_2)$. Where $ct = (ct_0, ct_1, ct_2)$, $rs_0 = (a_0^{(0)}, b_1^{(0)}, c_1^{(0)})$, $(a_1^{(1)}, b_1^{(1)}, c_1^{(1)})$, and $(a_1^{(2)}, b_0^{(2)}, c_0^{(0)})$. Being a valid transcript it must satisfy the corresponding verification equations. We compare the case of $ch = 1$ and $ch = 2$, we see ct_0 is part of both of them. Hence, they must be equal. Comparing them yields,

$ct_0 = H(\mathbf{a}_1^{(1)} \parallel \mathbf{v} - \mathbf{F}(\mathbf{a}_1^{(1)}) - \mathbf{G}(\mathbf{b}_1^{(1)}, \mathbf{a}_1^{(1)}) - \mathbf{c}_1^{(1)}) = H(\mathbf{a}^{(2)_1} \parallel \mathbf{G}(\mathbf{b}_0^{(1)}), \mathbf{a}_1^{(1)}) + \mathbf{c}_0^{(1)}$.
It implies $a_1^{(1)} = a_1^{(2)}$ and $v = F(a_1^{(1)}) + G(b_1^{(1)}, a_1^{(1)}) + c_1^{(1)} + G(b_0^{(1)}, a_1^{(1)}) + c_0^{(2)}$
If we do the above analysis for other two cases of $(ch = 0, ch = 2)$ and $(ch = 0, ch = 1)$, we get more relationship between the parameters. After suitable substitution and elimination within the set of relations, we get $v = F(a_0^{(0)} + a_1^{(1)})$. The result simply says $a_0^{(0)} + a_1^{(1)}$ is a preimage of v. Hence, it is a witness.[QED]
• Argument for proof of knowledge: We assume MQ problem is intractable. Suppose $\mathcal{A}$ can impersonate with significantly better probability than $k = 2/3$. If this is practical, then we can rewind $\mathcal{A}$ on the same commitment ct but three different challenges ch = 0, ch = 1 and ch = 2. This generates three valid transcript (ct, 0, rs0); (ct, 1, rs1); and (ct, 2, rs2). These transcript forms the special set of transcripts that can help to extract a witness sk using the method mentioned above. This amounts to solving MQ problem. Hence, this is a proof of knowledge.

Knowledge Error: If an adversary $\mathcal{A}$ initiate the protocol with a fake $s' \neq s$, she can successfully impersonate as $\mathcal{V}$ (who possess secret s) with some probability. A careful look at the verification process reveals that the case $ch = 1$ is the only one where the knowledge of secret s is indispensable for successful verification, as $v = \mathbf{F}(s)$ is used there. Hence, $\mathcal{A}$ can successfully impersonate if $ch = \{0, 2\}$. This amounts knowledge error to be $k = \frac{2}{3}$. We want knowledge error to be negligibly close to 0. But our k is comparatively big. In such cases, it is required to run the IDS r-times in parallel. This causes the knowledge error $k^r \to 0$ for $r \gg 1$.

Digital signature from 3-pass IDS: Fiat-Shamir transformation (FST) changes a interactive proof of knowledge into a digital signature. The same technique is used to create MQ-DSS as below:
• Key-Gen: Signer runs key generation algorithm to create (sk, pk). $sk = \hat{x}$ and corresponding $pk = (\mathbf{F}, v)$. Details of the Hash function used in FST and number of round r of IDS-run are also made public.
• Signing: To sign a message **M**, generate commitment ct for r-rounds of IDS. Now, Hash function do the job of random challenge ch that we expect from verifier $\mathcal{V}$, i.e, $ch = H(M, ct, pk)$. Now signer generate response rs as per IDS protocol. Finally we have signature on M as $\sigma = \langle ct, rs \rangle$.
• Verification: We compute $ch = H(M, ct, rs)$, and check if the tuple (ct, ch, rs) forms a valid transcript or not.
$\star$ $\underline{Security\ argument\ for\ MQ - DSS}$: The security of MQ-DSS stems from the security of underlying IDS and the Hash function used in FST. Any attempt to an efficient scheme for forgery would either be possible if he find a valid transcript or the he know efficient way to find collision in the hash function. Another case would be H is not behaving like a random function. We have discussed that the finding a valid transcript is equivalent to solving MQ problem. And, we have assumed that the utilised Hash function is collision resistant(CHRF) and acts as random function. Thus MQ-DSS is secure.[QED]

Exercise: 5-Pass IDS (Knowledge Error, and its Pros and Cons)[5]

A 5-pass IDS is another interactive identification protocol. The main different between 3-pass and 5-pass the length of transcript generated. In 3-pass three layers of interaction generate (ct, ch, rs) while in 5-pass we have 5 layers of interaction (ct, ch_1, rs_1, ch_2, rs_2). Here ct is initial commitment by $\mathcal{P}$, and $ch_1 \in_R \mathbb{F}_q$ is the first challenge from $\mathcal{V}$. Based on ch_1, $\mathcal{P}$ sends his first response rs_1. Then $\mathcal{V}$ sends his second challenge $ch_2 \in_R \{0, 1\}$. Finally $\mathcal{P}$ sends rs_2 which is evaluated by $\mathcal{V}$ and he accordingly declare the verification result. [See Fig.1 below]
Knowledge error k : We need to inspect the verification protocol closely to get the idea of k. Verification takes place when $\mathcal{P}$ sends $rs_2 \in_R \{0, 1\}$ to $\mathcal{V}$.
• If $ch_2 = 0$, parse $rs_2 = r_0$, and check $ct_0 \stackrel{?}{=} H(r_0, ch_1 r_0 - t_1, ch_1 \mathbf{F}(r_0) - e_1)$
• If $ch_2 = 1$, parse $rs_2 = r_1$, and check $ct_1 \stackrel{?}{=} H(r_1, ch_1(v - \mathbf{F}(r_1)) - \mathbf{G}(t_1, r_1) - e_1)$
We can see the knowledge of secret key s is indispensable only when $ch_2 = 1$ as $v = \mathbf{F}(s)$ is involved here. Hence, we already have success probability of $\frac{1}{2}$ using fake secret s' for the case of $ch_2 = 0$. We can improve this further by guessing in advance what could $\mathcal{V}$ send as $ct_1 \in_R \mathbb{F}_q$. This guess has success chance equal to $\frac{1}{q}$, where q is the cardinality of the field $\mathbb{F}_q$. Why this guessing would help in verification goes as follow: With a correct guess of what challenge ct_1 would $\mathcal{V}$ send, he design his initial commitment in such a way that the requirement of secret sk is no more essential. A honest $\mathcal{V}$ always compute commitment $ct_1 H(r_1, G(t_0 r_1) + e_0)$, but the $\mathcal{A}$ will use guess technique and compute ct_1 in his favour as $ct_1 = H(r_1, ch_1(v - \mathbf{F}(r_1)) - \mathbf{G}(t_1, r_1) - ct_1 F(r_0) + e_0)$ and $e_1 = ch_1 F(r_0) - e_0$. We can see the Fig.1 and correlate that it lead to successful verification.

[5]https://eprint.iacr.org/2016/708.pdf

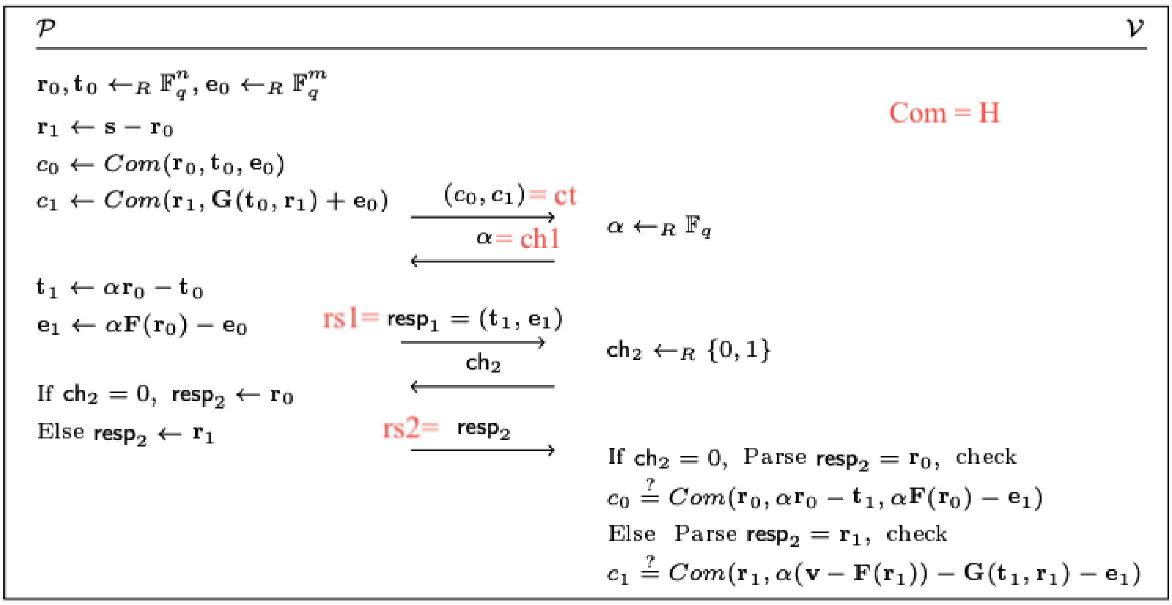

Fig. 1. Sakumoto et al. 5-pass IDS

This finally implies knowledge error $k = \frac{1}{2}[1] + \frac{1}{2}[\frac{1}{q}] = \frac{1}{2} + \frac{1}{2q}$. This implies 5-pass has lesser knowledge error k if $q \geq 4$.

Pros and Cons in signature construction from 5-pass IDS: We will take 3-pass based MQ-DSS as benchmark to compare 5-pass IDS based MQ-DSS.

• Knowledge error k criteria: 3-pass has fixed knowledge error of $\frac{2}{3}$. While 5-pass has $k = \frac{1}{2} + \frac{1}{2q}$. Hence, it can do better if $q \geq 4$. Since on running r rounds, knowledge error goes like k^r. Hence, small k implies comparatively smaller r required to achieve negligible error. Also the value of r directly proportional to the size of signature generated. Hence, 5-pass has an advantage over 3-pass. It is important to note that we can't make q arbitrary large as it will increase the transcript size per round too. The optimal q is achieved for $\mathbb{F}_q = \mathbb{F}_{31}$. [6]

• Extractor for computing witness: We have seen three accepted special transcript is sufficient for getting a witness for 3-pass IDS. But, 5-pass IDS requires five such transcripts to extract a witness. Hence, another layer of complexity is added for adversary $\mathcal{A}$.

Acknowledgement

• Most of the idea is taken from the classroom lecture slides.
• Fig.1 is taken from the paper Schwabe et.al(2016).
• A few ideas from Post-Quantum Cryptography 2017 Summer School on Post-Quantum Cryptography 2017.
• This document is pdf version of the LATEX code written on Overleaf.

THANK YOU

[6] https://eprint.iacr.org/2016/708.pdf ,Page-21