Dear all,

TL;DR: The security proof of HiMQ-3 (Theorem 4) is flawed.

The HiMQ-3 submission document claims that the HiMQ-3 signature scheme is EUF-CMA secure provided that it is hard to find a solution for a system of quadratic equations in the HiMQ-3 family. In other words, the claim is that if the scheme is UF-KOA secure (universal forgery under key-only attack), then the scheme is also EUF-CMA secure.

The proof of this claim is to be found in [1] (Theorem 4.1), where the same claim is made for the ELSA signature scheme. The proof is very similar to the classic proof of [2] for the security of a hash-and-sign signature scheme based on a trapdoor permutation. However, the trapdoor function used by the HiMQ-3 scheme is not a permutation, and this causes the proof to fail.

The proof programs a random oracle by sampling random x, and returning P(x), where P is the public key. In the trapdoor permutation setting this is a valid approach, because there is no way to distinguish (x,P(x)) from (P^{-1}(y),y), for x and y uniformly distributed variables on the domain and codomain of P respectively. When P is no longer a permutation (as is the case for HiMQ-3 and ELSA) this might no longer be the case. (In fact, P^{-1}(y) is not even uniquely defined) This means that the adversary is no longer guaranteed to function correctly in the simulated environment and that the proof fails.

Kind regards,
Ward


EUF-CMA security on multivariate signature schemes was discussed in [3]. There, it is described how to modify the signature scheme to achieve EUF-CMA in the random oracle model. Likewise, it seems that HiMQ-3 may also achieve EUF-CMA.

Kind regards,
Ryo


2018年5月3日木曜日 5時42分28秒 UTC+9 Ward Beullens:

Dear all,

TL;DR: The security proof of HiMQ-3 (Theorem 4) is flawed.

The HiMQ-3 submission document claims that the HiMQ-3 signature scheme is EUF-CMA secure provided that it is hard to find a solution for a system of quadratic equations in the HiMQ-3 family. In other words, the claim is that if the scheme is UF-KOA secure (universal forgery under key-only attack), then the scheme is also EUF-CMA secure.

The proof of this claim is to be found in [1] (Theorem 4.1), where the same claim is made for the ELSA signature scheme. The proof is very similar to the classic proof of [2] for the security of a hash-and-sign signature scheme based on a trapdoor permutation. However, the trapdoor function used by the HiMQ-3 scheme is not a permutation, and this causes the proof to fail.

The proof programs a random oracle by sampling random x, and returning P(x), where P is the public key. In the trapdoor permutation setting this is a valid approach, because there is no way to distinguish (x,P(x)) from (P^{-1}(y),y), for x and y uniformly distributed variables on the domain and codomain of P respectively. When P is no longer a permutation (as is the case for HiMQ-3 and ELSA) this might no longer be the case. (In fact, P^{-1}(y) is not even uniquely defined) This means that the adversary is no longer guaranteed to function correctly in the simulated environment and that the proof fails.

Kind regards,
Ward


Dear HiMQ-3 team,

There seem to be contradictions between the description of the third layer of the central map and the matrices presented in the analysis of known attacks (figure 2 of the supporting documentation).

In the description, it is written that the polynomials of the third layer of the central map are of the form:

\[ f(x) = \sum_{i,j} \beta_{i,j} x_i x_j + \theta(x) + \theta'(x) + \epsilon x_{v1+o2+k} \]

where the \( i,j \) in the sum are between \( v+1 \) and \( v1 \).

For the definition of \( \theta \) and \( \theta' \), it is written that the coefficients are such that symmetric matrix associated to the quadratic part of \( f \) has full rank, which implies that the quadratic part of \( f \) involves all \( n \) variables.

However, not all variables appear in \( f \). For the \( k \)-th polynomial of the third layer, \( x_{v+1}, \ldots, x_{v1} \) appear in the sum, \( x_{v1+1}, \ldots, x_{v2} \) appear in \( \theta \) (assuming the modulo is \( o2 \) and that 1 is added to the result) and \( x_{v2+1}, \ldots, x_n \) appear in \( \theta' \) (assuming again that 1 is added to the subscript). All the other variables, save for \( x_k \) that appears in \( \theta \) and \( \theta' \), are not in \( f \).

Moreover, with the definition of the third layer given in the description of the central map, we get matrices with non-zero coefficients only in the square corresponding to the sum and on line \( k \) and column \( k \) resulting from \( \theta \) and \( \theta' \) (\( x_k \) appear in the products \( x_k x_i \) for several different \( i \) between \( v1+1 \) and \( n \)).

(the theoretical secret key size provided also suggests that they are more coefficients that the one given in the description),

Sincerely,

A-E. Louisy,

Student in cryptography at Versailles University
Dear A-E. Louisy,

Thank you for your comments.

There is a typo. The current formulas

\[ \Theta_i(x) = \sum_{j=1}^{v_1} \gamma_{i,j} x_{v_1+(i+j-1) \pmod o_3}, \]
\[ \Theta_i'(x) = \sum_{j=1}^{v_2} \gamma_{i,j}' x_{v_2+(i+j-1) \pmod o_3}, \]

should be changed to

\[ \Theta_i(x) = \sum_{j=1}^{v_1} \gamma_{i,j} x_{v_1+i \pmod 2}, \]
\[ \Theta_i'(x) = \sum_{j=1}^{v_2} \gamma_{i,j}' x_{v_2+i \pmod 3}, \]

Note that $1 \leq A \pmod B \leq B$ for an integer $A$ and a positive integer $B$,

in our definition.

Kind regards

Kyung-Ah Shim

******************************************************************************

Answer to Our Security Proof.

Due to the use of the multivariate quadratic map requiring additional random Vinegar variables, our trapdoor function is not permutation and the signature distribution is not uniformly distributed as presented in [1]. The authors [1] make the distribution of signatures uniform by using a random salt to the message being hashed and re-choosing a random salt instead of Vinegar variables.

We can use the same way to prove unforgeability of our scheme. For it, we need to propose a modified version: the modified signing algorithm is the same as the original one except that

-choose a random $r \in \{0, 1\}^R$, compute $H(m, r)=h$.

-If one of the linear systems has no signature then choose another random $r'$ and try again.
Then the signature is \((\tau, r)\).

In Verify algorithm, to verify a signature \((\tau, r)\) on a message \(m\), check whether the equation \(\cal P(\tau) = H(m, r)\) holds or not.

In the security proof, the H-query should be changed as:

For \(H\)-queries, the tuples in \(H\)-list are of the form \((m_i, c_i, \tau_i, r_i, P(\tau_i))\). When \(\cal A\) queries \(H\) at \(m_i \in \{0, 1\}^*\),

i) If the query already appears on \(H\)-list in a tuple \((m_i, c_i, \tau_i, r_i, P(\tau_i))\) then \(\cal B\) returns \(H(m_i, r_i) = P(\tau_i)\).

ii) Otherwise, \(\cal B\) picks a random coin \(c_i \in \{0, 1\}\) with \(\Pr[c_i = 0] = \frac{1}{q_S + 1}\).

- If \(c_i = 1\) then \(\cal B\) chooses a random \(\tau_i \in \mathbb{F}_q^n\) and \(r_i \in \{0, 1\}^R\), adds a tuple \((m_i, c_i, \tau_i, r_i, P(\tau_i))\) to \(H\)-list and returns \(H(m_i, r_i) = P(\tau_i)\).

- If \(c_i = 0\) then \(\cal B\) adds \((m_i, c_i, r^*, *, \eta)\) to \(H\)-list from the instance and returns \(H(m_i, r^*) = \eta\), where \(\eta\) is the given MQ-instance.

For Sign Queries. When \(\cal A\) makes a Sign-query on \(m_i\), \(\cal B\) finds the corresponding tuple \((m_i, c_i, \tau_i, r_i, P(\tau_i))\) from \(H\)-list.

- If \(c_i = 1\) then \(\cal B\) responds with \((\tau_i, r_i)\).

- If \(c_i = 0\) then \(\cal B\) reports failure and terminates.

Then the distribution of the outputs \(H(m_i, r_i)\) of our random oracle is identical to the distribution of \(\cal P(\tau), \tau \in \mathbb{F}_q^n\), since \(\tau\) is uniformly distributed over \(\mathbb{F}_q^n\) and it is a valid signature satisfying \(\cal P(\tau) = H(m, r)\).

The rest of the proof is the same as that in [2].
