Slide 1:

Hallo, everyone, I am Han Zhao, thank you for waiting me for a long time. Now, it’s my turn and my topic is “Post-quantum secure PUF authentication using LPN”.

Slide 2:

Now, I will go into this topic in these following parts. Firstly, I want to describe the introduction of this topic, it contains the concept and the motivation of the theme. And then, I will introduce the construction of this authentication system in detail, it refers to two stages, namely the enrollment phase and the authentication phase. After that, I will give a summary and show you my schedule roughly. Finally, if you have questions or suggestions about this topic, maybe we can discuss it together.

Slide 4:

Firstly, I introduce the concept of LPN, Learning Parity with Noise. This problem can be described as mathematical type as followed, selecting a secret s in Galois filed 2 randomly with the length of n then selecting a string A in Galois filed 2 randomly as well, the length of A is also n, after that, selecting a bit random noise and it follows Bernoulli distribution with the parameter epsilon. That means, the probability that the noise is equal to one is epsilon. Finally, the output b is equal to the product of A and s plus noise. This is only one sample, if we collect m samples, then the dimension of matrix A is m\*n and the length of error is m, but the length of secret is always n and normally n is less than m. So, at last, the goal of this problem is to extract s given public known matrix A and the outputs b.

Slide 5:

Ok, then, the motivation or the background of my topic, the first point is based on the fundamental in theory, the LPN problem is equivalent to the problem of decoding random linear code in coding theory. Until now, there is no algorithm that can solve s in polynomial time, so this problem can be considered hard.

Slide 6:

I selected 3 different algorithms for solving LPN problem, the result shows that the computational complexity of the LPN problem depends on the length of the secret n and the noise distribution. The larger n is, the harder the problem is.

Slide 7:

On the other hand, the second is the main point, in practice there are many cryptographic applications based on LPN, such as user authentication, in the past, strong PUF were used for authentication, their inputs are very long, and the corresponded outputs are very short. Its security is based on information theoretical complexity, and nowadays it’s not secure anymore, because some of its outputs are repetitive, so it can be analyzed out by machine learning. However, comparing to strong PUF-based authentication LPN-based authentication is more secure, so far there is no quantum computer that can calculate the LPN problem in polynomial time, that means, LPN problem has no known quantum attacks, so it’s suitable for post-quantum authentication-system.

Slide 8:

After the introduction of the background, now I go into the main part of the topic, the construction of the authentication system, it refers to two phases, enrollment phase and authentication phase.

Slide 9:

In the first stage, there are an encoding-module and Hash module, the Challenge-Response-Pairs are collected and stored in database. In the second stage, there are a decoding module, the extracted Challenge-Response-Pairs are matched with the reference in the database. If they are identical, then the authentication succeeds, otherwise, it fails.

Slide 10:

It’s the module of the enrollment phase. This part is encoding module, and this Hash module. Firstly, let us see the input of the encoding module, TRNG stands for True Random Number Generator, it’s used for producing random number, namely the secret, 0 or 1, binary number. And then, POK, namely the weak PUF, produce the random noise e, r\_noise is from the extra environment, the both are added into the encoding module as noise, matrix A is also random, and b is the output of the encoding module, now we define that the hash function of s is the response and b and the hash function of s,b is challenge. In this process, POK is used as one-time-pad and s is also random, so the message b is unique. This method is different from the traditional encryption method, there is no encryption-module, the traditional method, such as AES and RSA, they adopt encryption algorithm to exchange information, the ciphertext is usually short. But here the text can be longer, namely the value of m. So, this structure is more secure than the traditional.

Slide 11:

In the authentication stage, the server sends the identical challenge b and the hash function of s,b, if the device can return the corresponded response, then the authentication succeeds. During this process, before returning the hash function of s, the hash function of s,b should be matched firstly in case that one bit of b is changed by the adversary and this bit doesn’t influence the extracting of correct s. So, this step is very important and it can improve the security of this system. In the decoding module, given public known matrix A and b, the goal is to extract correct s, e-prime is the POK output in later time and it’s different from the former, this is so-called LPN problem.

Slide 12:

For the device, how to extract s, there are two methods, Gaussian elimination and error correction algorithm. Gaussian elimination is only suitable for solving linear equations, obviously, in this system this is not linear, unless we can find the stable POK bits, but the workload is very large, in hardware using Gaussian elimination algorithm is also very complex and can’t guarantee the accuracy of s, so I adopt error correction algorithm. As for the security analysis, maybe error correction code can cause security reduction, this is possible, but there is POK output, it’s random, so I think it’s secure and a trade-off decision must be made between the security and the accuracy.

Slide 13:

Which kind of codes can be used? There are several candidates, such as LDPC, BCH and so on, among them the parity check matrix of LDPC code is sparse, it’s useful for implementation in hardware, but its generator isn’t sparse, that means, the encoding process is very complex, as for Hamming code, the number of error correction is limited, there are about 10 to 15 percent errors, so I select Reed Muller code.

Slide 14:

Firstly, the generator matrix of Reed Muller code is very simple to construct and in decoding module, there can be no parity-check-matrix to correct error, that means, in hardware there is no need for the extra storage of this matrix. Besides, it has better error correction ability, especially for long code.

Slide 15:

The Plotkin-Construction method can be used to generate Reed Muller code, Reed Muller code can be made from two subcodes, namely u and v, u belongs to Reed Muller r, m-1, and v belongs to Reed Muller r-1, m-1. This is a relationship between the parameters r, m and the length of codeword n and the length of information bits. The Hamming distance of the codeword is equal to 2 square m minus r, so the long code has better error correction property.

Slide 16:

This is a tree of Plotkin-Construction.

Slide 18:

Slide 22:

Trapdoor：

Trapdoor is defined as c, c is the confidence information, the actual differential count between two ring oscillators. e is the sign of c, the larger c is, the more stable e is.

Hash function:

Hash function is a function that can be used to map data of arbitrary size to data of fixed size. It has the properties: One-wayness and Non-malleability. That means, even though somebody knows H(s), he cannot calculate s or according to b and H(s,b) he cannot calculate s. It can improve the security of the system.

Strong PUF vs Weak PUF:

Strong PUF: normally, strong PUF are used for authentication, their inputs are very long, and the corresponded output is very short. That means, the input and the output of the authentication are not same. Its security is based on information theoretical complexity, but nowadays it is not secure any more, because some of its outputs are repetitive so it can be analyzed by machine learning.

However,

Weak PUF: it is used to generate the symmetric key, the length of the weak PUF’s input and output are the same, that means, the weak PUF outputs can be used as intended error to produce one-time-pad. Combined with LPN, the security of the authentication system can increase quitely. Because LPN is based on computational complexity and Hardware. Until now

LPN has no known quantum attacks, so this makes it a suitable choice for the post-quantum authentication-system. Many researchers have gone into the solution of LPN for many years, however, LPN is still a NP problem, NP is the set of problems solvable by a nondeterministic algorithm whose running time is bounded by a polynomial in the length of the input.

Cryptosystem VS Authentication-system:

The traditional cryptosystem, such as AES and RSA, they adopt encryption algorithm to exchange information. The ciphertext is usually short. However, in this authentication system there is no encryption algorithm, the ciphertext is very long and only for once use, the weak PUF outputs are random and can’t be predicted, as well as that the secret is random.

Post-quantum cryptography refers to cryptographic algorithms (usually public-key algorithms) that are thought to be secure against an attack by a quantum computer.

Post-quantum cryptography is distinct from [quantum cryptography](https://en.wikipedia.org/wiki/Quantum_cryptography), which refers to using quantum phenomena to achieve secrecy and detect eavesdropping.

Notation:

A is uniformly public-known random matrix\\

s is chosen uniformly at random with length n\\

e is the added noise \\

Trapdoor：

Trapdoor is defined as c, c is the confidence information, the actual differential count between two ring oscillators. e is the sign of c, the larger c is, the more stable e is.

Hash function:

Hash function is a function that can be used to map data of arbitrary size to data of fixed size. It has the properties: One-wayness and Non-malleability. That means, even though somebody knows H(s), he cannot calculate s or according to b and H(s,b) he cannot calculate s. It can improve the security of the system.

How to extract correct s according to the challenge, namely the helper data and the public known matrix A? Firstly, the Gaussian elimination algorithm is recommended, but this algorithm is only suitable for solving linear equations. In this system, the intended error contains the POK output and the noise from the extra environment, so if we use Gaussian elimination algorithm here, the probability of extracting incorrect s is very high, unless we can find the stable POK bits, but this workload is very large, it seems impractical. Besides, the computation complexity of Gaussian is also very high, so we adopt error correction algorithm in the decoding module to check the errors and correct them. Then the accuracy of extracting correct s can increase.

Error correction codes:

We considered several kinds of error correction codes, such as Hamming code, LDPC code and so on. Among them, we decided to select Reed Muller code. There are several reasons. The construction of Reed Muller code is easy and there is no parity check matrix, only generator matrix. That means, in the decoding module, if I use the other code, such as LDPC code, the parity check code is sparse, it is very useful for decoding, but its generator matrix is not sparse, it can cause that its encoding is very complex. And generally, it is used for thousand long code.

Finite-state machine, FSM