Post-Quantum Digital Signatures with Attenuated Pulse Generator

Maksim Iavich[a], Razvan Bocu[b], Arturo Arakelian[c] and Giorgi Iashvili[a]

[a]School of technology Caucasus University Tbilisi, Georgia
[b]Department of Mathematics and Computer Science Transilvania University of Brasov, Brasov, Romania
[c]Informational technologies dept.University of Georgia Tbilisi, Georgia

Abstract

Quantum computations cause problems for classical systems. The perfect examples are effective quantum algorithms, which are causing problems for the most popular cryptosystems. These systems are considered safe for classical computers. Despite on releasing quantum computers all sensitive information should stay safe. This information should be encrypted in such way that will withstand quantum computers' attacks. Classical cryptography consists of problems and instruments: encryption, key distribution, digital signatures, pseudo-random number generator and one-way functions. For example, "RSA" is safe only when factorization is hard for classical computer, but quantum computers can easily solve this problem. Hash-based signature systems provide interesting alternatives. Hash-based signatures systems use cryptographically secure hash functions. Their security is based on the security of the concrete hash function. Using secure hash function is the minimal requirement for safe digital signature's system, which can be used for signing many documents using one secret key. In the article we provide the improved hash based digital signature scheme. The scheme uses the secure pseudo random number generator. As the seed for this pseudo random number generator it uses the truly random number received using attenuated pulse quantum generator. The security and the efficiency of the scheme is evaluated.

Keywords

cryptography, post-quantum, digital signatures, generation, pulse generator

1. Introduction

Nowadays digital signatures become main security technology for the internet and for other IT infrastructures. Digital signatures are widely used for identification and authentication in protocols. That's why safe signature algorithms are very important [1, 2]. It is rather very hard to make the new crypto-system. The goal is to create such system, which would satisfy all safety standards. During creating a new system we can only assume the computational resource. Supposedly the attacker uses classical computer with limited generation time. But, if he uses a quantum computer, then we have to identify which crypto systems are secure. Quantum computers use not the same computational methods. Some submodules like quantum "Fourier's" transformation will act faster on quantum computer than on classical computer. In cryptography public-key is used for securing transactions, its safety is based on the couple hard theories. Quantum computers can break these theories. Couple examples of such theories are: factorization and the equation of "Pell". Using these algorithms means that quantum computers can hack: "RSA", "Diffie-Hellman" and "eliptic curve cryptography", which nowadays are widely in use.

"Buchmann-Williams" is also widely used and considered as safe system. Nowadays the main question is, which cryptosystem will be safe against quantum computes' attacks. Nowadays signature algorithms that are used in practice are: RSA, DSA and ECDSA. They are not protected from quantum computers, because of they are based on factorization's algorithm. Hash-based signature systems provide interesting alternatives. As well as other signature systems, hash-based signatures systems are using cryptographical hash functions. Their safety is based on concrete hash function's safety. Using safe hash function is the minimal requirement for safe digital signature's system, which can be used for signing many documents using one secret key.

2. Hash-Based Signature Systems

The template Hash-based signatures schemes were presented by Ralph Merkle. First of all, he offered "Lamport and Diffie" one-time signature [3]. Despite on effectiveness of this signature scheme, size of signature is long. The one-time signature of "Winternitz" was offered. The main idea of this scheme is to sign

IVUS 2020: Information Society and University Studies, 23 April 2020, KTU Santaka Valley, Kaunas, Lithuania
miavich@cu.edu.ge (M. Iavich); razvan.bocu@unitbv.ro (R. Bocu); arturo.arakelyan@gmail.com (A. Arakelian); giiashvili@cu.edu.ge (G. Iashvili)

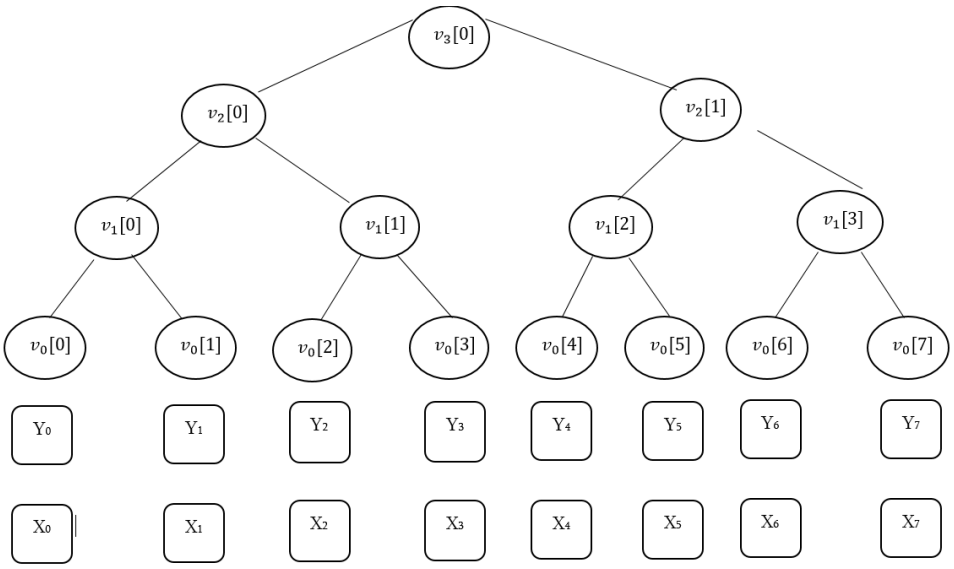

Figure 1: Mertkle Tree

multiple bits of one message simultaneously. In real life one-time signatures are not efficient, because using one key we can generate only one signature [4, 5]. In 1979 Ralph Merkle introduced solution of this issue. His idea is to use whole binary tree where public key is the root if this tree.

Merkle signature scheme (MSS) is working with any hash functions and with any one-time signature schemes For example, we have "$h : \{0,1\}^* \longrightarrow \{0,1\}^n$" which is a hash function also let's say we have some one-time signature scheme.

Signer selects H ∈ N, H ≥ 2. After that he chooses key pair, which will be generated.

Using these, he can sign 2^H documents. Signer generates 2^H times pair of one-time signatures (X_j, Y_j), $0 \leq j \leq 2^H$. X_j is signature's key and Y_j is a confirmation key.

Both of them are string of bits. The leafs of Merkle tree are

$$h(Y_j), \quad 0 \leq j \leq 2^H \tag{1}$$

Merkle tree leafs are calculating in such way: value of parent leaf is hash of concatenation of left and right children. Public key is a root of MSS. Private key of MSS is a 2^H one-time signatures' sequence:

$$\begin{gathered} k_p[j] = h(v_{h-1}[2j] || v_[h-1][2j+1]), \\ 1 \leq p \leq H, \quad 0 \leq j \leq 2^{H-h} \end{gathered} \tag{2}$$

The process is illustrated on the Figure 2.

2^H pairs of one time keys must be calculated to get the public key. Storing such a large number of keys is very not efficient in practice. The scientists are working on improving the scheme. They to integrate pseudo random number generator – PRNG into the scheme [6]. This is offered not to compute and not to save large amounts of one-time keys pairs.

CSPRNGs are pseudo random number generators, which are secure for use cryptography. A lot of PRNGs are not quantum resistant, so we offer an algorithm based on a secure hash function, as the entire the algorithm is based on it.

We offer to use HASH_DBRG, because it is rather efficient, is based also on hashing and is NIST standard. As a seed to pseudo random number generators it is recommended to use the true random number numbers [7, 8, 9]. To get the seed for this CSPRNG we offer to use the physical quantum random number generator (QRNG).

3. Attenuated Pulse Generators

Today, most of QRNGs are based on the quantum optics. Time of arrival generators are optical quantum random number generators - OQRNGs. In these generators during each measurement only one bit can be received. In order to improve the efficiency photon counting generators can be used. This type of generators are based on time effects. Attenuated pulse generators are based on a simplified version of the previ-

ous approaches with much less requirements for the detectors. Almost all the single photon detectors have rather limited photon number resolving capability and get the response by clicking or not clicking. Photon counting approaches rely on multiple clicks occurred in a long period of time. This period is divided into the smaller bins, which are concatenated. These approaches need a weak source that outputs zero or one photons in the bin and there is negligible probability to generate two or more photons in this short period of time. OQRNGs with a weak source and with the same probability of generating or not generating the photon. We need the complete system to produce a detection probability of 1/2. A superposition of the empty and single photon states in the same spatio-temporal mode can be represented so that the quantum state of our photon pulse is:

$$(|0\rangle + |1\rangle)/(2^{1/2}) \tag{3}$$

We get a 0 when the event is not detected and 1 when we get a click. The state does not have to get only one photon. The superposition:

$$1/2^{1/2}|0\rangle + \sum \alpha_k|k\rangle \quad \text{with} \quad \sum_{k=1}^{\infty} |\alpha_k|^2 = 1/2 \tag{4}$$

So we get ones from clicks and do not pay attention, if they are triggered by many or one photons. Coherent states give us this superposition and it is easy to reproduce them. The probability to get zero photons is:

$$p(n = 0) = e^{-|\alpha|^2} \tag{5}$$

The probability to find the photons is:

$$p(n \geq 1) = 1 - e^{-|\alpha|^2} \tag{6}$$

We have to find α for which $p(n = 0) = p(n \geq 1)$, which occurs for $\alpha = ln(2^{1/2})$.

The Poissonian source where

$$\lambda T = ln(2) \approx 0.693 \tag{7}$$

also gives the need probability of the detector.

Practically, the generator uses an efficient mean photon number at the detector $f\ \lambda\ T$, with an efficiency f.

This OQRNG can be balanced by fine tuning of a variable attenuator. On the other hand, the generator can follow up on the light source.

To get the random bit it can output 1 if $num_0 > 0$ and $num_1 = 0$ and outputs 0 if $num_0 = 0$ and $num_1 > 0$. Where and num_0 and num_1 are photon numbers in two detections. The outcomes with 2 successive empty pulses or two successive clicks are cancelled. We offer to use attenuated pulse generators as a seed of HASH_DBRG

4. The Offered Scheme

4.1. Key Generation

$$\text{Signer selects} \quad H \in N, H \geq 2 \tag{8}$$

After that, he chooses key pairs, which must be generated. To generate the signature keys, first of all the signer has to generate the seed. The seed must be generated using the quantum random number generator. He uses the attenuated pulse generator for this. Afterwards he uses the HASH_DBRG to generate the keys. This CSPRNG takes the seed received by attenuated pulse generator as the input and outputs the signature keys. Afterwards he generates the corresponding verification keys. So, the signer generates 2^H pairs of one-time signatures "$(X_j, Y_j), \quad 0 \leq j \leq 2^H$". "$X_j$" is signature's key and " Y_j " is a confirmation key. Using these keys he can sign 2^H documents. The keys are the strings of bits. To get the leaves of the tree, he hashes the verification keys by means of the the hash function:

$$h : \{0,1\}^* \longrightarrow \{0,1\}^n \tag{9}$$

To get the parent node, the concatenation of two previous nodes is hashed. The root of the tree is the public key of the signature - public. Merkle tree leafs are calculated in such way: value of the parent leaf is the hashed value of concatenation of left and right children. Public key is a root of MSS.

4.2. Message Signature

To sign a message, the signer hashes it and gets the hash of the n size. $h\ (m) = hash$, to sign the message, is used some one-time key X_{arb}. This key is calculated by means of CSPRNG using the same seed got from the attenuated pulse generator. Afterwards he uses the HASH_DBRG to generate the signature keys once more. This CSPRNG takes the seed received by attenuated pulse generator as the input and outputs the signature keys. The signature is a set of the one-time signature, one-time verification key, its index, and all fraternal nodes according to the selected arbitrary key with the index "arb"

$$\text{Signature} = (sig \| arb \| Y_{arb} \| auth_0, ..., \\ ..., auth_{H-2}, ..., auth_{H-1}) \tag{10}$$

4.3. Signature Verification

For the signature verification, the one-time signature is verified using the verification key, if the verification is successful, all the needed nodes are computed using "auth", index "arb" and Y_{arb}.

If the root of the tree is the same as the public key, the corresponding signature is correct

4.4. Advantages

The offered scheme does not save a large amount of one-time keys pairs in the memory. The scheme stores only the short seed of the attenuated pulse generator, which is a secure quantum random number generator, attenuated pulse generator. The advantage of using it approach is that during each measurement several random bits are received. As CSPRNG the system uses HASH_DBRG, which is NIST standard and is rather efficient, this CSPRNG uses the same secure hash function as the whole scheme does.

5. SECURITY

Algorithm of Merkle is almost the same, but the hash based CSPRNG is integrated, the seed for CSPRNG is generated using attenuated pulse generator. The offered CSPRNG is secure against the attacks of quantum computers. The seed is received using quantum random number generator, so the proposed algorithm of Merkles scheme is secure.

6. CONCLUSION

As the result, the secure digital signature scheme is got, and it is secure against quantum computers attacks. The scheme stores only the short seed of the attenuated pulse generator, which is a secure quantum random number generator.

The scheme uses HASH_DBRG as CSPRNG, this generator uses the quantum resistant hash function. This random number received by attenuated pulse generator is used for HASH_DBRG pseudo random number generator, which uses quantum secure hash function. This pseudo CSPRNG is very efficient and secure, and it is NIST standard. As a seed HASH_DBRG uses a random number received by means of attenuated pulse generator. The integration of attenuated pulse generator does not the influence on the efficiency, as it is used only for generating the seed for CSPRNG. So, in our approach the space need for storing the key is greatly reduces and the implementation speed is not effected. Our method also increases the security of the scheme and the scheme is safe against the attacks of quantum computers.

Acknowledgments

The work was conducted as a part of PHDF-19-519 and the grant financed by Caucasus University.

References

[1] G. C. Cardarilli, L. Di Nunzio, R. Fazzolari, M. Re, Tdes cryptography algorithm acceleration using a reconfigurable functional unit, in: 2014 21st IEEE International Conference on Electronics, Circuits and Systems (ICECS), IEEE, 2014, pp. 419–422.

[2] S. Battiato, G. M. Farinella, C. Napoli, G. Nicotra, S. Riccobene, Recognizing context for privacy preserving of first person vision image sequences, in: International conference on image analysis and processing, Springer, 2017, pp. 580–590.

[3] M. Ajtai, Generating hard instances of lattice problems, volume 13 of Quad. Mat., 2004, pp. 1–32. Dept. Math., Seconda Univ. Napoli, Caserta (2004). Preliminary version in STOC 1996. 8. Babai, L.: On Lovász lattice reduction and the nearest lattice point problem. Combinatorica, 6:1*13 (1986).

[4] A. Gagnidze, M. Iavich, G. Iashvili, Analysis of post-quantum cryptography use, 2017.

[5] G. M. Farinella, C. Napoli, G. Nicotra, S. Riccobene, A context-driven privacy enforcement system for autonomous media capture devices, Multimedia Tools and Applications 78 (2019) 14091–14108.

[6] J. Buchmann, L. García., E. Dahmen, M. Döring, E. Klintsevich, An improved merkle signature scheme. in: Barua r., lange t. (eds) progress in cryptology, volume 4329, 2006.

[7] A. Gagnidze, M. Iavich, G. Iashvili, Novel version of merkle cryptosystem, 2017.

[8] G. Lo Sciuto, S. Russo, C. Napoli, A cloud-based flexible solution for psychometric tests validation, administration and evaluation, in: CEUR Workshop Proceedings, volume 2468, 2019, pp. 16–21. URL: www.scopus.com, cited By :1.

[9] Langdon, B. William, Prng random numbers on gpu, 2017.