On some issues affecting the security of RSA and ECDSA digital signature schemes

Trieu Quang Phong, Do Dai Chi, Tran Duc Huy, Nguyen Ngoc Diep

***Abstract*— In this paper, we will consider 02 problems with traditional digital signature schemes. The first problem will be related to multi-target attacks, which are often considered on the family of hash-based digital signature schemes such as XMSS, HORST, and FORST that are proposed as candidates for post-quantum cryptography. The article will show the influence of this type of attack on the popularly used digital signature schemes such as RSA, and ECDSA, and then analyze and evaluate some prevention solutions. The remaining issue that is concerned in this paper will be related to the requirement of recalculation in digital signature schemes based on a discrete logarithmic problem such as ECDSA. There, we will suggest some checks and iterations of the computation in the signing algorithm to avoid trivially leaking information related to the signing key.**

***Tóm tắt*— Trong bài báo này, nhóm tác giả sẽ xem xét hai vấn đề trên các lược đồ chữ ký số truyền thống. Vấn đề đầu tiên sẽ liên quan đến các tấn công đa mục tiêu, vốn thường được xem xét trên họ các lược đồ chữ ký số dựa trên hàm băm như XMSS, HORST, FORST mà được đề xuất như các ứng viên cho mật mã hậu lượng tử. Bài báo sẽ chỉ ra ảnh hưởng của kiểu tấn công này lên các lược đồ chữ ký số đang được sử dụng phổ biến hiện nay như RSA, ECDSA và sau đó phân tích, đánh giá một số giải pháp ngăn chặn. Vấn đề còn lại được quan tâm trong bài báo sẽ liên quan đến yêu cầu về bước lặp trong các lược đồ chữ ký số dựa trên bài toán logarit rời rạc như ECDSA. Ở đó, nhóm tác giả sẽ đề xuất một số bước kiểm tra và lặp lại tính toán trong thuật toán ký để tránh việc lộ lọt thông tin liên quan đến khóa ký một cách dễ dàng.**

Từ khóa*— *tấn công đa mục tiêu; lược đồ chữ ký số; RSA; ECDSA; vấn đề bước lặp.

Keyworks*— *multi-target attack; digital signature scheme; RSA; ECDSA; recalculation problem.

This manuscript is received on November 22, 2022. It is commented on December 14, 2022 and is accepted on December 18, 2022 by the first reviewer. It is commented on November 24, 2022 and is accepted on March 23, 2023 by the second reviewer.

I. INTRODUCTION

Introduced in W. Diffie and M. E. Hellman's article in 1976 [8], digital signatures have become a subject of extensive interest and research. Recently, a variety of digital signatures have been proposed, yet, RSA, DSA, and ECDSA remain the most popular among all. Regarding national uses, RSA is the most widely used digital signature platform in e-government and e-commerce.

The basis of the above digital signature schemes is the traditional problems: integer factorization and discrete logarithm. However, with high science technology advances, namely the potential foundation of quantum computers, those schemes which depend on these problems are on the edge of being fully broken down. Therefore, new generations of digital signatures are developed to counter this potential threat. For convenience, we use the concept of "traditional digital signatures" to refer to those signature schemes based on integer factorization and discrete logarithm problems.

According to the recently published results of NIST, hash-based and lattice-based signatures are the most potential candidates for quantum-resistant digital signatures. However, lattice-based digital signatures are quite complex. Regarding hash-based digital signature schemes, a typical attack is introduced as a "multi-target attack" (related to finding the second pre-image resistance of hash) by Hulsing et al. in their research [6]. This type of attack deserves to study since its ability to reduce the security of the schemes by the number of created target signatures. From a broader view, this type of attack has been previously used by the authors [7] to show the insecure HMQV key exchange protocol in the post-specified peer model.

Because the hash function is also a key component in traditional digital signature schemes and provides some important properties such as collision resistance, preimage awareness in serveral current signature scheme [10]. Thus, this paper will first focus on analyzing the threat of multi-target attacks on some traditional digital signatures, including RSA and ECDSA. Then, the authors will consider several solutions to help mitigate the impact of these schemes.

Additionally, this paper also considers another aspect of the digital signature scheme based on the discrete logarithm, which is the iteration problem (i.e., in the signing process, if a special case occurs, for example, the signature component is 0, we will have to redo the process from the beginning). This problem has been studied and analyzed in several works [1, 2, 3, 4]. The goal of reviewing is to provide some special cases to be covered in the signature algorithm (otherwise, the verifier can easily recognize and then recover the signing key). This work will be demonstrated by ECDSA scheme analysis.

The rest of the paper is structured as follows: Part II will present the security implications of using the hash function over traditional digital signature schemes, with two representatives without using any hash function, such as RSA and ElGamal. Part III will provide analytical evaluations of the multi-target attack and its impact on some popular digital signature schemes RSA, ECDSA, and time stamping scheme in RFC 3161 (whose main component is a digital signature), consequently, evaluate some solutions that counter this type of attack. Part IV of the paper will focus on the iteration problem of the ECDSA digital signature scheme. Finally, the conclusion will be placed in Section V.

II. SIGNIFICANCE OF HASH FUNCTION IN DIGITAL SIGNATURE

In this part, the paper will remind the importance of the hash function in digital signature schemes through examples of schemes based on integer factorization and discrete logarithm problems which are RSA and ElGamal.

A. RSA signature scheme

RSA digital signature scheme is constructed based on integer factorization problem. In which, the algorithm uses a modulus parameter N as the multiplication of two big prime numbers p, q $(N = pq)$. The public keys for users will be $\langle N, e\rangle$ with corresponding signing keys $\langle N, d\rangle$, where $ed = 1\ mod\ \phi(N)$and $\phi(N) = (p-1)(q-1)$. Therefore, its signing and verification algorithms are described as follows:

Signing algorithm: for signing key $\langle N, d\rangle$ and message $m \in \mathbb{Z}_N^*$, the signature σ is generated by calculating:

$$\sigma = m^d\ mod\ N$$

Verification algorithm: With public key $\langle N, e\rangle$, signed message $m \in \mathbb{Z}_N^*$ and a signature $\sigma \in \mathbb{Z}_N^*$, providing the valid result if and only if:

$$m = \sigma^e\ mod\ N$$

In this description, RSA digital signature scheme is insecure. In particular, in [5], the author has shown two types of forgery attacks against this scheme. The first one is a no-message attack, where the attacker can prepare a forged signature $\sigma_0 \in \mathbb{Z}_N^*$ and then compute message $m_0 = \sigma^e\ mod\ N$. Apparently, this pair (m_0, σ_0) satisfied the verification equation, thus, it is a valid one. The second type follows the adaptively chosen message attack to forge a valid signature on some message $m \in \mathbb{Z}_N^*$. In detail, In detail, the attacker can request the Signing algorithm to get signatures σ_1, σ_2 on two required messages $m_1, m_2 \in \mathbb{Z}_N^*$ (such that $m = m_1 m_2$), and then produce a signature $\sigma = \sigma_1\sigma_2$. This signature is valid on message m with respect to public key $\langle N, e\rangle$ since $m = m_1 m_2 = \sigma_1^e\sigma_2^e = (\sigma_1\sigma_2)^e = \sigma^e\ mod\ N$.

Those mentioned types of attacks can be addressed by hashing the message m by a hash function H before generating its signature, i.e. $\sigma = H(m)^d\ mod\ N$ and the verifier will determine whether $H(m) =$

$\sigma^e \ mod \ N$ or not. By using the hash function, the first type of attack is infeasible since it requires finding a pre-image of the target to trace back, which is blocked by the hash properties. For the second type of attack, the forgery process also requires another difficult task finding m_1, m_2 which satisfy $H(m_1)H(m_2)$ with given m. Besides, this version of RSA is also proven to be secure against adaptively chosen message attacks [5].

B. ElGamal signature scheme

ElGamal is representative of the digital signature schemes based on the discrete logarithm problem. The basis parameters of this digital signature scheme include large primes p and a generator $g \in \mathbb{Z}_p^*$. User's parameters contain signing key $a \in \{1, \dots, p-2\}$ and public key $y = g^a \ mod \ p$ mod p. Signing and Verification algorithms in this scheme work as follows.

Signing algorithm: for signing key a, message m, the signature is generated as follow:

(1) Randomly choose an integer $k \in \{1, \dots, p-2\}$ satisfying gcd$(k, p-1)$=1.

(2) Compute $r = g^k \ mod \ p$

(3) Compute $s = k^{-1}(m - ar) \ mod \ (p-1)$

(4) Return a signature (r, s).

Verification algorithm: With public key y, message m and signature (r, s), provide the valid result if and only if:

$$g^m = y^r r^s \ mod \ p$$

This scheme is shown to be insecure against no-message attack (according to [9]). Indeed, by randomly picking $e \in \{1, \dots, p-2\}$, computing $r = g^e y \ mod \ p$, $s = -r \ mod \ (p-1)$, and $m = es \ mod \ (p-1)$, we have (r, s) is a valid signature on m with respect to public key y since:

$$y^r r^s = y^{-s}(g^e y)^s = y^{-s} g^{es} y^s = g^{es} = g^m \ mod \ p$$

To prevent this attack, we can use a hash function H in a similar way to the RSA case, i.e., using $H(m)$ instead of m to compute s. A similar use of the hash function is also applied for other digital signature schemes such as DSA, ECDSA, GOST R 34.10-2012,…

III. EXAMINING MULTI-TARGET ATTACK ON SEVERAL TRADITIONAL DIGITAL SIGNATURE SCHEMES

In the previous section, the paper has presented the importance of the hash function to the security of digital signature schemes. This also explains why using the hash function as part of the schemes is crucial. In this section, we will describe multi-target attacks against digital signature schemes RSA and ECDSA, representing digital signature schemes that rely on the hardness of integer factorization and discrete logarithm problems.

A. Multi-target attack on RSA scheme

In this section, the paper will study the multi-target attack on RSA digital signature scheme, then give some discussion and propose a solution to address the situation.

1. Multi-target attack method on RSA

First of all, reminding that this type of attack starts from a fact that exists two different message $m_1 \neq m_2$ satisfying $H(m_1) = H(m_2)$, then the valid RSA signature on m_1 is also valid on m_2. Indeed, assume that σ is a valid signature on m_1 with public key e, we have:

$$H(m_2) = H(m_1) = \sigma^e \ mod \ N.$$

This equality shows that σ is also the valid signature of m_2 with public key $\langle N, e \rangle$.

According to the above observation, we have if the attacker can collect some signed message m with its valid signature σ and then generate m' such that $H(m) = H(m')$, then (m', σ) is considered as a valid forgery of RSA digital signature scheme. Note that the latter condition is hard to occur since it is equivalent to finding the second pre-image of hash function H in general cases. However, if a large number of signed messages and corresponding signatures are collected, the possibility of finding a message whose hash value belongs to the set of the hash values of the collected messages is more real. Therefore, the probability of forging a forgery is increased. In particular, assuming that the

attacker has collected ℓ pairs of message/valid signature $(m_1, \sigma_1), (m_2, \sigma_2), \dots, (m_\ell, \sigma_\ell)$ corresponding to their public keys $\langle N_1, e_1\rangle, \langle N_2, e_2\rangle, \dots, \langle N_\ell, e_\ell\rangle$.

Therefore, if the attacker can find a m' that $H(m') = H(m_i)$, then:

$$H(m') = H(m_i) = \sigma_i^{e_i} \bmod N_i.$$

This equality shows that σ_i is also the valid signature of m' with public key $\langle N_i, e_i\rangle$.

2. Impact evaluation of multi-target attack on RSA

As analyzed, we can observe that finding a message m' that $H(m')$ belongs to the set of multi-targets $\{H(m_1), H(m_2), \dots, H(m_\ell)\}$ is ℓ times easier than being equal a single hash value. In other words, the security of RSA is inversely proportional to the number of message/signature pairs collected by the attacker. In the following, we will show 02 practical reasons why this type of attack should be prevented.

Number of Targets: The above attack shows that the more valid message/signature pairs the attacker can collect, the easier it will succeed in finding a forgery. Note that the collected pairs do not necessarily correspond to a unique public key, but must be treated with the same hash function. Thus, when the number of users is large, the insecurity of the digital signature scheme will also increase. An example to illustrate the impact of this attack is that if about 50% of the population of Vietnam (about 50 million people) has a demand to create at least one digital signature per day, then there are about $5 * 10^7 * 365 \approx 2^{34}$ signatures being created (per year). If all of those signatures are collected by an attacker (with the ability to hijack and access the host servers in which the digitally signed databases are stored), the risk of the digital signature scheme breakdown will become larger. In particular, when the security threshold of a digital signature scheme based on the second pre-image resistance of the hash function will be reduced by 34 bits. This number will increase according to the scale, demand, and usage time of digital signature users in the system.

Validity of forged signatures: For digital signature schemes, the signing key is a core component and should be kept secret throughout its lifecycle. However, if the users suspect the signing key is weak or compromised, it must be revoked using key revocation solutions. In that case, we need to ensure that revoked (or expired) signing keys will no longer be able to generate valid signatures. In order to achieve this goal, some solutions suggest using the Certificate Revocation List (CRL) or the Online Certificate Status Protocol (OCSP). These solutions will guarantee that a valid signature for a particular signing/public key pair can only be generated during the validity period of the corresponding certificate. This also means that, if an attacker can find a message/signature pair satisfying verification with some public key at the time that is no longer within the validity period of the corresponding certificate, this forgery is no longer considered valid.

Regarding RSA/ECDSA signature forgery based on multi-target attack formerly considered, the attack only provides a forged message/signature $\big(m', (r, s)\big)$ such that $H(m') = H(m)$ and m is a signed message with corresponding signature (r, s). Therefore, if (r, s) is only generated by legal users with signed message m in the certificate's validity period, then $\big(m', (r, s)\big)$ is also considered valid. Because the attacker only reuses a valid signature instead of creating a new one, therefore, this forgery can be done wherever in the future, not necessarily restricted to the certificate's validity period.

3. Solutions to prevent multi-target attack on RSA

In this section, we will evaluate several solutions that can be applied to prevent the risk of multi-target attacks.

Solution 1: Because multi-target exploits the hash component of the signature, i.e., finding the pre-image of one of the hash values belonging to a set of l targets is l times easier than that of a single hash value. Therefore, the first solution considers increasing the output bit-length of the hash function to guarantee that the

security threshold of the hash against the attacks (that find the pre-image for any member in the target set) achieves minimum expectation. For example, if every people in the world needs to generate 1000 signatures per day within 100 years (it is equivalent to $1000 * 100 * 8 * 10^9 * 365 \approx 2^{58}$ signature be created), and the minimum security threshold is 2^{256}, then the hash function should have a minimum output bit-length of $256 + 58 = 314$ (bit).

Second solution: Reminding that the multi-target attack is regarded as exploiting a set of hash values of messages signed by any users, or in other words, these hash values are not personalized. Therefore, we suggests that for each user $\hat{A}$ (with corresponding keys pairs $(\langle N_{\hat{A}}, e_{\hat{A}}\rangle, \langle N_{\hat{A}}, d_{\hat{A}}\rangle)$), the signing algorithm for message m will work as follows:

1. Compute $h = H(m \parallel \hat{A})$.

2. Generate signature $\sigma_{\hat{A}} = h^{e_{\hat{A}}} \, mod \, N_{\hat{A}}$.

The corresponding verification algorithm for signature $\sigma_{\hat{A}}$ on message m with public key $\langle N_{\hat{A}}, e_{\hat{A}}\rangle$ of user $\hat{A}$ will be:

1. Compute $h = H(m \parallel \hat{A})$

2. Check if $\sigma_{\hat{A}} = h^{e_{\hat{A}}} \, mod \, N_{\hat{A}}$.

By using this solution, the attacker's target set will only correspond to a specific user since even if the attacker can figure out the value m' and the identity of some user $\hat{B}$ such that $H(m' \parallel \hat{B}) = H(m \parallel \hat{A})$, where m is the message signed by user $\hat{A}$ with the corresponding signature $\sigma_{\hat{A}}$, then the attacker can only use signature $\sigma_{\hat{A}}$ as a valid signature on the message m' corresponding to the public key of $\hat{A}$ when $H(m' \parallel \hat{B}) = H(m' \parallel \hat{A})$, will result in $\hat{B}$ and $\hat{A}$ coincidence except for the negligible probability for occurring collision event of the hash function. In other words, the number of targets in the system will not depend on the number of users who need to get digital signatures in the system, but only on the number of signatures created by some specific individual targeted by the attacker. Therefore, this solution will be suitable for the group of users who less often have digital signing demand.

Solution 3: Although the second solution significantly limits the number of targets for an attacker to exploit, its main limitation is that the number of signatures of the target user can still increase due to their keys changing over time. Therefore, in this solution, we include the user's (still valid) public key to hash along with the signed message and the user's identity. Note that, including the public key and user identity in the calculation of the message hash step is not a new idea, as it was used in the design of the EdDSA signature scheme.

In more detail, for each user $\hat{A}$ having key pair $(pk_{\hat{A}}, sk_{\hat{A}}) = (\langle N_{\hat{A}}, e_{\hat{A}}\rangle, \langle N_{\hat{A}}, d_{\hat{A}}\rangle))$, the signing algorithm for message m will work as follows:

1. Compute $h = H(m \parallel \hat{A} \parallel pk_{\hat{A}})$.

2. Generate signature $\sigma_{\hat{A}} = h^{e_{\hat{A}}} \, mod \, N_{\hat{A}}$.

The corresponding verification algorithm for signature $\sigma_{\hat{A}}$ on message m with public key $pk_{\hat{A}} = \langle N_{\hat{A}}, e_{\hat{A}}\rangle$ of user $\hat{A}$ will be:

1. Compute $h = H(m \parallel \hat{A} \parallel pk_{\hat{A}})$.

2. Check if $\sigma_{\hat{A}} = h^{e_{\hat{A}}} \, mod \, N_{\hat{A}}$.

Similar to the argument for Solution 2, we can say that this solution will cause the number of targets that an attacker can collect and exploit to be limited to a particular user with its specific public key. Therefore, this solution is suitable for users with moderate digital signing demand. However, for objects that need to perform a large number of digital signing operations daily such as servers providing digital signing services, time stamping, etc., solution 3 is still not a complete solution.

With the above solutions and evaluations, we found that RSA signature combining solutions 2 and 3 would be suitable for a class of users who do not have a high demand for digital signing. In other cases, Solution 1 should be used (combing with Solutions 2 and 3 to reduce the output length requirement of the hash function).

B. Security of ECDSA signature scheme

ECDSA is an ElGamal-type signature scheme and operates on an elliptic curve. Here, we will analyze the multi-target attack against this scheme. Firstly, we describe the ECDSA scheme.

Given elliptic curve $E(\mathbb{F}_p)$ defined on a finite field $\mathbb{F}_p$ and a base point P of order n, where n a prime divisor of $\#E(\mathbb{F}_p)$, signing key is $d \in_R [1, n-1]$ and its corresponding public key is $(E(\mathbb{F}_p), P, n, Q)$, where $Q = dP$.

- *The signing algorithm (for $\hat{A}$ on message M):* Using its secret key, $\hat{A}$ generates digital signature for message *M* through following steps:

1. Pick a integer $k \in_R [1, n-1]$

2. Compute $kP = R = (x_R, y_R)$ and $r = x_R \bmod n$, if $r = 0$, go to step 1

3. Compute $h = H(M)$

4. Compute $s = k^{-1}(h + dr) \bmod n$; if $s = 0$, go to step 1

5. Return signature (r, s)

- *The verification algorithm (digital signature (r, s) of $\hat{A}$ on message M):*

1. Verify that r, s are in range $[1, n-1]$

2. Compute $w = s^{-1} \bmod n$.

3. Compute $h = H(M)$.

4. Compute $u_1 = hw \bmod n$.

5. Compute $u_2 = rw \bmod n$.

6. Compute $u_1 P + u_2 Q = (x_0, y_0)$, and compute $v = x_0 \bmod n$.

7. Return "valid" result if $v = r$.

It can be seen that, in the ECDSA signature scheme, the hash value $H(M)$ will be used to compute the signature components instead of using the message *M* itself like the ElGamal signature scheme. This helps resist forgery searches that act in the same manner as outlined in Section B Part II. However, just as in the case of RSA, such a manner would also result in ECDSA being subject to a multi-target attack. Specifically, we assume that the attacker obtains ℓ valid ECDSA message/signature pairs of $(m_1, (r_1, s_1))$,...,$(m_\ell, (r_\ell, s_\ell))$ corresponding to the public keys pk_1,…,pk_ℓ, with pk_i=$(E(\mathbb{F}_{p_i}), P_i, n_i, Q_i)$ and $i \in \{1, \dots, \ell\}$. Then, if the attacker can find a message m' such that $H(m') = H(m_i)$ for some $i \in \{1, \dots, \ell\}$, we have:

$$r_i = x_{H(m_i)s_i^{-1}P_i + r_i s_i^{-1} Q_i} = x_{H(m')s_i^{-1}P_i + r_i s_i^{-1} Q_i} \bmod n_i.$$

It implies that $(m', (r_i, s_i))$ satisfies the verification equation with the public key $(E(\mathbb{F}_{p_i}), P_i, n_i, Q_i)$. Thus, the attacker can provide a valid forgery to the ECDSA scheme.

Impact evaluation: With a similar evaluation presented in Section A-2 Part III, it can be seen that a multi-target attack on ECDSA is basically similar to that of RSA.

Solution: The solutions suggested for RSA to limit multi-target attacks can be applied to the case of ECDSA. In this section, we will consider another solution that is more suitable for the ECDSA scheme and other discrete logarithm-based signature schemes (such as GOST R 34.10-2012, DSA,...). In fact, when using this solution for the ECDSA scheme, we obtain the Pointcheval/Vaudeney signature scheme (which is mechanism number 4 for certificate-based signature schemes in the ISO/IEC 14888-3 standard).

In more detail, for the case of ECDSA, this solution only makes a slight modification in the computation of the hash function. There, the hash value h in step 3 of the ECDSA signing algorithm will be computed by $h = H(M \| r)$ (where r is the first component of the signature calculated in step 2) instead of $h = H(M)$. Since the component r is generated randomly with a negligible probability of duplication, it is also unlikely for two signed messages to have the same component r. This can be visualized through the following example: for n of 256-bit size (which means $n \approx 2^{256}$), the average number of signatures generated per person is 1 million signatures per day (which is equivalent to 2^{20} signatures), then the number of signatures of all people in the world after 1000 years will be about $2^{20} * 8 * 10^9 * 365 * 1000 \approx 2^{72}$ signatures, so the probability that there are two signatures with identical first components (which is in range {1,…,n}) is approximately $\frac{(2^{72})^2}{2n} \approx \frac{1}{2^{113}}$.

This also means that the probability that the attacker obtains a set with more than one target becomes negligible, thus causing the multi-target attack to reduce to a single-target attack which is equivalent to finding the second preimage of the hash function. Furthermore, the example analyzed above also shows that the ECDSA scheme with this solution is suitable even for those with high demand for digital signatures, which is a problem that Solutions 2 and 3 for RSA can not guarantee.

C. Multi-target attack on time-stamping scheme in RFC 3161

Timestamp is a solution for sensitive data that requires proof of its creation time (such as patents, property certificates, etc.). According to RFC 3161, the time-stamping process for data between a customer and a Timestamp Authority (TSA) is essentially done as follows:

1. The client computes a hash value $H(m)$ to send to the TSA as a time-stamping request. The purpose of sending a hash value instead of the digital data itself is to ensure the confidentiality and privacy of the data. Note that, in addition to the hash value $H(m)$, some other data that is also included in the request are *version number, nonce, reqPolicy*, etc.
2. After receiving a time-stamping request, the TSA will use its signing key to generate a signature σ_{TSA} on a value $H(m)||timestamp$ (where $timestamp$ represents the time at which the TSA time-stamped data).
3. Subsequently, the TSA returns to the client $(\sigma_{TSA}, timestamp)$ in the response to the client.
4. Client will store data m with $(\delta_{TSA}, timestamp)$ to secure m's integrity and confirmation time.

It can be witnessed that, through a multi-target attack, if an attacker can find a data m' with an identical hash value to authentic time-stamped data m with timestamp $(\sigma_{TSA}, timestamp)$ from the described process, then that timestamp is also valid on m'. Whence, replacing m with m' will break m's integrity.

Impact evaluation: Basically, time-stamping the user's digital data follows the mechanism of creating a digital signature of the server on the hash value of the data and the timestamp, so the solutions applying for RSA and ECDSA only help prevent TSA signature forgery against multi-target attacks, but does not actually prevent multi-target attacks on users' hash values. In other words, even if the solutions considered in Sections A-3 and Section B Part III are applied to the TSA's signature scheme, the above attack remains available.

Solution: Based on the solutions examined in the previous sections, in order to prevent multi-target attacks for time-stamping schemes, we propose a solution for both the server and the client simultaneously:

1. Increase the output size of the hash function used to hash data that is needed to be timestamped.
2. Apply solutions that limit multi-target attacks from forging signatures on the server side.

IV. THE PROBLEM OF ITERATION STEP IN THE ECDSA SCHEME

In ECDSA digital signature scheme's signing algorithm described in section B part III, if component r or s of the signature is equal to zero, we have to repeat the entire process. The reason for this has been argued in studies [1], [2], [4] Besides, in [3]., M. Braun and A. Kargl have considered a case where $u_1P = u_2Q$ while computing step 6 of the ECDSA verification algorithm with signature (r,s) and a message with hash value h. If that is the case, the signature keys can easily be disclosed to the verifier as follows:

$$u_1P = u_2Q \Leftrightarrow hs^{-1}P = rs^{-1}dP \Leftrightarrow h = rd \bmod n \Leftrightarrow d = hr^{-1} \bmod n$$

To address this issue, M. Braun and A. Kargl have suggested testing $h = rd \bmod n$ in the signing algorithm, which will repeat the process if this equality holds.

In this section, we will point out a few more cases that need to be aware of to avoid trivially exposing the user's signing key to the verifier.

- *Case* $k = 1$: The verifier will detect this case by testing $u_1P + u_2Q = (x_0, y_0) \stackrel{?}{=} P$ in step 6 of verification algorithm. If this is the case, the verifier will obtain the user's signing key as follows:

$$(u_1 - 1)P = -u_2Q,$$
$$\Leftrightarrow (u_1 - 1) = -u_2 d \ mod \ n$$
$$\Leftrightarrow d = u_2^{-1}(1 - u_1) \ mod \ n.$$

- *Case* $k = d$: The verifier will detect this case by testing $u_1P + u_2Q = (x_0, y_0) \stackrel{?}{=} Q$ in step 6 of verification algorithm. If this is the case, the verifier will obtain the user's signing key as follows:

$$(u_1 - 1)P = -u_2Q,$$
$$\Leftrightarrow (u_1 - 1) = -u_2 d \ mod \ n$$
$$\Leftrightarrow d = u_2^{-1}(1 - u_1) \ mod \ n.$$

In both considered cases, we can see that if the ephemeral secret key $k \in \{1, d\}$ then the signatures generated by it will expose information to the verifier to extract the user's signing keys. Note that, to detect such special cases, the verifier will need to test whether $u_1P + u_2Q = (x_0, y_0) \stackrel{?}{=} P$ hoặc $u_1P + u_2Q = (x_0, y_0) \stackrel{?}{=} Q$ or not in step 6 of the verification algorithm. However, such operations are fast and do not consume the computing power of the verifier.

To avoid such situations, we suggest a modification and addition of iteration cases to the signing algorithm of ECDSA as follows

Algorithm 1

1. Pick an integer $k \in_R [1, n-1]$

2. Compute $kP = R = (x_R, y_R)$ and $r = x_R \bmod n$, if $r \in \{0, x_P, x_Q\}$, go to step 1

3. Compute $h = H(M)$

3'. If $h = rd \ mod \ n$, go to step 1

4. Compute $s = k^{-1}(h + dr) \bmod n$, $s = 0$ go to step 1.

5. Return signature (r, s).

In our modification, we have used some addition iteration conditions ($r = x_P$ or $r = x_Q$) in step 2. With these conditions, we can cover cases where $k = 1$ or $k = d$, since they will cause the algorithm to redo the process. On the other hand, the test in step 3' follows [3]. We note that our modifications are also necessary for other digital signature schemes based on the discrete logarithm problem (such as GOST R 34.10-2012, EC-Schnorr, …).

V. CONCLUSION

In this paper, we have examined two issues related to the security of some traditional digital signature schemes, which are now commonly used, RSA and ECDSA.

The first problem is related to multi-target attacks on the RSA, ECDSA, and time-stamping schemes in RFC 3161. This type of attack deserves attention because it can reduce the security level of the scheme according to the number of signatures generated, especially in the case of RSA digital signatures which are widely used in the fields of e-government and electronic transactions in our country nowadays. To mitigate this type of attack, we have evaluated several possible solutions.

The remaining problem mentioned in our paper is related to iterations of the computation in the signing process of the ECDSA digital signature scheme. More specifically, we have pointed out some special cases in which if the elliptic point computed in step 6 of the verification algorithm matches the base point or the public key, the verifier can easily compute the signing key of the user who generated the verified signature. Accordingly, we have added a test condition in the signing algorithm of the ECDSA scheme which results in Algorithm 1.

REFERENCES

[1] Ronald L. Rivest, Martin E. Hellman và John C. Anderson, "Responses to NIST's Proposal", Communications of the ACM, July 1992, Vol.35, No.7, pp. 42-52.

[2] Markus Michels, David Naccache, and Holger Petersen, "GOST 34.10 – A Brief Overview of Russia's DIGITAL SIGNATUREA", Computers & Security 15(8):725-732, 1996.

[3] M. Braun, A. Kargl. "A Note on Signature Standardigital signature". IACR Cryptology ePrint Archive 2007 (2007): 357.

[4] Ian F. Blake, G. Seroussi, and Nigel P. Smart, edigital signature. "Advances in elliptic curve cryptography" (Chapter II). Vol. 317. Cambridge University Press, 2005.

[5] Katz, Jonathan, and Yehuda Lindell. "Introduction to modern cryptography" (Chapter 12). CRC press, 2020.

[6] A. Hülsing, J. Rijneveld, F. Song. (2016). Mitigating multi-target attacks in hash-based signatures. In Public-Key Cryptography–PKC 2016 (pp. 387-416). Springer, Berlin, Heidelberg.

[7] A. Menezes, and B. Ustaoglu. "Comparing the pre-and post-specified peer models for key agreement." Australasian Conference on Information Security and Privacy. Springer, Berlin, Heidelberg, 2008.

[8] W. Diffie and M. E. Hellman (1976). "New Directions in Cryptography". *In IEEE Transactions on Information Theory*, volume IT-22, no. 6, pages 644-654, November 1976.

[9] D. Pointcheval, and J. Stern (1996). "Security proofs for signature schemes". *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, Berlin, Heidelberg.

[10] Phong, T.Q. and Linh, V.T. 2022. The security of the BLT signature scheme with practical implementation issues. Journal of Science and Technology on Information security. 2, 14 (Jan. 2022), 35-44. DOI:https://doi.org/10.54654/isj.v2i14.146.

ABOUT THE AUTHOR

Trieu Quang Phong

Workplace: Institute of Cryptographic Science and Technology, Government Information Security Committee.

Email: phongtrieu53@gmail.com

Eudcation: The BS in Department of Mathematics, Hanoi University of Science 2014.

Recent research direction: The provable security for the signature schemes and the key exchange protocol.

Cơ quan làm việc: Viện Khoa học và Công nghệ mật mã, Ban Cơ yếu Chính phủ

Email: phongtrieu53@gmail.com

Quá trình đào tạo: Cử nhân Khoa Toán, Đại học Khoa học Tự nhiên Hà Nội năm 2014.

Hướng nghiên cứu hiện nay: Tính bảo mật có thể chứng minh được đối với sơ đồ chữ ký và giao thức trao đổi khóa.

Do Dai Chi

Workplace: Institute of Cryptographic Science and Technology, Government Information Security Committee.

Email: dodaichi2005@gmail.com

Eudcation: The BS in Department ofMathematics, Hanoi University of Science (2013), The MS of Cryptography in University of Limoges (2019).

Recent research direction: Digital signature scheme; Provable security; Public key cryptography.

Cơ quan làm việc: Viện Khoa học và Công nghệ mật mã, Ban Cơ yếu Chính phủ

Email: dodaichi2005@gmail.com

Quá trình đào tạo: Cử nhân Toán, Đại học Khoa học Tự nhiên Hà Nội (2013), Thạc sĩ Mật mã học tại Đại học Limoges (2019).

Hướng nghiên cứu hiện nay: Lược đồ chữ ký số; Bảo mật có thể chứng minh được; Mật mã khóa công khai.

Tran Duc Huy

Workplace: Institute of Cryptographic Science and Technology, Government Information Security Committee.

Email: audirs12@gmail.com

Eudcation: The BS in Information Technology, Saint Petersburg Electrotechnical University (2010). The MS in Information Technology, Saint Petersburg Electrotechnical University (2012). The PhD in Information Technology, Academy of the Federal Security Service Russian Federation (2020). Recent

research direction: Information security; Database Security; Blockchain.

Cơ quan làm việc: Viện Khoa học và Công nghệ mật mã, Ban Cơ yếu Chính phủ
Email: audirs12@gmail.com
Quá trình đào tạo: Cử nhân Công nghệ thông tin, Đại học Kỹ thuật Điện Saint Petersburg (2010). Thạc sĩ Công nghệ thông tin, Đại học Kỹ thuật Điện Saint Petersburg (2012). Tiến sĩ Công nghệ thông tin, Học viện Cơ quan An ninh Liên bang Nga (2020).
Hướng nghiên cứu hiện nay: An toàn thông tin; Bảo mật cơ sở dữ liệu; Chuỗi khối.

Nguyen Ngoc Diep
Workplace: Institute of Cryptographic Science and Technology, Government Information Security Committee.
Email: diepnngoc@yahoo.com
Eudcation: The BS in cryptographic techniques, Academy Of Cryptography Techniques (1995). The MS in Information Technology, Le Quy Don Technical University (2003). The PhD in Information Systems, Posts and Telecommunications Institute of Technology (2018).
Recent research direction: Information security; Blockchain.

Cơ quan làm việc: Viện Khoa học và Công nghệ mật mã, Ban Cơ yếu Chính phủ
Email: diepnngoc@yahoo.com
Quá trình đào tạo: Cử nhân về kỹ thuật mật mã, Học viện Kỹ thuật Mật mã (1995). Thạc sĩ Công nghệ thông tin, Đại học Kỹ thuật Lê Quý Đôn (2003). Tiến sĩ Hệ thống thông tin, Học viện Công nghệ Bưu chính Viễn thông (2018).
Hướng nghiên cứu hiện nay: An toàn thông tin; Chuỗi khối.