INTRODUCTION

Data is confidential and remains secure when sent on digital channels, is often known to unauthorized persons. This is what motivates the researchers to think of a reliable algorithm so that the information remains safe. Several algorithms have been developed and one of which has been discovered is the Shor algorithm which was invented by Peter Shor in 1995. Through this Shor algorithm, a quantum computer can decode a secret code that is currently commonly used to secure data transmission. This code is called an RSA code. RSA (Rivest–Shamir–Adleman) is a public-key cryptosystem that is widely used for secure data transmission. If encoded via RSA code, the data transmitted will be safe because the RSA code cannot be cracked in a short time. In addition, solving the RSA code requires the work of thousands of computers in parallel. So, this solving work is not effective. In this study, there is various literature that forms the basis of the author’s theory in conducting research, the literature consists of various scientific books and papers which according to research.

The following are some of the main literature on which the theory is based, namely: the application of digital signatures in government electronic systems to support e-government [1], a post-quantum digital signature scheme based on Supersingular Isogenies [2], an efficient signature scheme of Isogenies of Supersingular Elliptic Curves [3], and mathematical concepts and models based on Isogen-Based Cryptographic Mathematics [4] and data integrity and security using keccak and Digital Signature Algorithm (DSA) [5] and elliptic curve Diffie-Hellman cryptosystem for public exchange process [6], [7].

Our Contribution. In this paper, the use of information technology includes activities related to the development of cryptographic mathematical concepts and models. DSA is public-key cryptography that is used for authentication, data security, and anti-denial tools [8], the parameters used are dynamic public key and private key parameters, which have different values for each digital signature creation process [9]. The DSA algorithm is used for message signing, the SHA function is
also involved as a message digest generator of messages [10]. A process for optimizing digital signature hash function files, then management systems and work processes electronically, and public services so that documents can be easily and cheaply verified.

**Digital Signature Algorithm (DSA)**

A signature on electronic data is called a digital signature, but a digital signature is not a sign that is digitized by a scanner or electronic pen. A digital signature is a mathematical scheme that relies on the message content and the sender of the message to prove the authenticity of the message or document. This scheme is carried out as a guarantee that the data and information come from the correct source [11]. A digital signature consists of a series of hash functions that are generated from a specific hash function algorithm which is then encoded (encrypted) with an asymmetric key cryptographic algorithm. To verify this, the algorithm’s public key is used.

DSA is a standard for digital signatures that was formalized in August 1991 by NIST (The National Institute of Standards and Technology) which consists of two main components, namely: the digital signature algorithm and the Secure Hash Algorithm (SHA) [12].

DSA is included in the public key cryptographic algorithm, and not used for encryption. It means more specifically for digital signatures. DSA has two main functions, namely the signature generation and signature validity checks developed from the ElGamal algorithm. DSA uses two keys, namely the public key and the private key. The formation of signatures uses the private key, while to verify a digital signature we can use the public key which can be illustrated in Figure 1 [13].

SHA is a one-way hash function (a one-way hash) created by NIST and used in conjunction with DSA which is the standard for digital signature generation. The National Security Agency (NSA) states that SHA is a standard one-way that hash function which is a continuation of its predecessor, namely MD4 made by Ronald L. Rivest from MIT, who had found the collision and developed MD5, in other words, SHA is safe because it is designed in such a way computationally. You may find messages that correspond to the message digest.

The SHA algorithm accepts $2^{64}$ bit (2,147,483,648 gigabytes) input by producing a message digest that is 160 bits in length. SHA has six variants that refer to a family of one-way hash functions, namely SHA-0, SHA-1, SHA-224, SHA-256, SHA-384, SHA-512. The numbers behind the SHA indicate the length of the digest messages. In this research, only SHA-1 will be discussed. Following Table 1 the sizes in bits for the SHA family.

**Table 1. The digest SHA message sizes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Output Internal state</th>
<th>Block</th>
<th>Max message</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHA-0</td>
<td>160</td>
<td>160</td>
<td>512</td>
</tr>
<tr>
<td>SHA-1</td>
<td>256/224</td>
<td>256</td>
<td>512</td>
</tr>
<tr>
<td>SHA-224</td>
<td>512/384</td>
<td>512</td>
<td>1024</td>
</tr>
</tbody>
</table>

The steps for making a message digest with SHA-1 are generally similar to MD-5, which are as follows: First add padding bits, then increase the value of the original message length and initialize the MD buffer, and processing messages in blocks of 512 size.

**METHOD**

The public key cryptographic algorithm for DSA is the creation of digital signatures using the private key and for verification using the public key [14], [15]. DSA uses the hash SHA function to convert a message into a message digest that is 160 bits in size. Algorithm Digital Signature Parameters:

1. Public key $p, q, g$ with
   a. $p$ is a prime number 512 to 1024 and is in the form of 64 blinks;
   b. $q$ prime numbers are 160 bits and are a factor of $p - 1$;

![Figure 1. Generate and verification of a digital signature algorithm.](image)
c. \( g = h^{(p-1)/q} \mod p \) with \( 1 < h < p - 1 \);

d. \( y = g^x \mod p \)
2. The private key \( x \) which is an integer less than \( q \);
3. Message \( m \) which will be signed.

Before generating digital signatures, the user must have a key first, namely a private key and a public key. The procedure for generating keys in a digital signature is as follows:

**Algorithm** Key generation for the DSA.
This procedure will generate the public key \((p, q, g, y)\) and the private key \(x\) by:
1. Select \( p \) and \( q \) with \((p - 1) \mod q = 0\).
2. Compute \( g = h^{(p-1)/q} \mod p \) which is greater than 1.
3. Specify private key \( x \) with \( x < q \).
4. Compute public key \( y = g^x \mod p \).

**Algorithm** DSA signature generation
1. Compute the message digest of \( m \) messages with a hash function SHA-1, \( H(m) \).
2. Compute a random secret integer \( k < q \).
3. Signature of the message \( m \) is \( r \) and \( s \), compute:
   \[ r = (g^k \mod p) \mod q \]
   \[ s = (k^{-1}(H(m) + x.r)) \mod q \]
4. Sent message \( m \) along with the signature \( r \) and \( s \).

**Algorithm** DSA signature verification
1. Compute the message digest of \( m \) messages with a hash function SHA-1, \( H(m) \).
2. Verify that digital signature \( 0 < r < q \) and \( 0 < s < q \); if not, then reject the signature and compute:
   \[ w = s^{-1} \mod q \]
   \[ u_1 = (H(m).w) \mod q \]
   \[ u_2 = (r.w) \mod q \]
   \[ v = ((g^{u_1}.y^{u_2}) \mod p) \mod q \]
3. If \( v = r \), then the signature verified.

The process of encrypting and decrypting messages using the digital signature algorithm with the help of the program on the portal [https://asecuritysite.com/encryption/md5](https://asecuritysite.com/encryption/md5)

**RESULTS AND DISCUSSION**

The value of \( p, q \) are prime numbers that have a length limit of 512 to 1024 bits. The \( q \) is 160 bits, making DSA almost impossible to solve by software in other words difficult to solve. Therefore, DSA is implemented in a software device, where the bit length limits \( p \) and \( q \) are changed to a maximum value of \( p \) and \( q \) is \( 2^{512} \).

DSA based on the ElGamal algorithm has the following parameter properties:
1. \( p \) is a prime number with length \( L \) bits, in this case, \( 512 \leq L \leq 1024 \) and \( L \) must be a multiple of 64 bits and is public.
2. \( q \) 160-bit prime number is a factor of \((p - 1)\), in other words \((p - 1) \mod q = 0\), is public.
3. \( g \) parameter is public with \( q = h^{(p-1)/q} \mod p \) with \( h < (p - 1) \).
4. \( x \) is a number less than \( q \) and is private
5. \( m \) is the message to be signed
6. \( y \) is the public key with \( y = g^x \mod p \).
7. \( r, s \) is a signature, where the message is always accompanied by a signature, written \((m, s)\).
8. \( v, w \) a compute procedure to verify a signature if \( v = r \) then a signature is valid.

Examples of message characteristics change processes for MD5, SHA-1, and SHA-2 (SHA-224, SHA-256, SHA-384, SHA-512) which generates a hash signature, and the output is usually displayed in hex or Base-64 format. For example, the message is given as follows.

**Message:**

Nearly nine months after the pandemic, Indonesia finally welcomed the first shipment of the COVID-19 vaccine produced by China Sinovac Biotech at Soekarno-Hatta International Airport in Tangerang, Banten, on December 6. The first batch contained 1.2 million doses of vaccine. This is not an all-around cure solution for a country of 270 million people. Even so, the arrival of the first wave marks an important step in the country's efforts to develop a vaccine for COVID-19.
Generate hash:

<table>
<thead>
<tr>
<th>MD5</th>
<th>780AD7A4AF9E608CD5C58710960964FFA</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHA-1</td>
<td>E17FF7B6320F70368BAAFF6A8DDAEBFA66D0F3FB2</td>
</tr>
<tr>
<td>SHA-256</td>
<td>9E04339AA01D0AF7061C39A83D709235B7DA341901CEDC734FDD53AD7D9EB792</td>
</tr>
<tr>
<td>SHA-382</td>
<td>F4BD69D7B1381A0C535F7CA5D1A006BB9EA182B2EB942140218C467303D34ED6863ADE7023F2F4D76FAEC562C36914</td>
</tr>
<tr>
<td>SHA-512</td>
<td>F80D7D21175D5A7662539BEF77A8AFABDDA0F2849F31BCE2D3EF360E9D673A4794C16F36053DA7C82EB6E5BF974E9E31F5811A1DAD8C57B0E9969E0AA614A195</td>
</tr>
</tbody>
</table>

Next, we will generate a key pair in DSA, first select the prime numbers $p$ and $q$, which in this case $(p - 1) \mod q = 0$. For example $p = 59419$ and $q = 3301$ this satisfies $(59419-1) \mod 3301 = 0$. Then calculate $g = h^{(p-1)/q} \mod p$ in this case $1 < h < p - 1$, for example, $h = 100$, then $g = 100^{(59419-1)/3301} \mod 59419 = 18870$. Find the private key $x$ that satisfies $x < q$, for example, $x = 3223$ then compute the public key $y = g^x \mod p$ or $y = 18870^{3223} \mod 59419 = 2945$.

The next procedure is to generate a digital signature by calculating the hash value of the message $m$, for example, $H(m) = 4321$. Find a random number $k < q$, for example, $k = 997$ and $k^{-1} \equiv 2907 \pmod{3301}$ with its public parameters $p = 59419$ and $q = 3301$, $g = 18870$ and private parameters $x = 3223$. Compute digital signature, $r$ and $s$ as follows:

\[
\begin{align*}
    r &= (g^x \mod p) \mod q \\
    &= (18870^{997} \mod 59419) \mod 3301 \\
    &= 848
\end{align*}
\]

\[
\begin{align*}
    s &= (k^{-1}(H(m) + x \cdot r)) \mod q \\
    &= (2907(4321 + 3223(848))) \mod 3301 \\
    &= 7957694475 \mod 3301 \\
    &= 183
\end{align*}
\]

Send message $m$ and signature $(r, s)$ namely $(848, 183)$.

The next procedure is to verify the validity of the digital signature. First, calculate the value of the message $m$, for example, $H(m) = 4321$, then sign $(r, s) = (848, 183)$ as follows:

\[
\begin{align*}
    s^{-1} &= 469 \pmod{3301} \\
    w &= s^{-1} \mod q = 469 \mod 3301 = 469
\end{align*}
\]

\[
\begin{align*}
    u_1 &= (H(m), w) \mod q \\
    &= (4321.469) 2026549 \mod 3301 = 3036
\end{align*}
\]

\[
\begin{align*}
    u_2 &= (r, w) \mod q \\
    &= (848, 469) \mod 3301 = 397712 \mod 3301 = 1592
\end{align*}
\]

\[
\begin{align*}
    v &= (s u_1 + u_2) \pmod{q} \\
    &= (18870^{1592}.29245^{1592}) \mod 3301 \mod 3036 \\
    &= 3036.848 \mod 3301 \mod 848 \\
    &= 848
\end{align*}
\]

Since $v = r$ the signature is valid, the signature is verified to prove whether the message is genuine or not by the steps as outlined above; first, the digital signature $(r, s)$ is decrypted using the sender’s public key, producing the original message digest. Second, the receiver then converts message $m$ into message digest by using the DSA algorithm, which is the same as the hash function used by the sender. Finally, if the signature is received it is genuine and comes from the correct sender.

**CONCLUSION**

In this work, we introduce a digital signature data security system with the ElGamal algorithm applicable to the Digital Signature Algorithm (DSA). The built scheme has rich dynamics as confirmed by the software implementing the DSA key exchange algorithm and the encryption-decryption algorithm has been successfully built. The author also shows an example of digital signature encryption and decryption process by taking any numbers $p = 59419$ and $q = 3301$ to prove that the message can be formed and verified its authenticity.
REFERENCES


