A Novel Mathematical Formal Proof in Unreliability Protocol with XOR in Two's Complement System

Chenglian Liu[1][0000−0002−9086−9740] and Sonia Chien-I Chen[2][0000−0002−6296−4943]

[1] Software Engineering Institute of Guangzhou, Guangzhou 510990, China
liuzl@mail.seig.edu.cn
[2] Qingdao University, Qingdao 266061, China
drsoniachen@qdu.edu.cn

Abstract. Thangavel and Varalakshmi proposed an enhanced DNA and ElGamal cryptosystem for secure data storage and retrieval in cloud. They modified ElGamal algorithm which it calls enhanced ElGamal cryptosystem. We prove that their enhanced ElGamal scheme, which does not require two random numbers by data owner. Although the attacker is unable to find out what message the data owner gave to the data user. However, the attackers can still confuse the issue of sending messages to data users. On the other hand, this scheme can not against insider attack, therefore it is insecure.

Keywords: Enhanced ElGamal Cryptosystem · Forgery Attack · Jamming Attac · Redundancy.

1 Introduction

Digital signatures have become very important with the advent of electronic commerce. Digital signatures provide authentication and data integrity when agreement is required with signer and verifier. There have been several schemes using digital signatures that have been proposed. One scheme that was proposed by Shieh [16] in 2000 used two multi-signatures and was based on the method of Nyberg-Rueppel [12]. This method enabled the receiver to verify and decrypt the message. One advantage was that it used less bandwidth. Another advantage was that message redundancy and one way hash did not have to be used. The problem was that this method was not secure and was subject to forgery attack, thus requiring much crypto-analysis improvement [24]. A new digital signature method was presented by Zhang and Wang [25] that had message recovery, but did not require message redundancy or a one-way hash. One claim they made was that their message could resist forgery. There are the same vulnerabilities in articles such as Wong-Chan scheme [20], KCDSA scheme [7] [14], and Cramer-Shoup scheme [4]. Due to limited conditions, this study lists parts of good contributions, sud as Arbiter PUFs topic [1, 13, 15], Formal verification

topic [10, 17, 6], and Others [8, 23, 22]; but is a little different then what is discussed in this article, please see Table 1. However, in this paper, we will show that their method is still insecure and will not prevent a forgery attack.

Table 1. Related Literatures

Arbiter PUFs	Formal Method	Others
Becker[1]	Stefanowicz et al.[17]	Liu et al.[8]
Podeti et al.[13]	Liu & Venkatesh[10]	Yu & Ciesielski [23]
Santikellur & Chakraborty[15]	Khan[6]	Yang et al.[22]

2 The XOR Operation in Two's Complement

Exclusive Or (XOR) is sometimes used in cryptography as a mixing function with Feistel network systems or a one time pad. It can also be used to detect an overflow of a signed binary operation. If the left most bit of the result is not the same as the number of digits to the left, than that mean an overflow has occurred. If there is an overflow, XORing those two bits will produce a "1". For what it is worth, XORing can be used to swap two variables in computers by using the XOR swap algorithm. However, this is not used in practice, but is considered merely a curiosity. In the following, the author describes the practical issues of a bitwise XOR operation on two variables based on two's complement computer system. He demonstrates that when taking the algebraic properties of XOR into account, new attacks can occur. As an example, he uses two variants of public parameter [8], if XOR is interpreted as free symbol, such as parring, then according to [8] this scheme is insecure.

2.1 Two's Complement System

NOTATION:
$\oplus$: denote bitewise exclusive-or operation.
$()_{10}$: denote decimal expression.
$()_2$: denote binary expression.
$[Pr]$: express probability.

The two's compliment system is used in subtraction because the operands are always added together. Thus, there is no need for additional circuitry to determine if the sign of the second operand is plus or minus. Also, the two's compliment of zero is zero: inverting the "0"s produces all "1"s, and adding "1" gives back "0". Two examples are shown in Table 2 and Table 3.

In [18], Thijssen and Vink show how two's complement arithmetic can be based on mapping the binary arithmetic onto a set of residue classes modulo 2^n. In

Table 2. Example of both odd numbers.

$$(187)_{10} = (10111011)_2.$$
$$(241)_{10} = (11110001)_2.$$
$$(187)_{10} \oplus (241)_{10} = (01001010)_2.$$
$$(187)_{10} \oplus (241)_{10} = (74)_{10}.$$
$$(-87)_{10} = (1111111101000101)_2.$$
$$(-241)_{10} = (1111111100001111)_2.$$
$$(-187)_{10} \oplus (-241)_{10} = (0000000001001010)_2.$$
$$(-187)_{10} \oplus (-241)_{10} = (74)_{10}.$$

Table 3. Example of both even numbers.

$$(108)_{10} = (01101100)_2.$$
$$(116)_{10} = (01110100)_2.$$
$$(108)_{10} \oplus (116)_{10} = (00011000)_2.$$
$$(108)_{10} \oplus (116)_{10} = (24)_{10}.$$
$$(-108)_{10} = (1111111110010100)_2.$$
$$(-116)_{10} = (1111111110001100)_2.$$
$$(-108)_{10} \oplus (-116)_{10} = (00011000)_2.$$
$$(-108)_{10} \oplus (-116)_{10} = (24)_{10}.$$

the two's complement system the integers m from the domain D_2^n: $-2^{n-1} \leq m \leq 2^{n-1} - 1$ are mapped onto $[m] \pmod{2^n}$. This residue class is represented by $R_2^n(m)$, which is the smallest non-negative integer in the residue class $[m] \pmod{2^n}$. So, it can be written

$$0 \leq m \leq 2^{n-1} - 1 \Longleftrightarrow R_2^n(m) = m. \tag{1}$$
$$-2^{n-1} \leq m \leq -1 \Longleftrightarrow R_2^n(m) = 2^n + m. \tag{2}$$

The binary representation of the integer number m is the same to the integer number $R_2^n(m)$.

2.2 The XOR Operation

Chevaler et al. first proposed an NP decision procedure for protocol insecurity with XOR in 2005 [2], but they appled their framework to an intruder that exploits properties of certain encryption modes such as cipher block chaining (CBC).

1. The Boolean Algebra Expression

Table 4 describes the XOR truth table. The XOR operation can be expressed as In the remainder of this section, we assume A, B, C are n bits numbers and F_2 represents the set $\{0, 1\}$. We reexpress Equation (3)

$$A \oplus B = (\neg A \wedge B) \vee (A \wedge \neg B), \tag{3}$$

to

$$A \oplus B = \overline{A}B + A\overline{B}. \tag{4}$$

And rewrite the express as Equation (5).

$$A \oplus B = R_2^n(A) \oplus R_2^n(B),\ A,\ B \in D_2^n. \tag{5}$$

It is a very common component in digital circuit or logic, and often used to many fields such as adder, cryptosystem, image process and so on.

Table 4. The XOR truth table

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

2. The Galois Field Expression

Theorem 1. *Let $\oplus$ be an operation on the set X. It is called commutative if $A \oplus B = B \oplus A,\ \forall A,\ B\ \in X$.*

Proof. Set, $A \leftrightarrow R_2^n(A) = \sum_{i=0}^{n-1} a_i 2^i, a_i \in F_2,$

$$B \leftrightarrow R_2^n(B) = \sum_{i=0}^{n-1} b_i 2^i, b_i \in F_2.$$

So, $A \oplus B = R_2^n(A) \oplus R_2^n(B)$

$$= \sum_{i=0}^{n-1} a_i 2^i \oplus \sum_{i=0}^{n-1} b_i 2^i$$
$$= \sum_{i=0}^{n-1} (a_i \oplus b_i) \cdot 2^i.$$

For the same reason, $B \oplus A = \sum_{i=0}^{n-1} (b_i \oplus a_i) \cdot 2^i$.
Since $a_i, b_i \in F_2$, then, $a_i \oplus b_i \equiv a_i + b_i \pmod 2$,
$b_i \oplus a_i \equiv a_i + b_i \pmod 2$,
Hence $b_i \oplus a_i = a_i \oplus b_i$.
Therefore $A \oplus B = B \oplus A$.

Theorem 2. *Let $\oplus$ be an operation in the set X. It is called associative if $(A \oplus B) \oplus C = A \oplus (B \oplus C)$ for all $A, B \in X$.*

Proof. *set*, $A \leftrightarrow R_2^n(A) = \sum_{i=0}^{n-1}(a_i 2^i),\ B \leftrightarrow R_2^n(B) = \sum_{i=0}^{n-1}(b_i 2^i),\ C \leftrightarrow R_2^n(C) = \sum_{i=0}^{n-1}(c_i 2^i),\ a_i,\ b_i,\ c_i \in F_2$.

So, $(A \oplus B) \oplus C = (R_2^n(A) \oplus R_2^n(B)) \oplus R_2^n(C)$

$$= \sum_{i=0}^{n-1}(a_i \oplus b_i)2^i \oplus \sum_{i=0}^{n-1} c_i 2^i$$
$$= \sum_{i=0}^{n-1}[(a_i \oplus b_i \oplus c_i)]2^i$$
$$= \sum_{i=0}^{n-1}[(a_i + b_i + c_i) \pmod 2]2^i.$$

The same reason, we have,

$$A \oplus (B \oplus C) = \sum_{i=0}^{n-1}[(a_i + b_i + c_i) \pmod 2]2^i.$$

Thus, $(A \oplus B) \oplus C = A \oplus (B \oplus C)$.

Theorem 3. *If* $A = B$, *then* $A \oplus B = \overbrace{0000\ldots0000}^{n\ bits}$.

Proof. Set, $A \leftrightarrow R_2^n(A) = \sum_{i=0}^{n-1}(a_i 2^i),\ B \leftrightarrow R_2^n(B) = \sum_{i=0}^{n-1}(b_i 2^i),\ a_i,\ b_i, \in F_2$.

Then, $A \oplus B = R_2^n(A) \oplus R_2^n(B)$

$$= \sum_{i=0}^{n-1}[(a_i + b_i) \pmod 2]2^i.$$

If $A = B$, that is $a_i = b_i,\ 0 \le i \le n-1$,

Then, $a_i + b_i \equiv 0 \pmod 2$.

We have $A \oplus B = \overbrace{0000\ldots0000}^{n\ bits}$.

Theorem 4. *If* A, B *are odd integer,* $(A) \oplus (-A) = \overbrace{1111\ldots111}^{n-1\ bits}0$, *and* $(B) \oplus (-B) = \overbrace{1111\ldots111}^{n-1\ bits}0$, *then* $(A \oplus B) = (-A \oplus -B)$.

Proof. Because of the symmetry, we assume $A \ge 0$. So, $A \leftrightarrow R_2^n(A) = \sum_{i=0}^{n-1}(a_i 2^i)$, $a_i \in F_2$.

Suppose A is odd, so, $R_2^n(A) = 1 \cdot 2^0 + \sum_{i=1}^{n-1}(a_i 2^i),\ a_i \in F_2$,

$$-A \leftrightarrow R_2^n(-A) = 2^n - A$$

$= (2^n - 1) - A + 1 = \sum_{i=0}^{n-1} 2^i - \sum_{i=0}^{n-1} a_i 2^i + 1$, while $(a_0 = 1) = (1-1)2^0 + \sum_{i=1}^{n-1}(1-a_i)2^i + 1 \cdot 2^0 = 1 \cdot 2^0 + \sum_{i=1}^{n-1}(1-a_i)2^i$

Therefore, $A \oplus -A = R_2^n(A) \oplus R_2^n(-A) = [(1+1) \pmod 2] \cdot 2^0 + \sum_{i=1}^{n-1}[(1 - a_i + a_i) \pmod 2] \cdot 2^i$

$$= 0 \cdot 2^0 + \sum_{i=1}^{n-1} 2^i$$

$$= \overbrace{1111\ldots111}^{n-1 \quad bits}0.$$

For the same reason, $B \oplus -B = \overbrace{1111\ldots111}^{n-1 \quad bits}0$.
Then, $A \oplus (-A) = B \oplus (-B)$.
So, $(A \oplus (-A)) \oplus (B \oplus (-B)) = \overbrace{0000\ldots0000}^{n \ bits}$.
Promptly, $(A \oplus B) \oplus (-A \oplus -B) = \overbrace{0000\ldots0000}^{n \ bits}$.
Thus, $A \oplus B = -A \oplus -B$.

Theorem 5. *If A, B are even integers, where $4 \mid A$, B and $8 \nmid A, B$, then $A \oplus B = -A \oplus -B$.*

Proof. We also assume, $A \geq 0$, then set
$A \leftrightarrow R_2^n(A) = A = \sum_{i=0}^{n-1}(a_i 2^i)$, $a_i \in F_2$.
Since A is even number and $4|A$, but $8 \nmid A$
Therefore $A \leftrightarrow R_2^n(A) = 0 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + \sum_{i=3}^{n-1} a_i 2^i$, $a_i \in F(2)$,
$-A \leftrightarrow R_2^n(-A) = 2^n - A$

$$= 2^n - 1 - A + 1$$

$$= \sum_{i=0}^{n-1} 2^i - (0 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + \sum_{i=3}^{n-1} a_i 2^i) + 1 \cdot 2^0$$

$$= 1 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + \sum_{i=3}^{n-1}(1 - a_i)2^i + 1$$

$$= 0 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + \sum_{i=3}^{n-1}(1 - a_i)2^i.$$

So, $A \oplus -A = R_2^n(A) \oplus R_2^n(-A)$

$$= (0 \oplus 0) \cdot 2^0 + (0 \oplus 0) \cdot 2^1 + (1 \oplus 1) \cdot 2^2 + \sum_{i=3}^{n-1}((1 - a_i) \oplus a_i)2^i$$

$$= 0 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + \sum_{i=3}^{n-1} 2^i$$

$$= \overbrace{1111\ldots1}^{n-3 \quad bits}000.$$

For the same reason, $B \oplus -B = \overbrace{1111\ldots1}^{n-3 \quad bits}000$.
Explicitly $A \oplus -A = B \oplus -B$.
Then $(A \oplus -A) \oplus (B \oplus -B) = \overbrace{0000\ldots0000}^{n \ bits}$.
According from commutative law, we get $A \oplus B = -A \oplus -B$.

Theorem 6. *If A, B are even integers, where $4 \nmid A$, B but $4 \mid |A - B|$, then $A \oplus B = -A \oplus -B$.*

Proof. We also assume, $A \geq 0$, then set $A \leftrightarrow R_2^n(A) = A = \sum_{i=0}^{n-1}(a_i 2^i)$, $a_i \in F_2$
Since A is even number and $4 \nmid A$, so

$$A \leftrightarrow R_2^n(A) = 0 \cdot 2^0 + 1 \cdot 2^1 + \sum_{i=2}^{n-1} a_i 2^i,\ a_i \in F_2.$$

$-A \leftrightarrow R_2^n(-A) = 2^n - A$

$$= (2^n - 1) - A + 1$$

$$= \sum_{i=0}^{n-1} 2^i - A + 1$$

$$= (1 - 0) \cdot 2^0 + (1 - 1) \cdot 2^1 + \sum_{i=2}^{n-1}(1 - a_i)2^i + 1$$

$$= 0 \cdot 2^0 + 0 \cdot 2^1 + \sum_{i=2}^{n-1}(1 - a_i)2^i.$$

Namely $A \oplus -A = R_2^n(A) \oplus R_2^n(-A)$

$$= (0 \oplus 0) \cdot 2^0 + (1 \oplus 0) \cdot 2^1 + \sum_{i=2}^{n-1}[(1 - a_i) \oplus a_i] \cdot 2^i$$

$$= 0 \cdot 2^0 + 1 \cdot 2^1 + \sum_{i=2}^{n-1} 2^i$$
$$= \overbrace{1111\ldots11}^{n-1 \quad bits}0.$$

For the same reason, we have $B \oplus -B = \overbrace{1111\ldots11}^{n-1 \quad bits}0$.
Hence $A \oplus -A = B \oplus -B$.

Then $(A \oplus -A) \oplus (B \oplus -B) = \overbrace{0000\ldots0000}^{n \quad bits}$.
This is to say $A \oplus B = -A \oplus -B$.

3 Review of Unreliable Zhang-Wang Scheme

A signature scheme was proposed by Zhang and Wang [25] in 2005 that did not use a one-way hash function. In this scheme, the signer would randomly select their private key 'x' where $gcd(x, p-1) = 1$, and then would compute the public key $y \equiv g^x \pmod p$. In this scheme, there are two phases: the signature generation phase and verification phase which are described in the next section.

3.1 Signature Generation Phase:

Suppose the signer wants to sign the message M, and then he executes the following steps:

Step 1. The signer computes

$$s \equiv (y + M)^{M \pmod{p-1}} \pmod p. \tag{6}$$

Step 2. The signer chooses a random number $k \in Z_{p-1}^*$ and computes

$$r \equiv M \cdot s \cdot g^{-k} \pmod p. \tag{7}$$

Step 3. The signer computes t where

$$s + t \equiv x^{-1} \cdot [k - (r \oplus s)] \pmod{p-1}. \tag{8}$$

Step 4. The signer transmits parameters $(s,\ r,\ t)$ of M to the verifier.

3.2 Verification Phase:

When the verifier receives parameters $(s,\ r,\ t)$ by signer, hey can check the validity of the signature by doing the following:

Step1. The verifier computes

$$M' \equiv y^{s+t} \cdot r \cdot g^{r \oplus s} \cdot s^{-1} \pmod p. \tag{9}$$

Step2. The verifier checks

$$s \equiv (y+M)^{M \pmod{p-1}} \pmod{p}. \tag{10}$$

If it holds, it shows that the signature $(s,\ r,\ t)$ is valid.

Proof.

$$\begin{aligned} M' &\equiv y^{s+t} \cdot r \cdot g^{r \oplus s} \cdot s^{-1} \pmod{p}. \\ &\equiv y^{s+t} \cdot M \cdot s \cdot g^{-k} \cdot g^{r \oplus s} \cdot s^{-1} \pmod{p}. \\ &\equiv g^{k-(r \oplus s)} \cdot M \cdot g^{-k+(r \oplus s)} \pmod{p}. \\ &\equiv M \pmod{p}. \end{aligned} \tag{11}$$

3.3 Our Attack Method

We analyze the probability of the parameters r and the s.

$$(r,s) \Rightarrow \begin{cases} r,s \text{ are both odd numbers, the } [Pr = \frac{1}{4}]. \\ r,s \text{ are one odd and one even, the } [Pr = \frac{1}{2}]. \\ r,s \text{ are both even numbers, the } [Pr = \frac{1}{4}]. \end{cases}$$

A forger wants to fake a validation signature, then she executes the following steps:

Step 1. Set

$$r' = -r. \tag{12}$$

Step 2. Set

$$s' = -s. \tag{13}$$

Step 3. Set

$$t' = 2s + t. \tag{14}$$

Step 4. Foger sends $(r',\ s',\ t')$ signatures to Bob, and successful forge the signature.

Proof.

$$\begin{aligned} M' &\stackrel{?}{\equiv} y^{s'+t'} \cdot (r') \cdot g^{(r' \oplus s')} \cdot (s')^{-1} \pmod{p}. \\ &\equiv y^{s'+t'} \cdot M \cdot (s') \cdot g^{-k} \cdot g^{(r' \oplus s')} \cdot (s')^{-1} \pmod{p}. \\ &\equiv g^{k-(r' \oplus s')} \cdot M \cdot g^{-k+(r' \oplus s')} \pmod{p}. \\ &\equiv M' \pmod{p}. \end{aligned} \tag{15}$$

4 Review of Unreliable Hwang et al.'s Scheme

In 2002, Hwang et al. [5] proposed an ElGamal-like cryptosystem for enciphering large messages scheme, the detailed as following.

4.1 The ElGamal Cryptosystem

The ElGamal [3] cryptosystem based on discrete logarithms and proposed in 1985. Let p is a large prime number, and g is primitive root where $g \in \mathbb{Z}_p$, and compute the public key $y_i \equiv g^{x_i} \pmod{p}$. The x_i denotes secret key. Here p, g and y are public information, the x_i and r are private information. If user u_i want to deliver the message $m(0 \leq m \leq p-1)$ to u_j, u_i randomly chooses an integer r and then encrypts m as below:

$$b \equiv g^r \pmod{p}. \tag{16}$$

$$c \equiv m \cdot y_i^r \pmod{p}. \tag{17}$$

u_i sends (b, c) to u_j. When u_j receives (b, c), u_j decrypts c as follows:

$$m \equiv c \cdot (b^{x_j})^{-1} \pmod{p}. \tag{18}$$

The cipher c depends on two issues: the original plaintext m, and a random integer r. A second random number r is mapped from another cipher text c, but from the same plaintext m. The ElGamal cryptosystem has two restrictions: the first one is the random number r which cannot be repeated, and the second restriction is that the message r must be less than $p-1$.

4.2 The Hwang et al.'s Scheme

Let 'p' be a large prime number and let 'g' be an element of $GF(p)$. Each user u_i will randomly chose a private key x_i (which is an element of $\mathbb{Z}_p$). The user then computes the public key $y_i \equiv g^{x_i} \pmod{p}$. Variables "$p$", "$g$", and "$y$" are public information. Any user u_i who wants to send a message m_i would use the following steps:

Step 1. Slice plaintext m_i into t pieces $m_1, m_2, \cdots, m_t$. Each piece is 512 bits long.

Step 2. Generate two random integers r_1 and r_2, where $1 < r_1, r_2 \leq p-1$, and compute b_1 and b_2 as follows:

$$b_1 \equiv g^{r_1} \pmod{p}. \tag{19}$$

$$b_2 \equiv g^{r_2} \pmod{p}. \tag{20}$$

Step 3. Compute $C_j, j = 1, 2, \cdots, t$ as follows:

$$C_j \equiv m_j \cdot (y_i^{r_i} \oplus (y_i^{r_2})^{2^j}) \pmod{p}. \tag{21}$$

Step 4. Send $\{b_1, b_2, C_j, j = 1, 2, \cdots, t\}$ to the receiver through a public channel.

After receiving $\{b_1, b_2, C_j, j = 1, 2, \cdots, t\}$ from the sender, the receiver recovers the plaintext m_i from following:

$$m_j \equiv C_j \cdot (b_1^{x_i} \oplus (b_2^{x_i})^{2^j})^{-1} \pmod{p}. \tag{22}$$

4.3 Security Analysis

A practical secure anonymous user authentication scheme was proposed by Lyuu et. al [11] and Wang et.al [19] which described several attacks. In Hwang's scheme, there are vulnerabilities when the XOR operation is combined with the two's complement number system. The related articles can be found in [8] [9] [21]. The authors describe follow situations.

$$(y_i^{r_1}, (y_i^{r_2})^{2^j}) \Rightarrow \begin{cases} \text{both odd numbers, the } [Pr = \frac{1}{4}]. \\ \text{one odd and even numbers, } [Pr = \frac{1}{2}]. \\ \text{both even numbers, the } [Pr = \frac{1}{4}]. \end{cases}$$

The attacker can easy to fake the valid parameters $(y_i^{r_1}, (y_i^{r_2})^{2^j})$ where $y_i^{r_1} \oplus (y_i^{r_2})^{2^j} \stackrel{?}{\equiv} (-y_i^{r_i} \oplus -(y_i^{r_2})^{2^j})$. She does follow steps:

Step 1. Set $-r = -y_i^{r_1}$.
Step 2. Set $-s = -(y_i^{r_2})^{2^j}$.

$$\begin{aligned} C_j &\equiv m_j \cdot (r \oplus s) \pmod{p} \\ &\stackrel{?}{\equiv} m_j \cdot (-r \oplus -s) \pmod{p}. \end{aligned} \tag{23}$$

From Theorem 1 to Theorem 6, we obtain $C_j \equiv m_j \cdot (r \oplus s) \pmod{p} \stackrel{?}{\equiv} m_j \cdot (-r \oplus -s) \pmod{p}$ are both odd numbers or even numbers where they matches specified rules. Now, it clearly describes from equation (23). The attacker successfully executes the forgery attack.

4.4 Other Unreliable Examples

The Cramer-Shoup Strong-RSA Signature Scheme Revisited, proposed by Fischlin [4] in 2003, gave Type II and Type III forgery attacks. Here,

$$h_1^{-\alpha_j} h_2^{-(\alpha_j \oplus H(m_i))} y_j^r \equiv x \equiv h_1^{-\alpha} h_2^{-(\alpha \oplus H(m))} y^r \pmod{n}. \tag{24}$$

We see the scheme format is based on the '$(A \oplus B) \equiv (-A \oplus -B)$' style in which two variables are numeric. We can, therefore, use our methodology to fake a valid parameter/signature under two's complement number system. In [7] and [14], these situations also existed.

5 Conclusions

The XOR operation is a good way to prevent the "multiplicative property of algebra" attacks. There are three advantages: 1) reduced costs, 2) easy hardware implementation and 3) increased speed. However, according to our analysis, they are both insecure. Mathematical methods are one way to determine if the scheme is correct. A mathematical proof can determine if the resulting "logical inference" is consistent with the previous stage, but cannot guarantee if there are defects in the "logical inference" process. In this paper, the author has used some examples such as the Zhang Wang signature scheme, Hwang's scheme, and the Cramer Shoup Strong RSA signature scheme revised because they used exclusive or bitwise operation.

References

1. Becker, G.T.: The gap between promise and reality: On the insecurity of XOR arbiter PUFs. In: Güneysu, T., Handschuh, H. (eds.) Cryptographic Hardware and Embedded Systems–CHES 2015. pp. 535–555. Springer Berlin Heidelberg, Berlin, Heidelberg (2015)
2. Chevalier, Y., Küsters, R., Rusinowitch, M., Turuani, M.: An NP decision procedure for protocol insecurity with XOR. Theoretical Computer Science **338**(1-3), 247–274 (June 2005)
3. ElGamal, T.: A public-key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transaction on Information Theory **IT-31**(4), 469–472 (July 1985)
4. Fischlin, M.: The Cramer-Shoup Strong-RSA Signature Scheme Revisited. In: Desmedt, Y. (ed.) Public Key Cryptography (PKC) 2003. vol. 2567, pp. 116–129 (2002)
5. Hwang, M.S., Chang, C.C., Hwang, K.F.: An ElGamal-like cryptosystem for enciphering large messages. IEEE Trans. on Knowl. and Data Eng. **14**(2), 445–446 (March 2002)
6. Khan, W., Kamran, M., Naqvi, S.R., Khan, F.A., Alghamdi, A.S., Alsolami, E.: Formal verification of hardware components in critical systems. Wireless Communications and Mobile Computing **2020**, 7346763 (February 2020), https://doi.org/10.1155/2020/7346763
7. Li, F., Xu, C.: An improved identity-based KCDSA signcryption scheme. In: The First International Symposium on Data, Privacy, and E-Commerce (ISDPE). pp. 230–232 (November 2007). https://doi.org/10.1109/ISDPE.2007.128
8. Liu, C., Chen, S., Sun, S.: Security of analysis mutual authentication and key exchange for low power wireless communicationsi. Energy Procedia **17, Part A**, 644–649 (2012)
9. Liu, C., Zhang, J.: Security analysis of zhang-wang digital signature scheme. Communications of the CCISA **15**(4), 24–29 (October 2009), (Chinese version)
10. Liu, Z., Venkatesh, R.: Methods and Tools for Formal Software Engineering, pp. 31–41. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
11. Lyuu, Y.D., Wu, M.L.: Cryptanalysis of an elgamal-like cryptosystem for enciphering large messages. WSEAS Transactions on Information Science and Applications **1**(4), 1079–1081 (October 2004)
12. Nyberg, K., Rueppel, R.A.: Message recovery for signature schemes based on the discrete logarithm problem. Advances in Cryptology-EUROCRYPT'94 (May 1994)
13. Podeti, R., Patri, S.R., P., M.: Highly reliable XoR feed arbiter physical unclonable function (XFAPUF) in 180Â nm process for IoT security. Microprocessors and Microsystems **87**, 104355 (2021)
14. Ryu, J.H., Jeong, Y.S., il Seo, D.: Identity based KCDSA signcryption. In: The 8th International Conference on Advanced Communication Technology (ICACT). vol. 2, pp. 1369–1374 (February 2006). https://doi.org/10.1109/ICACT.2006.206227
15. Santikellur, P., Chakraborty, R.S.: A computationally efficient tensor regression network-based modeling attack on XOR arbiter PUF and its variants. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **40**(6), 1197–1206 (2021). https://doi.org/10.1109/TCAD.2020.3032624
16. Shieh, S.P., Lin, C.T., Yang, W.B., Sun, H.M.: Digital multisignature schemes for authenticating delegates inmobile code systems. IEEE Transactions on Vehicular Technology (July 2000)

17. Stefanowicz, A., Kyle, J., Grove, M.: Proofs and mathematical reasoning. University of Birmingham (2014), https://www.birmingham.ac.uk/Documents/college-eps/college/stem/Student-Summer-Education-Internships/Proof-and-Reasoning.pdf
18. Thijssen, A.P., Vink, H.A.: Two's complement arithmetic. Contents of Proceedings Electronics 1998 **3**(1), 150–156 (1998)
19. Wang, M.N., Yen, S.M., Wu, C.D., Lin, C.T.: Cryptanalysis on an ElGamal-like cryptosystem for encrypting large messages. In: Proceedings of the 6th WSEAS International Conference on Applied Informatics and Communications. pp. 418–422. World Scientific and Engineering Academy and Society (WSEAS) (2006)
20. Wong, D.S., Chan, A.H.: Mutual authentication and key exchange for low power wireless communications. In: The IEEE Military Communications Conference (MILCOM). Communications for Network-Centric Operations: Creating the Information Force. (2001)
21. Xu, L., Liu, C., Wang, N.: Comment on an improved signature without using one-way hash functions. In: 2010 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC). pp. 441–443 (October 2010)
22. Yang, J., Tong, X., Liu, C., Chen, S.C.I.: A novel mathematical formal proof in Zhang-Wang's cryptographic algorithm. Converter **2021**(2), 449–458 (2021)
23. Yu, C., Ciesielski, M.: Formal analysis of galois field arithmetic circuits-parallel verification and reverse engineering. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **38**(2), 354–365 (2019). https://doi.org/10.1109/TCAD.2018.2808457
24. Zhang, F.: Cryptanalysis of Chang et al. signature scheme with message recovery. IEEE Communications Letters **9**(4), 358–359 (April 2005)
25. Zhang, J., Wang, Y.: An improved signature scheme without using one-way hash functions. Applied Mathematics and Computation **170**(2), 905–908 (November 2005). https://doi.org/9533 10.1016/j.amc.2004.12.028