# A note on "a lightweight mutual and transitive authentication mechanism for IoT network"

Zhengjun Cao and Lihua Liu

**Abstract**. We show the authentication mechanism [Ad Hoc Networks, 2023, 103003] fails to keep user anonymity, not as claimed.

**Keywords**: Chebyshev polynomials, Mutual authentication, Anonymity, Key agreement, Impersonation attack

## 1 Introduction

The security of cryptosystems based on Chebyshev recursive relation, $T_n(x) = 2x\,T_{n-1}(x) - T_{n-2}(x)$, relies on the difficulty to find the large degree $n$. Notice that

$$T_n(x) = \cos(n \arccos x), x \in [-1, 1],\ n = 0, 1, 2, \cdots,$$

Hence, we have

$$\begin{aligned} T_n(T_m(x)) &= \cos(n \arccos(\cos(m \arccos x))) \\ &= \cos(nm \arccos x) = T_{nm}(x) \end{aligned}$$

which is just the so-called semi-group property of Chebyshev polynomials. For example,

$$\begin{aligned} &T_0(x) = 1, \quad T_1(x) = x, \\ &T_2(x) = -1 + 2x^2, \quad T_3(x) = -3x + 4x^3 \\ &T_4(x) = 1 - 8x^2 + 8x^4, \\ &T_5(x) = 5x - 20x^3 + 16x^5 \\ &T_6(x) = -1 + 18x^2 - 48x^4 + 32x^6, \\ &\quad \cdots \end{aligned}$$

By the above polynomial equality, we have

$$T_n(T_m(a)) = T_{nm}(a) \bmod N$$

for some integers $a$ and $N$. The seed $a$ and degree $n$ can be kept secret so as to construct chaotic maps [1].

Z. Cao is with Department of Mathematics, Shanghai University, Shanghai, China.
L. Liu is with Department of Mathematics, Shanghai Maritime University, Shanghai, China. Email: liulh@shmtu.edu.cn

Very recently, Krishnasrija et al. [2] have presented a mutual and transitive authentication mechanism for IoT network based on chaotic maps. The Krishnasrija-Mandal-Cortesi authentication and key agreement scheme is designed to meet many requirements, such as mutual authentication and key agreement, perfect forward secrecy, user anonymity (untraceablity), and resistance against replay attack, man in the middle attack, offline password guessing attack, privileged insider attacks, known session key secrecy attack, and stolen smart device attack. Though the Krishnasrija-Mandal-Cortesi scheme is interesting, we find it fails to keep user anonymity.

## 2 Review of the Krishnasrija-Mandal-Cortesi authentication mechanism

In the considered scenario, there are three entities: trusted devices, new devices willing to join the network, and gateway. The network needs to handle two different connection requests: device-to-gateway, and device-to-device. The involved notations are listed below (Table 1).

Table 1: Notations and descriptions

| Notation | Description |
|---|---|
| $ID_n$ | identity of $n$-th user |
| $PW_n$ | password of $n$-th user |
| $T_i$ | the $i$-th timestamp |
| $T_x(\cdot)$ | Chebyshev polynomial of degree $x$ |
| $N, N_t, N_p, B_g$ | random nonces |
| $\oplus$ | bitwise XOR |
| $a\|\|b$ | concatenation of strings $a$ and $b$ |

Its registration phase can be described as below.

(1) The user inputs the identity $ID_n$ and password $PW_n$. Then pick a nonce $R$ and a timestamp $T_1$ to compute

$$PID = ID_n \oplus R, \;\; PIN = PW_n \oplus T_1.$$

Send $\{PID, PIN, R, T_1\}$ to the gateway via a secure channel.

(2) Upon receiving the registration request, the gateway generates the timestamp $T_2$ to check $T_1$. Then pick two random numbers $s, x$ and a big prime $z$ to compute

$$ID_n = PID \oplus R, \;\; T_s(x) = 2xT_{s-1}(x) - T_{s-2}(x),$$
$$z' = z \oplus PID, \;\; CK = H(PIN \oplus PID) \oplus z'$$

Store $\{ID_n, s, x, T_s(x), z\}$. Send $\{CK, s, T_s(x), T_2\}$ to the device.

(3) The user generates the timestamp $T_3$ to check $T_2$. If true, compute $v = z \oplus H(ID_n||PW_n)$ and store $\{v, x, T_s(x)\}$ in the device.

Its authentication and key agreement phase can be described and depicted as below (Table 2).

(1) The user inputs $ID_n, PW_n$. Compute $z_n = v_n \oplus H(ID_n||PW_n)$. Check if $v_n = H(z_n \oplus T_{s_n}(x_n)$.

Pick nonce $N, r_n$ and timestamp $T_1$ to compute

$$k_{rs} = T_{r_n}(T_{s_n}(x_n)),\ e = (v_n \| N) \oplus k_{rs},\ AID_n = ID_n \oplus H(T_1)$$

Send $\{AID_n, e, T_{r_n}(x_n), T_1\}$ to the trusted device via an open channel.

(2) The trusted device generates the timestamp $T_2$ to check $T_1$. Then pick nonce $N_t, N_p$ to compute $ID_n = AID_n \oplus H(T_1)$,

$$AID'_n = ID_n \oplus H(T_2),\ AID_t = ID_t \oplus H(T_2),$$
$$M = (e \| T_{r_n}(x_n) \| N_p \| N_t \| AID'_n \| ID'_t),\ M_t = M \oplus H(SK_t)$$

Send $\{AID_t, M_t, T_2\}$ to the target gateway.

(3) The gateway generates the timestamp $T_4$ to check $T_3$ and $ID_t = AID_t \oplus H(T_2)$. Then compute

$$M_t \oplus H(SK_t) = e \| T_{rn}(x_n) \| N_p \| N_t \| AID'_t \| ID'_t$$

Check $ID'_t = ID_t \oplus H(SK_t)$. If so, compute $ID_n = AID'_n \oplus H(T_2)$, $k_{rs} = T_{s_n}(T_{r_n}(x_n))$, $(v_n \| N) = e \oplus k_{rs}$. Check if $v_n = H(z_n \oplus T_{s_n}(x_n))$. Then compute $N'_t = H(N_t)$, $N'_g = H(N) \oplus k_{rs}$. Set the session key as $SK_{gn} = H(k_{rs} \| N)$. It then computes

$$N'_p = N_p \oplus H(z_n \oplus T_{s_n}(x_n)),\ M_g = (N'_t \| N'_g \| N'_p) \oplus SK_t$$

Send $\{M_g, T_3\}$ to the trusted device.

(4) The trusted device generates the timestamp $T_4$ to check $T_3$. Then compute $(N'_t, N'_g, N'_p) = M_g \oplus SK_t$. Check $N'_t = H(N_t)$. If so, compute the session key $SK_{tn} = H(N'_g \| N_p)$. Send $\{N'_g, N'_p, T_4\}$ to the user.

(5) The user generates the timestamp $T_5$ to check $T_4$. Then check $N'_g = H(N) \oplus k_{rs}$. If so, compute $N_p = N'_p \oplus v_n$. Set the session keys as $SK_{tn} = H(N'_g \| N_p)$, and $SK_{gn} = H(K_{rs} \| N)$.

The correctness is due to that

$$k_{rs} = T_{r_n}(T_{s_n}(x_n)) = T_{s_n}(T_{r_n}(x_n))$$

## 3 The loss of user anonymity

In the considered security model, the user anonymity refers to the ability to determine a user's true identity. The untraceablity means that an adversary cannot determine a target user identity and check if two sessions are being run by the same user. In the scheme, a user with the identity $ID_n$ needs to pick a timestamp $T_1$ and compute the pseudonym

$$AID_n = ID_n \oplus H(T_1) \tag{1}$$

Then send $\{AID_n, e, T_{r_n}(x_n), T_1\}$ to the target trusted device.

Notice that the communication channel between the user and the trusted device is assumed to be a common open channel, not a secure channel. So, an adversary can capture the message $\{AID_n, e, T_{r_n}(x_n), T_1\}$ to recover the parameters $AID_n$ and $T_1$. The adversary then invokes the

public hash function $H(\cdot)$ to retrieve the true identity by computing

$$ID_n = AID_n \oplus H(T_1) \tag{2}$$

Table 2: The Krishnasrija-Mandal-Cortesi authentication and key agreement scheme

| Device: $\{v_n, x_n, T_{s_n}(x_n)\}$ | Trusted device: $\{v_t, x_t, T_{s_t}(x_t)\}$ | Gateway: $\{ID_n, s_n, x_n, T_{s_n}(x_n), z_n\}$, $\{ID_t, s_t, x_t, T_{s_t}(x_t), z_t\}$, |
|---|---|---|
| Input $ID_n, PW_n$. Compute $z_n = v_n \oplus H(ID_n \| PW_n)$. Check if $v_n = H(z_n \oplus T_{s_n}(x_n)$. Pick nonce $N, r_n$ and timestamp $T_1$ to compute $k_{rs} = T_{r_n}(T_{s_n}(x_n))$, $e = (v_n \| N) \oplus k_{rs}$, $AID_n = ID_n \oplus H(T_1)$. | | |
| $\xrightarrow[\text{[open channel]}]{AID_n,\ e,\ T_{r_n}(x_n),\ T_1}$ | | |
| | Generate the timestamp $T_2$ and check $T_1$. Pick nonce $N_t, N_p$. Compute $ID_n = AID_n \oplus H(T_1)$, $AID'_n = ID_n \oplus H(T_2)$, $AID_t = ID_t \oplus H(T_2)$, $M = (e \| T_{r_n}(x_n) \| N_p \| N_t \| AID'_n \| ID'_t)$, $M_t = M \oplus H(SK_t)$. | |
| | $\xrightarrow{AID_t,\ M_t,\ T_2}$ | |
| | | Generate the timestamp $T_4$ and check $T_2$. Check $ID_t = AID_t \oplus H(T_2)$. Compute $M_t \oplus H(SK_t) = e \| T_{r_n}(x_n) \| N_p \| N_t \| AID'_t \| ID'_t$. Check $ID'_t = ID_t \oplus H(SK_t)$. Compute $ID_n = AID'_n \oplus H(T_2)$, $k_{rs} = T_{s_n}(T_{r_n}(x_n))$, $(v_n \| N) = e \oplus k_{rs}$. Check $v_n = H(z_n \oplus T_{s_n}(x_n))$. Then compute $N'_t = H(N_t)$, $N'_g = H(N) \oplus k_{rs}$, $SK_{gn} = H(k_{rs} \| N)$, $N'_p = N_p \oplus H(z_n \oplus T_{s_n}(x_n))$, $M_g = (N'_t \| N'_g \| N'_p) \oplus SK_t$. |
| | Generate the timestamp $T_4$ and check $T_3$. Compute $(N'_t, N'_g, N'_p) = M_g \oplus SK_t$. Check $N'_t = H(N_t)$. If so, compute $SK_{tn} = H(N'_g \| N_p)$. | |
| | $\xleftarrow{N'_g,\ N'_p,\ T_4}$ | |
| | | $\xleftarrow{M_g,\ T_3}$ |
| Check the timestamp $T_4$. Check if $N'_g = H(N) \oplus k_{rs}$. Compute $N_p = N'_p \oplus v_n$, $SK_{tn} = H(N'_g \| N_p)$, $SK_{gn} = H(K_{rs} \| N)$. | | |

The original argument claims that: *In the transitive protocol, the adversary cannot get user identity $ID_n$ from the communication message directly. The new device identity $ID_n$ is hidden in auxiliary identity $AID_n$ while sending it to a trusted device and gateway.*

The argument is flawed because it wrongly treats the hash function $H(\cdot)$ as a confidential thing. It has confused a general *hash function* with a *keyed hash function* [3]. We want to stress that a general hash function is publicly accessible. An adversary can invoke it to derive the fingerprint corresponding to a target input.

By the way, the trusted device identity $ID_t$ is eventually exposed. In fact, the adversary who has captured the message $\{AID_t, M_t, T_2\}$ via the open channel, can recover the identity by computing

$$ID_t = AID_t \oplus H(T_2) \tag{3}$$

## 4 Conclusion

We show that the Krishnasrija-Mandal-Cortesi authentication and key agreement scheme cannot provide user anonymity, because it has confused the general hash function with a keyed hash function. The findings in this note could be helpful for the future work on designing such schemes.

## References


[1] Z. Cao, L. Liu, A note on the insecurity of cryptosystems based on Chebyshev polynomials, Int. J. Bifurc. Chaos 32(3) (2022), 2250044:1-8.

[2] R. Krishnasrija, A. K. Mandal, A. Cortesi, A lightweight mutual and transitive authentication mechanism for IoT network, Ad Hoc Networks 138 (2023), 103003.

[3] A. Menezes, P. Oorschot, S. A. Vanstone, Handbook of applied cryptography, CRC Press 1996.