# Extension of PCL Theory and Its Application in Improved CCITT X.509 Analysis

Lei Yu[1], Zhi-Yao Yang[2], and Ze-Peng Zhuo[2]
(Corresponding author: Lei Yu)

School of Computer Science and Technology, Huaibei Normal Universtiy[1]
School of Mathematical Sciences, Huaibei Normal University [2]
Huaibei, Anhui 235000, China
(Email: yulei@chnu.edu.cn)

(Received Sept. 5, 2019; Revised and Accepted Dec. 6, 2020; First Online Apr. 3, 2020)

## Abstract

In the original PCL theory, due to the lack of a strict definition and inference rules of the relations among message subterms, the protocol analysis process was described as having neither rigor nor formalization, which seriously affected the accuracy of the analysis results. Secondly, The temporal ordering between the actions of the principals is the key basis for judging whether the principals correctly perform the roles of the protocol or not. The analysis on it based on timestamp mechanism which can directly reflect the temporal ordering of the actions of the principals, will greatly reduce the complexity of protocol analysis. However, there are no verification or inference rules based on the timestamp mechanism for the temporal relationship between the acts of protocol principals in the existing PCL theory. Accordingly, this paper is aimed to extend the PCL theory from two aspects: Message subterms relationship and timestamp mechanism. First, the inference rules of message subterms are given on the basis of a strict definition of the relations between message subterms. Secondly, based on the defined timestamp relations and the original PCL inference system, the rules for judging the temporal ordering of the receiving and sending behavior of protocol principals are given. To verify the validity of PCL extension theory, the conciseness of PCL in protocol security analysis and the correctness of improved CCITT X.509 protocol, a formal description of the improved CCITT X.509 protocol is given by using cue calculus language, and a formal description of the security properties of the protocol is given by using PCL logic. And then, the security analysis of the protocol is given by using the extended PCL theory in three areas: Authentication, confidentiality and data integrity. The process and results of protocol analysis show that the extended PCL theory can effectively reduce the complexity of protocol analysis, and the improved CCITT X.509 protocol can meet the goal of protocol security attribute design.

*Keywords: CCITT X.509; Formal Analysis Method; Protocol Composition Logic; Security Protocol*

## 1 Introduction

Security protocol has become the basis of Cyberspace Security [20]. It is very complex to design a correct security protocol. Protocol defect analysis has become the main method and means of security protocol design [4, 16, 23]. At present, formal method has been proved to be the most scientific, rigorous and effective method of security protocol defect analysis [1–3]. Protocol Composition Logic (PCL) [7, 9], proposed by Datta, Derek and Michell in 2003, is a formal design and analysis method of security protocols based on Floyd-Hoare logic. The formal proof system of PCL consists of protocol modeling system, protocol logic and proof system. In the protocol modeling system, PCL uses cryptographic primitives to describe the basic elements of the protocol, such as sending and receiving messages. Cords calculus links the security properties of the protocol with the execution semantics of the protocol. It can not only formally describe the protocol itself, but also accurately describe the security properties of the protocol. In the protocol logic system, PCL uses standard logic concepts such as predicate logic and model operators to eliminate the influence of informal factors such as "belief" and "jurisdiction" on the correctness of protocol analysis results. In the proof system, PCL adopts honest rules and does not need explicit inference about intruder's behaviors, greatly reducing the complexity of protocol analysis process. In addition, the logical inference system of PCL can ensure the security analysis of parallel and sequential combination of protocols. Therefore, PCL has scientific and rigorous inference system, as well as flexible and efficient analysis methods, compared with other formal analysis methods. PCL has been broadly extended and improved by researchers [8, 21, 22, 25] in recent years. So far, PCL has been widely used in formal design and analysis of protocols [10, 14, 15, 18]. However, there are still many flaws in the PCL theory, such as inadequate formal theory of message algebraic space, limitations of honesty theory, *etc.* [6].

When the PCL theory is adopted to analyze security protocols, some issues are found, e.g. the message space theory of PCL is less systematic; the definition of message structure, message types and the relations among message subterms are not rigorous; and the inference rules of the relation among message subterms are missing. In the process of protocol analysis, $contains(a, b)$ only be used to assert the subterms relations between different messages. The inference of the subterm relations of messages is latent, subjective and informal. In order to prevent message replay attacks, timestamp mechanism is often used to ensure the freshness of messages in the process of security protocol design. For example, timestamp mechanism is used in CITT X.509, Denning-Sacco, Wide-Mouth Frog, and Kerberos. Timestamp not only prevents message replay attacks, but also potentially establishes the temporal relationship between the actions of principals. In the PCL logical inference system, the temporal relationship between the behaviors of protocol principals is the key basis to judge whether the protocol is executed correctly. The logical inference and establishment of the temporal relationship between the actions of protocol principals are not only the basis of correctness analysis of protocol security properties, but also the key of PCL analysis method that determines the complexity of protocol analysis. Timestamp can directly reflect the timing relationship of the action of the principals, so the complexity of protocol analysis will be greatly reduced, if the timestamp is bound to the action of the principals to analyze the temporal relationship of the action of the principals [5]. In the existing PCL theory, there are no verification or inference rules based on timestamp mechanism for the temporal relationship of the actions of protocol principals. Therefore, this paper is focused on expanding the original PCL theory from two aspects - message subterms relationship and timestamp mechanism, in order to further improve the theoretical basis of PCL, expand the application scope of PCL theory, improve the formalization of protocol analysis, and enhance the efficiency and accuracy of protocol analysis.

CITT X.509 [12] is a security protocol based on public key cryptosystem. In its design, not only random values but also timestamp mechanism are used. The security properties of CITT X.509 consist of authentication, confidentiality and data integrity. The actions of principals contain many basic message operation types, such as encryption, decryption, signature, verification and so on. The diversity of CITT X.509 in security mechanism, security objectives and message operation types requires higher theoretical basis and inference system of formal methods. Since the publication of CITT X.509, some security defects in authentication and confidentiality have been detected by various security protocol analysis methods [11, 13, 17–19, 24]. Researchers have addressed a variety of improvement schemes by reconstructing message structure. Based on the improved scheme of CITT X.509 given by literature [29], the security proof of improved CITT X.509 protocol is delivered by using extended PCL logic to verify the efficiency of extended PCL logic and the correctness of improved CITT X.509 protocol.

## 2 Protocol Composition Logic

### 2.1 Symbols and Terminology

The basic symbols and terms used in this paper are as follows.

1) $\rho$ : The role in the protocol;
2) $\hat{X}$: Principal that performs protocol role;
3) $t$: term;
4) $T_{stamp}$: The set of timestamps;
5) $m(X, Y, t)$: Formatted message terms. $X$ is the sender of the message, $Y$ is the receiver of the message, and $t$ is the content of the message;
6) $K_X$: Key set of principal $\hat{X}$;
7) $k_X, k_X^{-1}$: The public and private keys of principal $\hat{X}$;
8) $K_{XY}$: Shared key of principal $\hat{X}$ and $\hat{Y}$;
9) $\{t\}_k$: Encryption of term $t$ with key $k$;
10) $|t|_k$: Signature of term $t$ with key $k$;
11) $gh$: Connection of term $g$ and $h$;
12) $\alpha, \beta$: Actions of the principal;
13) **a,b**: Action formula;
14) $S$: Strands;
15) $P$: Threads;
16) $n$: Random value;
17) $\top$: True.

### 2.2 Protocol Programming Language

PCL uses a protocol programming language based on Cords calculus to describe protocol message interaction. The formal definitions of message operation and message sequence are given below.

1) *new* $t$: Generate a new term $t$;
2) *send* $u$: Send a term $u$;
3) *receive* $u$: Receive a term $u$;
4) *match* $u, u$: Match a term to a patter;
5) $x :=$ *sign* $u, k$: sign the term $u$ with $k$;
6) $verify\ u, u, k$: Verify the signature;

7) $x := enc\ u, k$: Encrypt the term $u$ with $k$;

8) $x := dec\ u, k$: Decrypt the term $u$ with $k$;

9) $x := gh$: Tuple the term $g$ and $h$;

10) $[\alpha; \cdots ; \alpha]_P$: Actions sequence of $\hat{P}$.

### 2.3 Protocol Logic

1) Action formulas.

$$\begin{aligned}\mathbf{a} ::= {} & Send(X,t)|Receive(X,t)|New(X,t)|\\ & Encrypt(X,t)|Decrypt(X,t)|Sign(X,t)|\\ & Verify(X,t)|Match(X,t)|Tuple(X,t).\end{aligned}$$

2) Logic formulas.

$$\begin{aligned}\phi ::= {} & \mathbf{a}|Has(X,t)|Fresh(X,t)|Gen(X,t)|\\ & FirstSend(X,t,t')Honest(X)|t=t|\\ & Contains(t,t')|\phi \wedge \phi|\neg\phi|Start(X)|\mathbf{a} < \mathbf{b}.\end{aligned}$$

3) Modal formulas.

$$\theta ::= \phi S \phi.$$

### 2.4 Inference System

According to the function of inference formula, the inference formula of PCL is divided into seven types. The proof of inference formula is detailed in reference [8].

1) Protocol actions.

**AA1** $\top[\alpha]_X \mathbf{a}$

**AA2** $Start(X)[\ ]_X \neg\mathbf{a}(X)$

**AA3** $\neg Send(X,t)[\alpha]_X \neg\ Send(X,t)$

**AA4** $\top[\alpha; \cdots ; \beta]_X \mathbf{a} < \mathbf{b}$

**AN1** $New(X,t) \wedge New(Y,t) \supset X = Y$

**AN2** $\top[new\ t]_X Has(Y,t) \supset Y = X$

**AN3** $\top[new\ t]_X Fresh(X,t)$

**AN4** $Fresh(X,t) \supset Gen(X,t)$

2) Possession axioms.

**AM1** $\top[\ ]_X Has(X, K_X)$

**AM2** $(x, \cdots)[\ ]_X Has(X,x)$

**ORIG** $New(X,t) \supset Has(X,t)$

**REC** $Receive(X,t) \supset Has(X,t)$

**TUP** $Has(X,a) \wedge Has(X,b) \supset Has(X,ab)$

**ENC** $Has(X,t) \wedge Has(X,k) \supset Has(X,\{t\}_k)$

**DEC** $Has(X,\{t\}_k) \wedge Has(X,k^{-1}) \supset Has(X,t)$

**PROJ** $Has(X,ab) \supset Has(X,a) \wedge Has(X,b)$

**GEN1** $Has(X,t) \wedge Has(X,k) \wedge Send(X,\{t\}_k)$
$\supset Gen(X,\{t\}_k)$

**GEN2** $Gen(X,\{t\}_k) \supset Has(X,t)$

**GEN3** $Gen(X,\{t\}_k) \supset Has(X,k)$

3) Encryption and signature.

**SEC** $Honest(\hat{X}) \wedge Decrypt(Y,\{x\}_{k_{\hat{X}}}) \supset \hat{Y} = \hat{X}$

**VER** $Honest(\hat{X}) \wedge Verify(Y,|x|^{-1}_{k_{\hat{X}}}) \wedge \hat{X} \neq \hat{Y} \supset$
$\exists X.Sign(X,|x|^{-1}_{k_{\hat{X}}}) \wedge Send(X,m)$
$\wedge\, Contains(m,x)$

4) Preservation axioms.

**P1** $Persist(X,t)[\alpha]_X Persist(X,t)$
where $Persist \in \{Has, FirstSend, \mathbf{a}, Gen\}$

**P2** $Fresh(X,t)[\alpha]_X Fresh(X,t)$
where $\neg Contains(t,a)$

5) Temporal ordering.

**FS1** $Fresh(X,t)[send\ t']_X FirstSend(X,t,t')$
where $Contains(t',t)$

**FS2** $FirstSend(X,t',t) \vee \mathbf{a}(Y,t'') \supset$
$Send(X,t') < \mathbf{a}(Y,t'')$
where $X \neq Y \wedge Contains(t'',t)$

6) Generic rules.

**G1** $$\frac{\theta[P]_X\phi \qquad \theta[P]_X\psi}{\theta[P]_X\phi \wedge \psi}$$

**G2** $$\frac{\theta[P]_X\psi \qquad \phi[P]_X\psi}{\theta \wedge \phi[P]_X\psi}$$

**G3** $$\frac{\theta[P]_X\phi}{\theta'[P]_X\phi'}$$
where $Contains(\theta',\theta) \wedge Contains(\phi',\phi)$

**G4** $$\frac{\phi_1[P]_X\phi_2 \qquad \phi_2[P']_X\phi_3}{\phi_1[PP']_X\phi_3}$$

7) Honesty rule.

**HON$_Q$** $$\forall\rho \in Q \cdot \forall P \in BS(\rho)$$
$$\frac{Start(X)[\ ]_X\phi \quad \phi[P]_X\phi}{Honest(\hat{X}) \supset \phi}$$

### 2.5 Initial Configuration of Protocol

**Definition 1.** *Let $C$ be initial configuration of protocol $Q$,$C$ is determined by:*

*1) A group of principals, some of which are designated as honest.*

*2) A cord space constructed by assigning roles of $Q$ to threads of honest principals.*

*3) One or more intruder cords, which may use keys of dishonest principals.*

*4) A finite number of buffer cords, enough to accommodate every send action by honest threads and the intruder threads.*

# 3 Extension of PCL

## 3.1 Subterm Relations

In the existing PCL theory, only one attribute assertion $Contains(a, b)$ is given to indicate that message $a$ is a subterm of $b$, but it does not give a strict definition of message subterm relationship, nor does it give the relevant inference rules of message subterm relationship. When in using PCL for security protocol analysis, the inference of message subterm relationship is latent and subjective, and this makes the protocol analysis process lack of rigorous theoretical basis and formal methods, and directly affects the correctness of protocol analysis results.

**Definition 2.** *Let t,g,h be message terms and k be the key of the protocol principle, the message subterm relationship can be defined recursively as follows:*

1) $Contains(t, t)$*,Message term is its own subterm;*

2) $Contains(t, \{h\}_k)$*,If and only if* $Contains(t, h) \vee t = \{h\}_k$*;*

3) $Contains(t, |h|_k)$*,If and only if* $Contains(t, h) \vee t = |h|_k$*;*

4) $contains(t, gh)$*,If and only if* $Contains(t, g) \vee Contains(t, h)$*.*

The inference rules of message subterm relations can be given from Definition 2:

**STR1** $gh \supset Contains(g, gh) \wedge Contains(h, gh) \wedge Contains(gh, gh)$

**STR2** $\{t\}_k \supset Contains(t, \{t\}_k) \wedge Contains(\{t\}_k, \{t\}_k)$

**STR3** $|t|_k \supset Contains(t, |t|_k) \wedge Contains(|t|_k, |t|_k)$

**STR4** $Contains(t, g) \wedge Contains(g, h) \supset Contains(t, h)$

## 3.2 Timestamp Mechanism

Timestamp is the main mechanism to guarantee the freshness of message terms in security protocols. The design of CCITT X.509 protocol adopts timestamp mechanism. In the existing PCL inference system, there is no rule of verification and inference based on timestamp mechanism to judge the temporal relationship of the behaviors of the principals, and it is impossible to formally express and inference the timestamp mechanism in the protocol correctly. In order to reduce the complexity of protocol analysis and improve the efficiency of protocol analysis, it is necessary to extend the logic inference system of PCL from the aspect of timestamp mechanism.

Timestamp exists in the form of message subterms, which are bound to the actions of the principals. How to formalize the relationship between message terms and the actions of protocol principals, and the temporal relationship between different actions based on timestamps are not addressed in the existing PCL theory.

**Definition 3.** *Let $m$ be the message term of action **a**, and then $m$ is defined as $term(\mathbf{a})$, i.e. $m = term(\mathbf{a})$.*

**Definition 4.** *Let $t_1$ and $t_2$ be timestamp constants created at protocol runtime:*

1) *if $t_1$ is created before $t_2$, the relationship between $t_1$ and $t_2$ is defined as $t_1 < t_2$;*

2) *if $t_1$ and $t_2$ are created at the same time, the relationship between $t_1$ and $t_2$ is defined as $t_1 = t_2$.*

**Property 1.** *Let $t$ be the timestamp constant created by the protocol runtime and $t_{sys}$ the current time, then $t <= t_{sys}$.*

**Theorem 1.** *Let $\hat{X}, \hat{Y}$ be the principal role of the protocol; $t_1$ and $t_2$ be timestamp constants, and $t_1 < t_2$, $m_1$ and $m_2$ be terms and $FirstSend(X, t_1, m_1) \wedge FirstSend(X, t_2, m_2)$. Then $Send(X, m_1) < Send(X, m_2)$.*

**Theorem 2.** *Let $\hat{X}, \hat{Y}$ be the principal role of the protocol; $t$ be timestamp constants, **a** be the action assertion, $m$ be term and $Contains(m, t) \wedge term(\mathbf{a}) = m$, $FirstSend(X, t, m)$. Then $Send(X, m) <= \mathbf{a}$.*

Theorems 1 and 2 can be proved by Definition 2, **AA1** and **AA4**, which are not discussed here.

Corollary 1 can be derived from Theorem 2.

**Corollary 1.** *Let $\hat{X}, \hat{Y}$ be the principal role of the protocol, $t$ be the timestamp, $t_{sys}$ be the current time, $m$ be the term and $Contains(m, t)$. Then $Send(X, m) < Receive(Y, m)$.*

From Theorem 1, Theorem 2 and Corollary 1, the following inference rules can be given.

**TT1** $$\frac{FirstSend(X, t_1, m_1) \wedge FirstSend(\hat{Y}, t_2, m_2)}{(t_1 < t_2) \supset Send(X, m_1) < Send(Y, m_2)}$$

where $t_1, t_2 \in T_{stamp}$

**TT2** $$\frac{FirstSend(X, t, m)}{Contain(term(\mathbf{a}), t) \supset Send(X, m) < \mathbf{a}}$$

**TT3** $$\frac{FirstSend(X, t, m)}{Send(X, m) < Receive(Y, m)} \quad \text{where } t \in T_{stamp}$$

# 4 Improved CCITT X.509

## 4.1 Improved CCITTX.509 Modeling

The improved CCITTX.509 protocol execution process is shown in Figure 1.

CCITTX.509 protocol is based on public key cryptosystem. It has two roles: $A$ initiator and $B$ responder. $T_a$ and $T_b$ are the timestamps produced by $A$ and $B$, $N_a$ and $N_b$ are the random numbers generated by $A$ and $B$, $X_a$ and $Y_a$ are the data generated by $A$, $X_a$ and $X_b$ are the data generated by $B$, $k_A$ and $k_A^{-1}$ are the public and private keys of $A$, and $k_B$ and $k_B^{-1}$ are the public and private keys of $B$.

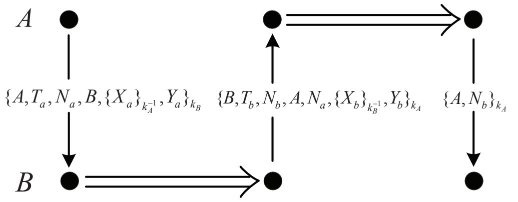
A
B
$\{A,T_a,N_a,B,\{X_a\}_{k_A^{-1}},Y_a\}_{k_B}$
$\{B,T_b,N_b,A,N_a,\{X_b\}_{k_B^{-1}},Y_b\}_{k_A}$
$\{A,N_b\}_{k_A}$

Figure 1: Graph of improved CCITT X.509 protocol

**Definition 5.** *Let $Init_{X.509}$ and $Resp_{X.509}$ be the initiator and responder roles of the improved CCITTX.509 respectively. $\hat{A}$ and $\hat{B}$ are the principals of the protocol roles, then $Init_{X.509}$ and $Resp_{X.509}$ are defined as below:*

```
$Init_{X.509} \equiv (\hat{A}, \hat{B})[$
        $new\ T_a;$
        $new\ N_a;$
        $new\ X_a;$
        $t_1 := enc\ X_a, k_A^{-1};$
        $new\ Y_a;$
        $t_2 := \{A, T_a, N_a, B, t_1, Y_a\};$
        $m_1 := enc\ t_2, k_B;$
        $send\ m_1;$
        $receive\ m_2;$
        $t_3 := dec\ m_2, k_A^{-1};$
        $match\ t_3, \{B, T_b, N_b, A, N_a, t_4, Y_b\};$
        $verify\ t_4, X_b, k_B;$
        $t_5 := \{A, N_b\};$
        $m_3 := enc\ t_5, k_A;$
        $send\ m_3;$
        $]_A <> .$
```

```
$Resp_{X.509} \equiv (\ )[$
        $receive\ n_1;$
        $r_1 := dec\ n_1, k_B^{-1};$
        $match\ r_1, \{A, T_a, N_a, B, r_2, Y_a\};$
        $verify\ r_2, X_a, k_A;$
        $new\ T_b;$
        $new\ N_b;$
        $new\ X_b;$
        $r_3 := sign\ X_b, k_B^{-1};$
        $new\ Y_b;$
        $n_2 := enc\ \{B, T_b, N_b, A, N_a, r_3, Y_b\}, k_A;$
        $send\ n_2;$
        $receive\ n_3;$
        $r_4 := dec\ n_3, k_B^{-1};$
        $match\ r_4, \{A, N_b\};$
        $]_B <> .$
```

## 4.2 Protocol Attribute Modeling

CCITTX.509 protocol is designed to share $Y_a$ and $Y_b$ on the basis of mutual authentication of principals. The protocol uses timestamps $T_a$ and $T_b$, as well as random values $N_a$ and $N_b$ to ensure the freshness of message terms. Encryption with private key signature ensures the integrity of $X_a$ and $X_b$, and encryption with public key ensures the confidentiality of $Y_a$ and $Y_b$. Therefore, the main security properties of the protocol include authentication, secrecy and data integrity.

1) Authentication.

**Definition 6.** *Let $\hat{A}$ be the principal of $Init_{X.509}$ and $\hat{B}$ be the principal of $Resp_{X.509}$. If the protocol satisfies the mutual authentication between $\hat{A}$ and $\hat{B}$, $\phi_{AUTH}(\hat{A})$ is the authentication of $\hat{A}$ to $\hat{B}$, and $\phi_{AUTH}(\hat{B})$ is the authentication of $\hat{B}$ to $\hat{A}$, then:*

$$\begin{aligned}\phi_{AUTH}(\hat{A}) \equiv \exists B.(((&Send(A, m_1) < Receive(B, m_1))\\ &\wedge (Receive(B, m_1) < Send(B, m_2))\\ &\wedge (Send(B, m_2) < Receive(A, m_2))\\ &\wedge (Receive(A, m_2 < Send(A, m_3))).\end{aligned}$$

$$\begin{aligned}\phi_{AUTH}(\hat{B}) \equiv \exists A.(((&Send(A, n_1) < Receive(B, n_1))\\ &\wedge (Receive(B, n_1) < Send(B, n_2))\\ &\wedge (Send(B, n_2) < Receive(A, n_2))\\ &\wedge (Receive(A, n_2 < Send(A, n_3))).\end{aligned}$$

2) Data secrecy.

**Definition 7.** *Let $\hat{A}$ be the principal of $Init_{X.509}$ and $\hat{B}$ be the principal of $Resp_{X.509}$. $\phi_{SEC}(\hat{A})$ denotes that $Init_{X.509}$ satisfies the confidentiality of $Y_a$ and $Y_b$, and $\phi_{SEC}(\hat{B})$ denotes that $Resp_{X.509}$ satisfies the confidentiality of $Y_a$ and $Y_b$. Then:*

$$\phi_{SEC}(\hat{A}) \equiv \exists Z.Has(Z, (Y_a, Y_b)) \supset (Z = A \vee Z = B)$$

$$\phi_{SEC}(\hat{B}) \equiv \exists Z.Has(Z, (Y_a, Y_b)) \supset (Z = A \vee Z = B)$$

3) Data integrity.

**Definition 8.** *Let $\hat{A}$ be the principal of $Init_{X.509}$ and $\hat{B}$ be the principal of $Resp_{X.509}$; $\phi_{INTE}(\hat{A})$ denotes the integrity of $X_b$ to $\hat{A}$, $\phi_{INTE}(\hat{B})$ denotes the integrity of $X_a$ to $\hat{B}$, then:*

$$\begin{aligned}\phi_{INTE}(\hat{A}) \equiv \exists Z.Sign(Z, \{X_b\}_{k_Z^{-1}}) \wedge Send(Z, m)&\\ \wedge Contains(m, \{X_b\}_{k_Z^{-1}}) \supset Z = B&\end{aligned}$$

$$\begin{aligned}\phi_{INTE}(\hat{B}) \equiv \exists Z.Sign(Z, \{X_a\}_{k_Z^{-1}}) \wedge Send(Z, m)&\\ \wedge Contains(m, \{X_a\}_{k_Z^{-1}}) \supset Z = A&\end{aligned}$$

# 5 Analysis of Improved CCITT X.509

## 5.1 Authentication Analysis

**Proposition 1.** *Let $C$ be initial configuration of improved CCITTX.509, $\hat{A}$ and $\hat{B}$ be the principal of the* *initiator and responder roles respectively. If principal* $\hat{A}, \hat{B} \in Honest(C)$*, then* $\phi_{AUTH}(\hat{A})$ *is true.*

**Proof 1.**

$$\boldsymbol{AM1}, \boldsymbol{AM2} \quad (\hat{A},\hat{B})[\;]_A Has(A,A) \wedge Has(A,B) \wedge Has(A,k_A^{-1}) \wedge Has(A,k_B) \tag{1}$$

$$\boldsymbol{AN2}, \boldsymbol{AN2} \quad \top[new\ n_a]_A Has(A,N_a) \wedge Fresh(A,N_a) \tag{2}$$

$$\boldsymbol{AN2}, \boldsymbol{AN3} \quad \top[new\ T_a]_A Has(A,T_a) \wedge Fresh(A,T_a) \tag{3}$$

$$\boldsymbol{TUP}, \boldsymbol{STR1} \quad \top[t_2 := \{A,T_a,N_a,B,t_1,Y_a\}]_A\ Tuple(A,t_2) \supset Contains(N_a,t_2) \wedge Contains(T_a,t_2) \tag{4}$$

$$\boldsymbol{STR2}, \boldsymbol{STR4} \quad \top[m_1 := enc\ t_2,k_B]_A Encrypt(A,t_2) \supset Contains(N_a,m_1) \wedge Contains(T_a,m_1) \tag{5}$$

$$\boldsymbol{2}, \boldsymbol{5}, \boldsymbol{AA1}, \boldsymbol{FS1} \quad Fresh(A,N_a) \wedge Contains(N_a,m_1)\ [send\ m_1]_A FirstSend(A,N_a,m_1) \tag{6}$$

$$\boldsymbol{2}, \boldsymbol{5}, \boldsymbol{AA1}, \boldsymbol{FS1} \quad Fresh(A,T_a) \wedge Contains(T_a,m_1)\ [send\ m_1]_A FirstSend(A,T_a,m_1) \tag{7}$$

$$\boldsymbol{AA1}, \boldsymbol{REC} \quad \top[receive\ m_2]_A Receive(A,m_2) \wedge Has(A,m_2) \tag{8}$$

$$\boldsymbol{AA1}, \boldsymbol{AA4} \quad \top[dec\ m_2,k_A^{-1}/send\ m_3]_A\ Decrypt(A,m_2) < Send(A,m_3) \tag{9}$$

$$\boldsymbol{1}, \boldsymbol{8}, \boldsymbol{9}, \boldsymbol{DEC}\ Has(a,m_2) \wedge Has(a,k_A^{-1}) \wedge Honest(\hat{A}) \supset Contains(N_a,m_2) \wedge Contains(T_b,m_2) \tag{10}$$

$$\boldsymbol{7}, \boldsymbol{10}, \boldsymbol{FS2} \quad FirstSend(A,N_a,m_1) \wedge Contains(N_a,m_2) \wedge Honest(\hat{B}) \supset (Receive(B,m_1) < Send(B,m_2)) \wedge FirstSend(B,T_b,m_2) \tag{11}$$

$$\boldsymbol{5}, \boldsymbol{6}, \boldsymbol{10}, \boldsymbol{TT3} \quad FirstSend(A,T_a,m_1) \wedge Receive(B,m_1) \wedge Contains(T_a,m_1) \supset (Send(A,m_1) < Receive(B,m_1)) \tag{12}$$

$$\boldsymbol{7}, \boldsymbol{10}, \boldsymbol{11}, \boldsymbol{TT3} \quad FirstSend(B,T_b,m_2) \wedge Receive(A,m_2) \wedge Contains(T_b,m_2) \supset (Send(B,m_2) < Receive(A,m_2)) \tag{13}$$

$$\boldsymbol{AA4}, \boldsymbol{P1}, \boldsymbol{G4}, \boldsymbol{HON}_Q \quad (\hat{A},\hat{B})[Init_{X.509}]_A \supset (Receive(A,m_2) < Send(A,m_3)) \tag{14}$$

$$\boldsymbol{11}, \boldsymbol{12}, \boldsymbol{13}, \boldsymbol{14}, \boldsymbol{HON}_Q\ (\hat{A},\hat{B})[Init_{X.509}]_A Honest(\hat{B}) \supset \phi_{AUTH}(\hat{A}) \tag{15}$$

**Proposition 2.** *Let C be initial configuration of improved CCITTX.509,* $\hat{A}$ *and* $\hat{B}$ *be the principal of the initiator and responder roles respectively. If principal* $\hat{A}, \hat{B} \in Honest(C)$*, then* $\phi_{AUTH}(\hat{B})$ *is true.*

The conclusion of Proposition 2 is true, and the proof process is similar to Proposition 1, which is no longer repeated.

According to the proof of proposition 1 and 2, the improved CCITTX.509 protocol satisfies authentication.

## 5.2 Secrecy Analysis

**Proposition 3.** *Let C be initial configuration of improved CCITTX.509,* $\hat{A}$ *and* $\hat{B}$ *be the principal of the initiator and responder roles respectively. If principal* $\hat{A}, \hat{B} \in Honest(C)$*, then* $\phi_{SEC}(\hat{A})$ *is true.*

**Proof 2.**

$$\boldsymbol{AM1}, \boldsymbol{AM2} \quad (\hat{A},\hat{B})[\;]_A Has(A,A) \wedge Has(A,B) \wedge Has(A,k_A^{-1}) \wedge Has(A,k_B) \tag{16}$$

$$\boldsymbol{AN2}, \boldsymbol{AN2} \quad \top[new\ n_a]_A Has(A,N_a) \wedge Fresh(A,N_a) \tag{17}$$

$$\boldsymbol{AN2}, \boldsymbol{AN3} \quad \top[new\ Y_a]_A Has(A,Y_a) \wedge Fresh(A,Y_a) \tag{18}$$

$$\boldsymbol{AA1}, \boldsymbol{STR1} \quad \top[t_2 := \{A,T_a,N_a,B,t_1,Y_a\}]_A\ Tuple(A,t_1) \wedge Contains(Y_a,t_2) \tag{19}$$

$$\boldsymbol{STR2}, \boldsymbol{STR4} \quad \top[m_1 := enc\ t_2,k_B]_A Encrypt(A,t_2) \wedge Contains(t_2,m_1) \supset Contains(N_a,m_1) \wedge Contains(Y_a,m_1) \tag{20}$$

$$\boldsymbol{20}, \boldsymbol{FS1}, \boldsymbol{AA3} \quad Fresh(A,N_a) \wedge Contains(N_a,m_1)\ [send\ m_1]_A FirstSend(A,N_a,m_1) \tag{21}$$

$$\boldsymbol{20}, \boldsymbol{FS1}, \boldsymbol{AA3} \quad Fresh(A,Y_a) \wedge Contains(Y_a,m_1)\ [send\ m_1]_A FirstSend(A,Y_a,m_1) \tag{22}$$

$$\boldsymbol{AA1}, \boldsymbol{REC} \quad \top[receive\ m_2]_A Receive(A,m_2) \wedge Has(A,m_2) \tag{23}$$

$$\boldsymbol{AA1}, \boldsymbol{DEC} \quad Has(A,k_A^{-1})[t_3 := dec\ m_2,k_A^{-1}]_A\ Decrypt(A,m_2) \supset Contains(N_a,m_2) \wedge Contains(Y_b,m_2) \wedge Has(A,Y_b) \wedge Has(A,N_a) \tag{24}$$

$$\boldsymbol{21}, \boldsymbol{22}, \boldsymbol{FS2} \quad FirstSend(A,N_a,m_1) \wedge Contains(N_a,m_2) \wedge Honest(\hat{B}) \supset (Receive(B,m_1) < Send(B,m_2)) \wedge FirstSend(B,Y_b,m_2) \tag{25}$$

$$\boldsymbol{HON}_Q \quad Honest(\hat{B})[\;]_A Has(B,A) \wedge Has(B,B) \wedge Has(B,k_B^{-1}) \wedge Has(B,k_A) \tag{26}$$

$$\boldsymbol{24}, \boldsymbol{25}, \boldsymbol{DEC} \quad Receive(B,m_1) \wedge Has(B,k_B^{-1}) \wedge Contains(Y_a,m_1) \supset Has(B,Y_a) \tag{27}$$

$$\boldsymbol{23}, \boldsymbol{24}, \boldsymbol{ENC} \quad FirstSend(B,Y_b,m_2) \supset Has(B,Y_b) \tag{28}$$

$$\boldsymbol{18}, \boldsymbol{27}, \boldsymbol{P1} \quad (\hat{A},\hat{B})[Init_{X.509}]_A(\exists Z.Has(Z,Y_a) \supset (Z = A \vee Z = B)) \tag{29}$$

$$\boldsymbol{18}, \boldsymbol{28}, \boldsymbol{P1} \quad (\hat{A},\hat{B})[Init_{X.509}]_A(\exists Z.Has(Z,Y_b) \supset (Z = A \vee Z = B)) \tag{30}$$

$$\mathbf{\mathit{29, 30, HON}}_Q \quad (\hat{A},\hat{B})[Init_{X.509}]_A \phi_{SEC}(\hat{A}) \tag{31}$$

**Proposition 4.** *Let $C$ be initial configuration of improved CCITTX.509, $\hat{A}$ and $\hat{B}$ be the principal of the initiator and responder roles respectively. If principal $\hat{A}, \hat{B} \in Honest(C)$, then $\phi_{SEC}(\hat{B})$ is true.*

The same principle can prove the correctness of the conclusion of proposition 4, which is omitted here.

Propositions 3 and 4 verify that the improved CCITTX.509 protocol can satisfy the confidentiality.

### 5.3 Data Integrity Analysis

**Proposition 5.** *Let $C$ be initial configuration of improved CCITTX.509, $\hat{A}$ and $\hat{B}$ be the principal of the initiator and responder roles respectively. If principal $\hat{B} \in Honest(C)$, then $\phi_{INTE}(\hat{A})$ is true.*

**Proof 3.**

$$\begin{aligned}
&\mathbf{\mathit{AM1}}, \mathbf{\mathit{AM2}} \quad (\hat{A},\hat{B})[\ ]_A Has(A,A) \wedge Has(A,B) \\
&\qquad \wedge Has(A,k_A^{-1}) \wedge Has(A,k_B) && (32)\\
&\mathbf{\mathit{AM1}}, \mathbf{\mathit{AM2}} \quad Honesty[\ ]_B Has(B,A) \wedge Has(B,B) \\
&\qquad \wedge Has(B,k_B^{-1}) \wedge Has(B,k_A) && (33)\\
&\mathbf{\mathit{AN1}}, \mathbf{\mathit{AN3}} \quad \top[new\ N_a]_A Has(A,N_a) \\
&\qquad \wedge Fresh(A,N_a) && (34)\\
&\mathbf{\mathit{AA1STR2}}, \mathbf{\mathit{STR4}} \quad \top[m_1 := enc\ t_2, k_B]_A \\
&\qquad Encrypt(A,t_2) \wedge Contains(t_2,m_1) \\
&\qquad \supset Contains(N_a,m_1) && (35)\\
&\mathbf{\mathit{36}}, \mathbf{\mathit{FS1}}, \mathbf{\mathit{AA3}} \quad Fresh(A,N_a) \wedge Contains(N_a,m_1) \\
&\qquad [send\ m_1]_A FirstSend(A,N_a,m_1) && (36)\\
&\mathbf{\mathit{AA1}}, \mathbf{\mathit{REC}} \quad \top[receive\ m_2]_A Receive(A,m_2) \\
&\qquad \wedge Has(A,m_2) && (37)\\
&\mathbf{\mathit{AA1}}, \mathbf{\mathit{DEC}} \quad Has(A,k_A^{-1})[t_3 := dec\ m_2, k_A^{-1}]_A \\
&\qquad Decrypt(A,m_2) \supset Contains(N_a,m_2) && (38)\\
&\mathbf{\mathit{37}}, \mathbf{\mathit{39}}, \mathbf{\mathit{FS2}}, \mathbf{\mathit{HON}}_Q \quad FirstSend(A,N_a,m_1) \\
&\qquad \wedge Contains(N_a,m_2) \wedge Honest(\hat{B}) \\
&\qquad \supset (Receive(B,m_1) < Send(B,m_2)) \\
&\qquad \wedge FirstSend(B,Y_b,m_2) && (39)\\
&\mathbf{\mathit{AA1}}, \mathbf{\mathit{AA4}} \quad \top[receive\ m_2/t_3 := dec\ m_2, k_A^{-1}]_A \\
&\qquad Receive(A,m_2) < Decrypt(A,m_2) \\
&\qquad \supset Has(A,t_3) \wedge Contains(m_2,t_3) && (40)\\
&\mathbf{\mathit{AA1}}, \mathbf{\mathit{STR1}} \quad \top[match\ t_3, \{B,T_b,N_b,A,N_a,t_4,Y_b\}]_A \\
&\qquad \wedge Match(A,t_3) \supset Has(A,t_4) \\
&\qquad \wedge Contains(t_4,t_3) && (41)\\
&\mathbf{\mathit{AA1}}, \mathbf{\mathit{HON}}_Q \quad Honest(\hat{B})[verify\ t_4, X_b, k_b]_A \\
&\qquad Verify(A,t_4) \supset Sign(B, t_4 := |X_b|_{k_B^{-1}}) && (42)\\
&\mathbf{\mathit{41}}, \mathbf{\mathit{42}}, \mathbf{\mathit{STR4}} \quad Contains(m_2,t_3) \wedge Contains(t_3,t_4) \\
&\qquad \supset Contains(m_2,t_4) && (43)\\
&\mathbf{\mathit{40}}, \mathbf{\mathit{43}}, \mathbf{\mathit{44}}, \mathbf{\mathit{HON}}_Q \quad (\hat{A},\hat{B})[Init_{X.509}]_A Honesty(\hat{B}) \\
&\qquad \supset \exists Z.(Sign(Z, |X_b|_{k_Z^{-1}}) \wedge Send(Z,m_2) \\
&\qquad \wedge Contains(|X_b|_{k_Z^{-1}}, m_2)) \supset (Z = B) && (44)
\end{aligned}$$

**Proposition 6.** *Let $C$ be initial configuration of improved CCITTX.509, $\hat{A}$ and $\hat{B}$ be the principal of the initiator and responder roles respectively. If principal $\hat{A} \in Honest(C)$, then $\phi_{INTE}(\hat{B})$ is true.*

The same principle can prove the correctness of the conclusion of proposition 6, which is omitted here.

Propositions 5 and 6 verify that the improved CCITTX.509 protocol can guarantee the integrity of sum.

The proof of propositions 1 to 6 shows that the improved CCITTX.509 protocol can meet the security attribute design goals of authentication, secrecy and data integrity.

### 5.4 Comparison with the Traditional Method

In traditional methods used in the analysis of CR [8], Otway-Rees [14] and NSL [22] based on PCL, the analysis of action sequence of protocol principals is mainly based on the judgment of freshness of random value $N_a$ and $N_b$. The main inference rules used in protocol analysis are **FS1**, **FS2**, **P1** and **P2**. In order to illustrate the validity of the principal action sequence judgment rules based on the timestamp mechanism, table 1 gives a detailed comparison with the traditional methods in judging parameters and inference rules used in a challenge response round of the protocol, as well as the value range of proving steps. Because of the rigor and intuitiveness of the proof process, the value range given in Table 1 are only steps to reasonably prove the action sequence of the protocol, and the value range is not very accurate, which is only a reference for the comparison of the complexity of two analysis methods.

From the comparison results of the two methods in Table 1, in a challenge response round of the protocol, new method only needs about 5 to 10 steps to determine the action sequence of the protocol principals. Compared with the traditional method, it simplifies the steps of protocol analysis and effectively reduces the complexity of protocol analysis. The improvement and application of **STR1, STR2, STR3** and **STR4** in message subitem relationship make protocol analysis more scientific and rigorous.

## 6 Conclusions

Formal analysis process and results of the improved CCITTX.509 protocol using the extended PCL show that the inference rules of the relations among message subterms make the protocol analysis process more rigorous and formalized. Compared with the methods used in literature 5, the logical inference rules based on timestamp

Table 1: Comparison of two methods in a challenge response protocol round

| **Method** | **Protocol** | **Parameter** | **Inference Rules** | **Steps(n)** |
|---|---|---|---|---|
| Traditional Method | $Otway - Rees$<br>$CR, NSL$ | $N_a, N_b$ | $\mathbf{FS1}, \mathbf{FS2}$<br>$\mathbf{P1}, \mathbf{P2}$ | $8 \leq n \leq 15$ |
| Methods in this paper | $CCITTX.509$ | $T_a, T_b$ | $\mathbf{FS1}, \mathbf{FS2}$<br>$\mathbf{TT1}, \mathbf{TT2}, \mathbf{TT3}$<br>$\mathbf{STR1}, \mathbf{STR2}$<br>$\mathbf{STR3}, \mathbf{STR4}$ | $5 \leq n \leq 10$ |

mechanism can greatly simplify the steps of protocol authentication analysis and effectively reduce the complexity of protocol analysis. Logical inference rules based on timestamp mechanism further improve the theoretical system of PCL. This method can be used to effectively analyze the authentication objectives of security protocols designed based on timestamp mechanism. In addition, according to the formal analysis process of CCITTX.509 protocol, it is obvious that propositional hypothesis analysis method further standardizes the PCL formal analysis method, making the protocol analysis process more intuitive and clear. The description method of security protocol based on protocol thread programming language and logical inference system based on the behavior assertion of protocol principal make PCL more formal and logical than other formal analysis methods [13, 17, 24].

Although PCL is highly formal in protocol description and logical inference system, the process of protocol analysis is relatively simple and intuitive. However, the assumption of honest rules makes it impossible for PCL to analyze attack types from within the protocol [6, 8]. Meanwhile, the language description system of PCL is not perfect and the standard definition of message algebraic space is not in place yet. These defects limit the modeling and analysis ability of PCL. Improving the message algebraic space theory of PCL and enhancing the ability of protocol description and analysis will be the main research goal in the next stage.

## Acknowledgments

This study was supported by the National Natural Science Foundation of China (61902140 and 61300048), Science Foundation for The Excellent Youth Scholars of Anhui University (gxyq2017154), Anhui Provincial Natural Science Foundation (1608085MF159 and 1908085QF288) and in part by Natural Science Foundation of Anhui University (KJ2014A231, KJ2015A315,KJ2018A0678, KJ2018A396). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

## References

[1] T. Y. Chang, M. S. Hwang, and C. C. Yang, "Password authenticated key exchange and protected password change protocols," *Symmetry*, vol. 9, no. 8, pp. 134, 2017.

[2] X. Chen and H. Deng, "Analysis of cryptographic protocol by dynamic epistemic logic," *IEEE Access*, vol. 7, pp. 29981–29988, 2019.

[3] V. Cheval, V. Cortier, and B. Warinschi, "Secure composition of pkis with public key protocols," in *IEEE 30th Computer Security Foundations Symposium (CSF'17)*, pp. 144–158, Aug. 2017.

[4] S. F. Chiou, H. T. Pan, E. F. Cahyadi, and M. S. Hwang, "Cryptanalysis of the mutual authentication and key agreement protocol with smart cards for wireless communications," *International Journal Network Security*, vol. 21, no. 1, pp. 100–104, 2019.

[5] N. Chumakova, V. Olenev, and I. Lavrovskaya, "Conformance testing of the STP-ISS protocol implementation by means of temporal logic," in *The 21st Conference of Open Innovations Association*, pp. 71–78, 2017.

[6] C. J. F. Cremers, "On the protocol composition logic PCL," in *Proceedings of ACM Symposium on Information, Computer and Communications Security*, pp. 66–76, 2008.

[7] A. Datta, A. Derek, J. C. Mitchell, and D. Pavlovic, "A derivation system for security protocols and its logical formalization," in *The 16th IEEE Computer Security Foundations Workshop*, pp. 109–125, 2003.

[8] A. Datta, A. Derek, J. C. Mitchell, and A. Roy, "Protocol composition logic (PCL)," *Electronic Notes in Theoretical Computer Science*, vol. 172, pp. 311–358, 2007.

[9] N. A. Durgin, J. C. Mitchell, and D. Pavlovic, "A compositional logic for proving security properties of protocols," *Journal of Computer Security*, vol. 11, no. 4, pp. 677–722, 2003.

[10] T. Feng, Y. Yi, and J. Ma, "Secure authenticated key agreement protocol for wmen based on protocol composition logic," in *IEEE 3rd International Conference on Communication Software and Networks*, pp. 284–288, May 2011.

[11] K. Gaarder and E. Snekkenes, "Applying a formal analysis technique to the CCITT X.509 strong two-way authentication protocol," *Journal of Cryptology*, vol. 3, no. 2, pp. 81–98, 1991.

[12] C. I'Anson and C. Mitchell, "Security defects in ccitt recommendation x.509: The directory authentication framework," *Computer Communication Review (CCR'90)*, vol. 20, pp. 30–34, Apr. 1990.

[13] H. Jiang, G. Zhang, and J. Fan, "Structure analysis and generation of X.509 digital certificate based on national secret," in *Journal of Physics: Conference Series*, vol. 1187, pp. 042067, Apr 2019.

[14] J. F. Ma, L. F. Lu, X. D. Duan, "Improvement and formal proof on protocol Otway-Rees," *Journal on Commuications*, vol. 33, pp. 250–254, Sep. 2012.

[15] L. Lu and J. Ma, "Formal analysis model of security protocol based on PCL," in *International Conference on Computer Application and System Modeling (ICCASM'10)*, Oct. 2010. DOI: 10.1109/ICCASM.2010.5620624.

[16] C. H. Ling, S. M. Chen, and M. S. Hwang, "Cryptanalysis of Tseng-Wu group key exchange protocol," *International Journal Network Security*, vol. 18, no. 3, pp. 590–593, 2016.

[17] J. F. Liu and M. T. Zhou, "Analysis of X.509 authentication protocol via authentication test," *Computer Engineering and Applications*, vol. 08, pp. 23–25, Aug. 2006.

[18] P. Liu and P. Zhou, "Formal analysis of improved EAP-AKA based on protocol composition logic," in *The 2nd International Conference on Future Computer and Communication*, May 2010. DOI: 10.1109/ICFCC.2010.5497694.

[19] S. Mendes and C. Huitema, "A new approach to the X.509 framework: Allowing a global authentication infrastructure without a global trust model," in *Proceedings of the Symposium on Network and Distributed System Security*, pp. 172–189, Feb. 1995.

[20] B. Meng, J. T. Lu, D. J. Wang, and X. D. He, "Survey of security analysis of security protocol implementations," *Journal of Shandong University (Natural Science)*, vol. 53, no. 1, pp. 1–18, 2018.

[21] A. Roy, A. Datta, A. Derek, J. C. Mitchell, and J. P. Seifert, "Secrecy analysis in protocol composition logic," in *Advances in Computer Science*, pp. 197–213, 2006.

[22] J. Song, M. Xiao, K. Yang, X. Wang, and X. Zhong, "LoET-E: A refined theory for proving security properties of cryptographic protocols," *IEEE Access*, vol. 7, pp. 59871–59883, 2019.

[23] C. H. Wei, M. S. Hwang, and A. Yeh-Hao Chin, "A secure privacy and authentication protocol for passive RFID tags," *International Journal of Mobile Communications* , vol. 15, no. 3, pp. 266–277, 2017.

[24] L. Yu, Y. Y. Guo, and M. M. Jiang, "Improvement of strand space theory for application of minimal element method," *Quarterly Journal of Indian Pulp and Paper Technical Association*, vol. 30, pp. 94–105, Mar. 2018.

[25] Z. You, J. T. Li, and X. Xie, "Extension and application of protocol composition logic," in *The 2nd International Conference on Computer Engineering and Technology*, Apr. 2010. DOI: 10.1109/ICCET.2010.5485720.

# Biography

**Lei Yu** was born in 1978. He received the MS an BS degree in computer science and technology from Huaibei Normal University of China. Currently, he is an assistant professor and MS supervisor in the school of computer science and technology, Huaibei Normal University, China. His major research interests include cryptography and information security. He has published many papers in related journals.

**Zhi-Yao Yang** was born in 1995. He received the B.S.degree from Information College of Huaibei Normal University in 2017. He has been with the School of Mathematical Science, Huaibei Normal Universtiy, where he is a postgraduate student. His research interests include cryptography and information theory.

**Ze-Peng Zhuo** was born in 1978. He received the M.S.degree from Huaibei Normal University in 2007, and the Ph.D. degree from Xidian University in 2012. Since 2002, he has been with the School of Mathematical Science, Huaibei Normal Universtiy, where he is currently a professor. His research interests include cryptography and information theory.