International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 24 (2015) pp 44271-44277
© Research India Publications. http://www.ripublication.com

Formal Verification on Signcryption Re-Cryptography: Secure and Efficient Approach towards Trust Problem

Gautam Kumar
Department of Computer Science & Engineering Jaypee University of Information Technology Waknaghat, H. P. -173234 INDIA
gautam.kumar@mail.juit.ac.in

Hemraj Saini
Department of Computer Science & Engineering Jaypee University of Information Technology Waknaghat, H. P. -173234 INDIA
hemraj1977@yahoo.co.in

Abstract
Cryptography is a discipline of computer science, which is directing the requirement specifications for satisfactory protection mechanism with efficient and smooth functioning in the real world. Signcryption is one of the most promising primitives of cryptography that was proposed by Y. Zheng (1997), that has rationally combines digital signature and encryption in a single step, for lowering the less computational and communications cost when compared with the cost of separate signature and encryption schemes. The concept of proxy re-cryptography first proposed by Blaze at Eurocrypt (1998), and formalized by Ateniese and Hohenberger (2005). They defined the model using two approaches like proxy re-signature and proxy re-encryption. In this manuscript, we directed towards a probably secure and efficient approach regarding the trust problem for third party, who is not directly involved 'called proxy', can be solved using signcryption re-cryptographic approach. In modern era of cryptography, this is one of the new diverse trend and motivating issues. To solve the cryptological problems such as trust and ciphertext access control problems into a single location so that researchers can evaluate their suitability for various applications. Research interest focuses on situations under a cryptographic key management by a semi-trusted proxy with special information where data encrypted under one cryptographic key need to be re-encrypted. Further, proposed work has simulated on AVISPA/SPAN, using the automated formal verification tool.

Keywords: Signcryption, Proxy Re-Cryptography, Trust Problem, Trusted Server Problem, AVISPA, SPAN.

1. Introduction

Diffie-and-Hellman (1976) [1] has first proposed the idea of public key cryptographic protocol wherein the public key infrastructure (PKI) has been developed for generating and maintaining the public-keys using the corresponding certificates. However, the PKI suffers from heavy management of public keys and certificates. An alternative solution is Shamir's identity-based crypto systems (IBC). However, shortcoming of IBC is the key escrow problem [2]. The key escrow is a key exchange process in cryptography where a key is held or escrow, by a third party. The key is compromised or lost by its original user(s) may be used to decrypt encrypted matter, and allowing restoration of the primary matter to its unencrypted state. Somewhat the third party involved is risky in escrow systems. Key escrow enables to provide a backup source for cryptographic keys. The modern cryptography in an interdisciplinary approach of computer science focusing on the trust problem is solved using the proxy re-cryptographic primitive. The concept of proxy re-cryptography was first proposed by Blaze, Bleumer, and Strauss (1998). This approach was formalized by Ateniese and Hohenberger (2005), consists of two methods such as: proxy re-encryption and proxy re-signature. Where, the goal of proxy re-encryption is to securely enable the re-encryption of cipher texts from one key to another, without relying on honest parties. Similarly, the goal of proxy re-signature to securely enable the signature signed by one to transform to another signature on the same message duly signed for another without relying on trustworthy parties. In (2006) they proposed enhanced few proxy re-signature schemes and also discussed its several potential applications related to the same. They predicted that proxy re-encryption and proxy re-signature will play an important role. Since then, researchers are sparked to give fairly light in this area. That's why some schemes excellently have been proposed, especially, the IEEE P1363.3 standardization group is establishing the standard for proxy re-encryption, which will certainly give power on researching in the field of proxy re-cryptography [3]. A semi-trusted is an entity to convert cipher texts addressed to those that can be decrypted by using some special information.

For primitives of the proxy re-cryptography such as, signcryption proxy re-signature (SCPRS), signcryption proxy re- encryption (SCPRE), and security models are motivated for the same [4]-[5].

In this manuscript, a more optimized notion of signcryption with proxy re-cryptographic definition and its formal verification have presented, and its efficiency motivation has specified. Finally, it provides directions for further research in this area in the concluding section.

1.1 Trust Problem

To solve the trustworthy problem within the domain of fully trusted authority to build the absolute trust relationship is challenging issues. The public-key infrastructure certificate authority releases a public-key certificate, which is signed by the trusted authority to bind with the identity [6]. It is used to verify that a public key belongs to an individual. However, how to build offshore trust relationships between honest, trusted authority domains is a difficult task in practical problem. The goal is to solve this problem, to set up a transfer server, ‘called proxy’, who is allowed to transfer certificates between the authorities, and the proxy can’t generate new certificates. Instead of that, it requires extra abilities of the proxy in some concrete applications. Sometimes it is desired that certificates of authority only transfer in a single direction known as unidirectional transformation. Bidirectional transformation is known to authorize in both direction. On the other hand, the requirements in the trusted domains further extended the process that continues from one of the many more proxies transfer is known as multiuse. Trust problem is a significant asset of a new cryptographic primitive called proxy re-signature to solve the above.

1.2 Trusted Server Problem

This problem has emerged with the cloud computing that reduces the cost of hardware and software resources in computing fields. Almost all cloud storage servers are exerting and responsible for sensitive information, like electronic storage user’s data, and the cloud access server over the data access. It is usually required that the cloud access control server is fully trusted, but this requirement can’t meet in practice for two reasons. One is that the provider(s) of control service can’t be assumed to be fully trusted, because that it could be corrupted in some situations.

A possible solution is to store the encrypted plaintext at the server of cloud storage. The trusted server problem can be easily solved through this. The encrypted cipher texts need to be shared by others, and the access control server has no right to perform decryption; it is a challenging problem. Under this condition, the following solution can be conceived: let the Encryptor authorize the access control server the right to transform the cipher texts so that the delegated users can decrypt the resulting cipher texts, but the access control server can’t decrypt the cipher texts. If the access control server under the authorization of the Encryptor can transform the cipher texts stored on the cloud storage server into a new form with the same plaintexts that can only be decrypted by the designated receivers. It is regarded the access control server, Encryptor, designated receivers, and authorization messages as the proxy, delegator, delegatee, and re-encryption keys, this is a particular case of proxy re-encryption [7]-[8].

1.3 Ciphertext Access Control Problem

Assume that the data owner intends to store a private message that is accessed by a specific set of users. The most motivating solution is that the data owner laid downs the data in plaintext form in the repository storage server, and the user’s access rights are specified by access control lists that are created by the data owner and performed by the access control server. The users specified by the access control lists and verified by the access control server, can access the message. However, trust and security issues of the servers are always serious in practice.

A trivial method would be to store the data into ciphertext form in the servers. However, the current encryption system can’t allow the ciphertext to be efficiently shared among a user group. It is becoming an urgent to develop a flexible and efficient method to share data directly based on encrypted plaintexts and also includes the access control policy. Fortunately, Bethencourt proposed such a cryptographic primitive called ciphertext-policy attribute-based encryption (CP-ABE) [9], which initiates a new direction in solving the ciphertext access control problem [10].

2. Signcryption

Signcryption is one of the cryptographic primitives, proposed by Y. Zheng (1997), which logically combines digital signature and encryption in a single step for achieving less communication and computational cost [11]. The practical application of signcryption in real life is like killing two birds with one stone. He has also proposed an elliptic curve (EC) based scheme on it that saved 58% of calculative cost and 40% of communication cost when it is compared with the individually EC-based signature-then-encryption schemes [12]. This brings savings in communication and computation. There are various and huge applications of signcryption are available that are widely used for electronic commerce in sheltered and substantiated transactions, invulnerable and validated message delivery, safe and authenticated multicast inclusive video fast, conferencing, compact, non-repudiated key transport and unforgeable.

Since then there are many other schemes have been proposed throughout the years, having its own problems and limitations, with offering different levels of security and computational costs. Through the encryption algorithm, confidentiality is achieved, whereas integrity is provided using authentication techniques. Authentication techniques categorized in two forms such as public-key digital signatures and private key authentication algorithms [13].

Signcryption has the intention that should satisfy this condition: costs of both ‘signature and encryption’ are too less compared with separate cost of signature and encryption. These can also be interpreted in a number of ways, such as: (i) this scheme is more computationally efficient than the other native combination of public-key encryption and digital signatures. (ii) These are to produce a cipher text which is shorter than a naive combination of a public key encryption ciphertext and a digital signature. (iii) These are also to provide finer security guarantees and/or finer functionality than a native combination of public-key encryption and digital signatures.

The digital signature (DS) is a fully mathematical scheme that demonstrates the authenticity of a message digests. This DS scheme generally consists of the three steps:

i. The key generation that selects a personal key at random from the possible set of particular keys, that output's private key and its corresponding public value.
ii. On behalf of the message and private key produces the signature and
iii. After that the verification phrase occurs on the message, public keys and signature.

A signcryption scheme that includes DS as well as encryption consists typically into five phases, such as: Setup, Key Generation by Sender, Key Generation by Responder, Signcryption, and finally Unsigncrypt.

Recently, there have been many areas where signcryption applications are widely accepted due to its ability to connect to the Internet, tiny digital phone such as PDAs, wireless transport-layer security handshake protocol, reduced bandwidth and its decreased computational load [14]. Also, the second major application is unforgivable key establishment over ATM networks.

3. Proxy Re-Cryptography

This is used to solve the trust problem, instead of that there are many applications such as digital-right management (DRM) that prevents the illegal redistribution of digital content. In 2006, Taban [15] proposed an entirely new interoperability architecture or modern module in the existing DRM called the domain interoperability manager (DIM). DIM applies a unique signature scheme and a particular public key encryption scheme. The traditional signature and public key encryption don't support transformation, but using proxy re-cryptography; this can be easily implemented. This scheme contains the two phases as: proxy re-signature and proxy re-encryption. Each phase contains its own properties and definition. A pictorial proxy re-cryptography digests approach has shown in figure 1.

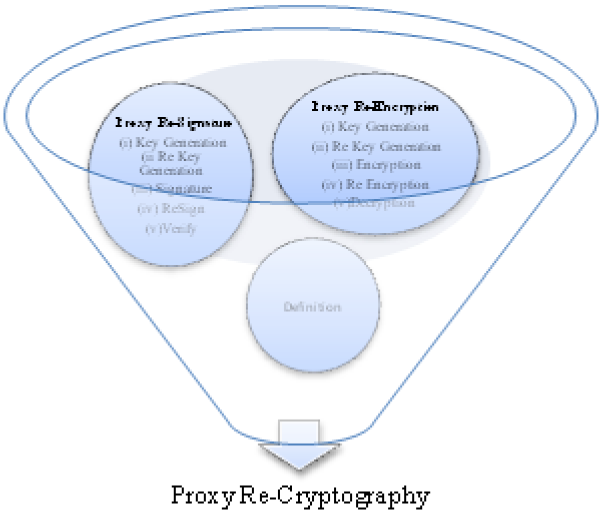


Figure 1: Proxy Re-Cryptography Digest

3.1 Proxy Re-Signature (PRS)

In this scheme, a delegate's signature transforms his/her signature using a semi-trusted proxy to a delegatee's on the same message by using some additional information. The proxy can't generate an arbitrary signature on behalf of either the delegate or delegatee.

3.1.1 Properties of Proxy Re-Signature

i. A Unidirectional or Bi-directional: The proxy is to allow for re-signature key either in uni-directional or bidirectional transformation.

ii. Multiuse: In this case, the proxy transforms the signature can be re-transformed again by a proxy. Even so, the signature does not transform a single use.

iii. Private Proxy: In private proxy, the re-signature key to a secret in scheme because anyone can compute re-signature by observing re-signature process passively in public proxy scheme.

iv. Transparent: The scheme should be see-through so the user(s) does not know the existence of proxy.

v. Key-Optimal: In this, a user is required to protect and store only a small constant amount of secrets, no matter how many signature delegations the user gives acceptance.

vi. Non-interactive: The parties involved are an ideal and not required during the commission process.

vii. Non-transitive: Other than the two, signature can't be generated from at any case for the same.

viii. Temporary: The right of re-signing is interim. This can be done by either revoking the right or expire the right.

ix. Collusion resistance: Via proxy, the delegator consigns the signing rights to the entrust delegate, instead keeping the decryption rights for the same public key.

3.1.2 Definition of Proxy Re-Signature

The proxy re-signature follows the following five steps:

i. Key Generation: The security parameter λ takes as input, and that returns a verification key pk and a signing key sk.

ii. Re-Key Generation: It takes as an input delegate key pair (pk_A, sk_A) , and a delegatee key (pk_B, sk_B), and returns a re-signature key $rk_{A\leftarrow B}$ for the proxy. If the scheme is unidirectional, the delegates signing key are not included in the input. But in the case of bidirectional, the proxy can be easily obtained $rk_{B\leftarrow A}$ from $rk_{A\leftarrow B}$. In many bidirectional schemes $rk_{A\leftarrow B} = 1/rk_{B\leftarrow A}$.

iii. Signature: It takes as input a signing key sk, a positive integer l, and a message m from message space, and returns a signature σ at level l. If this scheme is single use, then $l \in \{1,2\}$.

iv. Re-signature: It takes as input a re-signature key $rk_{A\leftarrow B}$, and a signature σ_A, taking place message m under pk_A, on level l, and returns the signature σ_B on the same message m under pk_B, at level $l+1$ if verify $(pk_A, m, pk_B, l) = 1$, or reject otherwise. If the scheme is single use $l = 1$.

v. Verify: This takes as input of verification key pk, the message m from the message space, the signature σ and a positive integer l, and returns 1 if σ is a valid signature under pk at level l or otherwise.

4. Signcryption with Proxy Re-encryption

The proxy signcryption scheme has the general condition, which divided into three parties such as delegate signer, proxy signer and the delegatee recipient. In this scheme, the delegate signer generates a proxy credential to the signing authority to a proxy signer. The proxy then after generates signcrypted message using a secret key and its own proxy credentials.

Finally, the proxy sends the signcrypted message to an assigned recipient through a network. After receiving the signcrypted message, the recipient recovers the content from the same and also verifies its validity. If any dispute occurs, the recipient is free to announce the signature of proxy for public verification.

The notion of signcryption [16] with proxy re-encryption [17] have presented here. This scheme consist proxy re-encryption, authenticity and confidentiality in a very efficient way. This primitive have various such applications, as:

(i) Email is the best candidate for applying signcryption. An application of signcryption of proxy re-encryption (SCPRE) is to allow and forwarding the message for authentication using signcrypted to be directed to a person when the original receiver is unavailable.

(ii) Another well-known application for secure and authentic distributed storage that can be extended whenever the content stored for authentication is desirable.

The signcryption of proxy re-encryption scheme follows the following steps:

i. Setup: The algorithm accepts a security parameter I and outputs a master secret key s.

ii. Extraction: The algorithm accepts an identity ID_u, and outputs the secret key S_u.

iii. Extract-rekey: It accepts two ID_1 and ID_2 , and outputs the rekey from ID_1 and ID_2.

iv. Signcryption: The signcryption accepts messages m, and two identities ID_1 and ID_2, and outputs the signcryption for m from ID_1 and ID_2

v. De-signcrypt: This accepts a signcryption message φ and identity ID_r, and outputs the de-signcryption of φ by ID_r.

vi. Re-encryption: It accepts a signcryption φ, and an identity ID_r, and outputs the re-encrypted signcryption φ' of φ to ID_r.

vii. De-re-encrypt: This accepts a second-level signcryption φ' and ID_d, and outputs the de-signcryption of φ' by way of ID_d .

4.1.1 The Scheme of signcryption proxy re-encryption (SCPRE)

The SCPRE scheme is derived from the identity-based signcryption scheme; the presented scheme is as follows:

Setup

Let I be the security parameter of the system. Let G_1 and G_2 be two prime ordered groups of order $q = \theta(2^I)$, where G_1 be represented additively, and G_2 be represented multiplicatively. Let P be a generator of G_1.

Let $e : G_1 \times G_2 \rightarrow G_2$, be a bilinear pairing. We assume that the Bilinear Computational Diffie-Hellman (BCDH) assumption holds in $< e, G_1, G_2 >$.

It uses four hash functions H_0, H_1, H_2 and H_3, where
$H_0: \{0,1\}^* \rightarrow G_1$,
$H_1: G_1 \times \{0,1\}^n \rightarrow Z_q^*$.
$H_2: G_2 \rightarrow \{0,1\}^{n+t}$
$H_3: G_1 \times \{0,1\}^* \rightarrow G_1$

The n is the number of bits in the message, and t is the number of bits used to represent an element in G_1.

The private key generator (PKG) chooses the master secret key $s \in R\, Z_q^*$, and sets the master public key $P_{pub} = sP$.

The published public parameters are $< e, G_1, G_2 , n, q, P, P_{pub}, H_0, H_1, H_2 >$. Each user have his/her identity ID_u, and public key. He/she gets two secret keys S_u, and $S_{u||delegatee}$, by providing ID_u and $ID_{u||"delegatee"}$.

Extract (IDu)

The public key generator (PKG) computes the secret key as $S_u = s.H_0(ID_u)$, where $H_0(ID_u)$, is generally denoted as Q_u

Signcrypt ($\boldsymbol{m, S_A, ID_B}$)

User A is to signcrypt a message m from delegator A to delegate B by using steps as:

1. Choose $r \in R\, Z_q^*$
2. Compute $X = rQ_A$ and $h = H_1(X||m)$
3. Compute the signature $Z = (r + h)S_A$
4. Choose $k \in R\, G_2$
5. Compute $Z = e(S_A, Q_B)^r$, and set $\lambda = w.k$
6. $y = H_2(k) \oplus (m||Z)$
7. The signcryption is $\emptyset =< X, y, ID_A >$.

De-signcrypt ($\boldsymbol{\emptyset =< X, y, \lambda, ID_A >, S_B}$)

The delegatee receiver B, after receiving the signcryption $\emptyset$, does the following.

1. $w = e(X, S_B)$
2. Compute $k = \lambda.w^{-1}$
3. Recover $m||Z = y \oplus H_2(k)$
4. $h_1 = H_1(X||m)$
5. If $e(Z, P) = e(P_{pub}, X + h_1.Q_A)$, then $< m, (X, Z), ID_A >$ This is the output as the message and signature. Otherwise, ⊥ is output.

Rekey-Extract ($\boldsymbol{S_B, ID_C}$)

B sends $rk_{B \rightarrow C} =< -S_B + H_3(e(S_B, Q_{(c||delegate)}))\ >$, to the proxy.

Re-encrypt ($\emptyset =< X, y, \lambda, ID_A >, rk_{B \rightarrow C}, ID_B, ID_C$)

The proxy computes re-encrypted signcryption $\emptyset' =< X, y, \lambda. e(X, rk_{B \rightarrow C}), ID_A, ID_B >$, and send $\emptyset'$ to C.

De-re-encrypt ($\emptyset' =< X, y, \lambda', ID_A, ID_B >, S_{c||delegatee}$)

On receipt of a level 2 signcryption, C decodes the algorithm as follows:

1. $w = e\left(X, H_3, \left(e\left(Q_B, S_{c||delegatee}\right)\right)\right)$
2. Compute $k = \lambda' w^{-1}$
3. Recover $m||Z = y \oplus H_2(k)$
4. $h_1 = H_1(X||m)$
5. If $e(Z, P) = e(P_{Pub}, X + h_1 Q_A)$, then output$< m, (X, Z), ID_A >$, else output ⊥.

The long-term goal is to collect a number of new proxy re-encryption and re-signature schemes into a single location so that researchers can evaluate their suitability for various applications.

5. Formal Validation Using AVISPA/SPAN Tool

AVISPA [18] is one of the formal tool for automatic validation and verification, that are pertinent for the Internet security applications and its protocols. It offers a significant expressive formal language for specifying protocols with their safety measures that has modularized into different four back-ends under the perimeter, structured shown in figure 2. Its accomplishment is based on the automatic analysis techniques. The High Level Protocol Specification Language (HLPSL) is used for describing security protocols and specifying the intended security properties, as well as to formally validate them. The HLPSL specification first translated into Intermediate Format (IF) through translator HLPSL2IF. Where the IF is a lower-level language and, that, is directly interpreted for back-ends tool. The IF objective has formulated for developers with the implication to use as their input language analysis. This happens automatically and is transparent to the user [19]. Now, the IF specification analyzed at the back-ends for the satisfied or violated security goals. The AVISPA Tool comprises four back-ends such as: On-the-fly Model Checker (OFMC) [20], Constraint Logic-based Attack Searcher (CL-AtSe) [21], SAT-based Model Checker (SATMC) [22]-[23], and Tree-Automata Based Protocol Analyzer (TA4SP) [24].

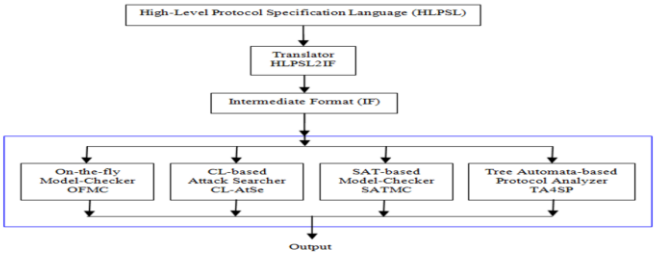
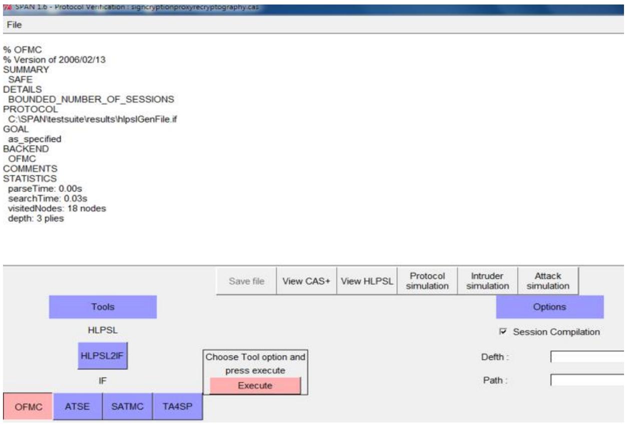


Figure 2: **AVISPA Structure**

An impressive SPAN tool comes with simple editing protocol specifications of web graphical interfaces of AVISPA, and in addition to this it contains honest agents for protocol simulation, intruder simulation for honest agents and an attack simulation. Where SPAN either to accept HLPSL or CAS+ specification as an input. But AVISPA only HLPSL only. In a broader sense SPAN is more robust than AVISPA.

- ✓ Protocol Simulation is used for simulating the protocol and building a particular Message Sequence Chart (MSC) corresponding to the HLPSL specification (Animation: based with no intruder).
- ✓ Intruder Simulation for simulating the protocol with the active/passive intruder (Animation: based on to build your own attack by hand).
- ✓ Attack Simulation for automatic building of MSC attacks from the output of either On-the-Fly Model Checker (OFMC) or Constraint Logic based Attack Searcher (CL-ATSE) tools**.** Attack simulation in this, like the same layout as intruder simulation, but attacks are automatically built using OFMC/CL-AtSe facilities.

This means a security protocol animator for High Level Protocol Specification Language (HLPSL) and CAS+ specifications. HLPSL is a used language for specifying the cryptographic protocols for AVISPA toolset and CAS+ is a light evolution of CASRUL language for SPAN.

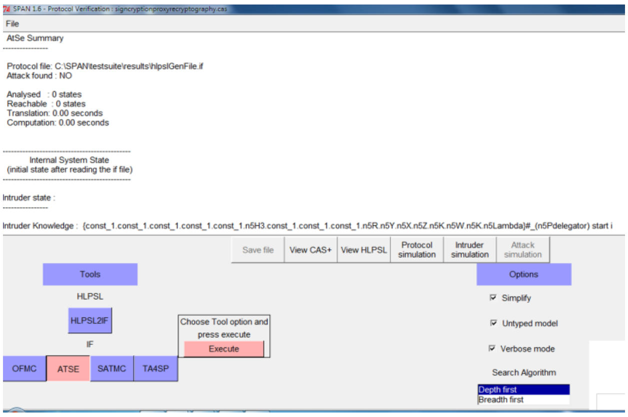


```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  C:\SPAN\testsuite\results\hlpslGenFile.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.03s
  visitedNodes: 18 nodes
  depth: 3 plies
```

☑ Session Compilation

Figure 3: **SPAN on OFMC Back End**

We simulated signcrypted proxy re-cryptographic approach in CAS language and shown its sender pattern principal information executed on OFMC back end tool. It is a useful debugging tool to check manually that your protocol specification allows agents to execute all the steps required for honest run of the principals, resultant in the form of SAFE state, depicted in figure 3.



```
AtSe Summary
---------------

 Protocol file: C:\SPAN\testsuite\results\hlpslGenFile.if
 Attack found : NO

 Analysed   : 0 states
 Reachable  : 0 states
 Translation: 0.00 seconds
 Computation: 0.00 seconds


-------------------------------------------
          Internal System State
 (initial state after reading the if file)
-------------------------------------------

Intruder state :
---------------

Intruder Knowledge :  {const_1.const_1.const_1.const_1.const_1.n5H3.const_1.const_1.const_1.n5R.n5Y.n5X.n5Z.n5K.n5W.n5K.n5Lambda}#_(n5Pdelegator) start i
```

☑ Simplify

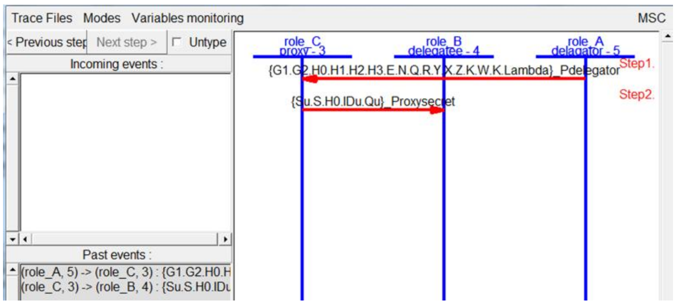
☑ Untyped model

☑ Verbose mode

Search Algorithm
Depth first
Breadth first

Figure 4: **SPAN on AtSe Protocol Check**

CL-ATSE is a set of constraints, used to find attacks on protocols. The translation and checking are fully automatic and internally performed by the same i.e. no external tool is used. Its back-end uses a slightly different format for some aspects of attack traces than OFMC does. For example, it writes an interpretation of the IF facts as tests or actions in the attack trace. This has executed the same on AtSe tool, shown in figure 4, which is presentation with negligible possibility of attack.



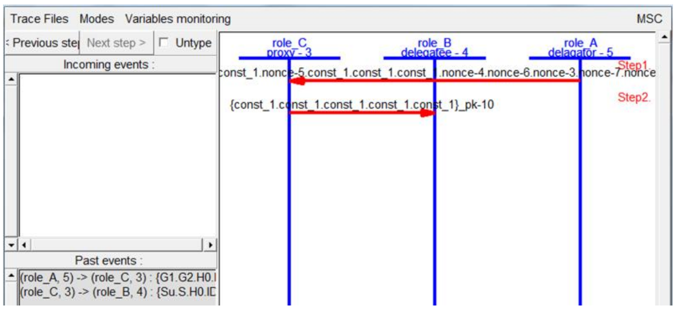
{G1.G2.H0.H1.H2.H3.E.N.Q.R.Y.X.Z.K.W.K.Lambda}_Pdelegator

{Su.S.H0.IDu.Qu}_Proxysecret

(role_A, 5) -> (role_C, 3) : {G1.G2.H0.H
(role_C, 3) -> (role_B, 4) : {Su.S.H0.IDu

Figure 5: **Sender pattern principal**

The specification has automatically simulated in the proposed approach between delegator and delegatee via a third party of proxy. Here in figure 5, the pattern of sender principal has shown according to the above provided definition. The delegator, sends the message to proxy, where secret via proxy is added and sent to the delegatee where it is deciphered.



const_1.nonce-5.const_1.const_1.const_1.nonce-4.nonce-6.nonce-3.nonce-7.nonce

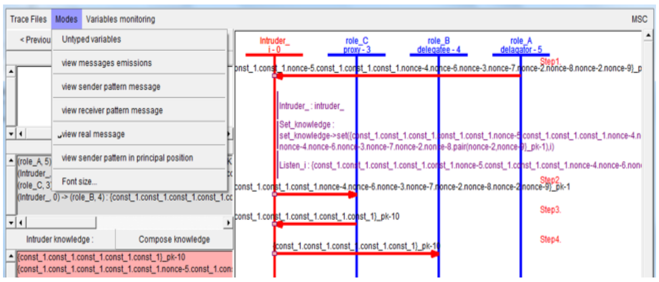
{const_1.const_1.const_1.const_1.const_1}_pk-10

(role_A, 5) -> (role_C, 3) : {G1.G2.H0.I
(role_C, 3) -> (role_B, 4) : {Su.S.H0.ID

Figure 6: **Real Type of Sending Messages View**

This permits to translate a CAS+ specification for fast and simple specification of security protocols; interactively building a Message Sequence Chart (MSC) [25]-[26] of protocol execution; automatically build attacks on MSC lying on HLPSL and CAS+ specifications; and interactively build specific attacks on specifications using the intruder mode. But, originally message are sent in the form of encrypted form, where it is like to be impossible to decrypt, depicted in figure 6.

The definition has simulated with the Intruder with its knowledge, in figure 7, with the real sender pattern principle.



onst_1.const_1.nonce-5.const_1.const_1.const_1.nonce-4.nonce-6.nonce-3.nonce-7.nonce-2.nonce-8.nonce-2.nonce-9}_p

Intruder_ : intruder_

Set_knowledge :
set_knowledge->set({const_1.const_1.const_1.const_1.const_1.nonce-5.const_1.const_1.const_1.nonce-4.n nonce-4.nonce-6.nonce-3.nonce-7.nonce-2.nonce-8.pair(nonce-2,nonce-9)}_pk-1),i)

Listen_i : {const_1.const_1.const_1.const_1.const_1.nonce-5.const_1.const_1.const_1.nonce-4.nonce-6.non

const_1.const_1.const_1.nonce-4.nonce-6.nonce-3.nonce-7.nonce-2.nonce-8.nonce-2.nonce-9}_pk-1

const_1.const_1.const_1.const_1.const_1}_pk-10

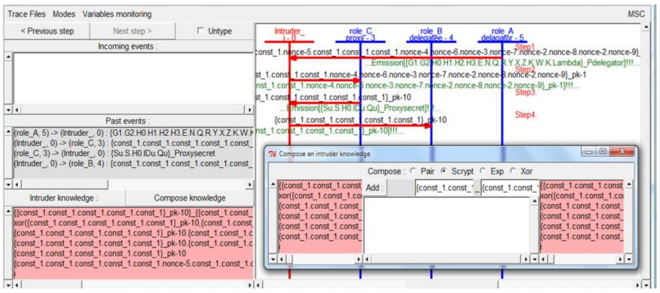
{const_1.const_1.const_1.const_1.const_1}_pk-10

(role_A, 5)
(Intruder_
(role_C, 3)
(Intruder_, 0) -> (role_B, 4) : {const_1.const_1.const_1.const_1.cc

{const_1.const_1.const_1.const_1.const_1}_pk-10
{const_1.const_1.const_1.const_1.const_1.nonce-5.const_1.con

Figure 7: **Intruder Simulation with knowledge on real messages**

Further, in the last but not the least, the various additional composition behaviors are also available, exposed in figure 8. SATMC's is used to check the executability that includes functionality to confirm the executability of a HLPSL specification. SATMC is particularly strict about the proper use of types in HLPSL specifications; this feature can thus be very useful for finding errors relating to typing that may lead to non-executability of a protocol specification.



const_1.nonce-5.const_1.const_1.const_1.nonce-4.nonce-6.nonce-3.nonce-7.nonce-2.nonce-8.nonce-2.nonce-9}_
...Emission[{G1.G2.H0.H1.H2.H3.E.N.Q.R.Y.X.Z.K.W.K.Lambda}_Pdelegator]!!!...
st_1.const_1.const_1.nonce-4.nonce-6.nonce-3.nonce-7.nonce-2.nonce-8.nonce-2.nonce-9}_pk-1
nst_1.const_1.nonce-4.nonce-6.nonce-3.nonce-7.nonce-2.nonce-8.nonce-2.nonce-9}_pk-1]!!!...
st_1.const_1.const_1.const_1.const_1}_pk-10
...Emission[{Su.S.H0.IDu.Qu}_Proxysecret]!!!...
{const_1.const_1.const_1.const_1.const_1}_pk-10
nst_1.const_1.const_1.const_1}_pk-10]!!!...

(role_A, 5) -> (Intruder_, 0) : {G1.G2.H0.H1.H2.H3.E.N.Q.R.Y.X.Z.K.W.K
(Intruder_, 0) -> (role_C, 3) : {const_1.const_1.const_1.const_1.const_
(role_C, 3) -> (Intruder_, 0) : {Su.S.H0.IDu.Qu}_Proxysecret
(Intruder_, 0) -> (role_B, 4) : {const_1.const_1.const_1.const_1.const_

{{const_1.const_1.const_1.const_1.const_1}_pk-10}_{{const_1.const_
xor({const_1.const_1.const_1.const_1.const_1}_pk-10,{const_1.const_
{const_1.const_1.const_1.const_1.const_1}_pk-10.{const_1.const_1.c
{const_1.const_1.const_1.const_1.const_1}_pk-10.{const_1.const_1.c
{const_1.const_1.const_1.const_1.const_1}_pk-10
{const_1.const_1.const_1.const_1.const_1.nonce-5.const_1.const_1.c

Figure 8: **With Intruder Real Type Pattern with Emissions**

The TA4SP proves secrecy properties with an unbounded number of sessions. From the practical point of view, this works completely automatic and supported by two (2) tools such as Timbuk and its extensional part. The analysis of four back-ends are harmonized to each others in a sense for some common back-ends procedure, but these are not equivalent that should return different results.

6. Conclusion and Future work

This is a motivation in the new direction of cryptography using the approach proxy re-cryptography for secure signcryption based protocol. Today, this is one of the most highly demanding cryptographic applications in the recent scenario and various challenging issues in the applied cryptography to preserve the strong connection between mathematics and information security. Signcryption using this approach is the new paradigm for tremendous demanding of cost effective, high performance, application for short-memory devices and so on. In addition, we would like to point out some of the future works such as (i) to collect for the long-term schemes using proxy re-cryptography into a single location through researchers can evaluate their suitability for various applications. (ii) The approach for modern cryptography with security requirements have arisen in different distributed environments as the attacks may come either from internal or external objects, (iii) proxy re-cryptography should be in the standard model and collusion-resistant.

References

[1] Diffie, W. and Hellman, M.E., "New Directions in Cryptography," IEEE Transaction on Information Theory, vol. 22, No. 6, pp. 644-654, 1976. DOI: 10.1109/TIT.1976.1055638

[2] Islam, S.K.H., and Biswas, G.P., “Provably secure certificateless strong designated verifier signature scheme based on elliptic curve bilinear pairing,” Journal of King Saud University, vol. 25, 51-62, 2011. DOI: 10.1016/j. jksuci.2012.06.003

[3] Cao, Z., New Directions of Modern Cryptography. Proxy Re-cryptography, CRC Press. Taylor & Francis Group, Chap. 2, 2013.

[4] Chadrasekhar, S., Ambika, K., Rangan, C. P., “Signcryption with Proxy Re-encryption,” Journal of Cryptology, IACR Eprint, 1-19, 2008. DOI: http://eprint. iacr.org /2008 /276

[5] Introduction to NISTIR 7628 guidelines for smart grid cyber security. National Institute of Standards and Technology (NIST), 2010 [Online]. Available at: http://www.nist.gov/ smart grid/ upload/nistir7628_total.pdf.

[6] Ateniese, G., Fu, K.., Green, M., Hohenberger, S., “Improved Proxy Re-encryption Schemes with Applications to Secure Distributed Storage,” Proceedings of the 12th Annual Network and Distributed Systems Security Symposium NDSS’05, San Diego, California, 2005. Available at: http://spar.isi.jhu.edu/~mgreen/proxy.pdf

[7] Shao, J., Cao, Z., “Multiuse unidirectional identity-based proxy re-encryption from hierarchical identity-based encryption,” Journal Information Sciences, 206, 83-95, 2012. DOI: dl.acm.org/citation.cfm?id=2228798

[8] Blaze, M., Bleumer, G. and Strauss, M., “Divertible protocols and atomic proxy cryptography,” In EUROCRYPT 1998, volume 1403, LNCS, pp. 127–144, 1998. DOI: 10.1007/BFb0054122

[9] Green, M., Ateniese, G., “Identity-Based Proxy Re-Encryption,” Applied Cryptography and Network Security Conference, vol. 4521, pp. 288-306, 2007. Available: http://eprint.iacr.org/eprint-bin/cite.pl?entry= 2006/473

[10] Li, Fagen., Khan, M. K., Alghathbar, K., Takagi, T., “Identity-based online/offline signcryption for low power devices,” Journal of Network and Computer Applications, 35 (1), 340-347, 2012. DOI: 10.1016/j. junkie.2011.08.001

[11] Zheng, Y., “Digital signcryption or how to achieve Cost (Signature & Encryption) << Cost (Signature) + Cost (Encryption),” Advances in Cryptology-CRYPTO’97, LNCS, Springer-Verlag, 1294, 165-179, 1997.

[12] Baek, J., Steinfeld, R., Zheng, Y., “Formal Proofs for the Security of Signcryption Public Key Cryptography,” (PKC 2002), LNCS, Springer-Verlag, 2274, pp. 80-98, 2002.

[13] Wang, Z., Du, X., Sun, Y., “Group Key Management Scheme of Proxy Re-cryptography for Near Space Network,” IEEE Int’l conf., 1-5, 2011.

[14] Vuillume, C., Okeya, K., Takagi, T., “Short-Memory Scalar Multiplication for Koblitz Curve,” IEEE transactions on Computers, 57 (4), 481-489, 2011.

[15] Taban, G., Cardenas, A. A., and Gligor, V. D., “Towards a secure DRM Architecture,” DRM’06, October 30, Alexandria, Virginia, USA. DOI: 10.1.1.85.360, 2006.

[16] Zheng, Y., “Signcryption and Its Applications in Efficient Public Key Solutions,” Information Security Workshop (ISW’97), Lecture Notes in Computer Science, Springer-Verlag, Vol.1397, pp.291-312, 1998. DOI: 10.1.1.101.3826

[17] Ohata, S., Kawai, Y., Matsuda, T., Hanaoka, G., and Matsuura, K., “Re-encryption Verifiability: How to Detect Malicious Activities of a Proxy in Proxy Re-encryption,” Springer, Journal of Cryptology, 2015. Available: https://eprint.iacr.org/2015/112.pdf

[18] AVISPA-Automated Validation of Internet Security Protocols [online] Available: http://www.avispa-project.org

[19] Sperschneider, V. and Antoniou, G., “Logic: A Foundation for Computer Science,” 1st ed., Addison-Wesley Longman, USA, 1991.

[20] Basin, D., Modersheim, S., and Vigano, L., "OFMC: A Symbolic Model-Checker for Security Protocols, “International Journal of Information Security,vol. 4, pp.181-208, 2005. DOI: 10.1007/s10207-004-0055-7

[21] Turuani, M., “The CL-Atse Protocol Analyzer,” Lecture Notes in Computer Science, vol. 4098, F. Pfenning, (Eds.) in RTE, pp. 277-286, 2006. DOI: 10.1007/11805618_21

[22] Armando, A. and Compagna, L., “Automatic SAT-Compilation of Protocol Insecurity Problems via Reduction to Planning,” in Proceedings of FORTE 2002, Lecture Notes in Computer Science, vol. 2529, pp. 210–225, 2002. DOI:10.1007/3-540-36135-9_14

[23] Armando, A. and Compagna, L., “Abstraction-driven SAT-based Analysis of Security Protocols,” in Proceedings of TAST, Lecture Notes in Computer Science, vol. 2919, pp. 257-271, 2004. DOI:10.1007/978-3-540-24605-3_20

[24] Boichut, Y., Kosmatov, N., and Vigneron, L., “Validation of Prouve Protocols using the Automatic Tool TA4SP,” in 3rd Taiwanese-French Conf. on Information Technology, pp. 467-480, 2006.

[25] Harel, D. and Thiagarajan, P.S., “Message Sequence Charts, UML for Real: Design of Embedded Real-time Systems,” ACM Digital Library, pp. 77-105, 2003. Available at: http://dl.acm.org/citation.cfm?id=886349

[26] Dolev, D., and Yao, A., “On security of Public key protocols,” IEEE Transactions on Information Theory, vol. 29, 1983. DOI: 10.1109/TIT.1983.1056650.