Mechanism of Minimum Credential Disclosure with Privacy Preserved in Trust Builder2 Framework

Dhiraj Shrestha and Li Jianli

***Abstract*—Present credential disclosure technique requires disclosing more than one credential to establish a trust during Automated Trust Negotiation (ATN), which is time consuming and requires additional burden of credential management. An integrated credential approach is proposed for solving above mentioned problem. This approach is primarily designed for TrustBuilder2 framework. Credential integration approach requires incorporating all the available credentials of an individual into single credential. The privacy of integrated credential is achieved by associating policy that setup conditions for credential disclosure. During ATN the entire content of the credential is not shown only those content which is required for establishing trust are exhibited. These approaches assure the privacy protection while undergoing Trust negotiation.**

***Index Terms*—Trust negotiation, digital credential, privacy, policy.**

I. Introduction

In the age of information technology, saving the resources available in the open environment, from intruders is a prime concern. Examining a trustworthiness of an individual before releasing a resource is another main concern.

Property-based digital credential [1] is on-line analogous of paper credential that people carry in their wallets [2]. It acts as a digital identity of an individual. A credential contains attributes about the credential owner asserted by the credential issuer, usually represented as name/value pairs [2]. The most widely used certificate standard is X.509v3. [X509, 1997]. ATN [2]-[4] is guided by a set of credentials and policies. We might be interested from where the credentials are issued? To summarize a complex process, credentials are issued by the same kinds of organizations that issue paper credential today. To avoid attack, credentials are issued typically offline and then either securely distributed to their new owners or made available for pickup in a semi public database [2]. To get a particular resources or services we need to fulfill certain requirements which are defined by the policies and here requirement means credential, this is how a trust is established in ATN. The main problem with existing trust negotiation system are, they need to undergo several rounds of credentials exchange for establishing a trust that prolongs the duration of trust establishment and in the meantime credentials contents are shown which are not essential for trust establishment but may lead to the information misuse in future. For ex. Tian wants to hire a motor vehicle, in order to hire, he needs to prove that he has a valid license, and a valid credit card, if he fulfills this condition he can hire a vehicle. In this case, if Tian has single credential which contains both information of credit card and vehicle license than he can establish a trust by disclosing a single credential instead of two, which saves the negotiation time, and removes extra burden of credential management.

The main objective of this paper is to provide guidance on how credential can be integrated and its disclosed number is limited along with privacy preserved in open and distributed environment? A mechanism for minimum credential disclosure in ATN is proposed. The main highlights of our work is listed below

1) Designing an approach for minimal disclosure of credential during ATN. To achieve this goal we use the concept of Single Credential in which all the available credentials of an individual is incorporated into single credential that minimizes the number of credentials to disclose and removes the extra burden of an individual to manage different credentials.
2) Associating a policy with integrated credentials for ensuring security from intruders and IOManipulation module of TrustBuilder2[5]-[7] framework is redefined in order to remove the feature of credential transparency while undergoing ATN, i.e. , only those fields of credential is disclosed to the peer which are required for satisfying policies, other attributes / value pairs will be hidden. This is an approach for preserving privacy.
3) Our credential is an uncertified credential as there are no certifying authorities and is analyzed in TrustBuilder2 framework.

II. Related Work

There is mainly two different credential disclosure policies Eager strategy and parsimonious strategy [4], Eager Strategy is simple and efficient and leads to successful negotiation whenever necessary whenever possible. While in parsimonious strategy exchange credentials request that focus on credential exchange, gaining disclosure minimality.

Ryan D. Jarvis in his research work proposed selective disclosure principle [8] to prevent the unnecessary disclosure of irrelevant but privacy-sensitive credentials attributes. Credentials that are setup for selective attribute disclosure are called selective disclosure credentials. When a selective disclosure credential is disclosed, the owner can hide certain attributes in such a way that the negotiation partner cannot find out what their attribute-values are, but still verify that the credential as a whole is valid. The first work on selective disclosure was done by Holt et al., which was later extended by the work of Jarvis. They proposed a system based on bit-commitment and blind signatures.

Manuscript received July 28, 2012; revised September 29, 2012.

D. Shrestha is with the Department of Computer Science and Engineering, Kathmandu University, Dhulikhel, Nepal (e-mail: dhiraj@ku.edu.np).

L. J. Li is with the College of Computer Science and Technology, Harbin Engineering University, Harbin, China (e-mail: lijianli@hrbeu.edu.cn).

Bertino et al., proposed a similar system as of Jarvis without blind signature. They called them privacy-enhanced credentials [9] which are designed to be incorporated in the Trust-X trust negotiation framework. Privacy enhanced credentials are different from selective disclosure credentials in that the credential contents are explicitly separated from the credential self.

The earlier version of TrustBuilder2 was TrustBuilder1 which was implemented in java and it supports the use of X.509 certificates to encode attributes and XML to represent policies written using the IBM Trust Policy Language (TPL) [10]. The IBM Trust Establishment (TE) compliance checkers is used to determine whether a certain set of credentials satisfies a given policy. TrustBuilder1 has been embedded into an implementation of Transport Layer Security (TLS) and several other protocols to demonstrate the applicability of trust negotiation in existing systems. But the main drawback with TrustBuilder1 was it supports only one credential format, one policy language and one trust negotiation strategy.

Later on, TrustBuilder2 was introduced; it is a fully-configurable and extensible framework for developing and experimenting with trust negotiation protocols and system components. The functionality of the TrustBuilder2 can be easily extended by developing various plug-ins that can be loaded by the TrustBuilder2 runtime system and be used in place of, or in addition to, the system components provided with TrustBuilder2**.**

III. Mechanism for Minimum Credential Disclosure with Preserved Privacy

In most of the existing ATN system, the credentials are stored independently; to get resources we need to exhibit more than one credential which is simply time consuming, it will be effective if we are able to integrate all the available credential of an individual into a single credential. This is an approach for minimizing the disclosed number of credential.

The next highlight of our work is preserving privacy; we give two approaches for preserving privacy of a credential. The first approach is associating the policy within the credential which ensures its disclosure to valid peer. The second approach is redefining the IOM anipulation module of TrustBuilder2 framework. The redefinition of module assures limited disclosure of attributes and its corresponding value of a credential. But never means it won't be able to satisfy the requirements mentioned in a policy. Only those contents of the credential which are required for satisfying the policy are disclosed and the remaining content is hidden. These are an approach through which we can preserve the privacy. We will be using both i.e. assigning a policy to credential whose contents needs to be preserved and when the credential are used for ATN, only the limited viewing of credential contents is allowed.

To verify it we will use Jess (Java Expert System Shell) [11] for defining credentials and policies.

IV. Clouseau Compliance Checker

As we are using TrustBuilder2 framework for verification, which uses CLOUSEAU compliance checker to find all satisfying sets of credentials for a given policy. The CLOUSEAU compliance checker is able to efficiently find all satisfying sets for a given policy because it reformulates this problem as an instance of the many pattern/many object pattern match problem. That is, if we treat policies as patterns and credentials as objects, the problem of findings all satisfying sets for a given policy reduces to finding all ways that our "objects " can match the specified "patterns." The CLOUSEAU compliance checker was built on top of Jess.

A. Representing Objects

CLOUSEAU represents policies as Jess rules which place the constraints on the credentials and credential chains needed to access various resources. The file jess_defs.clp included in the config/jess directory of the Trustbuilder2 distribution specifies these formats. The few portion of the jess_defs.clp is shown below in the table.

```
; Simple Credential Representation
(deftemplate credential
  " Representation of credential fields"
  (slot id)
  (slot issuer)
  (slot subject)
  (slot fingerprint)
  (slot owned (default false))
  (slot map (default (new java.util.HashMap))))

;; Credential chain representation
   (deftemplate credential-chain
  " Simple credential chain representation"
  (multislot credentials))
```

Fig. 1. Lines of jess_defs.clp showing credential and credential chain representation in Jess.

As shown in Fig. 1 the credential consists of 6 fields, as per the requirements number of fields can be increased or decreased. It consists of unique identifier, an issuer name, a finger print, a Boolean indicating the proof-of-ownership which is by default set to false, and a data structure mapping the other field names used by the credential to their values. We aren't considering finger print field of a credential description in our ex.

B. Defining Policies

After the credential has been defined the next step is to define the policies to control the access of resources and credentials. In CLOUSEAU, policies take the form of one or more Jess rules. The left hand side of each rule places constraints on the credentials and claims required to satisfy the policy. The right hand side can either assert some intermediate result that can be used by other rules making up the policy, or can assert that the policy has been satisfied. The ex. of policy is shown in Fig. 6. The Jess definition for policy satisfaction is defined by a "satisfaction" object type in jess_defs.clp is shown in Fig. 2.

```
(deftemplate satisfaction
 "Holds information about policies that get satisfied"
  (slot resource-name)
  (multislot claims)
  (multislot credentials))
```

Fig. 2. Lines of jess_defs.clp showing Jess definition for policy satisfaction in Jess.

A satisfaction object holds a name describing the

resource, whose access policy is satisfied, a collection of claims and credentials.

V. Integration of Newly Defined Credential in TrustBuiler2 Framework with Privacy Preserving

In this section the integration of newly defined credential and policies into the TrustBuilder2 framework will be described and we compare its result with the previous non integrated credential representation. We will begin this section by describing the scenario in which the trust establishment has to be made:

A. The Scenerio

Our scenario was designed to mimic a trust negotiation scenario that might take place in one branch (Acme Springfield) of a national-scale corporation (Acme Fabrication). In this scenario, an employee wants to access a file server containing sensitive files related to "Project_x." The policy protecting the Project_x file repository states that an authorized entity must be a full-time employee of Acme Spring field that has a sensitive document training certification, works in department 2400 – 2499 and was granted an "access exception" for Project_x by either Alice or Bob. This policy is thought to be a reasonable example of a negotiation that one might see in a large corporation as it is much simpler than managing a long access control list, but also includes provisions for the explicit white-listing of people who are not authorized by the blanket policy. Furthermore, entities on the white list can easily be traced back to the employee authorizing them.

The client in our scenario has a valid employee ID stating that he is a fulltime employee in department 2442 of Acme Spring field, a sensitive documents training credential, and an access exception issued by Bob.

In the above case to establish a trust user should possess three different credentials, first credential showing that s/he is a full time employee working in department 2442, second credential showing that s/he has got sensitive document training, and third credential showing that s/he has got an access exception issued by Bob or Alice. In this example instead of using three credentials we integrate these three credentials into one. The access to this credential is preserved by assigning policy to it. The policy incorporated within credential defines that for exhibiting a credential the server must prove, it is operated by Acme Springfield as shown in Fig. 8.

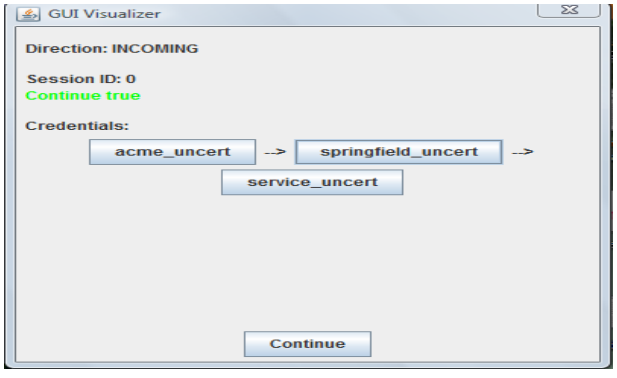

Fig. 3. Snapshot of credential chain satisfying policy of Fig. 8.

; 1) Credential of Sensitive Document Training
loaderClass = edu.uiuc.cs.TrustBuilder2.test.UncertifiedCredentialLoader
issuer = O=Acme Fabrication,C=USA
subject = CN=Charlie,O=Acme Springfield,C=USA
attr1 = Type
value1 = Sensitive Document Training
rid = docs_uncert
pid = policy_springfield_service

;2) Credential of Employee
loaderClass = edu.uiuc.cs.TrustBuilder2.test.UncertifiedCredentialLoader
issuer = O=Acme Springfield,C=USA
subject = CN=Charlie,O=Acme Springfield,C=USA
attr1 = Type
value1 = Employee
attr2 = EmpType
value2 = FTE
attr3 = Org
value3 = 2442
rid = emp_uncert

; Credential of Access Exception issued by Bob
loaderClass = edu.uiuc.cs.TrustBuilder2.test.UncertifiedCredentialLoader
issuer = CN=Bob,O=Acme Springfield,C=USA
subject = CN=Charlie,O=Acme Springfield,C=USA
attr1 = Type
value1 = Exception
attr2 = Project
value2 = Project X
rid = exn_uncert
pid = policy_springfield_service

Fig. 4. Original three different credentials of charlie

```
loaderClass =
edu.uiuc.cs.TrustBuilder2.test.UncertifiedCredentialLoader
issuer = O=Acme Springfield,C=USA
subject = CN=Charlie,O=Acme Springfield,C=USA
attr1 = Type
value1 = Employee
attr2 = EmpType
value2 = FTE
attr3 = Org
value3 = 2442
attr4= access Exception
value4= Alice
attr5= project
value5= Project X
attr6 = training
value6 = Sensitive Document Training
rid = Charlie_integrated_uncert
pid = policy_springfield_service
```

Fig. 5. Integrated credential of Charlie

```
(defrule fte-docs-24xx
(employeeId (empType "FTE") (org ?org) (Training ?training)
(access Exception ?excaption) (chain $?c1))
(test (and (<= 2400 ?org) (> 2500 ?org)))
(test (eq "Sensitive Document Training" ?training))
(or (test (eq "Bob" ?excaption))
    (test (eq "Alice" ?excaption)))
=>
 (assert (satisfaction (resource-name project_x) (credentials
(?c1)))))
```

Fig. 6. A policy defined for disclosing project_x.

Fig. 7 illustrates the example negotiation scenario graphically. The first two messages exchanged during the negotiation contain configuration information used by TrustBuilder2 to establish the parameters for the negotiation session. The second message sent by the client indicates his interest in accessing the file server associated with Project_x. The second message sent by the file server releases the policy protecting this file server to the client. The client can satisfy this policy, but is not willing to disclose his integrated credential having security clearance or access exception unless the server can prove that it is operated by

Acme Spring field. As such, the third message sent by the client discloses the release policy protecting the integrated credentials which contains this information. Server then discloses the credential chain that identifies the file service as operated by Acme Spring field. The client verifies this credential chain and the proof-of-ownership associated with the leaf credential in the chain and then discloses his integrated credential which consists of Access Exception issued by Bob, Sensitive Document Training and his employee ID to the file server. The file server verifies the proof-of -ownership associated with the credentials and then grants the client access to the service.

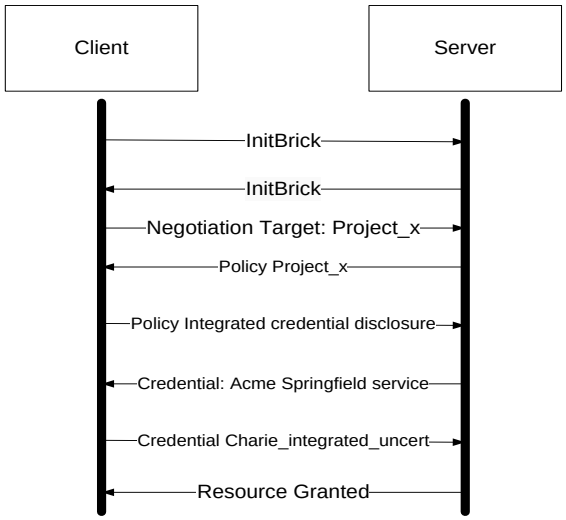

Fig. 7. A simplified view of trust negotiation for our example

```
(defrule rule-springfield-service
(credential (id ?i1) (issuer "O=Acme Fabrication,C=USA")
(subject "O=Acme Fabrication,C=USA"))

(credential (id ?i2) (map ?m2)
(issuer "O=Acme Fabrication,C=USA")
(subject "O=Acme Springfield,C=USA"))
(test (eq "Branch Office" (?m2 get "Type")))

(credential (id ?i3) (map ?m3) (subject ?s3)
(issuer "O=Acme Springfield,C=USA"))
(test (eq "Service" (?m3 get "Type")))

(credential-chain (credentials $?c))
(test (is-root ?i1 ?c))
(test (is-nth ?i2 2 ?c))
 (test (is-leaf ?i3 ?c))
=>
 (assert (satisfaction (resource-name creds) (credentials ?c))))
```

Fig. 8. Policy defined for disclosing integrated credential

While integrating the credentials we assume that as Charlie works in Acme Springfield then this company will held the responsibility of credential integration and after credential integration it will be referred as an issuer. Charlie after each time obtaining the new credentials will go to the Acme Springfield for integration. The Fig. 4 shows the credentials of Charlie before they were integrated and after the credential is been integrated by Acme Springfield. Fig. 5 shows the integrated version of Charlie's Credential.

After the credential is integrated policy is written accordingly to safeguard the credential. The new defined policy for credential disclosure is shown in Fig. 8. We are giving two approaches for preserving privacy during ATN. The first approach is associating policy within the credential Fig. 8 shows policy for preserving integrated credential. The peer must be able to satisfy the condition defined in the policy before credential is disclosed. The second approach is redefining IOM anipulation module of TrustBuilder2 framework. Presently, during ATN the framework discloses the entire content of credential, which may result information misuse. To overcome this problem we have disabled credential visibility feature by redefining the module. This ensures limited disclosure of credential content. When a credential is used during trust negotiation, only those contents which are essential for satisfying trust are disclosed the remaining contents are hidden. These two approaches provide enough means for preserving privacy.

VI. Experiment

We used the trust negotiation scenario of section V.A for conducting an experiment. In experiment, a client application made a TCP connection to a server application and carried out the trust negotiation described by Fig. 7 using an Object Output Stream to write Trust Messages to the remote server and an Object Input Stream to read response Trust Messages. When the negotiation succeeded, the client will automatically disconnect from the server. This entire process was repeated 100 times. The client and server applications were both executed from the system command prompt using JDK1.6.0, Jess 7.1p2, Bouncy castle crypto API supporting JDK 1.5 or higher and TrustBuilder2 framework. This experiment was designed to study the average execution time of a trust negotiation session and space occupied by credential along with preserved privacy.

In our experiments, the TrustBuilder2 objects used by the client and server processes supports only the use of X.509 credentials encoded as X509 Credential Brick objects. All X.509 credentials used during this scenario has encoded RSA key pairs. Further, each credential was represented as a unique X.509 certificate with its own key pair. Both the client and server processes support the use of the CLOUSEAU compliance checker. The strategy used by both parties was the variant of the TrustBuilder1 relevant strategy that is implemented by the Maximum Relevant Strategy class included in the TrustBuilder2 distribution. Credential chains were built using the Simple Chain Builder class and verified using the Root to Leaf verifier class. The IOM anipulation Module are enabled at both the client and server. The experiments described above were run using a single machine, as we were more interested in the computational costs of the trust negotiation. The machine that we used had Intel Pentium(R) D 2.8GHz processor, 1.5 GB RAM, and was running Microsoft Windows XP Service Pack 2.

VII. Result

We have conducted the experiment in above mentioned environment, using both independent credentials and integrated credential. The result of the experiment is shown in table I. The result shows that after integrating credentials the average execution time for aforementioned trust negotiation session is decreased by 6% and the space

requirement in disk was reduced by 30.5%.

TABLE I: RESULT OF EXPERIMENT

Credential Type	Average Execution Time (ms)	Occupied Space in disk (Bytes)
Independent Credential	4510	787
Integrated Credential	4239.2	547

After redefining the IOM anipulation module of TrustBuilder2 framework the limited disclosure of credential content is achieved. Only those attributes/value is disclosed which are required for satisfying the policy. The remaining fields were hidden which ensures credential content security. In the above ex. out of eight fields only the six fields of a integrated credential is disclosed which are issuer, subject, attribute emptype and its value, attribute org and its value, attribute training and its value, attribute access exception and its values in order to satisfy the policy defined in Fig. 8 for receiving the file Project_x.

VIII. CONCLUSION

The In this paper we have used TrustBuilder2 open framework to verify our findings, when we integrate the credentials, average time required to conduct the trust negotiation is faster and space required for credentials storage are lesser in compare to using separate credential. In this case we are integrating only three credentials, and we feel the difference. In the real life scenario where there are number of credential of same individual if we are able to integrate and use it, the time required to complete the trust negotiation will be faster and space required for credentials storage will be much lesser through which we can enhance the performance of ATN. Along with it extra burden of managing separate credentials can also be removed. In another finding by associating policy and redefining module only those attributes/ value pairs will be disclosed which are required to establish trust, the remaining will be hidden through which we can preserve the privacy of an individuals and organizations credentials.

ACKNOWLEDGMENT

Dhiraj Shrestha thanks Chinese Scholarship Council, College of Computer Science and Technology of Harbin Engineering University for providing all the necessary support without which this work wouldn't have been realized. He also thanks his parents and his lab friends for all their direct and indirect support. Li Jianli thanks College of Computer Science and Technology of Harbin Engineering University for providing the research environment to accomplish this work.

REFERENCES

[1] E. Bina, V. Jones, R. McCool, and M. Winslett, "Secure access to data over the internet," *Proceedings of the Third ACM/IEEE International Conference on Parallel and Distributed Information Systems*, Austin, Texas, Sept., 1994

[2] M. Winslett, T. Yu, K. E. Seamons, *et al.*, "Negotiating trust on the web," *IEEE Internet Computing*, vol. 6, no. 6, pp. 30-37, 2002.

[3] W. H. Winsborough and N. Li. "Protecting sensitive attributes in automated trust negotiation," *Anon. ACM Workshop on Privacy in the Electronic Society, Washington, DC, United States: Association for Computing Machinery*, pp. 41-51, 2002.

[4] W. H. Winsborough, K. E. Seamons, and V. E. Jones, "Automated trust negotiation," *Anon. DISCEX'00. Las Alamitos, CA, USA: IEEE Comput. Soc*, pp. 88-102, 1999.

[5] A. J. Lee, M. Winslett, and K. J. Perano, *TrustBuilder2: A Reconfigurable Framework for Trust Negotiation*.

[6] A. J. Lee. *TrustBuilder2 User Manual version 0.1*, May 21, 2007.

[7] A. J. Lee and K. J. Perano, *The TrustBuilder2 Framework for Trust Negotiation.*

[8] R. Jarvis, "Protecting sensitive credential content during trust negotiation," *M. Sc Thesis*, Brigham Young University. 2003.

[9] E. Bertino, E. Ferrari, and A. C. Squicciarini, "Privacy-preserving trust negotiations," *Anon. Revised Selected Papers*. Berlin, Germany: Springer-Verlag, pp. 283-301, 2004.

[10] A. Hertzberg, Y. Mass, J. Michaeli, D. Naor, and Y. Ravid, "Access control meets public key infrastructure, or: Assigning roles to strangers," *IEEE Symposium on Security and Privacy*, May 2000.

[11] *Jess the Rule Engine for the Java Platform*, Version 7.1 p2, Ernest Friedman-Hill Sandia National Lab, November 5, 2008.

Dhiraj Shrestha received the B.E in Computer Engineering from Pokhara University, Nepal, in 2004. He received M.E in computer Applied Technology from Harbin Engineering University, China, in 2011. Currently, he is working as a faculty in Department of Computer Science and Engineering at Kathmandu University. His research interests include Information Security, Automated trust negotiation, and E-health.

Li Jianli is an Assistant professor in the College of Computer Science and Technology, Harbin Engineering University, Harbin, China. He has graduated in 1985 from South China University of Technology. His research interests include information security standards and technologies, P2P trust model technology, automated trust negotiation.