Model for Token Based Secure Transaction in ATM Networks.

Sonika Katta, Dinesh Goyal, Ruchi Dave,Naveen Hemrajani
Dept. of CSE, Suresh Gyan Vihar University, Jaipur, Rajasthan, India.

***Abstract*--Security is a key issue in Data Warehouses. Data warehousing poses its own set of challenges for security. One major challenge is that enterprise data warehouses are often very large systems, serving many user communities with varying security needs. Thus, the data warehouses require a flexible and powerful security infrastructure. The Security applications, must prevent unauthorized users from accessing or modifying data; the applications and underlying data must not be susceptible to data theft by hackers; the data must be available to the right users at the right time; and the system must keep a record of activities performed by its users. A successful authentication takes place if a user can prove to a server that he/she knows the shared secret without actually transmitting that secret across the wire. Here we use the Token-based authentication for access the data in Data Warehouse Server. Token Based authentication is a security technique that authenticates users who are attempting to log in to a server, a network, or some other secure system.**

Keywords: Data warehouse, security, Authentication, Token-Based authentication.

I. INTRODUCTION

E-commerce provides the capability of buying and selling products, services and information on the Internet by using electronic payment systems. [1] In electronic payment systems the exchange of value is represented by the exchange of data. It is easy, cheap and fast to transfer data, but the main challenge is security. [2][3] Contemporary electronic payment systems may be classified into two groups: .account-based, credit-debit systems, and token-based, electronic currency systems. Both groups have important characteristics such as trust, security, reliability, easy of use, efficiency, flexibility, convertibility, interoperability, etc. In this paper we focus on the security of the system.

Credit card transactions have become a de facto standard for Internet and Web based payments. There is millions of credit card transactions processed each day. The system is distributed, heterogeneous and hierarchical. The main participants are: Issuer, Acquirer, Payment Gateway, Merchant, Certification Authority, and Cardholder or Buyer. Payment devices such as ATM (Automatic Teller Machine), POS (Point Of Sale) terminal, and kiosk may be located at merchant site. Communication between an acquirer and an issuer is based on ISO 8583 standard for bank card originated messages. Enterprises are seeking ways to simplify and reduce the scope of the Payment Card Industry's Data Security Standard (PCI DSS) compliance by shrinking the footprint where cardholder data is located throughout their organization. Compliance with the PCI DSS is a combination of documented best practices and technology solutions that protect cardholder data across the enterprise. This paper explores the use of tokenization as a best practice in improving the security of credit card transactions in the ATM's, while at the same time minimizing the cost and complexity of PCI DSS compliance by reducing audit scope. The scope of PCI DSS compliance for any organization is significant both in terms of effort and cost. In a PCI DSS audit, all systems, applications and processes that have access to credit card information, whether encrypted or unencrypted, are considered in scope. ATM's uses the Token Based Authentication for secure transaction.

II. AUTHENTICATOR BACKGROUND

This section provides an introduction to authenticators and related security matters.

A. Authenticator Definitions We use the term *password* to include single words, phrases, and personal identification numbers (PINs) that are closely kept secrets used for authentication. There are many studies showing the vulnerabilities of password-based authentication schemes [4]–[5]. The basic problem with passwords can be explained succinctly: a memorable password can often be guessed or searched by an attacker and a long, random, changing password is difficult to remember. [6] An *identity token*, *security token*, *access token*, or simply *token*, is a physical device that performs or aids authentication. This can be a secure storage device containing passwords, such as a bankcard, remote garage door opener, or smart card. This can also be an active device that yields *one-time passcodes*, either *time-synchronous* (changing in synchrony with a master at the host) or *challenge–response* (responding to a one-time challenge). Token security defenses include tamper-resistant packaging and special hardware that disables the token if it is tampered with or if the number of failed authentication attempts exceeds a chosen threshold. When we refer to "token" in this paper, the general concept will be a digital token key, secure device accessed at the client end via a password to obtain a token that is transmitted to the host's mobile phone for authentication. A *token is* a secret key like a password, except it is machine-generated or machine-stored, so it can be longer, more random, and perhaps changing. Token key is a combination of the alphanumeric elements. A biometric purports to inextricably link the authenticator to its

owner, something passwords and tokens cannot do, since they can be lent or stolen. When used to verify the person involved in a transaction, an inextricable link can offer the property of *non-repudiation*. This property provides proof of a transaction such that the involved parties cannot subsequently reject the transaction as unauthorized or claim not to have participated in the transaction.

B. Security Definitions

Security systems and methods are often described as *strong* or *weak*. [2] A door with a lock offers stronger security than one with no lock. A credit card number alone offers "weak" defense against repudiation because a user can easily deny a credit card charge by claiming that his credit card number was stolen. However, a credit card number *plus* a signature has a "strong" defense (meaning "stronger" defense than without a signature) because the user leaves evidence of his presence by his signature. [3] It is more difficult to measure security in absolute terms. One way to measure absolute strength and weakness of security systems is as follows. A strong system is one in which the cost of attack is greater than the potential gain to the attacker.

C. Types of Authenticators

Authentication factors are usually grouped into three categories: 1) What you know (e.g., password); 2) what you have (e.g., token); and 3) who you are (e.g., biometric). This is a good mnemonic scheme and unlikely to fall from use, but it is not without problems. [7][8] For instance, a password is not strictly *known*: it is memorized. Therefore, it can be forgotten, either in the short term or over a longer period. Biometrics are definitely not "who you are" any more than hair color or body build indicates your true self. A biometric is simply one feature of your appearance. [9] We prefer the following authenticator labels: knowledge-based, object-based, and ID-based.

1) **Knowledge-Based Authenticators** ("what you know")—characterized by secrecy or obscurity. This type includes the memorized password. [10] It can also include information that is not so much secret as it is "obscure," which can be loosely defined as "secret from most people." Mother's maiden name and your favorite color are examples in this category. Security drawback of secrets is that, each time it is shared for authentication, it becomes less secret. [11]

2) **Object-Based Authenticators** ("what you have")—characterized by physical possession. Physical keys—which we call metal keys to distinguish them from cryptographic keys—are tokens that have stood the test of time well. [12] A security drawback of a metal house key is that, if lost, it enables its finder to enter the house. This is why many digital tokens combine another factor, an associated password, to protect a lost or stolen token. There is a distinct advantage of a physical object used as an authenticator; if lost, the owner sees evidence of this and can act accordingly.

3) **ID-Based Authenticators** ("who you are")—characterized by uniqueness1 to one person. A driver's license, passport, credit card, university diploma, etc., all belong in this category. So does a biometric, such as a fingerprint, eye scan, voiceprint, or signature. [13][14] For both ID documents and biometrics, the dominant security defense is that they are difficult to copy or forge.

III. RELATED WORK

A **security token** (or sometimes a *hardware token*, *hard token*, *authentication token*, *USB token*, *cryptographic token*, or *key fob*) may be a physical device that an authorized user of computer services is given to ease authentication. The term may also refer to software tokens. Security tokens are used to prove one's identity electronically (as in the case of a customer trying to access their bank account). The token is used in addition to or in place of a password to prove that the customer is who they claim to be. The token acts like an electronic key to access something. Several systems have been designed where a physical object is used to access digital information that is stored outside the object. We introduce a schema with three types of physical objects that can be linked to digital information:

- *Containers* are generic objects used to move information between different devices or platforms;
- *Tokens* are used to access stored information, the nature of which is physically reflected in the token in some way; and
- *Tools* are used to manipulate digital information.

ATM's must meet the requirements of PCI DSS (Payment card data decision support system) are increasingly embracing the compliance benefits of tokenization. By using tokenization, businesses are reducing the number of locations where they are retaining cardholder information. Requirements mandate that access to keys be restricted to the fewest number of custodians and that keys be stored securely in the fewest possible locations. With the help of tokens, ATM's get the high level security. With tokenization, encryption is performed centrally when credit card values are tokenized, and keys are centralized on a secure server, optimally addressing these requirements.

1. *Centralized data vault*

Under the tokenization model, encrypted payment card data is stored only in a central data vault and tokens replace the credit card values in applications or databases. Risk is reduced since credit cards, even in their encrypted state, are not available outside of the data vault, except when originally captured at the beginning of a transaction or, later, accessed from the data vault by authorized and authenticated applications and/or users. When a credit card is used for a purchase, the number is transmitted in real-time to the token server. A token representing that data is generated and returned to the calling application and takes the place of the credit card number in that application and all downstream uses of the data (for example, CRM systems, backup copies

of the application, exports to data warehouses and exports to fraud detection applications). Meanwhile, the credit card number is encrypted and stored in the central data vault. The clear text value is not stored anywhere. If the need to access the actual value arises (for instance, to process a merchandise return), the value can be decrypted in response to a request by those users or applications with proper authority, and then only for as long as it is needed to complete a transaction. This "round trip" protection limits the risk to the credit card data and ensures that it is an ever stored in the clear, and that the encrypted cipher-text version of the card data only resides within the secure data vault and in its corresponding back-up/disaster recover instances. The data vault is never exposed to any applications other than the token server.

2. *Tokens act as data surrogates*

With traditional encryption, when an application or database needs to store credit card data, the values are encrypted and cipher text is saved in the original location. With tokenization, a token – or surrogate value – is stored in place of the original data. The token is a reference to the actual cipher text, which is stored in a central data vault. Encrypted data usually takes up more space than the original values, which requires changes to the applications and/or databases. Tokens can be engineered to preserve the length and format of the original data, so they are almost completely non-invasive to databases and applications; requiring no changes to database schemas, application screens, business processes, etc. This significantly reduces the IT and business impact associated with PCI DSS compliance.

3. *Tokens are surrogates for masked data*

Tokens can be generated to preserve parts of the original data values. A typical pattern in PCI DSS scenarios is generating the tokens to maintain the original first two and last four digits of the credit card. A "Token Strategy" provides the flexibility to define the format of the tokens including what part, if any, of the original value to preserve. Frequently there are applications within the enterprise that need only the last four digits of a credit card to validate credentials. It's this scenario where the use of a format-preserving token allows users to perform their job, validating the last four digits of the credit card. Since the application does not contain credit card information - not even in an encrypted format - the entire application is removed from PCI DSS scope. A word of caution - proper network segmentation is still required. [1]

4. *One-to-one token/data relationship*

Certain types of tokenization can ensure that there is always a one-to-one relationship between the credit card number and the token that is generated so that you maintain referential integrity across multiple systems. [1] Maintaining referential integrity allows the data warehouse to perform transaction analysis with tokens, rather than credit card numbers, thus removing it from scope. Once again, network segmentation is also important, but done properly, the scope of the PCI DSS audit is reduced. And the consequences of a breach of the data warehouse have been minimized because any unauthorized access to the data warehouse only yields tokens and not actual credit card numbers.

5. *No relationship between tokens and data values*

With tokenization there is no mathematical relationship between a token and data value – the only relationship is referential. This is not the case when using encryption or hashing where an algorithm mathematically derives the output cipher text or hash based on the input data. Tokens can be passed around the network between applications, databases and business processes safely, all the while leaving the encrypted data it represents securely stored in a central data vault. Authorized applications that need access to encrypted data can only retrieve it from the tokenization engine, providing an extra layer of protection for credit card data and shrinking the "risk footprint" that needs to be managed and monitored by your security team. [1]

6. *Centralized key management*

With the use of tokenization, keys are limited to use by the centralized token manager, keeping the distribution of the keys to a minimum. This helps to minimize the scope of PCI DSS compliance and reduce the risk of a key compromise. [1]

IV. PROPOSED WORK

In this paper, we propose a model of a secure access of data from a data warehouse server. In this dissertation we use Token based Security Access Control for the ATM transaction. Initially, the financial service outlet and the user each have their own digital certificates. Token based Authentication is necessary to establish a binding between the identity and the public key of the remote entity. For each entity, the entity's identity, the public key, their binding, validity conditions and other Attributes are made unforgivable in digital certificates issued by the common root CA. We classify the procedure outline here in the following steps.

1. Mutual authentication and token generation.
2. Sending token on another network
3. Using token key for the transaction request
4. Establish Connection for Transaction
5. Transaction approval acknowledgement
6. Transaction done and token destroy

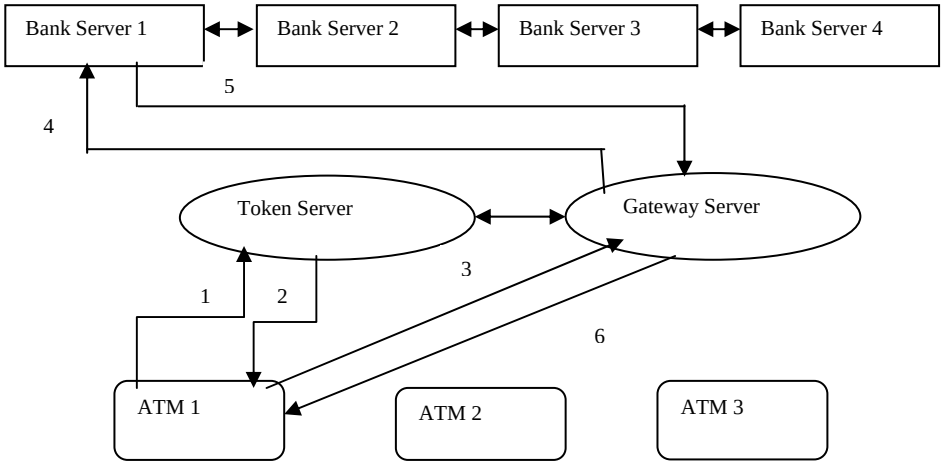

Figure 1 Mutual authentication and token generation. 2. Sending token on another network.
3. Using token key for the transaction request. 4. Establish Connection for Transaction.
5. Transaction approval acknowledgement. 6. Transaction done and token destroy

V. CONCLUSION

In this work, we have proposed a new transaction model for financial services which addresses the various concerns to the existing transaction model. As a result of the preferred secured on-line transaction and secret key distribution, the authentication tokens may be used for different purpose at different locations on a variety of systems. We also improved application level security system and method which offer secure on-line distribution of shared secret keys at the time of registration and also dynamic configuration of user entitlements, and which utilizes the shared secret keys to provide mutual authentication and generation of session keys to secure subsequent communication between holders of the shared secret keys. So, we propose a token based authentication for the ATM's transaction.

VI. FUTURE WORK

We discussed the most important scenario where the server public key is itself distributed to the user in a secure manner, by transmitting it to the user at the time of registration in the form of a certificate signed by the token issuer or a certification authority and verifiable by a public key embedded in the token prior to distribution. Since the certificate is signed using a private key known only to the token issuer or token certifier, the client software can ensure that the server public key has been authorized by the token issuer or we can say token server. In future this token based access model can be implementing in ATMs protocol layers.

REFERENCES:

[1] SANS Institute Infosec Reading Room, www.nubridges.com, 2009 nuBridges, Inc.

[2] D. Kahn, *The Codebreakers: The Story of Secret Writing*. New York: Scribner, 1996.

[3] G. Stocksdale. *NSA Glossary of Terms Used in Security and Intrusion Detection*

[4] R. Morris and K. Thompson, "Password security: A case history," *Commun. ACM*, vol. 22, no. 11, pp. 594–597, Nov. 1979.

[5] R. Pond, J. Podd, J. Bunnell, and R. Henderson, "Word association computer passwords: The effect of formulation techniques on recall and guessing rates,"

[6] K. P. Weiss, "Method and apparatus for positively identifying an individual," U.S. Patent 4 720 860, Jan. 19, 1988.

[7] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM Syst. J.*, vol. 40, no. 3, pp. 614–634, 2001.

[8] S. M. Bellovin and M. Merritt, "Encrypted key exchange: Passwordbased protocols secure against dictionary attacks," in *Proc. 1992 IEEE Computer Society Conf. Research Security and Privacy*, 1992, pp. 72–84.

[9] B. L. Riddle, M. S. Miron, and J. A. Semo, "Passwords in use in a university timesharing environment," *Comput. Security*, vol. 8, no. 7, pp. 569–579, 1989.

[10] D. L. Jobusch and A. E. Oldehoeft, "A survey of password mechanisms: Weaknesses and potential improvements," *Comput. Security*, vol. 8, no. 8, pp. 675–689, 1989.

[11] M. Bishop and D.V. Klein, "Improving system security via proactive password checking," *Comput. Security*, vol. 14, no. 3, pp. 233–249, 1995.*Society*. Dordrecht, The Netherlands: Kluwer, Nov. 1998.

[12] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1999, p. 490.

[13] T. Wu, "The secure remote password protocol," in *Proc. 1998 Internet Society Network and Distributed System Security Symp.*, pp. 97–111.

[14] Common Criteria for Information Technology Security Evaluation, Part 2, Security Functional Requirements (1999, Aug.).