**T.C.**

**ANTALYA BILIM UNIVERSITY**

**INSTITUTE OF POSTGRADE EDUCATION**

**CYBER SECURITY**

**MASTER'S THESIS**

# SECURITY OF AUTOMATED TELLER MACHINE (ATM)

**Shaikh Muhammad ABBAS**

**DECEMBER 2022**

**ANTALYA**

**T.C.**

**ANTALYA BILIM UNIVERSITY**

**INSTITUTE OF POSTGRADE EDUCATION**

**CYBER SECURITY**

**MASTER'S THESIS**

# SECURITY OF AUTOMATED TELLER MACHINE (ATM)

**Shaikh Muhammad ABBAS**

**DECEMBER 2022**

**ANTALYA**

**T.C.**

**ANTALYA BILIM UNIVERSITY**

**INSTITUTE OF POSTGRADE EDUCATION**

**SECURITY OF AUTOMATED TELLER MACHINE (ATM)**
**Shaikh Muhammad ABBAS**

This thesis was accepted by the jury (with unanimous vote/majority vote) on the date 17/01/2023 in CYBER SECURITY of CYBER SECURITY DEPARTMENT.

Asst Prof. Asli BAY (Supervisor)

Dr. Neşe KOÇAK

Prof Dr. Cafer ÇALIŞKAN

Director of the Institute

Prof. Dr. İbrahim Sani MERT

Thesis Submission Date: / /202

## DECLARATION

MS thesis of this study named “SECURITY OF AUTOMATED TELLER MACHINE (ATM)”, which I presented, I declare that scientific moral principles were followed in the preparation of this study, in case of benefiting from the works of others, reference is made by scientific norms, no falsification has been made in the data used, and that any part of this study is not presented as another academic study.

30/12/ 2022

Shaikh Muhammad ABBAS

**CONTENTS**

## ABSTRACT

## SECURITY OF AUTOMATED TELLER MACHINE (ATM)

**Shaikh Muhammad ABBAS**

**MS Thesis in CYBER SECURITY**

**Supervisor: Ass. Prof. Dr. Asli BAY**

**December 2022; 57 pages**

Automated Teller Machines, ATMs are known to be the first well-known machines that allow customers to access their bank accounts via electronic means. Formerly, the ATM only works as cash dispensers i.e. to deliver money in the form of banknotes to customers, and to debit the customer's bank account. Gradually, consumers' dependency on ATMs increases, and their trust builds up in ATMs for fulfilling their banking needs. ATMs are mainly operated by plastic cards with special features. It is quite easy to instantly withdraw money from ATMs at any time throughout the world. One can carry out a variety of financial operations via an ATM, including cash withdrawals, checking account balances, transferring money between accounts, paying insurance premiums, taking out small loans, and paying bills.

An ATM executes this vast array of activities via plastic, magnetic-stripe cards, and PINs. An ATM is dependent on an API to fulfill the banking needs of consumers. API aids an ATM to communicate with the bank servers and carrying out transactions. API handles every task associated with completing an ATM transaction from card verification, PIN generation, PIN verification, and cash withdrawals to balance inquiry. The use of APIs in ATMs emerges with benefits as well as challenges. However, certain security measures must be considered when using an API in ATM.

There was a time when thieves exclusively targeted ATMs for the cash they contained. Modern ATMs, also include consumer data, which is just as important as cash.

Although, there is no client data kept on an ATM. However, it does send and gather user data. Owners of ATMs are now faced with the difficulty of securing their devices against various forms of attacks. In this thesis, such ATM attacks have been discussed. Additionally, preventive measures to avoid such attacks have been suggested.



**KEYWORDS:** API, ATM, Cryptography, Cyber Security.

**COMMITTEE:** Dr. Neşe KOÇAK

Prof Dr. Cafer ÇALIŞKAN

Asst Prof. Asli BAY

# ÖZET

## OTOMATİK TELLER MAKİNASI (ATM) GÜVENLİĞİ

**Shaikh Muhammad ABBAS**

**Yüksek Lisans Tezi, SİBER GÜVENLİK Anabilim Dalı**

**Danışman: Dr. Asli BAY**

**Aralık 2022; 57 sayfa**

ATM'ler, müşterilere elektronik erişim sağlayan ilk iyi bilinen makinelerdir. İlk başta, ATM'nin birincil işlevi, banknot şeklinde nakit teslim etmek ve buna karşılık gelen bir banka hesabını borçlandırmaktı. Zamanla, tüketiciler bankacılık ihtiyaçlarını rahatça karşılamak için Otomatik Para Çekme Makinesi'ne (ATM) güvenmeye başlar. ATM'ler çoğunlukla özel özelliklere sahip plastik kartlarla çalıştırılır. Dünyanın her yerinde her an ATM'lerden anında para çekmek oldukça kolaydır. ATM aracılığıyla nakit çekme, hesap bakiyelerini kontrol etme, hesaplar arası para transferi, sigorta primleri ödeme, küçük krediler alma ve fatura ödeme gibi çeşitli finansal işlemler gerçekleştirilebilir.

Bir ATM, plastik, manyetik şeritli kartlar ve PIN'ler aracılığıyla bu geniş etkinlik dizisini yürütür. Bir ATM, tüketicilerin bankacılık ihtiyaçlarını karşılamak için bir API'ye bağımlıdır. API, bir ATM'nin banka sunucularıyla iletişim kurmasına ve işlemleri gerçekleştirmesine yardımcı olur. API, kart doğrulama, PIN oluşturma, PIN doğrulama ve nakit çekme işlemlerinden bakiye sorgulamaya kadar bir ATM işleminin tamamlanmasıyla ilgili her görevi yerine getirir. API'lerin ATM'lerde kullanımı, zorlukların yanı sıra faydalarla da karşımıza çıkıyor. Ancak, ATM'de bir API kullanırken belirli güvenlik önlemlerinin alınması gerekir.

Hırsızların, içerdikleri nakit para için yalnızca ATM'leri hedef aldıkları bir zaman vardı. Modern ATM'ler, nakit para kadar önemli olan tüketici verilerini de içerir.

Bununla birlikte, bir ATM'de tutulan müşteri verileri yoktur. Ancak, kullanıcı verilerini gönderir ve toplar. ATM sahipleri artık cihazlarını çeşitli saldırı biçimlerine karşı korumanın zorluğuyla karşı karşıya. Bu tezde, bu tür ATM saldırıları ele alınmıştır. Ek olarak, bu tür saldırıları önlemek için önleyici tedbirler önerilmiştir.



**ANAHTAR KELİMELER:** API, ATM, Kriptografi, Siber Güvenlik.

**JÜRİ:** Dr. Neşe KOÇAK

Prof Dr. Cafer ÇALIŞKAN

Asst Prof. Asli BAY

## SYMBOLS AND ABBREVIATIONS

**Abbreviations**

AES : Advance Encryption Standard

API : Application Programming Interface

APT : Advanced Persistent Threat

ATM : Automated Teller Machine

AWK : Acquirer Working Key

BCC : Bank Central Control

BDK : Base Derivation Key

BIN : Bank Identification Number

BIOS : Basic Input / Output System

CA : Certification Authority

CD : Compact Disk

CVK : Card Verification Key

CVV : Card Verification Value

DES : Data Encryption Standard

DUKPT : Derived Unique Key for Each Transaction

DVD : Digital Video Disk

EPP : Encrypting PIN Pad

FSM : Finite State Machine

HSM : Hardware Security Module

HTTP : Hypertext Transfer Protocol

IBAN : International Bank Account Number

IPEK : Initial PIN Encryption Key

ISO : International Standard Organization.

KCV : Key Check Value

KEK : Key Encryption Key

KSN : Key Serial Number

LMK : Local Master Key

MAC : Message Authentication Codes

MitM : Man-in-the-Middle

NFC : Near Field Communication

OWASP : Open Web Application Security Project

PAN : Primary Account Number

PCI : Payment Card Industry

PDK : PIN Derivation Key

PED : PIN Entry Device

PIN : Personal Identification Number

POS : Point of Sales

PVK : PIN Verification Key

PVV : PIN Verification Value

RSA : Rivest Shamir Adleman

SHA : Secure Hash Algorithm

SQL : Structured Query Language

SSL : Secure Sockets Layer

TRM : Tamper Resistant Security Module

USB : Universal Serial Bus

XOR : Exclusively-OR

ZMK : Zone Master Key

ZPK : Zone PIN Key

## LIST OF TABLES

## LIST OF FIGURES

## PREFACE

The hereby master's dissertation in Cyber Security covers the working of security of APIs in ATMs and its working. In this thesis, we will discuss the working scenario of an ATM that how it works, and its security-related issues in the literature. I want to sincerely thank my advisor, Ass. Prof. Dr. Asli Bay, for her direction, unending support, and insightful observations. I'm grateful that I had the chance to collaborate directly with her, benefit from her wisdom and expertise, and explore the world of Cyber Security. My mother, father, and siblings are the most significant part of my life. Without their contributions, encouragement, and assurance I would not have been able to make it this far. I dedicate this work to them with the aspiration that I will always be able to make my family proud and content. I hope they are proud of this dissertation!

# 1. INTRODUCTION

A computerized device known as an ATM (Automated Teller Machine) enables a cardholder or client of a banking or financial institution to carry out transactions about their accounts. There are now two different types of ATMs: the first is restricted to straightforward operations like withdrawals and general account balance information. The second type is more complex and includes extra features like depositing money into one's account or transferring funds to another account. Many approaches and procedures have been applied to ATM transactions in the last ten years, but to what extent are these tactics secure?

Security is considered first when sensitive information needs to be stored and transmitted over the internet where physical boundaries can no longer secure the information. To maintain safe communication between the ATM and Bank Central Control (BCC), encryption is a crucial, effective, and efficient component. By sending incomprehensible information, the information will be protected from unauthorized access. The appropriate cryptographic algorithm selection is crucial for secure communication that offers increased security, precision, and efficiency.

ATM uses one-pin security to provide secure financial transactions at the consumer's point. Although, it is not sufficient enough to secure the data supplied to the bank server via ATM and vice versa. The bank imposes security at three different stages to safeguard the ATM. These consist of:

(a) Security at the Physical level

(b) Security at the Software level

(c) Security at the Communication level

We will talk about these securities in this thesis. A new degree of security must be implemented to ensure the security of data transmission between an ATM and the banking system, as it is becoming more and more advanced.

ATMs typically employ a single key pin to validate a user's identity, but it is not enough to safeguard the data during a network transaction. Banks employ many types of encryption techniques to protect transactions along with securing sensitive information. One of them is the encryption of information. Through employing a distinct key and similarly encrypting the plaintext into encoded text, encryption is the cryptographic procedure created specifically to transform delicate information into a form that is difficult to interpret. In this paper, we will discuss different cryptographic standards and other measures adopted by banks for providing secure financial facilities. The arrangement of this thesis is as follows:

- Chapter 2, discusses the basics of an ATM and the components associated with the functioning of an ATM i.e. the Hardware related components as well as Software related components.

- Chapter 3, provides a brief explanation of an Architectural Model of an ATM Machine and the detailed working of an ATM of how an ATM transaction is processed from PIN encryption to its validation.

- Chapter 4, highlights the cybersecurity issues regarding ATM Machines and various In Chapter 5, we have discussed some of the ways of mitigating ATM attacks and security breaches.

- Finally, in Chapter 6, we finalized our Research Thesis and concluded our findings.

# 2. BACKGROUND

## 2.1. Automated Teller Machine (ATM)

The ATM is one of the chief innovations of the banking sector that provides banking operations outside the bank premises. It is an electronic banking outlet that facilitates its users to process cash transactions including deposits of cash, cash transfers, account inquiry, and utility bills payment, without the assistance of any representative from the bank.

### 2.1.1 Definition

In essence, an Automated Teller Machine is a banking terminal with few input/ output devices. (Bowen, 2000). To operate an ATM needs to link up with a financial server. All ATMs are connected to a unified network (Ghafari et al., 2014).

### 2.1.2. Working

An ATM begins its work as soon as the user inserts a plastic card (either a bank debit or a credit card) into the ATM's Magnetic Stripe Reader. ATM processes the magnetic card and transmits the identification code i.e. Primary Account Number (PAN) which is either embossed or laser-printed on the card to the bank's central control (BCC) through the host server. The ATM card holder is identified by this number. ATM then prompts the user to enter the Personal Identification Number (PIN) to authorize the user. The terminal's PIN Entry Device (PED) encrypts the PIN to ensure security before transmitting it to the card issuer. The encryption process is administered under a shared secret among the server or switch that the ATM is attached to. The PIN is subsequently sent to an adjacent switch, at which it is first decoded, then encoded again. The PIN may have gone through several encryptions and decryption iterations before it gets to the card issuer.

### 2.1.3. Transaction Steps

A simple ATM communication between the cardholder (user) and Bank Central Control (BCC) is presented in this section. ATM transaction stages are depicted in Figure 2.1.

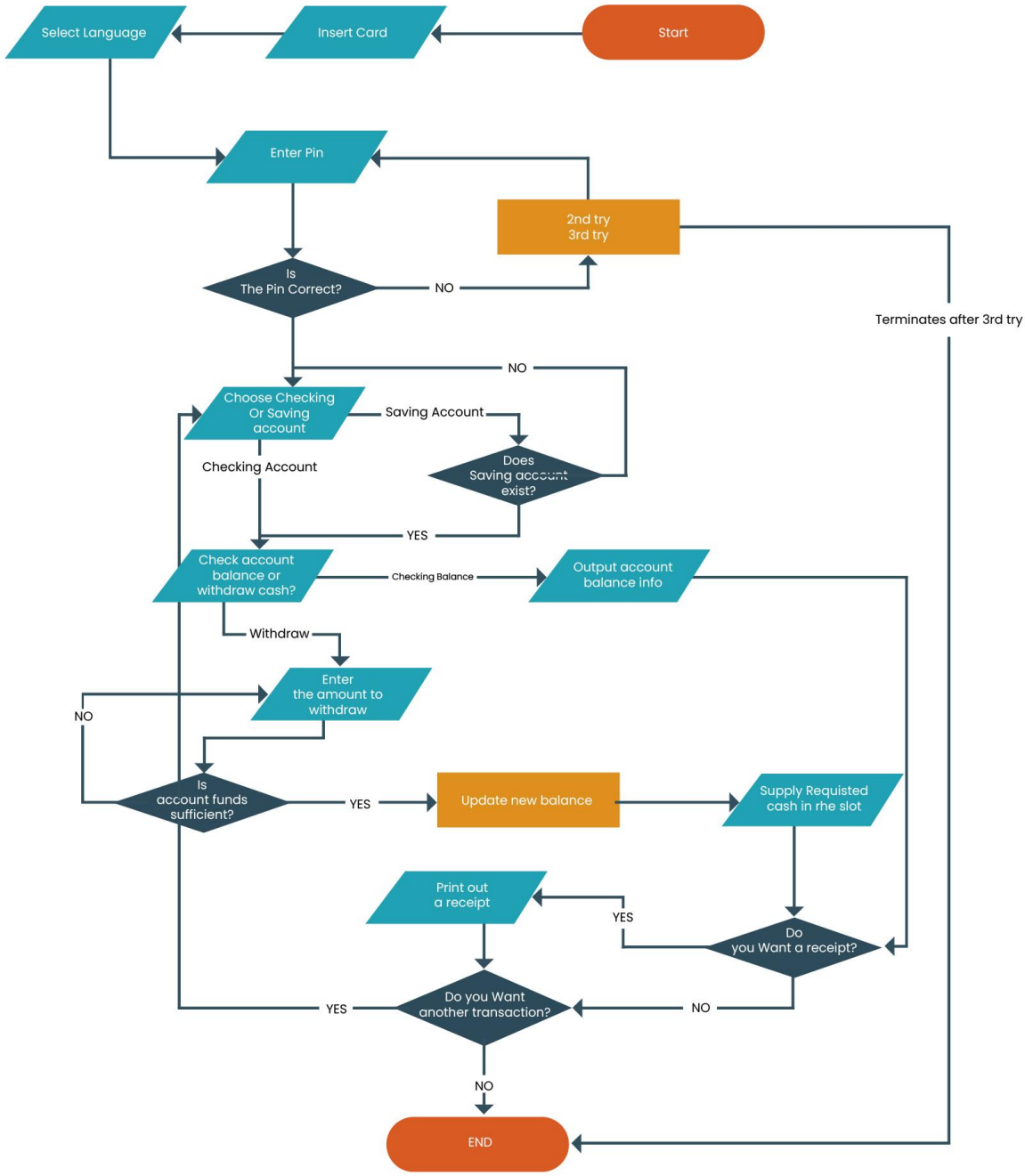


**Figure 2. 1.** ATM Transaction Steps

## 2.2. Cryptography

Cryptography is a crucial component for secure communication and the manipulation of information employing security services for secrecy, reliability, authorization, identity verification, and non-repudiation. It offers a means of sending sensitive data in an incomprehensible form, and only the authorized recipient can access this data by converting it back into the original text (Mushtaq et al., 2017).

Encryption is the process of using the key to transform plaintext into cipher text, and decryption is the process of undoing the encryption. Numerous alternative encryption

algorithms exist for converting plaintext into cipher text. The most widely used encryption algorithms include AES ((NIST), National Institute of Standards and Technology, 2001), RSA (United States Patent No. 4,405,829, 1977), DES ((NIST), National Institute of Standards and Technology, 1999), and 3DES (Barker & Mouha, 2017).

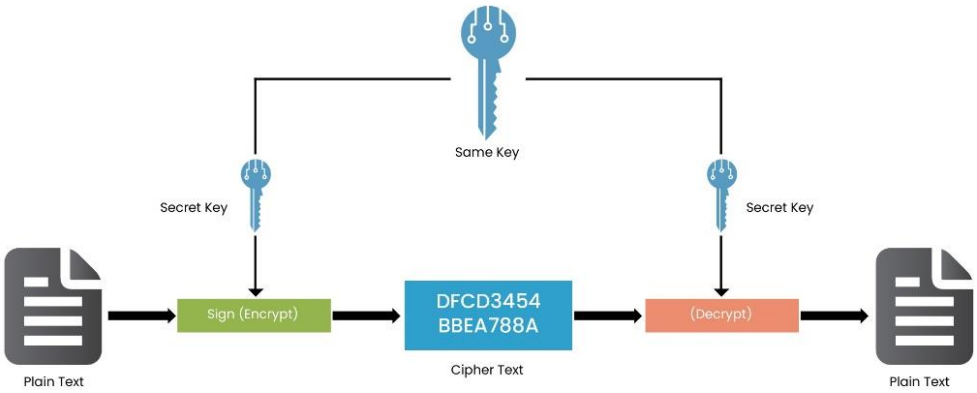


**Figure 2. 2.** Secret-key encryption

### 2.2.1. Cryptographic Security Mechanisms

The primary objectives of cryptography are five. All safety mechanisms needs to comprise a range of safety characteristics that can guarantee the system's privacy. The following five major categories fit these objectives:

- **Authentication:** the procedure of establishing one's identity. This indicates that prior to sending and receiving data through the system, the sender and receiver identities should be verified.
- **Privacy/confidentiality:** ensuring that only the intended recipient may read the message. The majority of people typically use this feature to assess a system's security. It indicates that nobody else can comprehend the message's content; only those who have been verified can.
- **Integrity:** reassuring the recipient that the message they have received hasn't been changed in any manner from the original. Packet checksums in IPv4 packets are the fundamental type of integrity.
- **Non-repudiation:** a way to confirm that the message's sender did so. This signifies that neither the sender nor the recipient can pretend that they did not transmit a particular communication.
- **Service Reliability and Availability:** Secure systems frequently experience intrusion attacks, which may have an impact on their usability and level of service to their users. Such systems offer a means of providing their consumers with the caliber of service they demand (Nema & M.A.Rizvi, 2015).

## 2.3. Authentication

The verb authenticate, which means to verify or confirm, is the root of the word authenticity, which means action. It refers to a claim that an artistic work is legitimate, genuine, or authentic. The process of confirming if someone or something is, who or what it is purported to be, is known as authenticating a person. There are numerous instances where this procedure takes place, like as

- The provenance, or source, of an artwork, which serves as its authentication,
- Before depositing a check, the procedure of inspection of a driving license as identification,
- The use of a passport photo to cross international borders.

In all of these scenarios, the element painting, the person giving check, or verification of the individual —is present or will be carried close to the entity in charge of establishing the authenticity and accepting or rejecting it.

Authentication refers to demonstrating your identity in the context of an ATM. What if a machine rather than a person is the entity doing the authentication? To conquer this the banks demand two proofs of the consumer's identity.

- A consumer's banking card, with a magnetically recorded Primary Account Number (PAN), and
- A set of personal numbers for separate identification, commonly referred to as a Personal Identification Number (PIN).

The consumer would place their ATM bank card with in the card reader space on the ATM, which would then read their PAN and prompt them to input their PIN using a keyboard. The ATM might confirm the connection among the PAN and PIN, thereby authenticating the user who presents it. The choice of transactions that would then be allowed was left up to the bank's discretion (Konheim, 2016).

## 2.4. Verification with Personal Identification Number and Primary Account Number Mechanism

It is vital to confirm that the PAN → PIN derivation is accurate. To achieve this the PAN and PIN were made operaionally related; that is, $PIN_i$ changes depending on the value that $PAN_i$ holds, where relation is complicated. Verification involves confirming the existence of this relationship between the users entered $PIN_i$ and the $PAN_j$ read from the inserted bank card.

## 2.5. The Personal Identification Number and Primary Account Number Security Mechanism

Cryptography was utilized to make the PIN-PAN connection E (SK, PANj) → PINj impossible to understand or interpret. When the notation E (SK,*) is employed, it means that the quantity * has been encrypted by means of a secret encryption key SK. Or we can infer that it signifies that PINj is somehow obtained from the encoding or encryption process employing PANj and secret key SK.

## 2.6. Programming Interface

A collection of instructions known as an Application Programming Interface (API) is used to bundle and present sophisticated functionality to external customers in an easy-to-use manner (Gorski & Iacono, 2016). These include initialization and management commands as well as any additional commands that directly affect the functionality. Clients who utilize these commands as the building blocks for software programs shouldn't be concerned with the supporting details as they are hidden. Here's an illustration: Without having to open network sockets or run checksums on incoming packets, a consumer may conveniently use the Host Connection command to establish a Hypertext Transfer Protocol interconnection. The Java Standard Development Kit API effectively illustrates the software API.

The aim of a security API is the same, but its commands are primarily intended to provide defense facilities and originate several implicit otherwise clearly stated security policies. These targets to restrict the command's behavior that ensures security. As, SecureTransfer (Integer x, Account acc1, Account acc2) is an API command in a banking system that moves from accl to acc2, x dollars. The command will possibly accomplish this by establishing a SSL connection with the banking server, however, this is hidden from the consumer. There is an underlying notion that the information provided to the banking server cannot be altered. This presumption represents one of this command's security policies. The Microsoft Cryptographic API is a nice illustration of a software security API.

### 2.6.1. The Security Application Programming Interface

Science-related fields that specialize in security protocol analysis are where the term "Security API" initially appeared. An application programming interface known as a security API applies a security policy to transactions between two entities using cryptography, according to Bond's definition (2004), "A security API (Bond, 2004) is a type of API that employs encryption to impose security rules on communications among two entities.". This would not apply to APIs that offer security functionalities without using encryption, lowering the risk of injection attacks like Cross-Site Scripting and SQL Injection (OWASP 2013), such as input validation libraries. Security APIs are a connection between an authorized and an insecure region, according to Steel (2011) (Steel, 2011).

## 2.7. Encrypting Personal Identification Number Pad

Financial institutions employ the PIN Entry Device of an ATM, an encrypting PIN pad, for a variety of secure and private financial operations. This machine is primarily used to conduct a credit, debit, or smartcard-related transactions, and while doing so, the cardholder's identifying code is often encrypted.


**Figure 2. 3.** Encrypting PIN Pad

Making sure that the owner's credit or debit cards are safely read is one of the encrypted pads' main goals. The ability for customers to type their passwords securely while having those passwords encrypted before being transferred to the appropriate financial institution is another crucial feature of the encrypting PIN pad.

In a similar spirit, the encrypting PIN pad was created to make PIN debit transactions hosted by various PIN networks easier to complete.

### 2.7.1. How Does Encrypting Personal Identification Number Pad Work?

PIN pad encryption is similar to ATM hardware or other types of payment terminals. With the PIN input component, a credit or debit card PIN can be securely read and sent from the card to financial institutions. However, when thinking about the chip card, what occurs in that scenario is that the encrypting PIN pad does a chip-based card verification.

An encrypting PIN pad contains hardware and software that serve as the safety features of the device and ensure that the passwords and encryption keys can be deleted quickly and easily if someone tries to obtain unauthorized access to the device.

The identity cipher is often encoded at the time of each transaction or entry, followed by the establishment of an encrypted PIN blocking code to thwart any unauthorized access. The PIN block code will automatically be removed whenever the encrypting PIN pads provide an encrypted PIN to the chosen payment terminal or financial institution, and upon receipt of the PIN by the receiver.

Most of the time, encrypting PIN pads only allowed the use of digits, while a few of them also allow for the use of alphabets.

But because encrypting PIN pads are so delicate, these gadgets are typically created and approved by the rules governing the payment and card industries. In light of this, such devices will certainly offer the necessary level of security during a transaction, from PIN entry to PIN encryption.

The current standard for managing and encrypting PIN pads is ISO 9564 (ISO/TC, Technical Committee, 2017).

### 2.8. Hardware Security Module (HSM)

The devices known as Hardware Security Modules (HSMs), also referred to as Tamper Resistant Security Modules (TRSMs), are used to carry out cryptographic operations such as the computation of specific values like Card Verification Values (CVVs) or Personal Identification Numbers (PIN), data anonymization, registration administration, and so forth.

These are physically protected and/or tamper-resistant gadgets, which means that any effort to penetrate the device will result in the rapid depletion of all sensitive data kept in its memory. Devices that enable high-speed encryption at the network level, or the ones that issue or sign certificates for a Certification Authority (CA), those that use time stamping, etc. Retail Point of Sales (POS) terminals that process “Chip and PIN” transactions and feature, a security core that is commonly referred to as an HSM are another excellent example.

The HSM is either a physical device for the virtual machine or a bus-connected device. While several communication protocols were typically supported in the past, most peripherally connected HSMs today communicate with the host machine via Ethernet or fiber cable. Customers would then select the protocol that best suits their need for transaction throughput and financial constraints. In addition to having a port for communication with the host computer, HSMs typically support a variety of other input/output methods, such as a smart card reader, keypad, central management connector, printer port, CD/DVD drive to load software, or a console to conduct HSM management or key ceremonies.

#### 2.8.1. Usage

An HSM is a piece of hardware or a Payment Card Industry (PCI) card that responds to instructions from an application via a vendor-specified application programming interface (API). The API can be easily modified to satisfy customer needs; this can be done by the customer or, more frequently, by the HSM vendor. A vendor private key is used to digitally sign orally or in writing HSM software or firmware, and the corresponding public key put in the HSM as part of the manufacturing process is used to verify the signature. Examples of applications for HSMs include the safeguarding of personal information (such as health records, databases, etc.), bulk encryption, and trusted third-party services (certificate authorities, signature authorities, etc.).Image 2.4 shows a basic three-tiered structure with an HSM.

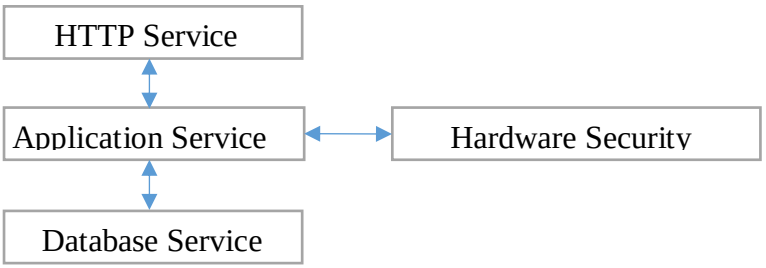


**Figure 2. 4.** A Basic Three-tiered Structure with an HSM

Before moving on to the next tier of the process, a message for the application server would be forwarded to the HSM for cryptographic processing. When financial records are encrypted the web server has received data and transmitted to the server. The data would then be encrypted on the application server and sent to the database server to be stored with the help of the HSM.

During the financial transaction, the server would send the banking details to be confirmed to the HSM, which would calculate an on-the-fly cryptographic value (like a Card Verification Value (CVV)] with the associated key Card Verification Key (CVK) and make a comparison with the value entered to confirm or reject the transaction. As mentioned before, our focus will be on HSMs utilized in the banking industry. One illustration is how they use "Chip and PIN" payment cards, like debit or credit cards. HSMs are mostly utilized with these cards in two areas:

- as part of the issuing process, data preparation, card personalization, and mailer printing for personal identification numbers (PINs);
- transaction handling

Several secrets or sensitive values, such as a PIN and CVVs (CVV/CVV2/iCVV), must be created and loaded onto the card to personalize it. There are cryptographic keys among these values (symmetric keys used during transaction processing and asymmetric keys, along with certificates, used for authentication purposes). These values are often created using specific cryptographic keys (PIN Verification Key (PVK) for a PIN and CVK for CVV values; we will refer to them as functional keys in general) during the preprocessing stage (the data creation creates the demographic data). The card personalization system receives these values and loads them into the card.

The card issuer uses an HSM during transaction processing to guarantee the integrity of transaction communications, at least in the case of online transactions. The PIN is encrypted by the ATM and forwarded to the issuer for verification, potentially via an acquiring organization, in the specific scenario where the card is used at an ATM to withdraw cash. For PIN translation and message integrity, an HSM would be employed by the acquirer.

The entire management of Chip and PIN cards is far more complex than the rather cursory explanation given above. An HSM is used in every step of this procedure because each step of the issuing and acquiring processes needs unique keys for unique

cryptographic operations. HSMs must accommodate a range of cryptographic techniques, including:

- DES ((NIST), National Institute of Standards and Technology, 1999) (although it is no longer used), 3-DES (Barker & Mouha, 2017), AES ((NIST), National Institute of Standards and Technology, 2001), RSA (United States Patent No. 4,405,829, 1977), SHA-1 (Information Technology Laboratory, National Institute of Standards and Technology (NIST), 2015), and SHA-256 (Information Technology Laboratory, National Institute of Standards and Technology (NIST), 2015), cryptographic algorithms;
- administration of cryptographic keys, including key creation, key elaboration, distribution, and storage ((PCI), Payment Card Industry, 2021);
- encryption of data, particularly PIN encryption methods ((PCI), Payment Card Industry, 2021);
- data integrity, including the creation and verification of Message Authentication Codes (MAC) and digital signatures (Barker E. , 2016);
- the creation and verification of CVV (Mavrovouniotis & Ganley, 2014);
- PIN verification, PIN translation, and PIN generation (Mavrovouniotis & Ganley, 2014);
- Chip cryptographic values production and verification, as well as key generation and transfer (Mavrovouniotis & Ganley, 2014).

All of the aforementioned functionality, including various PIN generation/verification, PIN block formats, key management strategies, MAC algorithms, and encryption modes, will be supported by a typical HSM used only in financial applications.

An HSM API will often have a large number of quite complex functions, frequently with a variety of alternatives. In the previously discussed Chip and PIN application, for instance, a single HSM command used by an acquirer could include PIN conversion, a format modification, content decoding using one key, followed by re-encryption using a different key (possibly using a different mode of encryption), and MAC verification followed by the creation of a new MAC.

A command for an HSM used in a payment transaction would typically be formatted as a command header comprising of command code and command data and a command trailer. Whereas the corresponding response message is formatted as a response header comprising of response code, error code, and response data, lastly the response trailer

For instance, an acquirer instruction that entails PIN translation, verification and generation of MAC: Errors in the aforementioned command could take many different forms, like:

- DES or 3-DES key parity problems (corrupted keys may have been stored);
- a MAC mode or PIN block format that isn't valid;
- a PIN block error, such as when the format of the plaintext PIN block is incorrect;
- failure of MAC verification;

- Syntax mistake in the command.

If an error is found, the HSM ought to provide the proper error code and move on to the subsequent transaction. As was already established, an HSM's protection of confidential data, particularly cryptographic keys and PINs, is a key function.

### 2.8.2. Physical Security of Hardware Security Module

HSMs' key role is to deliver high-grade cryptographic security, and the physical security of the device is a key component of that security. However, it must be emphasized that this is only one part of HSM security; procedural safeguards and assaults through the HSM's API are as crucial. One could argue that an HSM's physical security is the simple part because a physical attack will almost certainly be swiftly discovered in contrast to a logical or procedural attack, which may never be discovered!

An anti-counterfeit core, an HSM set of channels that houses every delicate aspect, is the foundation of an HSM's major defense against physical assault. The Host Master Key (HMK), which will be described later, and other plaintext cryptographic keys will typically be stored in battery-backed volatile memory provided by the security core, and all cryptographic processing will take place within the core system.

The secure memory's contents will be quickly wiped ("zeroized") should an attack on the core sub-system be discovered thanks to the core sub-tamper-resistant system's characteristics. HSM software/firmware is typically stored in a combination of ROM and E2PROM and won't be lost if the device is tampered with because of this. According to the ANSI X9.24-1 (X9.24-1, ANSI, 2009) standard:

Features that prevent effective fiddling with an HSM must include breaching without adjusting security parameters to zero, illegal modifications of the HSM's operating units, or the insertion of mechanisms or non-intrusive eavesdropping techniques to identify, record, or modify sensitive data. Such features must include one or more of the following:

- Irrespective of the power state of the HSM, tamper-detection systems must be activated;
- Physical obstacles to prevent effective tampering;
- Sufficient tamper resistance to make successful tampering take a long time (tampered device should be noticed before being returned to begin cryptographic functions before an HSM is missing from its approved location);
- The HSM is designed with a tamper-evident feature, which signifies that a successful infringement will cause physical harm to the device that will be apparent after it has been put back in its proper place but before cryptographic operations resume.

No matter how an HSM is used, rigorous physical restrictions must be in place because of its nature. This would typically imply that the HSM is situated in a zone with heightened security, secured within a locked cabinet, and subject to two physical controls, such as two keys, a key, and a combination, a key and a biometric or another comparable and successful strategy. Therefore, an attacker would find it very challenging to remove an HSM from its customary location without being noticed unless he is an individual with

physical access and the required rights. Due to this, dual control should be used, and the second person shouldn't necessarily be permitted to be close to the particular cabinet. Other HSM types, like the security core of a retail PIN pad, might not be subject to the same restrictions. As a result, the primary line of defense against physical attack is an HSM's tamper-detection circuitry, which must zeroize protected memory as soon as an attack is detected.

Attacks employing changes in voltage or current, low-temperature assaults, power analysis or timing attacks (part of a class of attacks known as "side channel" attacks), drilling, or other techniques to break the security core, and these are only a few of the attacks that need to be defended against.

The typical defenses against such attacks involve encasing the entire security core in epoxy resin and then covering it in some kind of fine-grained electronic mesh. The mesh will probably break if an assailant tries to break through the resin. The zeroization circuitry is quickly activated if the mesh is destroyed or damaged in any way.

Additional HSM defenses include solid vaults, microswitches, light-sensitive diodes, mercury tilt switches, temperature monitoring sensors, and detectors that may detect variations in voltage and current. Side channel assaults are unlikely to be successful unless the attacker can access the core (in which case one is probably unnecessary!).

Furthermore, HSM providers typically include defenses against these threats. Important information: It must be emphasized that the HSM can only be protected by these controls when they are turned on. Some of these controls can be turned on or off. Only when these controls are engaged is an HSM deemed to be an HSM (Mavrovouniotis & Ganley, 2014).

### 2.8.3. Key Types Employed by the Hardware Security Module

The keys that HSM processes are all encrypted using other Key Encryption Key (KEK) keys; it never uses plain keys. HSM's core tenet is that a real key value or a true functioning key plain value cannot be obtained. With HSM, all of your keys are cryptograms.

#### 1. Local Master Key - LMK

All additional keys used by the institution are safeguarded using LMKs, which serves as the HSM’s principal key. Banks typically use many HSMs as transaction volume rises. This does not imply that there are numerous LMKs. Per the site, there is just one LMK. It is each institution’s most important key.

Three representatives of the relevant institution each come up with three distinct critical components. A separate authority assigned by that organization is responsible for storing all of these distinct clear components, inserted into HSM with the aid of a chip card, and a KCV is produced. Each produces a smart card copy during this process as a backup. The HSM receives the components from each custodian and combines them to create the ZMK. The LMK is commonly created by simply XOR-ing the clear components.

Data is not encrypted using LMKs; instead, other keys are encrypted and decrypted when they enter and exit the HSM. Even if the data traffic between a client and the HSM is being monitored, LMKs are utilized to ensure that the clear values of any transmitted keys are secure.

**2. Zone Master Key - ZMK**

To trade encrypted data like PIN blocks, banks periodically requires to transfer keys to external parties like Visa or MasterCard. In that circumstance, Banks ought to employ the ZMK KEK.

For additional keys to be exchanged automatically, a Zone Master Key (ZMK) must be manually provided on communications link, between two (or more) connecting nodes (without the need for manual intervention). Lower-level keys are encrypted for transmission using the ZMK. One of the LMK pairings encrypts a ZMK underneath it for local storage. This is referred to as a ZCMK in the VISA environment.

**3. Zone PIN Key - ZPK**

A Zone PIN Key (ZPK) is an automatic data encryption key that encodes PINs to transmit them securely among the linked entities. Under a ZMK, a ZPK is encoded for communication and under one of the LMK pairings for local storage.

**4. Acquirer Working Key - AWK**

The service provider and acquiring bank employ the AWK data-encrypting key to encrypt PIN data.

**5. Issuer Working Key - IWK**

The PIN between the switch and the issuer is encrypted using the IWK. The PIN is kept secure throughout the transaction in this way.

**6. PIN Verification Key - PVK**

PIN verification info is created and verified using a PIN Verification Key (PVK), an information-encrypting key, thereby confirming a PIN's authenticity. A PVK is encoded for communication utilizing TMK or a ZMK, and for local storage, it is encrypted using one of the LMK pairs.

### 2.9. The Automated Teller Machine Attacks

Both physically and electronically, cybercriminals[1] attack ATMs to steal money for their gain or the benefit of a nation-state. These assaults frequently take place during holidays to avoid or postpone detection. One or more financial institutions may be used to create counterfeit payment cards in this situation.

[1] A person who engages in criminal activity by means of computers or the internet.

Various ATM attacks are described below:

### 2.9.1. Black Box Attacks

A black box (Umawing, 2019) is some kind of a device or altered circuit connected to a USB connection capable of executing ATM commands at the command of the racketeer. They can confirm transactions without using cards or obtaining authorization if they physically delink the cash dispenser and the ATM and attaches the black box to the ATM's computer. This attack's likely targets are retail ATMs located outside of a business.

### 2.9.2. Automated Teller Machine Skimming

ATM skimming is a sort of economic fraud in which "skimmers" are placed at ATM terminals to record credit or debit card data. The information from payment cards is stolen via ATM skimmers, which are attached to ATM card readers. They are designed to imitate a real ATM part.

### 2.9.3. Memory Scrapers

These are harmful software that inspects the ram of the device for the victim's financial card information during the execution by the ATM. By using stolen remote support credentials or other tactics, attackers frequently compromise ATM systems with malware in order to steal bank card data.

### 2.9.4. Man-in-the-Middle Attack

A Man-in-the-Middle (MitM) assault is carried out when a threat actor succeeds in intercepting and forward the traffic between two entities without either of them noticing. Additionally, certain MitM attacks change parties' communication, again without their knowledge.

### 2.9.5. Code Book Attack

A cryptanalysis method is known as a code book (Citizendium) assault. The term "code book assault" refers to the attempt to create a table listing which cipher texts correlate to which plaintexts when a block cypher is being attacked. The attacker tries to compile a listing of the full output stream up until it repeats in another variation that can be used against stream cyphers.

### 2.9.6. Automated Teller Machine Jackpotting

Jackpotting is a method used by cybercriminals to deceive an ATM into distributing cash, much like the black box assault. Jackpotting is a pretty simple crime that has become more prevalent over the years. Without a doubt, it will continue to rise.

# 3. HOW AUTOMATED TELLER MACHINE WORKS?

## 3.1 The Theoretical Framework of the Automated Teller Machine Mechanism

A front terminal in real time for 24/7 banking facilities, an ATM system is supported by a centrally controlled bank database and a financial sector server. As depicted in Figure 3.1, the ATM system's architecture consists of an ATM processor, a system clock, a remote account database, and several peripheral devices, including a card reader, monitor, keypad, bills storage, and bills disburser.

A Finite State Machine (FSM) is typically used to define the theoretical framework of an ATM system. A series of phases and phase transition mechanisms are adopted by the FSM that are represented by a transition illustration or a table to refer to the ATM system's fundamental behaviors. The most significant behaviors of ATMs can be depicted in a transition illustration, as demonstrated in Figure 3.2, based on theoretical framework of the ATM system demonstrated in the Figure 3.1.

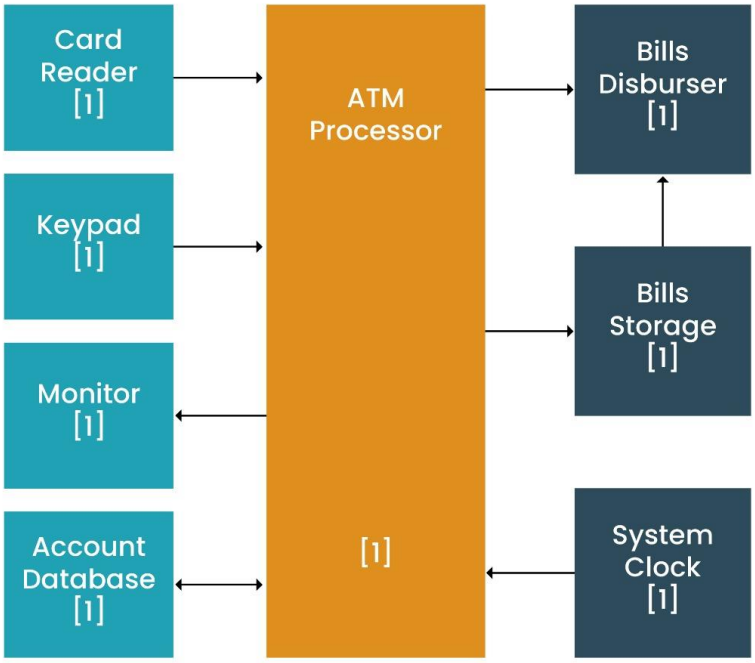


**Figure 3.1** The Theoretical Framework of the ATM System

ATMST A (S, ∑, s, F, δ)

Where

- The domain of the ATM is defined by the set S, where state0, state1,..., state7, ∈ S and these are defined as:

  state0 – System,

state1 – Welcome,

state2 - Check PIN,

state3 – Input withdraw amount,

state4 - Verify balance,

state5 - Verify bills availability,

state6 - Disburse bills, and

state7 - Eject card, respectively;

- The ATM may accept and handle a series of events that is event0, event1, …, event10 $\in \sum$ where:

  event0 - Start,

  event1 - Insert card,

  event2 - Correct PIN,

  event3 - Incorrect PIN,

  event4 – Request ≤ max,

  event5 – Request > max,

  event6 - Cancel transaction,

  event7 - Sufficient funds,

  event8 - Insufficient funds,

  event9 - Sufficient bills in ATM, and

  event10 - Insufficient bills in ATM;

- state0 represents the ATM's initial state; identical to state1 (Welcome);
- state1 ∈ F; F represents the collection of final states;
- According to the ATM's transition mechanism, $state_i+1$ = ($state_i$, $event_i$), the future state of the FSM is determined by the present state, $state_i$ and a particular incoming event, where:

$$\delta = f: S \times \sum \rightarrow S \qquad \text{(Wang et al., 2010)}$$

It is consistent with the theoretical description of the operations of the ATM as seen in Figure 3.2

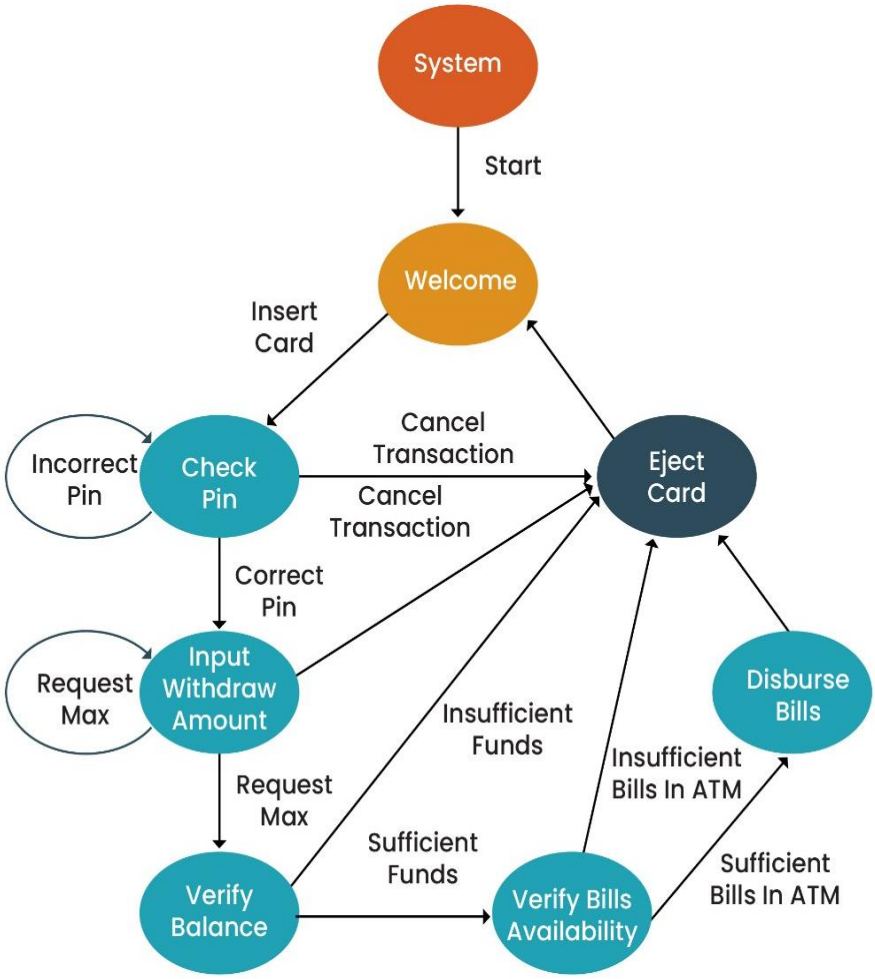


**Figure 3.2** The Transition Diagram of the ATM Behaviors

### 3.2 An Automated Teller Machine Network

An ATM network is simply depicted in Figure 3.3. The customer-facing device, which is often an ATM device, is on the far left. The bank that "operates" the device, or the acquiring bank, is tied to the ATM terminal. It should be noted that the user does not necessarily need to have an account with the acquirer, and in light of this illustration, we'll suppose that is not the case. As a result, the transaction must be sent to the bank where the user has an account (i.e. the card-issuing bank) (Clulow & Clulow, 2003).

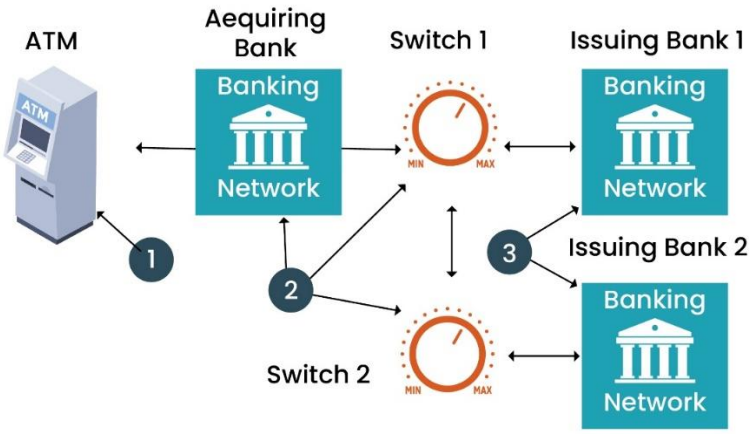


**Figure 3. 3** Simple Representation of ATM Network

To carry out the transaction the two entities i.e. the issuing bank and the acquiring need to communicate. The user initiates by inserting his or her ATM card and entering the PIN for authentication. The PIN is a very sensitive entity, wherein the PIN has to traverse from the ATM terminal, it has to go back to the issuer because the card has been issued by the issuing bank. And therefore only the issuing bank has the final authority to approve or reject the PIN entered. It is vital to ensure that the PIN when it traverses from one hop to the other, does that in a very secure way. When the PIN has to be transported it has to be converted into a block and that transportation happens from the acquiring side to the issuing side via the network. There are various ISO formats for generating PIN blocks that are being prescribed by PCI SSC (i.e. Standards for enhancing financial network security, such as the PCI Data Security Standard, are created, managed, educated about, and enforced through the PCI Security Standards Council, an international body that is open to all parties).

## 3.3 What Are Personal Identification Number Blocks?

### 3.3.1. Personal Identification Number Management

Cardholders have a 4- to 6-digit PIN to confirm that the individual presenting a payment card has permission to use it. PINs are often utilized in the payment card industry. The PIN is the cardholder verification technique that is most frequently utilized, even if the payment ecosystem offers other cardholder verification methods as well.

From the moment a cardholder inputs their PIN at a payment acceptance terminal until the issuer, or their agent, confirms the PIN is accurate, certain technical and procedural safeguards must be in place to guarantee PIN security (Haque, 2018). PIN management's objective is to decrease the likelihood of scam occurring within the payment card infrastructure by protecting the PIN from misuse, unauthorized disclosure, and exposure during its lifespan. Throughout the PIN's full life cycle, including its generation, distribution, initiation, maintenance, access, transfer, verification, disablement, and any additional operations, its confidentiality must be protected.

A vital security component for protecting the PIN is the use of PIN blocks.

### 3.3.2. Personal Identification Number Blocks

A PIN is protected by cryptography throughout the majority of its lifespan. As per ISO 9564-1

“The same PIN value is being linked to multiple accounts, the endorsed encipherment technique must make sure that the encipherment of the textual PIN value using a specific cryptographic key does not predictably produce the same enciphered value.”

The PIN is arranged into a PIN block to aid with this encoding.

PIN blocks are available in five different formats as per the ISO standard, shown in the following table. (Council, PCI Security Standards, 2021).

**Table 3. 1.** PIN Block Formats

| Format | Description |
|---|---|
| 0 | The account number field and the plain text PIN field are two 64-bit fields that are added together to create this PIN block. |
| 1 | The transaction field and the plain text PIN field are concatenated to create this PIN block. |
| 2 | For use with IC cards, the format 2 PIN block has been provided. The format 2 PIN block must only be used in offline settings; it cannot be used for PIN verification online. The filler field and the plain text PIN field are concatenated to create this PIN block. |
| 3 | With the exception of the fill digits, the format 3 PIN block is identical to the format 0 PIN block. |
| 4 | Two 128-bit PIN and PAN data fields are used to create Format 4, an enhanced PIN block format. It is possible to utilize a 128-bit block cypher (AES) with this format. |

### 3.3.3. Personal Identification Number Block Format

We need to ensure that the PIN is safe and secure when traveling from the acquirer to the issuer when a cardholder enters his PIN into any PED [POS, ATM, etc.]. Therefore, the first step is to create a block of data called a "Pin Block" using one of the standards for creating such data. This block of data combines the PIN and the card PAN. The ISO 9564 is the foundation of majority of PIN-block formats, even though there are several others that are widely utilized. The most common PIN block formats are as follows:

### 3.3.3.1. International Organization for Standardization 9564 – Format 0

ISO-0 serves as the most widely used PIN code format internationally. Its key feature is how it includes the PIN's association with a particular PAN in the block data. To retrieve the correct PIN from the block, the PAN must be supplied. (transferred with the PIN block). The PIN and the PAN are two pieces of data that are combined to form information found in ISO PIN Block 0 (Tushie, 2015).

The following are the PIN digits' meanings:

| Format | Cnt | P | P | P | P | P/X | P/X | P/X | P/X | P/X | P/X | P/X | P/X | X | X |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Format: indicates block format (ISO-0 = 0)

Cnt: number of PIN digits (4-12 (hex ‘C’))

P: PIN

P/X: PIN or FILL (hex ‘F’) as needed

| N | N | N | N | P | P | P | P | P | P | P | P | P | P | P | P |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

N: Null (0)

P: Right most 12 PAN digits excluding the check digit

**Computation Procedure:**

Create a PIN by using the following formula: L = length of PIN, P = PIN digit, and F = padding value "F"

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | L | P | P | P | P | P/F | P/F | P/F | P/F | P/F | P/F | P/F | P/F | P/F | P/F |

Take the 12 rightmost digits of the primary account number to prepare PAN (excluding the check digit)

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | PAN | PAN | PAN | PAN | PAN | PAN | PAN | PAN | PAN | PAN | PAN | PAN |

XOR both values

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | L | P | P | P | P | P/F | P/F | P/F | P/F | P/F | P/F | P/F | P/F | P/F | P/F |
| XOR | XOR | XOR | XOR | XOR | XOR | XOR | XOR | XOR | XOR | XOR | XOR | XOR | XOR | XOR | XOR |
| 0 | 0 | 0 | 0 | PAN | PAN | PAN | PAN | PAN | PAN | PAN | PAN | PAN | PAN | PAN | PAN |

**Example:**

| **PIN data** | 0 | 4 | 8 | 4 | 8 | 2 | F | F | F | F | F | F | F | F | F | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **PAN data** | 0 | 0 | 0 | 0 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 |
| **PIN block** | 0 | 4 | 8 | 4 | C | 7 | 9 | 8 | 7 | 6 | F | E | D | C | B | A |

After a decrypted ISO-0 PIN block has been obtained, the recipient must validate that the count spans from 4 to 12 ("C") and the format is "0".If not, the transmission was likely tampered with. The message has once more been corrupted if the PAN's XOR doesn't yield the right padding.

### 3.3.3.2. International Organization for Standardization 9564 – Format 1

This PIN block format accepts PINs with a length of 4 to 12 digits. The right side of a PIN with more than 12 digits are truncated.

There is no PAN to link the PIN to while utilizing ISO-1 PIN Block format. In the event that VISA PVV is adopted, this might mean that the PINs must be transferred to the PVV calculator because they are generated in one location (by association with a PAN) before the PVV calculation (Tushie, 2015)

The following are the PIN digits' meanings:

| Format | Cnt | P | P | P | P | P/X | P/X | P/X | P/X | P/X | P/X | P/X | P/X | X | X |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Format: indicates block format (ISO-1 = 1)

Cnt: number of PIN digits (4-12 (hex ‘C’)

P: PIN

P/X: PIN or FILL (random digits as needed)

**Computation Procedure:**

Create a PIN by multiplying L by the length, P by the PIN digit, and R by a random number.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | L | P | P | P | P | P/R | P/R | P/R | P/R | P/R | P/R | P/R | P/R | P/R | P/R |

Where:

The PIN length, L, is a 4-bit number ranging from X'4′ to X'C'.

R is a randomly chosen digit with a value ranging from X'0′ to X'F'. For preset transaction unique data, such a sequence number, this should typically be utilized.

**Example:**

PAN: 43219876543210987
PIN: 1234

PIN block: 141234CE8C767872

Even with identical PINs, the insertion of arbitrary padding—as opposed to consecutive repetitive padding—creates a distinctive encoded PIN block.

### 3.3.3.3. International Organization for Standardization 9564 – Format 3

The ISO-3 PIN Block format is an ISO-0 PIN Block with arbitrary padding in place of 0xF. It conceals the PAN digits which might present in the ISO-0 PIN Block as inverted digits and the block data includes the PIN for a particular PAN. (Tushie, 2015). The PAN must be known retrieve the accurate PIN from the block. The ISO-3 format is advised by several card companies for PIN transmissions. The PIN and PAN, two data elements, are XOR-ed to form PIN blocks of this format.

The following are the PIN digits' meanings:

| Format | Cnt | P | P | P | P | P/X | P/X | P/X | P/X | P/X | P/X | P/X | P/X | X | X |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Format: indicates block format (ISO-3 = 3)

Cnt: number of PIN digits (4-12 (hex ‘C’))

P: PIN

P/X: PIN or FILL (random hex digits (0x0-0xF) as needed)

The meanings of the PAN digits are as follows:

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| N | N | N | N | P | P | P | P | P | P | P | P | P | P | P | P |

N: Null (0)

P: Right most 12 PAN digits excluding the check digit

**Computation Procedure:**

Create a PIN. The PIN's length is L, its digit is P, and a random number between X'0′ and X'F' is R.

| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | L | P | P | P | P | P/R | P/R | P/R | P/R | P/R | P/R | P/R | P/R | P/R | P/R |

To prepare a PAN, subtract the 12 rightmost digits from the primary account number (excluding the check digit)

| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | PAN | PAN | PAN | PAN | PAN | PAN | PAN | PAN | PAN | PAN | PAN | PAN |

XOR both values

| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | L | P | P | P | P | P/F | P/F | P/F | P/F | P/F | P/F | P/F | P/F | P/F | P/F |
| XOR | XOR | XOR | XOR | XOR | XOR | XOR | XOR | XOR | XOR | XOR | XOR | XOR | XOR | XOR | XOR |
| 0 | 0 | 0 | 0 | PAN | PAN | PAN | PAN | PAN | PAN | PAN | PAN | PAN | PAN | PAN | PAN |

**Example:**

| **PIN data** | 3 | 4 | 8 | 4 | 8 | 2 | 2 | 5 | F | 3 | 9 | C | E | 4 | B | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **PAN data** | 0 | 0 | 0 | 0 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 |
| **PIN block** | 3 | 4 | 8 | 4 | C | 7 | 4 | 2 | 7 | A | 9 | D | C | 7 | F | 4 |

Once an ISO – PIN block has been encrypted, the recipient should confirm that the count spans from 4 to 12 ("C") and that the format is "3". If not, the transmission was likely tampered with.

PIN translation when done from one Format to the other where allowed must be within a secure Cryptographic device like HSM. One of the commonly used standards for encoding a PIN Block is ISO 9564-1 Format 0.

After getting the value of a clear PIN Block. The PIN Block must be converted with an encryption secret before sending it from the PED to the Acquiring Bank to deliver it safely and securely.

The pin is typically encrypted using the Triple DES method by newer ATMs before being sent to a host server for processing. The host system converts the pin from one encryption key to another after receiving it before sending it to the bank.

### 3.4. Personal Identification Number Encryption

A PIN (Personal Identification Number), which confirms the cardholder's identity, is used to protect ATM transactions. The card’s bearer and its security system are only aware of the PIN. Encryption is used to preserve the PIN by blurring and making it unreadable in the event that a third party intercepts the transaction while it is being transferred over the network. (Sholes, 2002)

The Data Encryption Standard (DES), which is detailed in ANSI X9.8, has been a national standard since 1977. To authenticate the identity of the cardholder, DES utilizes a single cryptographic key to scramble the PIN at the ATM and decrypt it once it is received by the processor.

Because of the small key size (64 bits), DES's robustness has come under scrutiny. With the use of a specially designed computer called the DES Cracker, a group by the name of the Electronic Freedom Foundation was able to crack DES in less than three days in 1998. The level of security needed for ATM transactions rises along with overall technological advancement. The level of security provided by triple DES technology is much higher. It can be added to the current Electronic Funds Transfer (EFT) network with the least amount of disturbance because it uses the same algorithm as a single DES. The deployment of Triple-DES is required to uphold public confidence in payment systems and to guarantee the integrity of private cardholder data.

The term "Triple DES" refers to a new specification, ANSI X9.52, which uses two 64-bit keys (effectively 128 bits) and applies them three times. Although it is done three times, the encryption process is precisely the same as with a single DES. The PIN is encrypted using the first key in Triple DES, decrypted using the second key, and then encrypted once more using the first key.

The technique for decrypting the PIN is the same when the processor receives a transaction produced by an ATM, except that it is carried out backward.

### 3.5 Triple Data Encryption Standard

The Triple-DES is based on a standard encryption technique that was first presented in 1975. The cipher block uses a straightforward 7-bytes key was susceptible to brute force assaults over time since computing performance improved. Then, in 1999, the new Triple-DES standard which tripled the key size of cipher and extended the life of DES by encrypting data in three runs.

### What is 3DES?

The 3DES algorithm uses the Data Encryption Standard (DES) cypher thrice for encipherment.

A symmetric-key algorithm built on a Feistel network is called DES. It is a symmetric key crypto, only one secret (i.e. key) is used in both the encoding and decoding procedures. The Feistel network makes each of these methods nearly identical, resulting in a more straightforward procedure.

Although the block and key of DES are indeed 8 bytes, the key only provides 7 bytes of security in reality. 3DES was developed as a more secure substitute for DES because of its shortened key length. Even though 3DES uses three separate keys and repeats the DES process thrice, it is only considered secure when 3 distinct keys are used.

### 3.5.1 3DES Algorithm

3DES Algorithm (Lake, 2022) requires triplet of 8-bytes keys, for a 24-byte full key size. The full 24-bytes key is supplied all at once as opposed to the 3 secrets (i.e. keys) are each being supplied separately. The 3DES Dynamic Link Library formerly separates the subscriber key into 3 sub - blocks, possibly filling every block to a length of 8 bytes. Initially, the content is encoded with the 1$^{st}$ key, then decoded with 2$^{nd}$ key, and finally secured using the last key. 3DES is a form of DES that uses the same encryption algorithm as regular DES but repeats it thrice.

### 3.5.2. Personal Identification Number Translation

PIN translation is arguably one of the payment methods that cause the greatest confusion because it is an encryption method. To continue serving as a secure method of transmitting PIN codes, the encryption procedure must maintain some level of complexity.

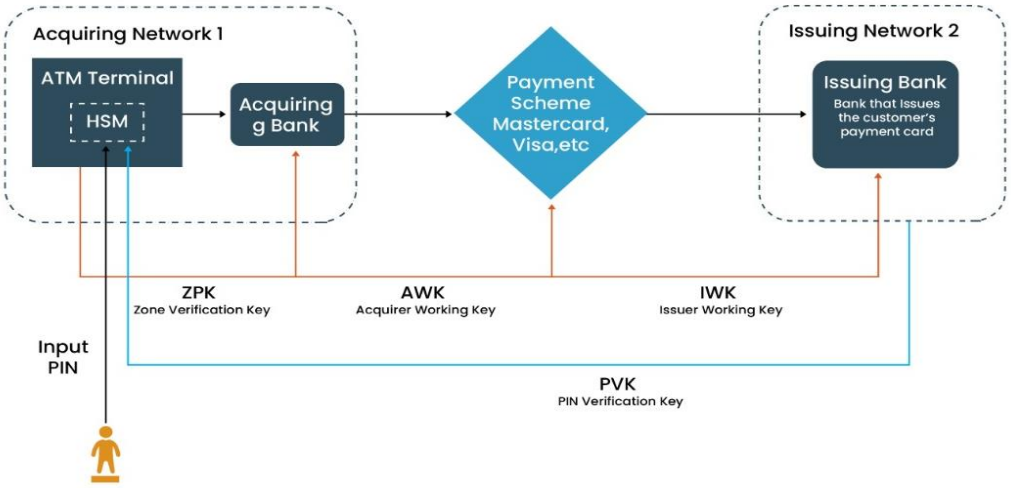


**Figure 3.4** Different Stages of PIN Translation

The PIN of the user will be transmitted from the ATM terminal to the acquiring bank, the payment scheme, and then to the issuing bank, which is the only place that can confirm whether the PIN is accurate. The PIN will be encrypted while passing through each of these points, and the encryption will change at each point to maintain the required level of security.

### Automated Teller Machine to Acquiring Bank

ZPK is the key used for the encryption between the ATM and the acquiring bank. This key was produced using the ZMK. Because both the ATM and the acquiring bank are familiar with the ZMK idea, they can exchange and decrypt the ZPK. Both encryption and decryption are not feasible without the master key (ZMK).

### Acquiring Bank to Payment Scheme

The PIN must now be added to the card system by the acquiring bank. The acquiring bank will first decode the code and then re-encrypt it using another key called AWK (Acquirer Working Key). The four-digit numerical code will be converted back to the 16-digit hexadecimal code before being encrypted again.

### Payment Scheme to Issuing Bank

The same thing occurs when the key is transferred from the payment scheme to the issuing bank for the final time. The key will once more be encrypted after being decrypted into its original 4-digit numerical form. The key will be converted into the IWK in this instance (issuer working key). The issuer will employ the PVK (Pin Validation Key), which was initially used to generate the PIN, in a final step to confirm that the PIN code entered in the POS is accurate.

The issuing bank will confirm by comparing the PIN to a customer account (or not). Through all of the instances, the verification will be provided as a code back to the ATM terminal.

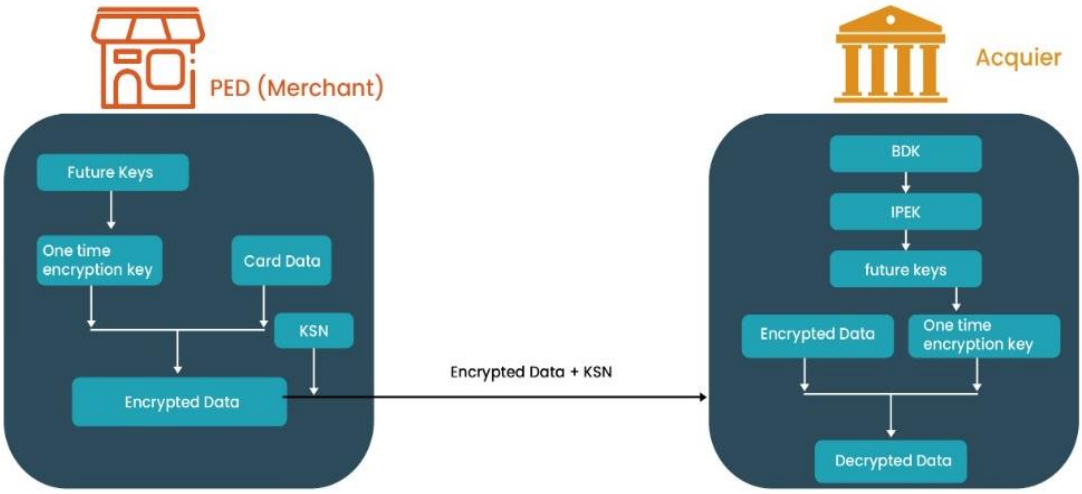


**Figure 3.5** How is your PIN Validated?

The PIN Block would then be passed from the card acquirer to the card issuer so that it could be validated. To do this, PIN Block may pass through several payment institutions (Zones) on the way to its final location (the issuer).

As a result, the acquirer will now exchange keys with the other entities or participants (a payment switch or a payment processor), who will then reply with a key that is [ZPK] encrypted under [ZMK]. The acquirer will be able to extract the clear ZPK and use that to encrypt the PIN Block since it already has the following institution's ZMK inserted in its HSM. The encrypted PIN Block will then be shared with the next institution.

The following institution, in a similar manner, will be able to translate PIN Block using the previously disclosed ZPK and will obtain Clear PIN Block. Repeating the identical steps, this entity will once more extract clear ZPK using the ZMK of the next institution (which has already been inserted into the HSM) and will once more encrypt the PIN Block before passing it on to the next institution (Kumar, 2021).

# 4. SECURITY OF AUTOMATED TELLER MACHINE

ATMs are now vulnerable to both physical (such as gas explosions) and logical threats (malware, software skimming, and black box). For several reasons, they have grown to be a desirable target for cyberattacks[2]. Both the cash in the ATM and the private data (credit/debit cards and PINs), which can also be used to withdraw cash, serve as incentives.

Attacks come in two flavors:

1. ATM Virus Attacks (logical[3])
2. ATM Attacks using a black box (logical/physical)

However, ATMs frequently have vulnerabilities that crooks use for their gain. ATMs are occasionally not well-managed, and little to no reasonable action is done to protect the data they contain.

The vast array of entities engaged in security, such as banking firms, distributors, vendors, manufacturers, etc., is another vulnerability. This could indicate that too many users have administrative access to ATM systems, which could potentially increase the danger of unauthorized access. Furthermore, there is no unified supervision of the actions of these diverse teams involved in ATM maintenance and support, who frequently come from third-party providers. As a result, there is a risk of significant security oversight gaps.

The hardware and software that make up an ATM environment are varied and complicated. Because of this, organizations struggle to implement proactive operating system and software update strategies and to have centralized, complete visibility of their security architecture. Even though banks are expected to comply with PCI requirements, outdated technology and software might lead to non-compliance.

Financial organizations must overcome several obstacles to keep ATMs open 24/7/365 while maintaining the highest level of security. On the one hand, they need to reduce the workload associated with hardware upkeep and software deployment while maintaining visibility and control over changes to software and hardware. On the other hand, it's important to implement security policies, abide by them, and ensure integrated visibility and control of the security situation.

## 4. 1. Why are Cybercrimes so Effective at Targeting Automated Teller Machines?

Cybercriminals have discovered that the ATM network is frequently one of the weakest places in a bank's security infrastructure. This is due in part to the expensive and challenging-to-update old hardware and software in ATM networks. The systems are now in a very risky situation as a result. For instance, despite Microsoft no longer providing support for Windows 7, many ATMs still use Windows 7 or are in the process of

[2] Any harmful activity that aims to gather information or to acquire, interrupt, reject, weaken, or destruct informational system resources.

[3] In logical attacks, the ATM's system is exploited and modified via ransomware or some other electrical equipment known as a "black box".

upgrading. Because the OS no longer receives Microsoft's security updates, there are known flaws that hackers might exploit to launch attacks. According to estimates, approximately 40% of ATMs worldwide are still using outdated operating systems (such as Windows XP-OS), which Microsoft has not supported since 2014. As a result, these machines are much more open to attacks.

The XFS interface, a standardized protocol intended to enable enterprise application software to run on manufacturers' Terminals or other hardware, serves as one of the primary attack mechanism on ATMs. Self-service programs are communicated with by the XFS layer using industry-standard APIs. Criminals have taken advantage of the fact that this gateway lacks an incorporated authentication mechanism to great effect. The XFS layer is a tempting target for attackers because they may use the virus to either collect card details (software skimming) or to "cash out" operations and obtain cash from hardware devices like ATM cash dispensers.

### 4.2. Potential Attacks Techniques

#### 4.2.1. Attacks on Hardware components

**The Fundamental Issue**

ATMs are a collection of various processing and money-handling components. Some of them have something to do with money either directly (like a dispenser that holds money in cassettes) or indirectly (like a Personal Computer (PC) that controls equipment). These gadgets are connected. The equipment within ATM boxes is regarded as reliable, and it is assumed that it cannot be tampered with or replaced with an unreliable one. However, frequently this technique is essential "security via obscurity," and devices lack the necessary safeguards to verify the unit endpoints' legitimacy (e.g. unprotected communication between the ATM core and ATM units.)

**Black Box Attacks**

1. A maliciously constructed device can be connected directly by an attacker to the card reader or cash dispenser. The ATM core (PC) is connected to all gear using serial or USB ports, and in a few unusual instances, SDC-bus. The communication between ATM units and among them must be secure and authenticated. A trusted zone is a network of connected entities. It should also be authenticated when entering a trust zone. Communication becomes insecure if a malicious device enters the trust zone, for example, if it can break encryption and transfer data in plain text without evading security measures. These protection methods are implemented in some ATM models, but banks do not employ them.

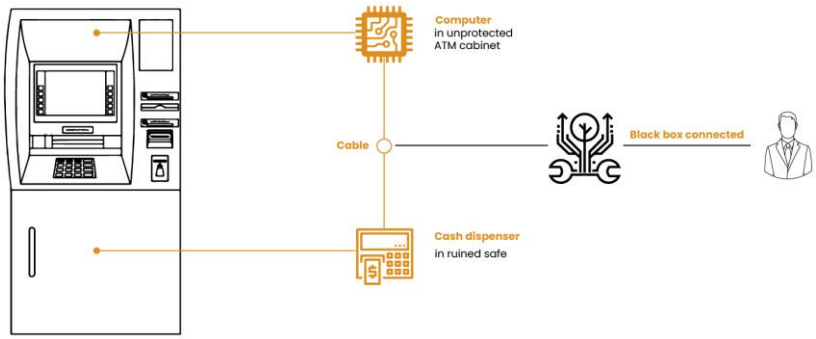


**Figure 4.1.** Hacking an ATM using Black Box

Black box (Krebs, Brian, 2015) assaults are a common way to steal cash, and in the upcoming years, there will be more these. An ATM assault that uses a black box focuses on deliberate ATM hardware tampering, such as the cash dispenser or card scanners. The cash dispenser from the ATM must be detached, and then a black box, an external electronic device, must be linked to it to conduct a this attack. In place of a card or transaction permission, the black box instructs the ATM dispenser to withdraw money.

2. The testing and debugging features that ATM manufacturers install can be left in the operational environment, where an attacker could exploit them to eject money.
   As new devices are developed, they initially have a variety of technological concerns and problems. As a result, manufacturers frequently want technical data about the state of hardware at the moment.
   Testing cash withdrawals are made possible by the development of test or service tools for maintaining ATM hardware units by ATM suppliers or third parties. The use of customized tokens and cassette manipulation protects this activity (service operators must open the safe door, use a safe key, and manipulate cassettes). However, older or altered versions of these test tools can be loaded on a laptop or microcomputer, which can then be used to expel cash by connecting it to the serial bus or the dispenser port. Through a hole cut in the plastic cover of the ATM, the attacker can sometimes connect a specially constructed device to the serial bus via ports (like EPP).
   An ATM's critically important hardware doesn't verify the legitimacy of its networking environment and applications (in this case, the ATM's primary application), black box attacks are usually achievable. According to PTSecurity (ptsecurity, 2018), 69% ATMs are vulnerable to Black Box attacks. Any connected devices will be able to issue commands to the cash dispenser.

### Attacks on Near-Field-Communication Devices

Before they are stable, newly installed NFC[4] readers, biometric data readers, or readers of other types of newly installed hardware in ATMs will need to undergo extensive testing. Attackers will have a great chance to investigate the new device and

[4] A wireless communication known as NFC, or near field communication, enables compatible devices to interact..

determine how to exploit its weaknesses in the future. Tons of ATMs and POS terminals which utilizes the NFC reader chips globally have been found to have vulnerabilities that have been dug up and reported over the past year by a researcher (Greenberg, 2021) and consultant at the security firm IOActive.

In such attacks, information is gathered from firmware which is being obtained from the target device utilizing hardware debugging ports. Such details can then be utilized for attacking targets with inactive debugging ports or even target members of the same device family.

A malicious POS terminal, a smartphone, or another device that has been carefully modified may be able to intercept the unencrypted authentication data supplied by NFC and utilize it for CNP transactions. Card information that has been stolen (by, for instance, skimming) is subsequently sold on dark web forums.

An attacker might, for instance, utilizes a specifically designed NFC sniffer to gather biometrics from a consumer's debit or credit card with an NFC chip. In areas with a lot of people, like the metro, he uses the NFC sniffer. The attacker approaches people closely and reads NFC chip data from cards. Cards lacking NFC are guarded against such attacks by design. If consumers utilize physical protection against wireless communication, an NFC card is protected (e.g. electric shield). The attacker then markets the card information on the dark web.

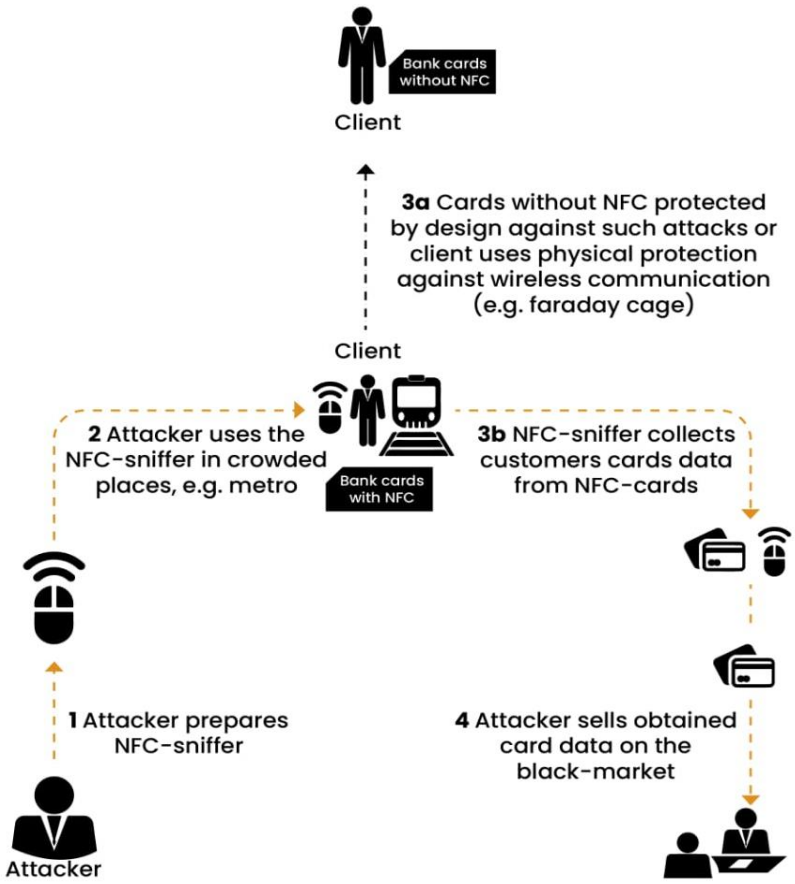


**Figure 4.2.** NFC Sniffer Attack Scenario

**Threats against Biometric Data and Devices**

The data exchanged by Biometric Devices[5] linked through USB or serial interfaces could be intercepted and exploited (Olga Kochetova, 2016). Identity fraud, the trading of biometrics on the illegal market, and the utilization of looted biometrics in certain systems are all problems. It poses a serious threat to cybersecurity:

Customers can alter card information or PINs after it has been compromised, but information used for biometric authentication cannot be modified and cannot be canceled after it has been compromised. A biometric data skimmer can potentially be used by an attacker to acquire consumer authentication information. The attacker chooses the ATM that will be used as a victim and connects a specially made device that has been preconfigured to skim biometric data. After obtaining biometric information (such as voice recordings), the attacker can employ it in a number of ways.

- For approving a different bank service (like online funds transfer);
- For carrying out illegal online transactions;
- For reselling biometric information on the illegal market.

Further attack method involves using a specialized gadget to obtain biometrics from a consumer's EMV-enabled debit/credit card, the attacker prepares a specially made gadget. He physically gains access to the customer's bank card by using social engineering methods (or just steals it). The customer's card data is then obtained by the attacker using a malicious device, however, if the card contains impenetrable security features, the intruder can only wipe off the biometric data from the chip during data retrieval, making it impossible to recover the data. The assailant trades acquired biometrics.

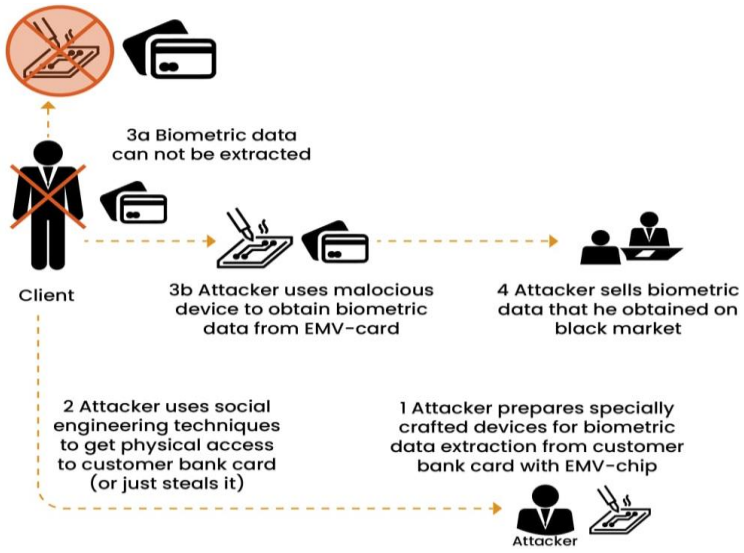


**Figure 4. 3.** Attacking Scenario of "Accessing Biometrics from EMV-Enabled Card using a Special Gear"

[5] A biometric device is built on technology that uses a person's distinct facial features, fingerprints, signatures, DNA, or iris pattern to identify them.

### 4.2.2. Threats against Software Components

#### The Fundamental Issue

Software is the foundation of all information processing. When discussing software, it is important to remember that any issues that could arise from an attack on hardware components could also arise from software-related issues. Not to mention, attacking software is considerably simpler because the attacker can find flaws by simulating all required parts using the software.

#### Malware Attacks

The software could be simple prey for malware due to zero-day vulnerabilities. The most cutting-edge malware attacks right now use a variety of techniques.

**1. Memory Scrappers**

These techniques' key component is memory scraping/searching for private customer data. This data consists of Track2 data, private data, and transaction log, among other things.

Based on information from law enforcement agencies and the targets themselves, a Carbanak attack (Vijayan, 2019) with more than 100 targets might result in total financial losses as high as $1 billion.

**2. Application Programming Interface - Specific Malware**

The common libraries and APIs of ATM vendors are being used by the next generation of malware. Relatively similar libraries used for legitimate communication with an ATM can be abused to gain access to consumer's confidentiality or to interact with hardware to conduct scam. If the software is not created with security in mind, it is secure to assume that an intruder may access an ATM in a similar manner as service technician or authorized representative.

The extensions for Financial Services, often known as XFS, is a specification which provides a server software architecture for transaction systems on the Microsoft Windows platform, specifically for endpoints like ATMs. With the help of a standard API, XFS aims to standardize software so that it may run on any hardware, independent of the maker.

The image below depicts a basic ATM architecture based on XFS.

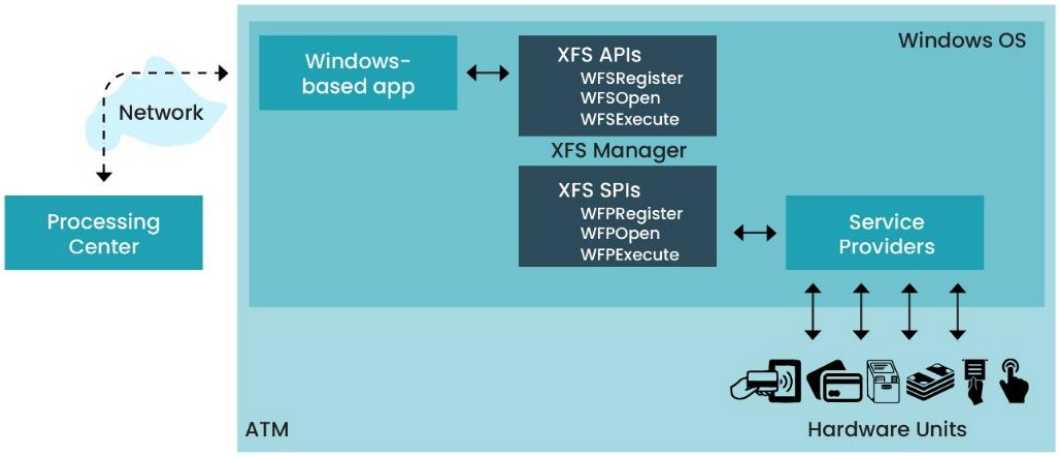


**Figure 4.4.** ATM Infrastructure's Basic Plan Based on Extensions of Financial Service

Since only the logic outlined in this standard can manipulate low-level objects, any application created with the XFS standard in mind can do so. And any malicious program may very well be that application.

The attackers can manipulate thanks to XFS (Kochetova et al., 2018)*:*

1. Cash dispensers - an attacker can open a safe using software, obtain cassette and cash control, and perform cash withdrawals without authorization.
2. Identification card devices - An attacker can use an EMV reader to obtain the payment history saved in the chip, as well as regulate the card insertion, ejection, and keeping procedures.
3. Devices with PIN keypads - An attacker can:
   To intercept the plain text PIN number, switch from secure mode to open mode. The intruder must query the PIN pad just after customer submits their PIN code for the open mode to conduct a MitM attack using a PIN device. The intruder is required to acknowledge when the buttons are pressed, however an erroneous PIN block is sent instead. Although, the server will deny the request, the attacker now has access to the target's PIN code (see pictures below).

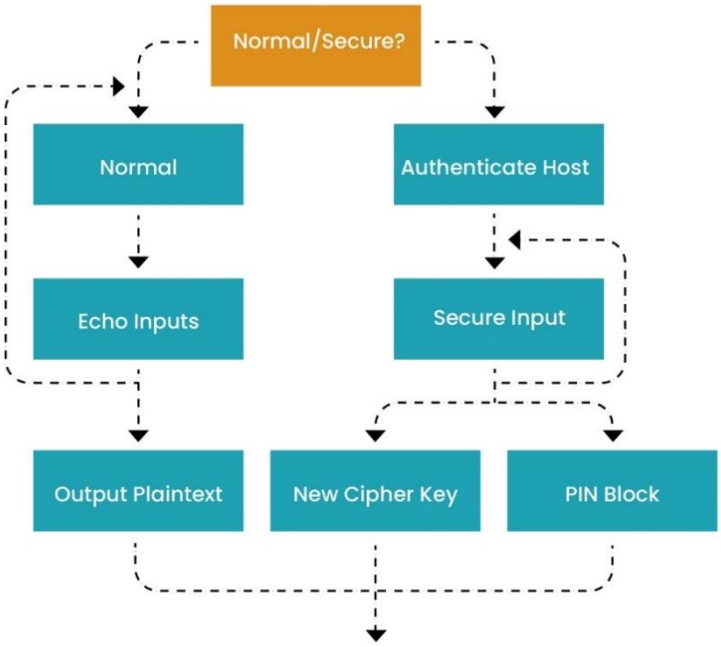


**Figure 4.5.** Functioning of PIN Devices in Open Mode and Secure Mode

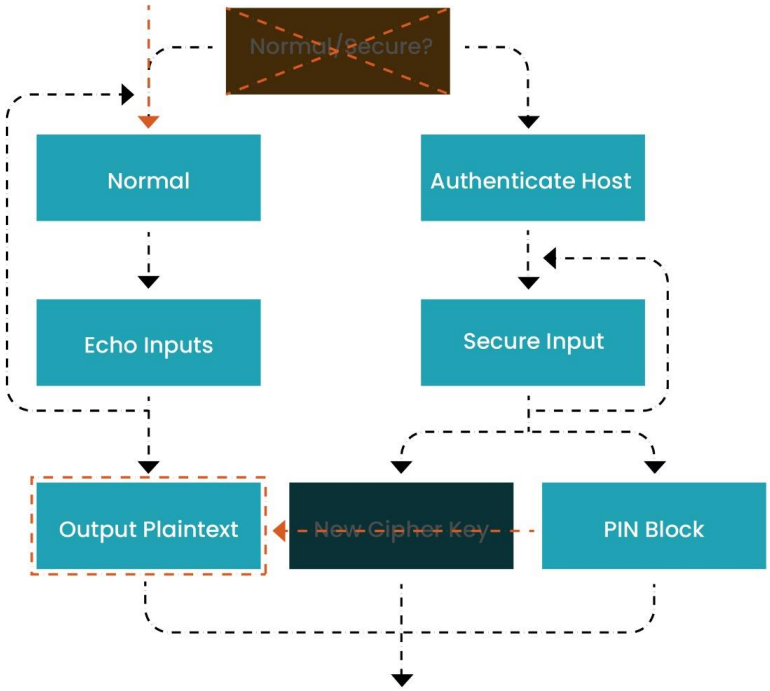


**Figure 4.6.** Sequence of PIN Device Utilization During a MitM Attack

### 3. Skimmer – New Generation Malware

Malware from the most recent generation is more advanced. It makes extensive use of hardware and architectural understanding of ATM. In the course of an investigation

into an incident. According to Kaspersky Lab (Kochetova et al., 2018), on the ATMs of one bank, upgraded Skimmer malware has been discovered.

By obtaining entry to the bank's internal network or through physical access to the ATM system, the Skimmer gang begins its operations. Backdoor. Win32.Skimmer infects an ATM's core, the executable in charge of the device's interfaces with the banking infrastructure, cash handling, and credit card processing, after successfully installing itself on the system.

- Modified Skimmer malware can currently perform the following functions:
- Directly ordering ATMs to disburse cash or masking contacts with fraudulent bank cards using specific information
- Interacting with smart cards, which includes receiving commands from swipe cards, modifying selves in reaction to information on a chip, sniffing confidential material, and launching MitM attacks
- Transmitting to the processing facility, all the assembled data so that it is sent there in readable form, transmitting offensive data to C&C servers or deactivating SSL encryption.

A lot of people received Skimmer between 2010 and 2013. Since its emergence, there have been up to nine separate malware families that have been used in assaults against ATMs. This includes the Tyupkin (GREAT, 2014) family, which was uncovered in March 2014 and quickly rose to fame. 49 variants of this virus have already been found by Kaspersky Lab, 37 of which target ATMs made by just one of the major manufacturers. The beginning of May 2016 saw the discovery of the most recent version.

## 4. Advanced Persistent Threat (APT) Attacks and Implications on Newer Technologies

It was widely reported in the ring in 2016 that the APT group (Zetter, That Insane, $81M Bangladesh Bank Heist? Here's What We Know, 2016) utilized the SWIFT network to steal 81 million US dollars from a Bangladeshi bank.

Attackers remained successful in entering Bangladesh's Central Bank, which is connected to the Federal Reserve Bank of New York through an account. The SWIFT system was utilized to carry out the attack, and it was later discovered that the attackers had employed a unique piece of malware they had developed themselves.

The episode that follows is the ideal illustration of an indirect attack on an ATM. In August 2015, thieves withdrew around half a billion rubles from the ATMs of different banks while using Master debit cards issued by the Russian bank "Kuznetsky." The UCS processing system's faulty configuration, which erroneously manages rolled-back transactions, and noncompliance with the requirements of the international payments systems were leveraged by the fraudsters.

This incident shows how interbank exchange system flaws and vulnerabilities can be used by attackers to compromise banks and their constituent parts. Although this kind of fraud chain is challenging to set up, if successful, attackers may be able to compromise dozens of ATMs and other parts of the banking infrastructure.

Carbanak is yet another noteworthy instance of an APT-style campaign that targeted financial organizations, but not exclusively. Attackers using Carbanak send phishing emails with CPL attachments. The administrative computer has the backdoor installed after a shellcode is run, and once it has entree, it "jumps" through the network till it locates a target—the ATMs.

ATMs could be instructed to remotely withdraw cash with no interaction with the Terminal itself while the cash is being picked up by straw men. By this technique, funds were transferred from the company to the criminals' accounts using the SWIFT network. Additionally, databases containing account information were changed to enable the creation of phoney accounts with a sizable balance, which was then used to collect funds through the use of straw men services.

Interception of card data in ATMs is another type of malware attack.

### 4.2.3. Attacks on the Network Layer

#### The Fundamental Issue

Due to the requirement that all ATMs be connected to financial systems in order to provide services, ATMs may be thought of as a point of entry for intruders to assault an ATM network or even an ATM processing facility. An attacker may use such attacks as a springboard to use other financial apps.

#### Man-in-the-Middle Attacks

Attackers utilize the ATM's security or network faults, or its externally accessible weaknesses, as a source of gateway for attack to carry out this attacks.

The connection layer may be compromised by an attacker in several ways:

1. **Lack of Network Segregation between Automated Teller Machines**

Additional ATMs that connect with the hacked ATM can be accessed by an attacker. Once they have it under their control, the attacker can then make cash withdrawals from all compromised ATMs (ptsecurity, 2018).

2. **Lack of Network Protection between the Automated Teller Machine and the Processing Center**

If the transmission link in both the ATM and the processing center is exposed and the server's handling is vulnerable, an unregistered user may hold entree to both a specific ATM as well as the processing center and other banks' services. (ptsecurity, 2018)

3. **Lack of Network Segregation between an Automated Teller Machine and other parts of the Bank’s Internal Network**

Because of network misconfigurations and segregation problems, an attacker may be able to access not just the processing center, the ATM administrator host, or even deeper - the hosts of bank offices or other banks' authentication systems.

ATMs communicate with other banking components over an insecure network, which makes it possible for the data being transferred can be intercepted and altered by an attacker This information might also include the particulars of each client's authentication. An attacker can use software interception if they have control of an ATM, and if the ATM's enclosure is not reinforced, they can also physically intercept the machines (Kochetova et al., 2018).

### 4.2.4. Attacks on the Security Application Programming Interfaces

#### The Fundamental Issue

The security APIs are referred to be a collection of two-party security protocols, individually consisting of a user info and an HSM reply. To carry out an assault, the attacker is free to create these protocols in any way he pleases. These APIs have been the target of multiple similar attacks in recent years. Some of these are key-management system attacks that can be found using methods akin to ones used for conventional security protocol analysis. Others are so-called "PIN Block Attacks," weaknesses in PIN processing. These assaults entail the assailant speculating roughly potential PIN figures hidden within an EPB. Assailants understanding of the PIN is influenced by the HSM's replies to different requests (Zetter, PIN Crackers Nab Holy Grail of Bank Card Security, 2009).

#### Attacks on Verification Application Programming Interface

The verification API attacks can be categorized into two types

**1. Decimalization Table Attacks**

Numerous PIN systems allocate preliminary client PIN numbers since an IBM proposal (the so-called "3624 scheme") by using a confidential PDK to encrypt a customer's PAN and decimalizing the outcome consuming a decimalization table. Every base 16 number is converted to a base 10 numbes using a decimalization table. This is how the "standard" decimalization table looks:

**Table 4. 1.** Standard Decimalization Table

| Hex Value | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Decimal Value | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 |

Therefore, if a customer's PAN under the PDK yields the first four digits in hexadecimal format 4A6B, the allocated PIN will be 4061. The verify function accepts the decimalization table as an argument since different financial firms employ various decimalization schemes. The customer may be able to update her PIN at an ATM under

some schemes. It is accomplished by fixing an offset, that when combined with the initial PIN digit-wise modulo 10, yields the customer's preferred PIN. Since it cannot be used to determine the proper client PIN without the original PIN, this offset is not regarded as being security-critical.

Attacks on decimalization tables (Bond & Zielinski, 2003) instead of figuring out the PIN's numbers, figure out the PIN's component digits first, later their relative placements. If an intruder possesses PIN that is encoded which, when paired with the common decimalization table and a specified offset, correctly validates inside the HSM, the HSM will state the PIN is accurate as such when the PIN Verify (.) function is used. Let's say the attacker changes the decimalization table as follows:

**Table 4. 2.** Altered Decimalization Table

| Old Value | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| New Value | 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 1 | 2 | 3 | 4 | 5 |

He then uses the updated dectab to make another call to PIN Verify (.) He will be able to tell there are no 0s in the PIN if the verification still succeeds. But if the verification now fails, he is aware that the PIN must have at least one 0 someplace. Finding out how many there are and where they are, is the current issue. The offset can be changed to achieve this. As the PIN is reported as right again, at each point, the attacker advances the offset by one. This displays where the PIN's 0s are located. The procedure is shown in the table below for a case in which the customer's PIN is 3060 and the offset is 0000.

**Table 4. 3.** Decimalization Attack

| Attacker set offset | Result from HSM | Knowledge of PIN |
|---|---|---|
| 0001 | Incorrect PIN | ???? |
| 0010 | Incorrect PIN | ???? |
| 0100 | Incorrect PIN | ???? |
| 1000 | Incorrect PIN | ???? |
| 0011 | Incorrect PIN | ???? |
| 0101 | Correct PIN | 0?0? |

A typical approximate of 16 API requests are required by the decimalization table attack to find a 4 number PIN.

**2. Check Value Attack**

Clulow (Clulow & Clulow, 2003) has provided details of an attack that leverages the "check value command" against the PIN verification API. One can employ this type of command to determine whether a key has been imported and is supported by many APIs. The command returns a 64-bit block of zeroes that has been encrypted with the provided key.

To execute the attack, the attacker might additionally be able to transmit a block of zeros as validation data to a command for verifying PINs similar to the one discussed above. Getting the PDK check value is the first step, after which the result's first four characters are decimalized using the conventional decimalization scheme. As an IPIN, save this.

Give the function PIN Verify (.) a PAN of 000000000000 and the EPB you need to decrypt now. Beginning at offset 0000 usually, the command will initially be unsuccessful. Increase the offset by 1 until the command indicates a successful verification. Assign the last offset the name OFFSET.

Now that we are aware that the PIN in the block properly checks when compared to IPIN + OFFSET (mod 10 for each digit), we can conclude that this PIN belongs to the client.

This attack would typically take one check value command call and 5000 PIN verify function calls for a four-digit PIN.

**Threats against the Translate Application Programming Interface**

**1. The codebook attack**

The translation API is used in this attack to remove variation data from the EPBs (Focardi et al., 2009). This equalizes EPBs with the same PIN, allowing common codebook attacks. Be aware that equal PINs typically result in distinct EPBs if EPBs contain arbitrary filling, as they do in formats 0 and 1 of ISO, or this before the PIN is encoded, the PAN is appended.

Switches and verification facilities are also vulnerable to assault.

The attacker starts by producing all EPBs in any ISO format every PIN that can have a set length. For instance, a group of intruders may attempt each of the 10,000 PINs at various terminals while intercepting the relevant EPB at the compromised switch.

After obtaining all 10,000 EPBs, the attackers convert them all into ISO’s format ‘0’ using a predetermined fixed PAN, such as PANA, at the switch where they were intercepted. Afterwards, they maintain a record with all of the EPB's and their associated

PINs. Real EPBs are translated into an ISO's format '1' upon arrival (the PAN is removed), and then reformatted into an ISO's format '0' message together with the selected PANA. By using the look-up table to search for this new encryption, the genuine PIN can now be found.

**2. Wrong-format attack**

This attack (Focardi et al., 2009) entails converting an encrypted message that has been received into an ISO's format 0 with various PANs to gather data based on the API's return value. To recover the PIN, the attacker specifically offers various PANs (PANAi), which are XOR-ED with the encrypted value. The API generates an error code when a non-decimal value is produced by such an XOR operation. This provides details on the actual PIN digits. In reality, by using this approach 16(n - 2) times, it is feasible to recover the (n - 2) rightmost PIN digits up to their parity for an n-digit PIN. For a four-digit PIN, for instance, we might find that the third and fourth digits are, respectively, 3 or 4 and 7 or 8.

Because the first two digits don't XOR-ED with the PAN, they cannot be recovered.

The attacker first translates a message into an ISO's format '0' with a PAN of all 0s (where the PIN starts from the third position), then asserts that the message is in VISA3 format., and requests to translate it into ISO's format '0' with a PAN of all 0s. This method can also recover the first two digits of the PIN. The operation's outcome is a two-position shift to the right of the PIN, allowing the attacker to use the prior attack on the complete PIN.

Four more calls can finally help you choose between the two values: Assuming there are 24 = 16 potential PINs and the pairs of values are {P1, P1 + 1}, {P2, P2 + 1}, {P3, P3 + 1}, and {P4, P4 + 1} the attacker can now send a message in ISO's format '0' posing as a VISA3 and including PAN E100000000000, where E1 = E $\oplus$ P1. The correct PIN digit value is P1, otherwise, it is P1 + 1 if the HSM displays a mistake.

The supplied PAN is XORED to the PIN, and if the PIN digit value is P1, we get P1 $\oplus$ E1 = P1 $\oplus$ E $\oplus$ P1 = E, which is non-decimal and results in an error (Focardi et al., 2009).

**Attacks on Functions allowing Personal Identification Number change**

The IBM 3624 (IBM, 2021) or the VISA PIN validation value (PVV) methods are the two acceptable PIN verification techniques used in the Financial PIN Processing API (Berkman & Ostrovsky, 2007).
The verify function's input for both approaches is as follows:

- An EPB containing the customer-provided PIN
- The account number for the client.
- A four-digit customer verification value with decimal places (called offset in the first method and PVV in the second).

The verification value for the customers is not secret. It is either stored on the customer's card or in a database together with the customer's account number as part of the customer's details.

Indicate the customer's verification value by V, the customer's account number by A, and the PIN stored in the EPB by P.

The EPB is decoded by the verify function, confirms its authenticity by examining the PIN block format, extracts P from the EPB, and then checks to see if V = f(P, A), where f is a function that depends on the issuer's secret key. The two approaches use various versions of the function f.

In order to facilitate customers to select their PIN online, the Banking Transaction API contains a pair of functions (one for each mechanism) that enable recalculating the customer's verification value once the customer's PIN updates. Calculate Offset and Calculate PVV are the names of the functions. The following input is provided to both functions:

- The consumer's chosen PIN's encrypted pin block
- The consumer's IBAN.
- The functions give back V = *f* (P, A), with P, A, V, and *f* remaining unchanged.

We observe that the use of the random issuer's key in f causes the value V in both fu nctions to be pseudo-random.

Regardless of *f*, the primary flaw with both functions is that the new PIN that is sent to them (packaged in an EPB) is not linked to the previous PIN.

Note that this binding must be checked by the function itself and not by the site's application because an attacker can carry out the attack by directly accessing the API. Also keep in mind that adding a parameter with an EPB that contains the customer's old PIN (and a way to verify it, such as the appropriate PVV), would not be sufficient because an attacker could capture the customer's actual EPB as it was being sent for verification and use it as the additional parameter.

Attacks on the calculate PVV function are discussed below. These attacks don't make use of any of the corresponding components (except for assuming that the value V is pseudo-random). As a result, all of our critiques of calculate PVV also apply to compute offset. We also go into particular attacks on compute offset. These assaults are more severe than general attacks because they make use of specific *f* features.

We must transmit an EPB and an account number to the calculate PVV to attack it. The account number would always belong to a different client than the one who created the EPB in every situation.

Before passing the EPB to the compute PVV function, we utilize the translate function to reformat it to ISO-1 (which is unrelated to any customer) to force the function to accept the non-matching parameters. There would be no conflict between the EPB parameter and the IBAN parameter because the IBAN is not a requirement for ISO-1. Noting that we can reformat the EPB to that format, limiting the compute PVV function to only accept

EPBs in that format would not prevent the attacks. We also point out that even if the PIN block reformatting functionality is deactivated, the attacks can still be used (although in a more limited manner).

We utilize the abbreviations V = PVV (E, A) and O = offset (E, A) when assaulting the calculate PVV function (and, conversely, calculate offset function) with an EPB indicated E and an account number A. We won't talk about reformatting any longer because all EPBs are converted to ISO-1 before utilizing the function.

**Attacks on the calculate PIN Verification Value function**

**1. Attacking the calculate PIN Verification Value Function in a Switch**

Consider each consumer EPB that arrives at the switch under attack. The attack (Bond M. K., 2004) finds a list of other EPBs with the same PIN for each such EPB (and its associated account number) (with high probability). Per attacked EPB, one or two HSM calls are necessary. We observe that the attack is equally applicable to facilities for verification.

We employ a 10,000-entry table. The values of computed P V Vs are used to index the table. The table's entries each have consumer EPBs (and their associated account numbers). All entries are initially empty.

We select B as the fixed account number. We demonstrate how to counter any consumer's EPB that arrives at the switch. Indicate the consumer's EPB with $E_c$.

1. V = PVV ($E_c$, B).

The PIN stored in the EPB of the consumer is $P_c$, and the PVV value calculated as V is equal to f ($P_c$, B).

2. Update the table entry that corresponds to the generated PVV value V with the consumer's EPB Ec.

For instance, if the value of $P_c$ is 1234 and the value of V is 5678, $E_c$ will be added to table entry 5678.

The only variables that affect the computed PVV value V are Pc, B, and the key k that the function f uses. k is not the issuer's key as necessary because the attack is conducted in a switch; instead, it might be any other random number (not known to the attacker).

Since all that is necessary for the attack is that the value V be a pseudo-random function of Pc, B, and k, which is the case based on our assumption on f, the value of k is irrelevant to the attack. V can be viewed as a Pc-only pseudo-random function because B and k are fixed.

Imagine we've used numerous EPBs to complete the aforementioned task. In reality, in stages 1 and 2, all EPBs that contain the same PIN value are placed into a single table

entry. A column entry might be void, include EPBs for more than one PIN, or contain nothing at all due to the random nature of the procedure.

Combinatorically, the procedure is analogous to tossing balls (PINs) into bins (table entries) and counting the balls (individual PINs) in each bin before asking questions about the results. It can be demonstrated that when the quantity of balls and bins is equal (10,000 in our case), there are typically fewer than 2 balls in a completely void bin. To put it another way, on average, EPBs that resulted in the same table entry correlate to less than two different PINs.

We are not finished yet because there is still a significant likelihood that EPBs in a single table item match several PINs. Using a different fixed account number C, we repeat the process for each element in the table concerning the EPBs in that entry. There is a considerable likelihood that EPBs from one table entry that finishes in another share the same PIN.

## 2. Attacking the calculate PIN Verification Value Function in a Verification Facility

This attack (Berkman & Ostrovsky, 2007) exposes the PVV that corresponds to the consumer's account number and a predetermined PIN for any account number linked to the attacked issuer. Using the specified PIN, a consumer's account can be withdrawn by changing the verification value on the card or in the database, depending on the system.

One HSM call is necessary for each account number targeted by the attack. Additionally, an EPB with a known PIN must be generated from an ATM. Attacks on all consumers' account numbers will be made using this one EPB.

We begin by creating an EPB in an ATM that contains a predetermined PIN. Attackers use this EPB, abbreviated Ea (for attacker's EPB), to target all consumers' account numbers. Calculate V = PVV for each consumer's account number Ac (Ea, Ac).

The consumer's account number and the PIN they have selected are represented by the PVV value calculated as V. The PVV is legitimate because the required issuer's key is utilized and the attack occurs in the verification facility. It is yet unclear how the attacker can use the PVV computed during the attack to replace the customer's original PVV used by the system.

The magnetic stripe of the card or a PVV database can both hold the clear PVV. When the PVV is kept on both, the database is used to retrieve it. As long as the consumer utilizes the initial PIN created by the issuer, the PVV is frequently only stored on the card in many implementations.

You can do the following to set the consumer's PVV to the computed PVV:

**Case 1:** The card is the sole place the PVV is kept. Create a card with the consumer's information on it, and change the PVV value on the magnetic stripe to the computed PVV. The consumer's true card and the fake card will both be valid in this situation.

It's vital to understand that in this situation, giving consumer a new PIN won't stop the attack because the fake card with the fake PVV would still be valid.

**Case 2:** This consumer's PVV entry is present in the PVV database. The attacker in this instance requires to write access to the PVV database. After that, the attacker has a few options:

- Remove the PVV entry (and then apply the steps described in Case 1).
- Update the consumer's entry in the PVV database with the attacker's calculated PVV. Create the entry if it doesn't already exist. The only card that will be accepted in this scenario is the fake one.

### 4.3. Reducing the Risk of Automated Teller Machine Attacks

It is advised to use a layered strategy to safeguard ATMs from malicious software and logical assaults. These layers work together to significantly lower the possibility of malware assaults on an ATM when integrated. The following are the defenses:

1. Physical access to the ATM

2. Offline protection

3. Online protection

4. Additional measures

#### 4.3.1. Physical Access to the Automated Teller Machine

Work on an ATM should only be done by authorized staff. More specifically

1. A method for ATM site staff to verify their authorization to work on the ATM should exist, and authorized service providers must have accreditation documents.

2. The PC is often located in the top box or compartment of an ATM. The access lock to the top box should be altered to prevent the use of the factory-supplied default master keys, or this location should be guarded with an intruder alarm to prevent unauthorized opening.

3. Cameras for surveillance monitoring should be installed so they can monitor the area around the ATM and capture any suspicious activity. If surveillance monitoring is employed, camera/video images should be kept somewhere than on the ATM, and a restart of the ATM shouldn't stop the cameras from working.

4. The area inside and surrounding the ATM should be well-lit.

#### 4.3.2. Offline Protection

It is possible to carry out logical assaults without utilizing the ATM operating system. Therefore:

**Basic Input / Output System Configuration**

Editing BIOS configuration should be password-protected, with the latter password being different from the vendor-provided default password.

1. Take into account strict password management guidelines. According to best practices, these passwords ought to be as challenging as the BIOS will allow.

2. Set the ATM hard disc as the single boot device in the BIOS.

3. Booting from removable media should be disabled by default.

4. Use a strong administrator password for your operating system.

5. Verify that AUTORUN has been completely and properly turned off.

**Hard Disk Encryption**

To prevent unauthorized alterations to the content of the hard drive, hard disc encryption should be used.

**Cash Dispenser Communications**

It is important to safeguard and secure the USB/serial communication between the dispenser and the PC against message tampering and injection.

1. The initial communication should involve authentication at the cash dispenser to prevent unauthorized devices from sending orders to the cash dispenser. Using the safe physically, for instance.

2. The protection of the communication shouldn't be circumvented, for example, by rolling back firmware or replaying messages.

**4.3.3. Online protection**

**Network**

All ATM network traffic should be protected with communication authentication and encryption. It is advised to implement MAC-ing and utilize TLS 1.2 or a VPN to provide cryptographic authentication of sensitive messages.

**Firewall**

It’s necessary to set up a firewall to censor all incoming communication to the ATM.

**Operating System**

To stop privilege abuse, the use of ransomware installation, welsh funds, and illegal utilization of resources like USB connectors, CD/DVD/tape drives, and an

operating system should be "hardened" or "parameterized." Consequently, the following should be used.

1. OS must strictly enforce application separation. For instance, runtime, service, and administrative unapproved usage of multiple services (OS, Platform, including XFS, and Applications) must always be avoided.

2. Applications and services that aren't in use should be deleted.

3. Create a procedure for safe software upgrades.

4. Make sure that the application is running in a restricted account with only the minimal privileges necessary, not root or administrator.

**Anti-Malware and Logical Protection**

It is best to use an anti-malware program designed specifically for ATMs and a logical approach based on "whitelisting" or "sandboxing" concepts.

**Universal Serial Bus Protection**

Unknown USB devices should not be allowed to be used.

### 4.3.4. Additional Measures

Additional measures can be implemented as follows:

**Automated Teller Machine Installation**

It is advised to do an antivirus scan before the authorized installation at the ATM or to begin with a clean install.

**Secure Software Delivery**

A procedure for safe and frequent software updates for all programs installed on the ATM should be devised.

**Fraud Monitoring**

1. Install a quick, real-time fraud detection system.

2. To prevent fraud, make sure your fraud detection system can spot suspect patterns of behavior.

**Automated Teller Machine Monitoring**

1. Ensure that there is efficient ATM monitoring in place and that every ATM opening is acknowledged centrally.

2. Security solution alerts should be watched over and responded to.

**Cash Refilling Cycles**

Enough cash should be inserted into the ATM to last for a shorter time.

**Test Vulnerability**

Conduct routine ethical hacking tests and vulnerability scans on the ATM and the ATM's network, including testing the availability of wireless access points.

**Look for Abnormalities**

Employees may randomly check ATMs for anomalies during maintenance or cash replenishment, on the ATM's facade or even inside its operational space.

**Segregation of Duties**

No single employee shall have unrestricted access to the ATM.

**Host Integrity Check**

It is advised to use the ATM to demonstrate to the host that the ATM security can return the same hash through the ATM's SW to the host (NCR Corporation, 2018).

# 5. DISCUSSION

Technology has advanced more quickly as a result of the COVID-19 epidemic, and we are all drawn to anything that allows us remote access. This includes a shift to digital transactions, which fraudsters are aggressively attempting to take advantage of.

With the widespread adoption of digital services, the significance of banking cybersecurity has increased. It is imperative to cover all bases and be able to protect customers in advance, regardless of the channel they choose. To stop and mitigate attacks, new procedures and barriers are required.

Therefore, it's crucial to consider banking and cybersecurity from several perspectives.

Due to its complexity and frequent use of out-of-date, unpatched operating systems, the ATM ecosystem is exposed. Because ATMs are built of complex hardware and software, updating them can be challenging from a systems and financial standpoint.

Another issue is that clients must have access to ATMs around-the-clock, thus downtime must be kept to a minimum. Finding the ideal moment to carry out testing and upgrading is impacted by this. The temptation to put off modifications or upgrades is strong because of the delicate balance between upgrades and availability. As a result, banks struggle to obtain an accurate picture of their total risk in terms of both potential surface attack regions and specific sites of vulnerability. ATM network enhancements also fall behind.

As a result, the management and upkeep of these machines are frequently dispersed. A lot of people have legitimate access to the systems and hardware, which raises the possibility of an attack. And the problem with defending against any assault is that an upgrade—which is incredibly expensive to deploy—is then required. Attackers have discovered ways to get around enhancements that vendors have put in place to address this.

Looking at cybersecurity solutions that are specifically designed to lessen the dangers of cyber-attacks linked with them can help solve all of these problems.

## 5.1. Automated Teller Machine Attacks Prevention

There are numerous ways to safeguard an ATM, however they often only offer protection from a single attack vector. The following techniques can be used to stop various ATM assaults.

**Automated Teller Machine Jackpotting:** End-to-end encryption is the most effective technique to reduce the risk of this form of attack. Basically, encrypting data on the HD, interactions among the banking system and the ATM, and linking among the ATM’s microcomputer and cash dispenser. The attack surface will be significantly reduced as a result of this, as well as with excellent networking security controls, and the endpoint least privileged methods.

**Card Skimming:** It can be effectively prevented by implementing extensive surveillance and tamper-evident technologies. For instance, ATM industrialists are adopting ways to efficiently restrict the ability of card skimmers to access consumer's data with a complex surveillance mechanism, enabling alarms and warnings to be sent to ATM owners with the ability to make the ATM unserviceable right away if notice any uncertainty that ATM attack is occurring.

**Network based Attacks:** Similar to attacks on other types of infrastructure, network-based assaults against ATMs should be secured by the following measures:

**Credentials Fortification**: To prevent illegal use of privileged accounts, access credentials should be stored securely, place access restrictions on them, and rotate them on a regular basis.

**Safeguarding Transactions**: To prevent malware from spreading from the network to target assets, keep sessions apart to prevent them from being attacked, from influencing or gaining access to the administrator's endpoint session that is operating on the system or vice versa.

**Mandate Terminal Security and Privileged User:** Reduce the attack surface by enforcing least privilege and terminal protection when blacklisting/whitelisting operations on ATM infrastructure.

**Ongoing Observation:** Follow up on any events or patterns of events that deviate significantly from the generated baselines for each network. Organizations must be able to promptly identify and rectify harmful behavior if, however a potential attacker succeeds in stealing authorization parameters and accessing intended resources, such as POS terminals.

**Man in the Middle Attack:** Ensuring that the financial system and the ATM can communicate with one another in safe manner and adheres to rigorous security guidelines is the best way to prevent MitM attacks.

**Memory Scrappers:** If the PCI rules were changed to require businesses to encrypt card data at the keypad, as they already are obligated to do, Memory scrapers might become obsolete.

**Logical Attacks:** Financial institutions and independent ATM operators can take some precautions to better secure their networks and make them more resistant to cyberattacks.

**Basic Input / Output System Security:** Operators should set the BIOS to exclusively boot from the primary hard disc during typical operations. Prior to deployment, the BIOS upgrades must be examined and tested. The BIOS settings must be password-protected when being edited.

**Create Access Controls Protecting all Credentials:** There must be a secure user account and password policy included with every ATM implementation. All default passwords must be changed, all user accounts and passwords for every ATM machine

must be unique, and passwords must be changed at least every 90 days. They must also include a minimum of one number and must be least 14 characters long, upper- and lowercase letters, and non-alphanumeric characters. These are the minimum password policy standards that must be adopted.

**Encipher Conversations among all Channels:** All networks must encrypt the transmission of sensitive cardholder data so that even if hackers are able to observe the data in transit, they cannot read it. Furthermore, this regulation is legally required by PCI DSS ((PCI), Payment Card Industry, 2021) Requirement 4.1, which specifies the use of robust cryptography and security procedures to protect the transmission of sensitive cardholder data.

**Install and Properly Configure a Firewall:** Only known, permitted communications that are required for the ATM environment must be allowed via the ATM firewall, and connections must be defined per program rather than per port.

ATM security is a complicated issue that requires multifaceted solutions. Only ATM makers or vendors can address many issues. There are numerous safeguards that banks should employ. Some can be lessened by regular ATM users. The lists below offer suggestions and potential defenses against assaults against ATM components (i.e. hardware, software, and network). These defenses can successfully reduce the likelihood of successful assaults and, consequently, fraud losses.

- Carry out routine ATM security evaluations
- Conduct routine visual inspections of ATMs;
- Keep an eye on the situation on the illicit market (e.g., via Threat Intelligence reports);
- Transient data encryption
- Utilize anti-skimming tools
- Implement integrity control and cryptographic protection for all hardware and PCs inside ATMs to ensure data transmission.
- Implement software whitelisting on ATMs
- Put software integrity check techniques in place
- Utilize robust encryption techniques for data storage Discard unused services and applications
- Implement PCI DSS-compliant firewall security. Only a small number of internal hosts and protocols must be permitted for all incoming and outgoing network connections for ATMs. Limit Internet-based direct access to the ATMs. Make sure the data link and network layer protocols used by the ATM network are secure from man-in-the-middle attacks.
- Authenticated dispensing should be used. Network connections between ATMs and the processing center, as well as interactions for both the ATM core and ATM units, both require encryption. It is necessary to confirm the veracity and integrity of these exchanges.
- Remove network configuration errors and security holes
- Utilize network access control methods (e.g. 802.1x)
- Using antivirus software and firewalls will help you prevent network threats.

# 6. CONCLUSION

This thesis covers the study of the Cybersecurity of an ATM. In this thesis, several risks that ATMs confront have been examined thoroughly. Additionally, certain countermeasures that can assist reduce these hazards have been recommended. ATM operators must assure that the area around the ATM is protected and they implement the suggested countermeasures for the ATM consumers. To fully benefit from the PCI DSS, ATM owners should adhere to all of the standard's recommended practices.

# REFERENCES

## Corporate Publications

Council, PCI Security Standards. (2021). *PIN Security–Requirements and Testing Procedures.* PCI Security Standards Council.

IBM. (2021). 3624 PIN Verification Algorithm. IBM.

IBM. (2021). Using Personal Identification Numbers (PINs) for personal authentication. IBM.

Information Technology Laboratory, National Institute of Standards and Technology (NIST). (2015, August). Secure Hash Standard (SHS). Gaithersburg, Maryland: FEDERAL INFORMATION PROCESSING STANDARDS.

ISO/TC, Technical Committee. (2017). Personal Identification Number (PIN) management and security — Part 1: Basic principles and requirements for PINs in card-based systems. The International Organization for Standardization.

Kaspersky. (February 16, 2015). *The Great Bank Robbery: Carbanak cybergang steals $1bn from 100 financial institutions worldwide.*

NCR Corporation. (2018). Guidance and recommendations regarding logical attacks on ATMs. European Law Enforcement Agency.

(NIST), National Institute of Standards and Technology. (1999). Announcing the DATA ENCRYPTION STANDARD. Federal Information Processing Standards Publications (FIPS PUBS).

(NIST), National Institute of Standards and Technology. (2001). Announcing the ADVANCED ENCRYPTION STANDARD (AES). Federal Information Processing Standards Publication 197.

(PCI), Payment Card Industry. (2021). PTS HSM Modular Security Requirements. PCI Security Standards Council.

ptsecurity. (2018). *ATM logic attacks: scenarios, 2018.* Positive Technologies.

X9.24-1, ANSI. (2009). Retail Financial Services Symmetric Key management, Part 1: Using Symmetric Techniques. ANSI X9.24-1.

## Single Author Articles

Barker, E. (2016, January). Recommendation Key Management, Part 1: General. *NIST Special Publication 800-57 Part 1*. National Institute of Standards and Technology (NIST).

Konheim, A. G. (2016). Automated teller machines: their history and authentication protocols. *Journal of Cryptographic Engineering*.

Olga Kochetova, A. O. (2016). *Future attack scenarios against ATM authentication systems.* Secure List.

Sholes, D. (2002). Triple DES and Encrypting PIN Pad Technology on Triton ATMs. Triton Systems of Delaware.

Steel, G. (2011). Formal Analysis of Security APIs. *Encyclopedia of Cryptography and Security.* Boston: Springer.

Thomas, A. (2020). Logical and Physical attacks on ATM Machines. Medium.com.

Tushie, D. (2015). *Pin Block Formats.* Retrieved 11 09, 2022, from http://icma.com/wp-content/uploads/2015/07/PinBlockFormats_SE1-15CM.pdf

Umawing, J. (2019). Everything you need to know about ATM attacks and fraud: Part 1. Malwarebytes Labs.

**Two Author Articles**

Barker, E. B., & Mouha, N. W. (2017). Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher. National Institute of Standards and Technology Special Publication.

Berkman, O., & Ostrovsky, O. M. (2007). The unbearable lightness of PIN cracking.

Bond, M., & Zielinski, P. (2003). *Decimalisation table attacks for PIN.* University of Cambridge Computer Laboratory.

Clulow, J., & Clulow, J. (2003). The Design and Analysis of Cryptographic Application Programming Interfaces for Security Devices.

Gorski, P., & Iacono, L. (2016). Towards the Usability Evaluation of Security APIs. In S. F. Nathan Clarke, *Proceedings of the Tenth International Symposium on Human Aspects of Information Security & Assurance (HAISA 2016)* (p. 314). Frankfurt: Lulu.com.

Nema, P., & M.A.Rizvi. (2015). Critical Analysis of Various Symmetric Key Cryptographic Algorithms. *International Journal on Recent and Innovation Trends in Computing and Communication, 3*(6).

**Articles with Three or more Authors**

Aggarwal, K., Saini, J. K., & Verma, H. K. (2013). Performance Evaluation of RC6, Blowfish, DES, IDEA, CAST-128 Block Ciphers. *International Journal of Computer Applications*.

Focardi, R., Luccio, F. L., & Steel, G. (2009). Blunting Differential Attacks. *In Proceedings of 14 th Nordic Conference on Secure IT Systems (NordSec'09).*

Ghafari, Z., Abazari, F., & Analoui, M. (2014). ATMSEC: A Secure Protocol for ATM . *Second Computer Science Conference on Computer and Information Technology.* Tabriz.

Kochetova, O., Osipov, A., & Novikova, Y. (2018). *Future Attack Scenarios Against Authentication Systems, Communicating With ATMS.* Kaspersky Lab.

Mushtaq, M. F., Jamel, S., Disina, A. H., Zahraddeen A. Pindar, N. S., & Deris, M. M. (2017). A Survey on the Cryptographic Encryption. *(IJACSA) International Journal of Advanced Computer Science and Applications,, 8*.

Wang, Y., Zhang, Y., Sheu, P. C.-Y., Li, X., & Guo, H. (2010). The Formal Design Model of an Automatic Teller Machine (ATM). *International Journal of Software Science and Computational Intelligence, 2*, 102-131.

**Book Section**

Mavrovouniotis, S., & Ganley, M. (2014). Hardware Security Modules. In *Secure Smart Embedded Devices, Platforms and Applications.* New York: Springer.

**Thesis**

Bond, M. K. (2004). *Understanding Security APIs.* University of Cambridge.

**Website Resources**

Auriga. (2020). Cybersecurity For The Next Generation Of Branch Banking.

Ayres, C. (2008). Hackers crack cash machine PIN codes to steal millions. Los Angeles: The Times.

Bowen, J. (2000, April 1). *How ATMs Work.* Retrieved 11 9, 2022, from https://money.howstuffworks.com/personal-finance/banking/atm.htm

Burroughs. (2018). ATM security and fraud. Burroughs.

Citizendium. (n.d.). *Code book attack.* Retrieved from Citizendium: https://citizendium.org/wiki/Code_book_attack

GREAT. (2014). *Tyupkin: manipulating ATM machines with malware.*

Greenberg, A. (2021). *NFC Flaws Let Researchers Hack ATMs by Waving a Phone.* New York: Wired.

Haque, I. A. (2018, July 1). *PIN Block Explained*. Retrieved 11 9, 2022, from https://www.linkedin.com/pulse/pinblock-explained-iftekharul-haque

Krebs, Brian. (2015). *Thieves Jackpot ATMs With 'Black Box' Attack.* Krebs On Security.

Kumar, N. (2021, April 26). *PIN safari: How is your PIN validated?* Retrieved November 09, 2022, from https://www.linkedin.com/pulse/pin-safari-how-your-validated-nishant-kumar/

Lake, J. (2022, Febraury 17). *What is 3DES encryption and how does DES work?* Retrieved November 9, 2022, from https://www.comparitech.com/blog/information-security/3des-encryption/

Rivest, R. L., Shamir, A., & Ademan, L. M. (1977, December 14). *United States Patent No. 4,405,829.*

Sancho, D., & Huq, N. (2017). Cashing in on ATM Malware (A Comprehensive Look at Various Attack Types). European Union’s law enforcement agency.

Vijayan, J. (2019). *Carbanak Attack: Two Hours to Total Compromise.* Dark Reading.

Zetter, K. (2009). PIN Crackers Nab Holy Grail of Bank Card Security. WIRED.

Zetter, K. (2016). *That Insane, $81M Bangladesh Bank Heist? Here's What We Know.* Wired.