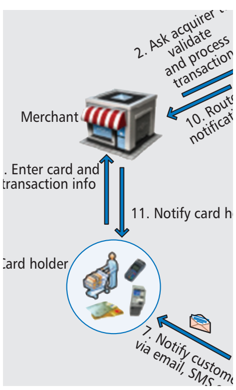
Enhancing Communication Adaptability between Payment Card Processing Networks

Financial organizations communicate with each other using ISO 8583 or its derivatives to complete the request and response cycle of card originated transactions originated from an ATM, POS, or the web. ISO 8583 is a broad standard. Its implementations slightly vary due to the flexibility available within the standard. The authors discuss the problem of adaptability of communication between payment card transaction processing entities due to this flexibility.

Adnan Noor Mian, Abdul Hameed, Muhammad Umar Khayyam, Farooq Ahmed, and Roberto Beraldi



Adnan Noor Mian and Abdul Hameed are with Information Technology University.

Muhammad Umar Khayyam is with TSSG Waterford Institute of Technology.

Farooq Ahmed is with National University of Computer and Emerging Sciences.

Roberto Beraldi is with Sapienza University of Rome.

[1] *3-D Secure* is branded as *Verified by Visa* by VISA and as *MasterSecure* by MasterCard http://www.visaeurope.com/en/cardholders/verified_by_visa, http://www.mastercard.com/us/personal/en/cardholderservices/securecode/index.html

Abstract

Financial organizations communicate with each other using ISO 8583 or its derivatives to complete the request and response cycle of card originated transactions originated from an ATM, POS, or the web. ISO 8583 is a broad standard. Its implementations slightly vary due to the flexibility available within the standard. This paper discusses the problem of adaptability of communication between payment card transaction processing entities due to this flexibility. We first provide an overview of different variations in ISO 8583 implementations and identify the interoperability issue based on our industry experience. We then propose a solution and suggest a way to standardize different implementations so that one organization can communicate with another without or with minimal changes in software. The suggestion is based on the exchange of meta data indicating how the target system is interpreting the header fields during the communication process.Finally, we discuss the benefits of the solution in which the vendors could not only avail the customization flexibility provided by ISO8583 but also ensure that their implementation of the standard is interoperable with others. This reduces the cost of interconnectivity with other partners when a network wants to expands its business.

Introduction

Financial transactions originating from credit, debit, gift, or other typed of cards are considered card-originated transactions. These transactions may include purchase, withdrawal, deposit, refund, reversal, balance inquiry, payments, inter-account transfers, etc. Such transactions are made via ATM, POS (Point of Sale) machines, web, and mobile applications, etc. A customer/cardholder inserts her card in an ATM or swipes it in a POS device and invokes a transaction after providing the necessary information, e.g. transaction amount and PIN (Personal Identification Number). Alternatively, the same information can be provided through a website or mobile-based application. Such user action initiates a card-based transaction that typically travels through a number of systems for its processing and results in either success or failure. Several entities are involved in the processing of the transaction and collectively constitute the payment processing chain. A message being exchanged between different systems carries information about the type of transaction, the card number, the merchant's information, requested amount, PIN, and so on. This is a type of two way communication. System A requests system B and system B responds to the transaction accordingly to system A. The response message contains similar attributes and a response code needs to be returned describing the status of the transaction as either approved, failed, or partially approved.

Figure 1 shows the interaction of entities and the sequence of steps involved in the validation of a payment transaction. The merchant validates the transaction and forwards a request message to the acquirer (step 2). The acquirer merchant's financial institution (MFI) validates the message and hands it over to the bankcard network, which routes it to the issuer processor/customer's financial institution (CFI) (step 5). The issuer, who holds the user account, processes the transaction request and replies with its status to acquirer (step 6). Additionally, the issuer may send a notification, e.g. through SMS or email to the card holder about the transaction done in his/her account (step 7). The acquirer also notifies the merchant about the status of the transaction (step 9), which then notifies the cardholder.

Since different vendors (e.g. VISA, MasterCard, etc.) provide services for such card-based transactions and different entities are involved in the entire process, a need to standardize message formats and communication flow for the exchange of financial data was seen as early as 1987, and a standard for exchanging card originated transactions was proposed by the International Standards Organization (ISO) under the code ISO 8583 [1]. The first version of it was published in 1987, and it was revised in 1993, 1998, and 2003. While the three revisions have made slight changes to the original draft, the basic message structure and communication flow remains the same. Most of the vendors still base their implementation on the 1987 draft [2].The standard defines the message structure and the communication flow among different entities of the transaction processing chain. Consumers may directly interact with ISO 8583-based devices (in the form of an ATM or POS devices), or their interaction may be routed indirectly through other protocols and standards built on top of ISO 8583 for value-added interaction. For instance, the ANSI X.59-2006 standard [3] and its derivative protocols such as CONSEPP [4] and ESCPS [5] are built on top of ISO 8583 to provide a secure electronic payment system in account-based environments (e.g. in web or mobile applications). Similarly, other protocols such as SET [6] and 3-D Secure[1] have been established for the same purpose with the support of major credit card vendors VISA and MasterCard, while other solutions such as IPS [7] have also been proposed to enhance security and usability. Underneath all these application-level solutions, ISO 8583 is at work.

Most vendors (VISA, MasterCard, etc.) differ in their implementation schemes of the standard. Moreover, standards such as ANSI X.59-2006 also require some customizations of ISO 8583. This does not pose as big a problem as long as every merchant supports only a single vendor. However, solutions such as jPOS [8] have made it quite easy for a merchant to support a diverse set of card-based services through a single integrated POS system. The issue with ISO 8583, however, is that it is a very flexible standard, which is simultaneously both good and bad. The standard defines the message structure and the information content that is necessary for the exchange of financial data along with a provision for custom fields from the flexibility perspective. That is the good part. However, it leaves various implementation details to the vendors, which raises interoperability issues. Hence, the standard provides a lot of flexibility, and that is probably why it has been extremely successful in facilitating various electronic payment schemes over the decades. However, this raises problems of interoperability in the modern dynamic environment when new and improved schemes continue to emerge [2].

Currently, if a financial service organization wants to make its own version of ISO 8583 interoperable with another organization, it has to review the target organization's implementation of the standard and make changes to its own implementation accordingly. For example, an acquirer may put a customized header in between the MTI and length indicator field. In this case the only option for the merchant is to make changes to its own implementation of the standard by incorporating the new header. This is not a scalable option, as the process has to be repeated for any organization that has recently entered the business circle.

In [2] a UML-based modeling technique has been discussed to address the interoperability issues in the VISA card processing network. The system provides a UML-based model for specifying the structure, semantic, and lifecycle of the data to prevent misunderstanding and incorrect population of message fields in the payment processing chain. A monitoring system is deployed that compares the field values of the message in every transaction against the specified values of the model and takes corrective actions if required. The problem with this approach is that a monitoring system is required to monitor and compare fields' values of every message of the transaction. Furthermore, the corrective steps required to address an interoperability issue are not addressed after they are identified.

Interoperability issues within card networks occur frequently and are pushing vendors to look for improved solutions [2]. One such effort is being conducted by ISO itself through another standard, ISO 20022 [9], which is aimed at standardizing protocols for all financial information exchanges, including card-based payments. ISO 20022 provides a common development platform based on UML modeling to define a number of financial business processes. Using this platform, financial institutions can develop a new message exchange protocol for various financial transactions that avoids confusion or misunderstanding of some common concepts. But there are some drawbacks to using this platform. ISO 20022 messages use an XML wrapper and each message carries a fare amount of XML data. Field-level constant information also needs to be shared with every single message and thus needs much more bandwidth. This increased bandwidth requirement of ISO 20022 becomes a bottleneck for millions of real-time transactions. On the other hand, ISO 8583 is simple and more precise, focusing only on the card-originated transactions. Further, ISO 8583 is lightweight and only small-sized bitmaps are exchanged in the messages. Consequently, we see that global adaptation of ISO 20022 is unlikely to occur in the near future. It is thus difficult for a financial institution to start its business based on ISO 20022 because no major payment network is currently accepting ISO 20022-based transactions. Though ISO 20022 may be used as a platform for avoiding misunderstanding of some common concepts in the financial process,it is however a paradigm shift and will take a long time to be globally adapted. The likely scenario is that both ISO 20022 and ISO 8583 will co-exist in order to support card-based payments [10]. The need, therefore, is to propose changes to ISO 8583 to make it more interoperable without compromising the flexible structure of the protocol.

This article is an attempt to identify the interoperability issues and propose changes to ISO 8583 to address these issues.Our proposed solution is based on the inclusion of meta data about the custom fields of the message that help to know how the target system is interpreting and using these fields in the processing chain. By using the current standard and suggesting enhancements to it, ISO 8583 will be become more adaptable, retaining all of its customization

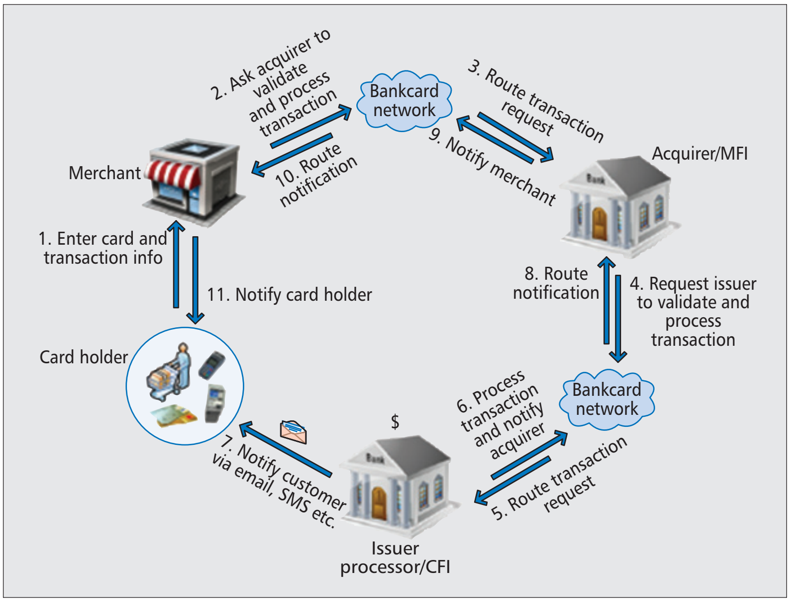


Figure 1. Basic process of a card-originated transaction.

The ISO 8583 standard is a message interchange specification that defines the format for exchanging transactional data in the entire payment processing chain. Transactions can be of various types, e.g. purchase, cash withdrawal, refund, etc.

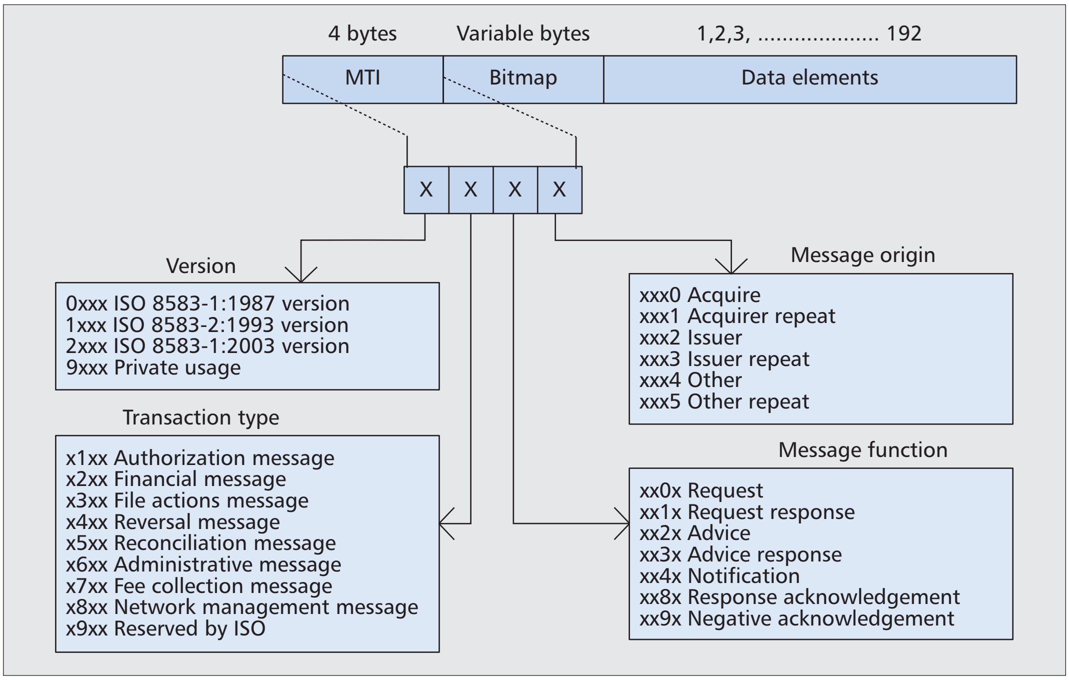


Figure 2. ISO 8583 message structure and MTI field.

flexibility intact. It will reduce the time and resources needed to make changes to existing implementations of the standard that are often needed when an organization wants to expand its network. Moreover, all entities involved in processing card-originated transactions would be on same page, and their software systems would be highly adaptable to welcome any other organization in their communication circle. Further, organizations would be able to focus on their business instead of the plumbing work for connecting with new partners.

The rest of the article is organized as follows. We provide a brief overview of the ISO 8583 message format. Then we present a gap analysis of different ISO 8583 implementations based on our experience in industry. We discuss the proposed solution to address these interoperability issues. We analyze and discusses the benefits and issues relevant to the solution, and we then conclude the article.

ISO 8583 Message Structure

The ISO 8583 standard is a message interchange specification that defines the format for exchanging transactional data in the entire payment processing chain. Transactions can be of various types, e.g. purchase, cash withdrawal, refund, etc. It also carries information regarding the merchant, her type of business, capture method, transaction amount and currency, etc. All of these are supported through the flexible message structure. The basic message structure has three components: MTI, bitmaps, and data elements, as shown in Fig. 2.

The message type identifier (MTI) is a small four digit numeric field that classifies the high-level function of the message. The first digit of the MTI identifies one of the three versions of ISO 8583. The second position indicates the type of transaction, e.g. authorization message, financial message, etc. The third digit of the MTI field indicates the function of the message. The fourth one is the origin of the message. Figure 2 shows various types of MTI messages with respect to the position of each field.

The MTI field is followed by bitmaps that are used to specify which data elements are present in the message considering the nature of transaction. A message should contain at least one bitmap that indicates the presence of data elements 1 to 64. This bitmap is called the primary bitmap. A message may also contain additional bitmaps in the form of a combination of primary and secondary (for data elements 1 to 128), or primary, secondary, and tertiary (for data elements 1 to 192).

Data elements contain the actual information about the transaction (e.g. card number, transaction amount, etc.). Each data element serves a specific purpose and its function is detailed in the standard. There are a total of 192 data elements, but not all are required in order to process a particular transaction. A data element is present only when its specific bit is on in the bitmap. This depends upon the nature of the transaction. Considering that the necessary data elements are made part of the message and their presence is indicated by setting the respective bits in the bitmap, the message length is thus controlled. Figure 3 shows a message with different data elements. The presenceof each data element is indicated by the related bit of the bitmap. Since the last data element present is 62, which is less than 64, only the primary bitmap is required. The absence of other bitmaps is also indicated by the OFF status of the first bit of the bitmap, which is ON only if this bitmap is followed by another one.

Data elements may be either fixed length or variable length. In order to control variable length, the tag length value (TLV) format may be used. It is a series of values of specified length, recognizable by a tag, all written in a single string. For instance, a string such as ad09city42USAag0229gd01M can represent useful information about a person, in this case address, age, and gender. The given string shows that there are three tags, ad, ag, and gd, respectively, representing address, age and gender. The length of the address is nine characters, thus the corresponding value of the address is city42USA. After nine characters, the next tag starts that represents the age whose length is two and hence the corresponding value is 29. Similarly, gender is represented using one character M, meaning male.

The issue with ISO 8583 is the flexibility accorded in the standard with respect to various implementation issues. For instance, the standard does not enforce any specific encoding scheme. A system is free to use any encoding from BCD, EBCDIC, ASCII, HEX, or BITSTRING for different types of its proprietary data fields. Similar flexibility is provided for other representations as well. Systems can use private fields to provide extra information for transaction processing, and different vendors use these in their own way. For instance, MasterCard sends field 48 (reserved for private data) in TLV format. VISA, on the other hand, uses field 48 with different formats depending on the transaction type and consequences. Similarly, some fields of the VISA message format have subfields represented by bitmaps.

Consequently, different vendors have slightly different formats all based upon ISO 8583 and they provide their own customized versions of the standard for development purpose. These customized versions are generally proprietary and not made available to larger audience. These customizations raise interoperability issues.

Gap Analysis

Based on our experience implementing ISO 8583 based solutions over the years for various network types including (but not limited to) MasterCard and VISA, we have done a gap analysis to find out where these implementations differ. We classify these implementation differences broadly into two categories: message level and field level.

Message Level Gaps

Message level gaps are the gaps that are related to the whole message and are not concerned with individual fields. Examples of such gaps are:

- According to the standard, the first two bytes of the message indicate the message length. This is followed by the MTI and the rest of the message. However, there is flexibility to place any informed data in between the message length and the MTI. Different vendors are free to add some reserved bytes in the message length and the MTI before transporting the message.
- A header may exist or not before the MTI. If it does, how can its length be specified in a standard way? An example of the header is a reject header indicating that the message received by another system is not in the correct form or information is not correct.

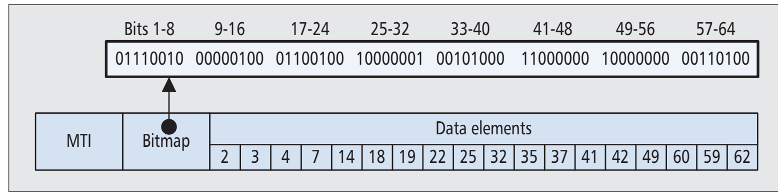


Figure 3. Example bitmap and data elements.

- The encoding of the messages may differ. It can be EBCDIC, ASCII, or any other encoding scheme. It may also vary by fields and subfields.
- The bitmap can be either packed or unpacked. An unpacked bitmap requires a byte each in order to represent the presence or absence of respective data elements. Thus 192 bytes will be required to represent the entire bitmap. A packed bitmap, on the other hand, uses one bit to represent the same information, thus reducing the size of the message.

Field Level Gaps

Field level gaps pertain to differences in representation of the data elements. They generally arise due to length and encoding issues. Examples of such gaps are:

- In case of variable length fields, a length is specified first before its content. Differences arise in the interpretation of length itself as to what it represents: the number of bytes or the number of elements in the field. In the case of bytes as a unit, length 8 means 16 values of some field of encoding hex. If the length is measured in the number of elements in the field, 16 in the length indicates 16 hex values and eight bytes of the original message. Even in a single message the length unit could vary. It can be either in bytes, in data per byte, or both (according to fields).
- A field can be a straight, TLV, or bitmap field. Straight means a simple data field with a constant number of sub-fields. TLV is discussed earlier in this document (section 2). A bitmap field is a field with a bitmap in its beginning, indicating the presence or absence of subfields in it. Multi usage is a field that may vary in its attributes or category in different transactions according to some context or condition.
- Moreover, the character set of a field can be numeric, alpha numeric, or alpha numeric special. Also, its encoding may vary from EBCDIC, ASCII, to others on a field to field basis.

Suggested Solution

As we have already seen, different networks may differ in their implementation of ISO 8583 and hence one vendor may send the information in a slightly different format while routing the information from one network to other. This will make interpretation of the message difficult. Our proposed solution is to keep the vendors flexible in using the standard in their own way as they do currently, but in order to make communication easy, we suggest sharing some meta information that can be made part of the standard. We propose two forms of meta data sharing: handshake level and transaction level.

Handshake Level Meta Data

Handshake level meta data would contain information for variations among systems both at the message level and the field level, only needing to be indicated once because in all upcoming messages that would remain same. Such metainformation may represent message level variations, as follows.

- **Message Beginning/Ending Symbol:** Currently, ISO 8583 does not suggest any symbol to represent the start or end of the message. It just asks for putting the length of the incoming message in the beginning of every message in two bytes. In the case of a noise in the message, the whole queue of messages for the transaction becomes disturbed. The issue could be solved by indicating the message boundaries through start and end symbols.
- **Reserved Bytes:** How many reserved bytes does a vendor send in the beginning? The number of bytes of proprietary data being used between the message length and MTI field should be indicated.
- **Presence of Header:** Is there any header present? If yes, then how much length does it have?
- **Encodings Used:** How is the message encoded? Which option (BCD, EBCD, ASCII, etc.) is used?
- **Bitmaps Information:** Does the message use packed or unpacked bitmaps? The type of bitmap should be indicated in the meta data.

Table 1 provides one possible format for representing these message level and field level meta indicators, but other ways may also be worked out. Similarly, meta information for field level variations, especially for private use fields, may include:

- **Length Unit:** What is the length unit for a specific private field? Is it fixed or variable? If variable, does the specified length represent the number of bytes or the number of elements?
- **Data Format:** Does it use the straight, TLV, bitmap, or multi usage format?
- **Character Set:** Does it use the numeric, alpha numeric, or alpha numeric special character set?
- **Encodings:** Which encoding scheme (BCD, EBCD, ASCII, etc.) does the field use?

The field level indicators are described in Table 1. Figure 4 shows an example of a header that could be used for exchanging the suggested meta indicators for message and field level during the handshake process. Meta indicators are shown by the dotted rectangle boxes within the message. On the receiving end, the header could be parsed to extract each item on the basis of the position and size of the item. The first part of the header is the message level meta data followed by field level meta data separated by a semi-colon (;) symbol. It should be noted that a header may contain meta indicators for multiple propriety fields separated by a semi-colon (;) symbol.

Level	Indicator	Symbols and description
Message	Message beginning/ending symbol	! marks beginning and ending of message
	Reserved bytes	XX where X represents a digit. Hence, 02 may represent two reserved bytes
	Header	XX where X represents a digit. Hence, 00 may represent absence of a header, where any other representation would give length of header in bytes, e.g. 10 means a 10-byte header
	Encodings	E for EBCDIC A for ASCII B for Binary V for Variable H for Hex 4 for 4-bit BCD
	Bitmaps	P for packed bitmap U for unpacked bitmap
	MTI	Offset of MTI
Field	Field ID	XXX where X represents a digit. For instance, 048 will refer to field number 48 for which the rest of the information is defined by the following indicators
	Length type	F for fixed length V for variable length
	Length unit	E for number of elements B for number of bytes
	Data format	S for Straight M for Multi-usage T for Tag Length Value B for Bitmap
	Field character set	N for Numeric A for Alphanumeric S for Alphanumeric special
	Field encoding	E for EBCDIC A for ASCII B for Binary V for Variable H for Hex 4 for 4-bit BCD

Table 1. Message and field level indicators.

Transaction Level Metadata

Transaction level meta data would contain information for variations that need to be indicated in every message because they can vary from transaction to transaction. Such information includes the following cases:

- **Reject Header:** Is there any reject header present? If yes, then what is its length?
- **MTI:** Where does the MTI lie? Flexibility of putting reserved bytes and headers plus reject headers makes the MTI lost until one parses the headers and reserved bytes first. But if one gets the MTI position or offset, one can proceed to parse the complete data elements from the message accordingly.
- **Field Data Format:** If the multi-usage field

format is used, then what usage is available in the current message.

Figure 5 shows an example header format for indicating transaction level variation. The message level and field level indicators are separated by a semi colon (;) symbol. Multiple field level meta data can be carried in the header. The indicator for a reject header is a two digit number where 00 represents the absence of a reject header; otherwise, it indicates its size. The size of the reject header is followed by offset of the MTI indicating the start of MTI in the message.

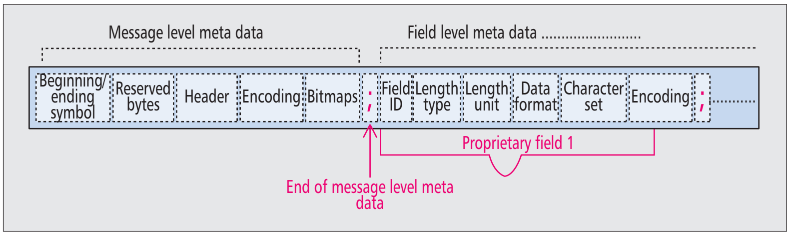


Figure 4. Handshake level meta data header.

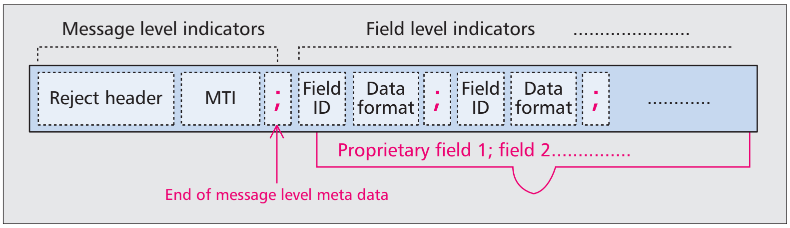


Figure 5. Transaction level meta data header.

Discussion

The solution proposed later has the following characteristics:

- Systems would have to share/receive meta data of private fields and customizations.
- For the first message at the sender side, the system would have to package and send information about handshake level and field level gaps to the receiver.
- For the first message at the receiver side, the system would have to understand the message, parse the information about handshake level and field level gaps, and save the information on a permanent storage.

This raises two issues when compared with existing implementations of ISO 8583: efficiency of the system and effort required to implement proposed changes.

Since extra information will need to be shared as metadata between the two systems for the solution to work, a question on efficiency can be raised. Considering the proposed message structure for metadata and encoding schemes presented in Fig. 4 and Fig. 5, there will be the addition of a few bytes only, and since most of this metadata will be exchanged once at the handshake level, there will be negligible impact on efficient transmission of transaction data.

Another important issue is the implementation effort required. This is surely a problem as many implementations of ISO 8583 are currently in use and will require changes. However, the changes itself are not very complex and can be easily implemented with little effort. This will be far more cost effective when compared with alternatives being explored to the interoperability problem in the form of other standards such as ISO 20022 [9], etc. In fact, these are the bankcard networks who publish their specifications of the customized ISO 8583 standard version. The other business entities, such as issuers, merchants, and processors, follow the specifications. So in order to achieve the objectives, first the bankcard networks such as MasterCard, VISA, Discover, American Express etc. have to agree on same page. They can document the final list of gaps and propose the meta data to be shared in the first message to issuers, merchants, or acquirers. Thus merchants, acquirers, and processors would be responsible for making changes in their system to receive that meta data in the message and store it in permanent storage to understand the upcoming messages accordingly.

Another possibility (even more suitable) is that instead of bankcard networks, ISO 8583 can impose the sharing of the meta data addressing all the gaps in the specification of a bankcard network. This will result in a new version of ISO 8583, also supporting previous versions for backward compatibility. Once adopted as part of the standard, the stakeholders would have to study the gaps in inter system implementations and comply with the proposed solution. Although it would require an additional effort to comply with the requirement of understanding or sharing the metadata, such an effort will result in the following benefits:

- It would reduce the cost of studying and incorporating the gaps for every new partner, resulting in fewer resources required in the form of time and effort.
- The system would be more reliable since handling different implementations for all entities involves multiple risk factors regarding implementation maintenance.
- All the entities involved in processing card originated transactions would be on the same page and their software systems would be highly adaptable to welcome any other organization in their communication circle.
- Organizations would be able to focus on their business instead of the plumbing work for connecting with new partners.

Conclusion

ISO 8583 is a widely used standard in financial transaction processing, but when a financial organization wants to expand its business network to some other organization, it requires making changes in its communication protocol with respect to all the customizations of the target system. Some changes in the standard can make it more interoperable so that once implemented, it can be used for any target system using th same ISO 8583 standard without losing the flexibility it provides. After doing gap analysis of different implementations of ISO 8583 based on our experience in industry, we identified certain fields that raise interoperability issues among different implementation of ISO 8583. To enhance the adaptability in the imple-

To enhance the adaptability in the implementations of this standard, we suggest a solution based on the exchange of meta data indicating how the target system is manipulating the custom fields.

mentations of this standard, we suggested a solution based on the exchange of meta data indicating how the target system is manipulating the custom fields. Two types of headers, handshake level and message level, can address the customizations of the standard being practiced. It would enhance the flexibility of communication and reduce significant waste of resources in porting existing implementations considering interoperability requirements.

References

[1] ISO 8583 Standard for Financial Transaction Card Originated Messages, 1987 version, 1993 version, 1998 version, 2003 version, http://www.iso.org.

[2] J. Bugajski and P. De Smedt, "Assuring Data Interoperability Through the Use of Formal Models of Visa Payment Messages," *Proc. 12th Int'l. Conf. Information Quality*, MIT, Cambridge, MA, USA, Nov. 9–11, 2007.

[3] American National Standard X9 .59-2006, Electronic Commerce for the Financial Services Industry: Account Based Secure Payment Objects, May, 24, 2006. http://webstore.ansi.org/.

[4] A. Levi and C. Kaya Koc, "CONSEPP: Convenient and Secure Electronic Payment Protocol Based on X9.59," *Proc. 17th Computer Security Application Conf. (ACSAC 2001)*, IEEE, 2001.

[5] C. Po Cheong *et al.*, "Designing an Efficient and Secure Credit Card-based Payment System with Web Services Based on the ANSI X9.59-2006," *J. Information Processing Systems*, vol.8, no.3, Sept. 2012.

[6] Secure Electronic Transaction Specification. Book 1: Business Description, Ver.1.0 1997, http://web.archive.org/web/20020802134102/http://www.setco.org/.

[7] Z. Duric, O. Maric, and D. Gasevic, "Internet Payment System: A New Payment System for Internet Transactions," *J. Universal Computer Science*, vol. 13, no. 4, 2007.

[8] jPOS Opensource Implementation of ISO 8583, http://www.jpos.org.

[9] ISO 20022 Financial Services - Universal Financial Industry Message Scheme, 2013, http://www.iso.org.

[10] D. Bruggink, P. Karsten, and C. R. W. de Meijer, "The European Cards Environment and ISO 20022," *J. Payments Strategy & Systems*, vol. 6, no. 1, June 2012.

Biographies

ADNAN NOOR MIAN (adnan.noor@itu.edu.pk) is currently working as an assistant professor in the Department of Computer Science at Information Technology University, Lahore, Pakistan. He has a Ph.D. in computer science and engineering from the University of Rome, "La Sapienza," Rome, Italy, and holds master degrees both in physics and computer science. His research interests include distributed algorithms, network protocols, pervasive computing, ad hoc and sensor networks, and application of AI techniques in these areas.

ABDUL HAMEED (abdul. hameed@itu.edu.pk) is a Ph.D. candidate in the Computer Science Department at National University of Computer and Emerging Sciences, Lahore, Pakistan. He has a master degree in computer science from National University of Computer & Emerging Sciences, Islamabad, Pakistan. His research interests include automation in network management, optimization, switching, network planning and diminishing, software defined networking, and the use of artificial intelligence techniques for computer network problems.

MUHAMMAD UMAR KHAYYAM (m.umar.khayyam@gmail.com) received his master degree from National University of Computer & Emerging Sciences in 2010, and an undergraduate degree in 2006 from COMSATS University, Lahore, Pakistan. Since 2006 he has been working in the software development industry in Pakistan and other countries. His area of interest is research and development of robust, scalable, maintainable and reliable real-time and enterprise software systems. He is currently working with the TSSG research group at Waterford Institute of Technology, Ireland.

FAROOQ AHMAD (farooq.ahmad@nu.edu.pk) received his master degree from National University of Computer and Emerging Sciences, Lahore, Pakistan in 2006. After that he worked as a software developer in the CS industry. Since 2011 he has been working as an assistant professor in the Computer Science Department of the same university, and he is also a Ph.D. candidate. His areas of interest are wireless ad hoc networks, distributed networks and protocols, and robotics.

ROBERTO BERALDI (roberto.beraldi@dis.uniroma1.it) received his Ph.D. in computer science from University of Calabria, Italy in 1996. Until 2002 he worked at Italian National Institute of Statistics. Since then he has been with the Department of Computer, Control, and Management Engineering at Sapienza University of Rome, Italy. He has published more than 70 papers in journals and main conferences related to wireless networks. His current research interest includes mobile and cloud computing, wireless networks, and distributed systems.