Tokenisation Blacklisting using Linkable Group Signatures

Assad Umar[1], Iakovos Gurulian[1] Keith Mayes[1], and Konstantinos Markantonakis[1]

Information Security Group, Royal Holloway, University of London.
Egham, Surrey TW20 0EX, UK.
{Assad.Umar.2011,Iakovos.Gurulian.2014,Keith.Mayes,K.Markantonakis}@rhul.ac.uk

Abstract. Payment cards make use of a Primary Account Number (PAN) that is normally used by merchants to uniquely identify users, and if necessary to deny users service by blacklisting. However, tokenisation is a technique whereby the PAN is replaced by a temporary equivalent, for use in mobile devices that emulate payment cards, but with reduced attack resistance. This paper outlines how tokenised payments contradict the process of blacklisting in open transport systems. We propose the use of a linkable group signature to link different transactions by a user regardless of the variable token. This allows the transport operator to check if a user's signature is linked to a previous dishonest transaction in the blacklist, while still maintaining the anonymity of the user.

1 Introduction

Card payments rely on the high levels of security and tamper-resistance provided by the chip embedded in the bank card. The chip provides secure storage for sensitive credentials such as the Primary Account Number (PAN) [1], as well as performing cryptographic operations. More recently, the use of contactless payments has risen significantly. There are now more than 81 million contactless bank cards on issue in the UK alone [2]. Contactless payments are quick and typically do not require cardholder verification, which makes them suitable for low value transactions. This opens up new use-cases for contactless payments such as transport. Major Transport Operators (TrOs) such as Transport for London (TfL) and the Utah Transit Authority (UTA) have moved from using proprietary smart card solutions exclusively, to accepting contactless credit/debit bank cards already in the user's possession. This model is generally referred to as the *'Open Ticketing Model'*. In open ticketing, the TrO typically relies on the PAN of the user, to determine the points of entry and exit that make up a complete journey.

In addition, by already having the infrastructure (terminals) to accept contactless bank cards, TrOs can also accept payments by Near Field Communication (NFC)-enabled devices with minimal or no changes. This is because both contactless cards and NFC devices comply with the ISO/IEC 14443 standards [3]. In fact, a terminal sees an NFC device in *card emulation mode* as if it were a regular contactless card. In this paper we focus on NFC device-based payments in transport.

Traditionally, an NFC device in card emulation mode relies on the Secure Element (SE) for enhanced security. It was envisaged that the host Operating System (OS) cannot guarantee the levels of security required by applications such as payment and transport. The SE is a small hardware tamper-resistant chip similar to the chip in a bank card in terms of functionality. The SE is typically embedded in the device, but could also be realised using the SIM card or an external memory card. The NFC controller routes messages received from a terminal to an application in the SE. The SE is tightly controlled by the Original Equipment Manufacturer (OEM) or by the Mobile Network Operator (MNO) in the case of a SIM card. This means only they can dictate who can provision an application on the SE and will usually charge a fee to do so. This adds an extra cost to NFC-based payments and adds to the complexity of the ecosystem.

However, *Host-based Card Emulation* (HCE) offers a drastic alternative to card emulation with an SE. HCE was first introduced by Cyanogenmod [4] and more notably by Google on Android 4.4 (KitKat) [5] onwards. It lets an application on the OS emulate a smart card. The NFC controller here routes messages directly to the application, bypassing the OS. Therefore security is traded for flexibility, because the guarantees of hardware-backed security are lost.

Different techniques have been proposed to manage the risk of HCE's reliance on software and make it acceptably secure for payments. More details in terms of the feasibility of these approaches as well their pros and cons can be found in [6]. Of significance to this paper is *tokenisation*. The idea of tokenisation is to replace the PAN in the user's device with a surrogate value that has a shorter life-span than the original PAN.

1.1 Problem Statement

The PAN has evolved from being just an account reference of the user. Merchants, in this case TrOs, rely on the PAN as a static value to uniquely identify users, and consequently blacklist them [7] [8]. However, in the case of tokenised payments, it is paradoxical for a merchant to rely on a non-static token for blacklisting. We highlight how tokenised payments in transport call into question the ability to blacklist dishonest users on the transport network. This variability in 'identity' exposes the TrO to attacks similar to the Sybil attack [9]. This is a potential problem for both academic proposals and real life implementations that rely on a static value to identify or distinguish users.

1.2 Proposed Solution

In this paper, we use linkable group digital signatures to propose a solution to the blacklisting problem [10] [11]. Linkable signatures have a property that lets a verifier link the signatures of a user on different messages, anonymously. We rely on the 'linkability' property to blacklist dishonest users, regardless of their non-static token. We also exploit the anonymity provided by the linkable signature, which is an important requirement for transport ticketing systems. Dishonest user in this

paper refers to a user travelling with no funds in the account, or an attacker using a stolen or compromised device. We are able to blacklist users regardless of their short-lived tokens while maintaining user anonymity. We test the feasibility of our solution by implementing it on an NFC mobile device.

1.3 Related Work

The work in [12] evaluates open ticketing using TfL and the Chicago Transit Authority (CTA) as case studies. The author mainly focused on the theoretical aspects of adoption, such as the issue of unbanked riders. In [13], linkable group signatures were used to detect the double usage of tickets; however, their proposal was based on a closed model and tickets were purchased well before the travel. To the best of our knowledge, the only academic open ticketing proposal is [14]. The authors proposed the use of bank cards and, specifically, using the PAN to identify users. However, the authors rely on Certificate Revocation List (CRL) to blacklist dishonest users. We shall discuss the problems with using CRLs below.

2 Transport Ticketing Systems

Transport ticketing systems can be classified into two broad categories: closed and open ticketing systems. Closed ticketing systems are proprietary systems that are typically 'card/device centric'; i.e. the card holds the logic, tickets, transaction value and other accounting related data used in the calculation of fares. In this model, the TrO is essentially its own 'bank'. Notable examples are the London Oyster card and the Hong Kong Octopus card. However, this paper focuses on the open ticketing systems described in more detail below.

2.1 Open Ticketing Systems

Open systems rely on the well established global payment infrastructure. This means users can make travel payments with contactless cards, mobile applications issued by the bank cards, or even digital wallets. Therefore the TrO in this model accepts payments like any other merchant. The TrO saves the cost of issuing the cards and managing the card system. It is considered that almost 10% of revenue generated on the London transport network goes to managing the Oyster card system [15].

Open ticketing can be realised in different ways. To that effect, the UK Cards Association (UKCA)[1] has designed a framework that outlines three contactless ticketing models as agreed by the card and transit industries [16]. This paper focuses on the *'Aggregated Pay As You Go'* model. In this model, the payment device is used multiple times and the price is not known at the beginning of the journey. Each usage of the device in a day is acknowledged and later aggregated at the back-office to determine the fare to be charged; and subsequently request for payments from the

[1] "The UK Cards Association is the trade body for the card payments industry in the UK, representing financial institutions which act as card issuers and acquirers"

user's bank through the payment network. It is important to highlight the concept of *delayed authorisation* of payments as it forms the basis upon which the aggregated pay as you go model relies.

In delayed authorisation, instead of requesting authorisation for every transaction as usual, the TrO only acknowledges the usage of the user's device at various points on the transport network (also known as a *TAP*) and sends the TAPs to the back-office. At the back office, all TAPs by the same user are aggregated at the end of the day. And only then does the TrO request for authorisation of funds. The apparent risk here is that a dishonest user can travel with no funds in the account since authorisation is not done at the time of travel. We refer to this as the *'first time travel risk'*. Currently, this risk is negotiated and accepted between the TrO and the bank issuers [16].

2.2 Blacklisting in Transport Ticketing

With over £200 million lost by UK transport operators in revenue due to dishonest users [17], blacklisting is an essential requirement for ticketing systems. Blacklisting becomes even more important for systems that rely on delayed authorisation due to 'first time travel risk'. This gives the TrO monetary incentives to deny the user travel until outstanding payments and possibly fines are settled. Therefore, a blacklisting solution must be able to uniquely identify a dishonest user and subsequently deny travel. In open ticketing, the two unique values that could potentially be used for blacklisting are the user's public keys through CRLs, or the PAN. Earlier solutions [14] have relied on CRLs. However, the use of CRLs for real life implementations has its challenges, the distribution of CRLs to merchants faces problems of efficiency. Furthermore, due to the strict timing requirements of transport ticketing systems, the look-up times of CRLs may prove to be too high. Also, with a huge number of transactions, it is impossible to update CRLs in an efficient way. More on CRLs can be found in [18].

2.3 EMV Payment Tokenisation

Tokenisation replaces the PAN with a short-lived surrogate value referred to as a 'token' [19]. The idea is to eliminate sensitive cardholder information, specifically the PAN, from the payment device as well as merchant terminals and replace it with a token. If the device is compromised, the token will be of minimal importance as it is only valid for a short time. EMVCo[2] has released a specification on the use of tokenisation for mobile payments [19].

EMVco introduces a new entity to the existing payment network known as the Token Service Provider (TSP). The TSP is responsible for generating, issuing, and provisioning payment tokens to legitimate token requests. The TSP is also responsible for maintaining the PAN-token mapping in the token vault, as well as the translation of PAN-token to legitimate requests. **Fig 1** above shows the transaction flow

[2] EMVCo, made up of six members; American Express, Discover, JCB, MasterCard, UnionPay, and Visa, facilitates worldwide interoperability and acceptance of secure payment transactions.

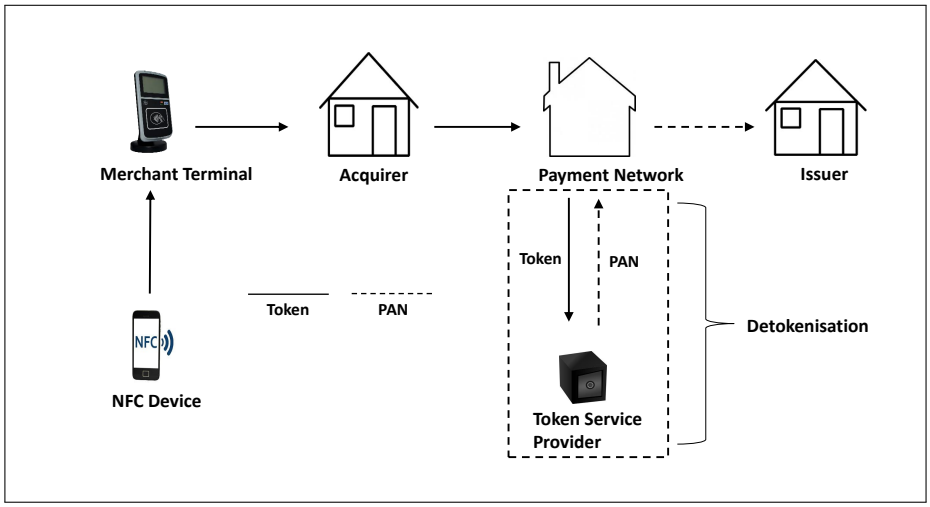

Fig. 1. Diagram Showing the Transaction Flow of a Tokenised Payment

in an EMV tokenised transaction. Methods of verifying the legitimacy of these requests, the token generation methods, and the way in which these tokens are provisioned to the device are out of the scope of this paper. We assume tokens will be generated and provisioned using global best practices. Also the validity period of tokens is out of scope; the amount of time for which a token is valid should be determined based on the perceived risk level.

3 Linkable Group Digital Signatures

Group signatures as first proposed by Chaum in [20] allow any member of a particular group to generate signatures anonymously. The verifier gets cryptographic assurances that a legitimate member of the group signed the message without revealing the signer's identity.

Group signatures with different properties have been proposed in the literature. In this paper, we use the linkable group signature first proposed in [11] (referred to as a list signature) and standardised by ISO/IEC in [10]. In its original construction, the signature supports linking signatures provided they were signed using the same linking tag. In [11] a time frame was used as the linking tag, allowing the linking of all signatures generated by a user within a given time frame. However, [10] shows the linking tag can also be any random value, as long as it is constant. This signature also supports revocation in case of dishonest users; it supports both private key revocation and verifier blacklist revocation. In the section below, we give an overview of the processes involved in this signature. For a detailed outline of the process and mathematical proofs, please refer to [10, 11].

3.1 Intractability Solutions

Strong RSA Assumption. First introduced in [21]; Let p' and q' be two distinct primes of equal length such that: $p = 2p' + 1$ and $q = 2q' + 1$ are also primes. The multiplicative group of quadratic residues modulo n denoted by $QR(n)$, is a cyclic group of order $p'q'$. Where $n = pq$, and is referred to as safe RSA modulus.

Decision Diffie-Hellman Assumption(DDH). Let g be the generator of a finite cyclic group G. The DDH assumption for group G states that it is hard to distinguish the DDH tuple: (g^x, g^y, g^{xy}) from random triples (g^x, g^y, g^z), for a random (x, y, z) modulo the order of group G.

In general, the DDH problem can also be constructed for arbitrary finite abelian groups. Therefore, if $G = QR(n)$, then G has composite order. If the group composition of G is known, then the DDH problem in G is reduced to the DDH problem in the components of G.

Notation	Meaning
$\boldsymbol{T_k}$	Token generated from the PAN
$\boldsymbol{t_{nt}}$	Timestamp at the point of entry
$\boldsymbol{t_{xt}}$	Timestamp at the point of exit
$\boldsymbol{t_X}$	Timestamp at the point of exit or entry
$\boldsymbol{R_n}$	Random nonce
$\boldsymbol{stn_{id}}$	Unique train station identity
$\boldsymbol{Sig_x}$	{} linkable signature with key x
$\boldsymbol{bsn}$	linking base
$\boldsymbol{T_4}$	linking tag
$\boldsymbol{A, e, x}$	signature key
$\boldsymbol{G_{pk}}$	group public key = (n, a, a_0, g, h, b)
$\boldsymbol{G_{mk}}$	group membership issuing key = (p', q')
$\boldsymbol{CHALL}$	$\{t_{nt}/t_{xt}\|\|R_n\|\|stn_{id}\}$
$\boldsymbol{TAP}$	$\{t_{nt}/t_{xt}\|\|R_n\|\|stn_{id}\|\|T_k\|\|T_4\}$

Table 1. Notations and Meanings

3.2 Phases

Below we describe the phases involved in a linkable group signature. We maintain the notations used in [10] and their meanings are given in **Table 1** above.

1. **Key Generation Phase:** The key generation is made up of two parts: Setup phase and the group membership issuing phase. In the setup phase, the Group Membership Issuer (GMI) creates the group public parameter, G_{pk}, and G_{mk}. The group membership issuing process is a protocol run between the GMI and a group member to create a unique signature key (A, e, x), where (x) is the private key and (A, e) is the group membership certificate for each group member. We assume the presence of a secure channel between the group member and the GMI.

a) Setup Phase We assume the existence of two hash functions H and H_Γ such that : $0,1^* \rightarrow 0,1^k$ and $H_\Gamma : 0,1 \rightarrow 0,1^{2lp}$ The GM chooses the group public parameters:$l_p, k, l_x, l_e, l_E, l_X, \epsilon$. GM also chooses random generators: a, a_0, g, h, b of QR(n). $G_{pk} = (n, a, a_0, g, h, b)$ and G_{mk} = (p', q')

b) Group Membership Issuing Phase At the end of this phase, the member knows a random $x \in [0, 2^{lx} - 1]$ and the manager knows a^x mod n and nothing more. Then the manager chooses a random prime $e \in [2^{lE} - 2^{le} + 1]$ and computes A = $(a_0 C_2)^{d1}$ mod n where $C_2 = a^x$ mod n and $d_1 = 1/e$ mod n. The manager sends A and e to the member. The member checks that $A^e = a_0 a^x$ mod n. The group member signature key is (A, e, x) and x is the private key.

2. **Signing Phase:** The signature process takes as input; the *(G_{pk})*, the group member signature key (A, e, x), a *linking base (bsn)* and the message to be signed and outputs a linkable signature Sig_x.

Algorithm 1 Signing

1: Compute f = $(H_\Gamma(bsn))^2$ (mod n)
2: Chooses random integers: $w_1, w_2, w_3 \in [0, 2^{2lp} - 1]$
3: Compute: $T_1 = Ab^{w1}$ (mod n),
$T_2 = g^{w1} h^{w2}$ (mod n),
$T_3 = g^e h^{w3}$ (mod n),
$T_4 = f^x$ (mod n).
4: Choose random integers:
$r_1 \in [0, 2^{\epsilon(le+k)} - 1]$,
$r_2 \in [0, 2^{\epsilon(lx+k)} - 1]$,
$r_3, r_4, r_5 \in [0, 2^{\epsilon(lp+k)} - 1]$
5: Choose random integers: $r_9, r_{10} \in [0, 2^{\epsilon(2lp+le+k)} - 1]$
6: Compute: $d_1 = T_1^{r1}/(a^{r2} b^{r9})$ (mod n)
$d_2 = T_2^{r1}/(g^{r9} h^{r10})$ (mod n)
$d_3 = g^{r3} h^{r4}$ (mod n)
$d_4 = g^{r1} h^{r5}$ (mod n)
$d_5 = f^{r2}$ (mod n)
7: Compute:
c = H$(a||a_0||g||h||T_1||T_2||T_3||T_4||d_1||...d_5||m)$
$s_1 = r_1 - c(e - 2^{lE})$, $s_2 = r_2 - c(x - 2^{lX})$,
$s_3 = r_3 - c\, w_1$, $s_4 = r_4 - c\, w_2$,
$s_5 = r_5 - c\, w_3$, $s_9 = r_9 - c\, e\, w_1$,
$s_{10} = r_{10} - c\, e\, w_2$
8: Set the signature as
$Sig_x = (c, s_1, s_2, s_3, s_4, s_5, s_9, s_{10}, T_1, T_2, T_3, T_4)$

3. **Verification Phase:** The verification process takes as input a message, bsn, a linkable signature, and G_{pk} corresponding to the group of the signer. It returns 1 if the signature is *VALID*, else it returns 0.

Algorithm 2 Verification

1: computes:

f $= H_\Gamma(b\,s\,n))^2$ (mod n)

$t_1 = a_0^c\, T_1^{s1-cl'}/(a^{s2-cL}\, b^{s9})$ (mod n) where $l' = 2^{lE}\, and\, L = 2^{lX}$

$t_2 = T_2^{s_1-cl'}/(g^{s9}\, h^{s10})$ (mod n) where $l' = 2^{lE}$

$t_3 = T_2^c\, g^{s3}\, h^{s4}$ (mod n)

$t_4 = T_3^c\, g^{s1-cl'}\, h^{s5}$ (mod n) where $l' = 2^{lE}$

$t_5 = T_4^c\, f^{s2-cL}$ (mod n) where $L = 2^{lX}$

2: computes:

c' = H($a||a_0||g||h||T_1||T_2||T_3||T_4||d_1||d_2||d_3||d_4||d_5||m$) If

c'=c, $s_1 \in [-2^{le+k}, 2^{\epsilon(le+k)} - 1]$,

$s_2 \in [-2^{lx+k}, 2^{\epsilon(lx+k)} - 1]$,

$s_3 \in [-2^{2lp+k}, 2^{\epsilon(2lp+k)} - 1]$,

$s_4 \in [-2^{2lp+k}, 2^{\epsilon(2lp+k)} - 1]$,

$s_5 \in [-2^{2lp+k}, 2^{\epsilon(2lp+k)} - 1]$,

$s_9 \in [-2^{2lp+le+k}, 2^{\epsilon(2lp+le+k)} - 1]$,

$s_{10} \in [-2^{lp+le+k}, 2^{\epsilon(2lp+le+k)} - 1]$ return 1 (valid signature) else return 0 (invalid signature)

4. **Linking Phase:** The linking process takes two valid linkable signatures and determines if they are linked, i.e. if they were signed by the same user.

Algorithm 3 Linking

Takes two **valid** linkable signatures:

$(c, s_1, s_2, s_3, s_4, s_5, s_9, s_{10}, T_1, T_2, T_3, T_4)$ and

$(c', s_1', s_2', s_3', s_4', s_5', s_9', s_{10'}, T_1', T_2', T_3', T_4'))$ If $T_4 = T_4'$ output 1 *i.e* they are linked, otherwise 0

5. **Revocation Phase:** The original construction of the signature supports two types of revocation: *Private Key Revocation* and *Verifier Blacklist Revocation*. In this paper we focus on the latter. In verifier blacklist revocation, the verifier generates a blacklist using T_4. So if the verifier needs to blacklist a dishonest signer, the signer's T_4 is added to the blacklist. Therefore the verifier can check if future signatures by the same signer are revoked by checking as follows: for each T_4', check $T_4' \neq T_4$. If any of the checks fail, output 0 (revoked), else, output 1 (valid).

4 Transport Ticketing Requirements and Adversary Model

Transport ticketing systems have both functional and security requirements. A general survey of electronic ticketing requirements can be found in [22]. Open ticketing models, however have fewer requirements than closed systems because most of the logic and fare calculation is moved to the back-office. We outline the requirements below. We also explain the capabilities and motivations of a determined adversary.

4.1 Adversary Model

The motivation for an adversary here is to abuse the 'first time travel risk' mentioned in section 2.1 above by maliciously, evading detection on the blacklist. The adversary could either be an attacker in possession of a stolen device, or a legitimate user trying to cheat the system. A determined attacker will try to avoid the blacklisting mechanism by producing a signature with a fake linking tag. According to the described Adversary model, we list the presumptive capabilities of the attacker below:

1. The attacker cannot break the linkable signature algorithm used in this paper.
2. The attacker is active, and can generate fake tokens and linking tags.
3. The attacker has access to the payment device, as well as a legitimate signing key.

4.2 Functional Requirements

1. Offline Verification: It should be possible to validate offline if the user is allowed to travel. This is because network connectivity cannot be guaranteed in some areas such as underground stations. Connecting to a back-end will also introduce latency to the overall transaction speed.
2. Efficiency: Transport ticketing systems should be very efficient in terms of passenger throughput. Therefore they are required to produce transaction speeds of 300–500 milliseconds from the time the user taps the device to the time the terminal grants or rejects access.

4.3 Security Requirements

1. Integrity: It should be possible to verify if a wrong ticketing credential is used. There should also be cryptographic evidence binding the user's transaction to a location at a particular time.
2. Anonymity: Although more of a privacy concern, the identity of the users of a transport system must not be revealed.
3. Exculpability: It should be impossible for any entity, including the group manager to falsely accuse a user of making a transaction at an entry or exit point on the transport network.
4. Blacklistability: It should be possible to build a blacklist of dishonest users (or compromised devices), and be able to deny them further usage of the transport network.

5 Proposed Model

In this section, we outline the general architecture of our proposed model. We define the entities involved, as well as the roles played by each entity. We also highlight the general assumptions made which form the basis of our proposed model.

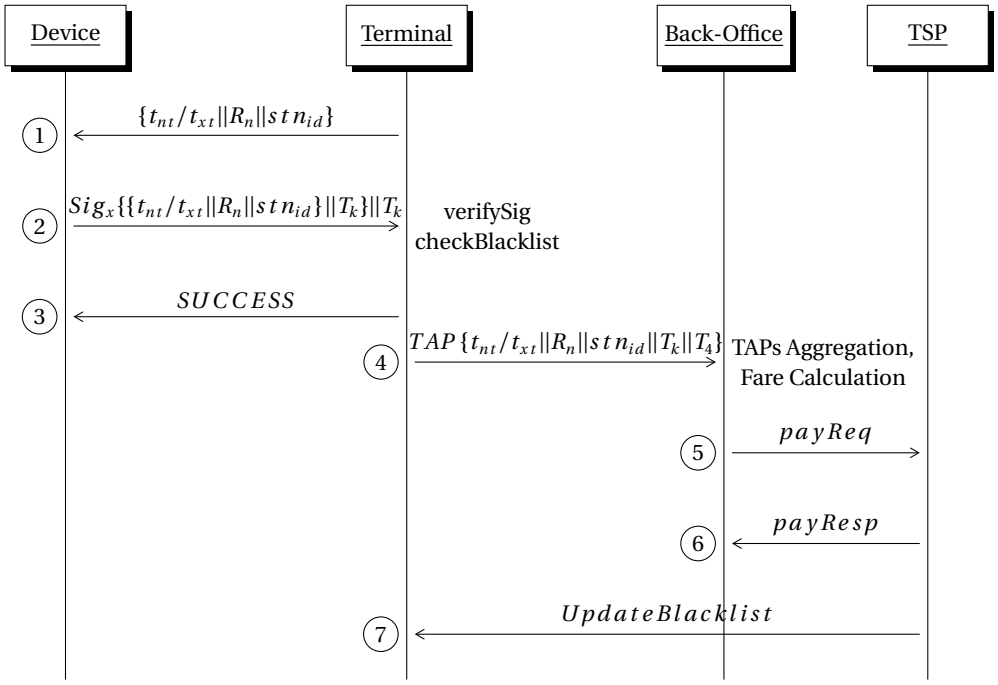

Fig. 2. Sequence Diagram Showing the Protocol of the Proposed Solution

We propose an open transport ticketing model which relies on EMV tokens provisioned to NFC devices. The protocol makes use of linkable digital signatures. Linkable digital signatures provide security features suitable for open transport ticketing models. By correctly verifying a user's signature, the TrO has assurance that the user belongs to a known group. More importantly for this paper, we use linkable signatures to solve the problem of blacklisting in tokenised payments. In case of a dishonest user, the TrO is able to link the signatures of the user on different tokens, while maintaining the anonymity of the user. **Fig 5**. shows the sequence of messages exchanged in the proposed protocol.

5.1 Assumptions

1. The transport application, and credentials including cryptographic material, shall be provisioned to the user's device using secure best practices such as using GlobalPlatform.
2. The payment networks act as TSPs, and shall subject users to necessary identification and verification (ID&V) prior to issuing new tokens.
3. Each user is part of a group of users depending on their payment network. For example, all MasterCard users are part of the same group.
4. The validity of the token in this paper is at least seven days.

5. We assume the maximum security features available via the platform/OS will be in place to store tokens, keys, and other cryptographic material. Therefore, in reality, the validity of the token will be based on the perceived residual risk.
6. The user is in possession of an NFC device with a payment application used for regular tokenised payments such as retail.
7. There is mutual trust between the TrO and the rest of the EMV ecosystem, and the terminals will be provided with the group public keys of the payment network.
8. Each train station has entry and exit gates, equipped with a terminal and a turnstile to grant or deny entry.

5.2 Entities

Below we describe the functions of the entities that make up the architecture of the proposed model.

User: A user in this context will already have a bank account and possibly a bank card. The user also has a NFC-enabled mobile device as well as a payment application provisioned to the device.

Transport Operator: The TrO is in charge of validating users before travel. The TrO also aggregates TAPs at the end of each day in the back-office to determine the fare, and subsequently apply for authorisation from the user's bank via the payment network. The TrO shall also maintain a blacklist of dishonest users in cases where the user has insufficient funds or in the case of a compromise.

Payment Networks: The payment networks will also serve as the TSP, and are in charge of provisioning new tokens to the users device, translating tokens back to PANs, and subsequently facilitating the authorisation of payments. The rationale behind the decision to use the payment network as the TSP is due to the fact that there are fewer payment networks globally, than banks[3]. Therefore this means that the TrO's terminal will have to keep a few group public keys for signature verification.

5.3 Phases

Our solution is divided into 4 phases: the *setup phase*, *validation phase*, *accounting phase* and the *blacklisting phase*. The specifics of the accounting phase are out of the scope of this paper. It is however important to mention as it precedes the blacklisting phase.

[3] The EMVco specification on tokenisation indicates that the payment networks can additionally act as the TSP, while still maintaining their primary roles in the EMV ecosystem

Setup Phase: This phase is executed between a user and the payment network. A user initiates this phase by opting to use the payment application on his device for transport payments. They both engage in an ID&V process to verify the user's identity and bank account, and check if the user's has any outstanding transport fares. The process is terminated if any of the checks fail. Otherwise they go through the key generation process as explained in section 1b. In the end, the user will have a unique signature key; (A, e, x), a token (T_k), and a TrO-specific (bsn). We assume the TSPs group public keys to be well known and are provided to the TrOs well before hand.

Validation Phase: This phase is illustrated in Fig5. We see that a user taps the device on a terminal at a train station, the terminal sends a challenge to the user. Challenge includes; timestamp (t_X), random nonce (R_n) and the station ID (stn_{id}). User concatenates the token (T_k) to the challenge, and signs as explained in section 2, using the (bsn) of the TrO. (t_x) could either be (t_{nt}) or (t_{xt}) for entry and exit gates respectively. The user concatenates (T_k) to the signed message and sends to the terminal. The terminal verifies the signature using (G_{pk}) as outlined in section 3. If the signature is valid, the terminal checks to see if the user's (T_4) is included in the blacklist. If it doesn't correspond with any (T_4) on the blacklist, the user is allowed to travel otherwise the user is denied travel. Afterwards, the terminal records a TAP. A TAP includes the challenge signed by the user, the users (T_k), and the amount to be charged which is determined in the *accounting phase* below. The blacklist check is only needed at the entry gates.

Accounting Phase: This is a back-office process where $TAPs$ of all users for the day are aggregated to determine the fare to be paid by the user. The TrO sends a payment request (payReq), which includes the (T_k) and the amount to be charged, to the payment network for authorisation[4].

Blacklisting Phase: This phase only becomes necessary in cases where an authorisation fails due to insufficient funds in the user's account. The TSP sends a *transaction decline* message to the TrO. The TrO then puts the users T_4 in its blacklist database and updates the terminals at the stations with the latest blacklist entries. A user's device can also be put in the blacklist in the case of compromise or a lost device.

5.4 Proof of concept

A proof of concept was developed to test the feasibility of our proposal and also analyse it against the requirements mentioned in section 4. A HCE-based Android application was installed on an NFC device for the digital signature implementation. We adapted an implementation of the digital signature in [23] which was part of an analysis of group signatures on mobile devices [24]. For the terminal, we had a Java application using the smartcard I/O Application Programming Interface (API) running on a PC; this acts as the terminal at a train station.

[4] The payment network, acting as the TSP, translates the token back to a real PAN and authorisation is processed as per normal EMV flow.

Device	Manufacturer	Operating System	RAM
Phone (Nexus 5)	LG Electronics	Android 5.1.1 (Lollipop)	2GB
Laptop	Dell	Windows 10	8GB
Reader	ACS(ACR1281U)	N/A	N/A

Table 2. Devices used in Proof of Concept

5.5 Lessons learned/Considerations

Support for extended Application Protocol Data Unit (APDUs)[5] on NFC devices is still not as extensive as smart cards. Therefore most NFC devices can only send normal APDUs with a maximum length of 256bytes. We realised this is a software-based restriction rather than the NFC controller's inability to handle bigger messages. Due to the size of the signature in our protocol, we modified the Android source to allow the device to send back the signature in one APDU, rather than in chunks.

Sending a substantial amount of data over the NFC channel may not always be efficient. Due to the size of the signature we used, we realised the time cost of compressing the message and decompressing at the terminal's side is trivial. We found the *BZip2* compression algorithm to be the most efficient.

Most of the parts of the signature can be precomputed; that is those parts that do not depend on the challenge from the reader.

5.6 Performance Analysis

The total size of $Sig_x\{\{t_{nt}/t_{xt}||R_n||stn_{id}\}||T_k\}||T_k$ is 3536bytes, with a 512bit key, plus additional 15bytes for concatenating the token to the signature in plain text. This is compressed to 1617bytes, providing 45.7% compression.

	CHALL	**Sign**	**Verify**	**Full Protocol**
Average	420.75	9.92	20.5	451.17
Min	405.55	7.65	17.32	430.45
Max	445.63	10.65	23.75	480.03

Table 3. Table Showing Transaction Times In Milliseconds (ms)

We took average timings of individual processes, as well as the total time it takes the full protocol to run over 100 iterations. The mobile device takes an average of 9.92ms to sign the challenge received from the reader and also compress the signature. The *Round Trip Time* (RTT), i.e. the time from when the reader sends the challenge to when it receives the response, takes on average 420.75ms. We refer to this as $CHALL$. It is important to note that about 90% of the RTT is spent on the

[5] Application Protocol Data Unit is the unit of communication between a device and a reader. APDUs are specified in ISO/IEC 7816 be

NFC communication link. The signature verification on the terminal side, including decompression of the received data, takes 20.5ms on average. The whole protocol takes on average 451.17ms.

For performance measurements of checking the blacklist, we rely on a comparative study of Database Management Systems (DBMS) in [25]. Each DBMS was populated with 1,000,000 records, and the timings for a *'select'* query for each was taken. The select query emulates the look up of a user's (T_4) from a blacklist database. SQL Server was the fastest and took 18ms, while the slowest was Oracle and took 23ms. These projections show that the delay introduced by searching a blacklisting is trivial and therefore, our protocol still runs within the accepted transaction time range for transport usage.

5.7 Requirements Analysis

We analyse our proposal against both the security and functional requirements mentioned in section 4.3 above. Our model meets the *offline verification* requirement because the terminal is able to verify a signature, as well as run the blacklisting function offline, i.e. without connecting to a back-office or relying on a third party. The protocol, as shown in Table 3 above, is within the acceptable transaction speed range, as stipulated by the *efficiency* requirement. It is worth noting that, currently, NFC devices in HCE only operate at the lowest NFC data rate of 106kbps[6]. We found out this limitation is also a software limitation and not the NFC controller's inability to operate at higher data rates. Therefore at higher rates, our solution is expected to be much faster.

In terms of security, for a signature verified to be valid, it is computationally hard for anyone except the group manager to reveal the identity of the actual signer. In the random oracle model, the proof of knowledge that is part of the signature can be proven in statistically zero knowledge. Also trying to identify a particular signer with certificate (A, e) requires the adversary to know if $\log_b T_1/A$, $log_g T_2$, and $log_g T_3/g^e$ are equal. This is assumed to be infeasible under the decisional Diffie-Hellman assumption. Therefore our protocol meets the *anonymity* requirement.

As shown in the key generation phase in 3.2, the group manager does not learn any new information about the user's private key (x), and at the end of the phase, the GM only learns a^x. Also, because (T1, T2 and T3) represent an unconditional binding commitments to to (A and e). This implies that if the factorization of n is feasible, the group signature is a proof of knowledge of the discrete logarithm of $A^/a0$ [26]. Therefore no entity, including the transport operator and the payment network acting as the group manager, can sign a message on behalf of a user as computing discrete logarithm is assumed to be infeasible. Therefore our protocol thereby meets the *exculpability* requirement because a user cannot be framed for a false transaction.

In addition, *integrity* is achieved because it is not possible for anyone without access to the private key (x) to generate a valid signature. Secondly, the TrO is able to verify that the signed message includes the correct challenge it had sent, thereby

[6] NFC supports data rate of 106, 212, 424, and 848kbps

cryptographically linking the user to that point on the transport network at that particular time. Hence creating the 'TAP'.

User *blacklisting* is achieved because a legitimate user cannot avoid detection on the blacklist by forging a false linking base. (T_4) is linked with (T_1) through the proof of knowledge and also the private key x. In addition, a legitimate user cannot repeatedly cheat the system by signing on a rogue token with a legitimate credential, because after the first payment request is declined,the TrO can blacklist the user with the corresponding (T_4).

6 Conclusion and Future Work

In this paper, we have looked at how a security solution – tokenisation, affects the unique identification of users in certain scenarios. In particular, we have highlighted how this calls into question user blacklisting in transport ticketing. We have shown how linkable group signatures can be used to link two transactions regardless of the changing token. This concept is used to create a blacklist of dishonest users.

We have also shown the feasibility of our solution by building a proof-of-concept which is analysed against the outlined requirements. Our solution can also be used in use-cases outside ticketing that rely on the static nature of PANs. For example, in retail to link different transactions of a user (with different tokens) for loyalty and promotional purposes.

As future work, we plan to investigate more efficient methods of achieving linkability while maintaining anonymity. We also plan to further improve the security of our proposed solution by implementing it on a Trusted Execution Environment (TEE).

References

1. Identification cards – Identification of issuers – Part 1: Numbering system. ISO/IEC 7812-1. Standard, International Organization for Standardization, Geneva, CH, 2015.
2. The UKCARDS Association. Card expenditure statistics, January.
3. International Organization for Standardization (ISO). Identification cards – Contactless integrated circuit cards – Proximity cards , 2008.
4. Doug Yeager. Added NFC Reader support for two new tag types: ISO PCD type A and ISO PCD type B, 2012.
5. Android Developer Guide. Host-based Card Emulation. `https://developer.android.com/guide/topics/connectivity/nfc/hce.html`.
6. MNFCC-14002. Host card emulation (hce) 101. Technical Report MNFCC-14002, SmartCardAlliance, August 2014.
7. Christian Radu. *Implementing Electronic Card Payment Systems*. Artech House computer security series. Artech House, 2003.
8. Samsung pay will transform the mobile wallet experience. Standard, Samsung Electronics Co, Ltd, 2016.

9. John R. Douceur. *Peer-to-Peer Systems: First InternationalWorkshop, IPTPS 2002 Cambridge, MA, USA, March 7–8, 2002 Revised Papers*, chapter The Sybil Attack, pages 251–260. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.
10. Information technology – Security techniques – Anonymous digital signatures. Standard, International Organization for Standardization, Geneva, CH, 2013.
11. Sébastien Canard, Berry Schoenmakers, Martijn Stam, and Jacques Traoré. List signature schemes. *Discrete Appl. Math.*, 154(2):189–201, February 2006.
12. Candace Elizabeth Brakewood. Contactless prepaid and bankcards in transit fare collection systems. June 2010.
13. GhadaArfaoui, Guillaume Dabosville, Sébastien Gambs, Patrick Lacharme, and Jean-François Lalande. A privacy-preserving NFC mobile pass for transport systems. *ICST Trans. Mobile Communications Applications*, 2(5):e4, 2014.
14. Jan-Erik Ekberg and Sandeep Tamrakar. Mass transit ticketing with NFC mobile phones. In *Trusted Systems - Third International Conference, INTRUST 2011, Beijing, China, November 27-29, 2011, Revised Selected Papers*, pages 48–65, 2011.
15. Transport Committee. The future of ticketing. Greater London Authority, 2011.
16. Briony Krikorian-Slade and Nicola Moir Adrian Burholt. Contactless transit framework. Standard, UK Cards Association, London, UK, 2016.
17. Annual Fraud Indicator. report, University of Portsmouth, Centre for Counter Fraud Studies, Portsmouth, England, 2016.
18. Craig Gentry. *Advances in Cryptology — EUROCRYPT 2003: International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4–8, 2003 Proceedings*, chapter Certificate-Based Encryption and the Certificate Revocation Problem, pages 272–293. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.
19. EMV Payment Tokenisation Specification. Standard, 2014.
20. S. Brands and D.Chaum. Distance-bounding protocols. In *Advances in Cryptology EUROCRYPT '93*, Norway, September 1993.
21. Niko Bari and Birgit Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *EUROCRYPT*, volume 1233 of *Lecture Notes in Computer Science*, pages 480–494. Springer, 1997.
22. Maciî Mut-Puigserver, M. Magdalena Payeras-Capellí, Josep-Lluís Ferrer-Gomila, Arnau Vives-Guasch, and Jordi Castellí-Roca. A survey of electronic ticketing applied to transport. *Comput. Secur.*, 31(8):925–939, November 2012.
23. Klaus Potzmader. ISO20008-2.2 Group Signature Scheme Evaluation on Mobile Devices, 2013.
24. Klaus Potzmader, Johannes Winter, Daniel Hein, Christian Hanser, Peter Teufl, and Liqun Chen. Group signatures on mobile devices: Practical experiences. pages 47 – 64, 2013.
25. Youssef Bassil. A comparative study on the performance of the top DBMS systems. *CoRR*, abs/1205.2889, 2012.
26. Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. pages 255–270. Springer-Verlag, 2000.