Securing Balise-based Train Control Systems using Cryptographic Random Fountains

J. Harshan‡, Sang-Yoon Chang⋆, Seungmin Kang†, Yih-Chun Hu* †
‡Indian Institute of Technology Delhi, India, †Advanced Digital Sciences Center, Singapore,
⋆University of Colorado, Colorado Springs, USA, *University of Illinois Urbana-Champaign, USA
Email: jharshan@ee.iitd.ac.in, schang2@uccs.edu, Seungmin.k@adsc.com.sg, yihchun@illinois.edu

***Abstract*—In modern train control systems, a moving train retrieves its location information through passive transponders called balises, which are placed on the sleepers of the track at regular intervals. When the train-borne antenna energizes them using tele-powering signals, balises backscatter preprogrammed telegrams, which carry information about the train's current location. Since the telegrams are static in the existing implementations, the uplink signals from the balises could be recorded by an adversary and then replayed at a different location of the track, leading to what is well-known as the *replay attack*. Such an attack, while the legitimate balise is still functional, introduces ambiguity to the train about its location, can impact the physical operations of the trains. For balise-to-train communication, we propose a new communication framework referred to as cryptographic random fountains (CRF), where each balise, instead of transmitting telegrams with fixed information, transmits telegrams containing random signals. A salient feature of CRF is the use of challenge-response based interaction between the train and the balise for communication integrity. We present a thorough security analysis of CRF to showcase its ability to mitigate sophisticated replay attacks. Finally, we also discuss the implementation aspects of our framework.**

I. Introduction

While traditional rail transportation systems facilitate movement of people across several thousands of kilo-meters such as inter-city services, relatively shorter-distance urban transportation systems such as Singapore Mass Rapid Transport (SMRT), Hong Kong's Mass Transit Railway (MTR), to name a few, have been the backbone of the state's economic growth by assisting millions of commuters reach their destination on-time at lower cost. Either inter-city or intra-city, train systems are critical infrastructures, wherein efficiency, punctuality and safety are of utmost importance. Despite stringent quality requirements, unexpected issues, be it at the signaling level, or at the equipment level culminate in reporting of incidents and accidents [4]. Furthermore, with the possibility of cyber-physical attacks looming large, renewed interest is seen in securing the rail transportation systems from various unforeseen and relevant attacks.

One such challenge for the rail operators is to protect the integrity of localization techniques, which are used to learn the position of the trains. Consolidation of trains' positions assists the control center to send relevant commands to the trains about the positions of the neighbouring trains in order to avoid collisions.

In contrast to global positioning system (GPS), which is popularly used for localization of general mobile applications (e.g., smartphone, ships, and ground/aerial vehicles), train systems often use its own infrastructure for the vehicle localization due to the following reasons: trains have a fixed operational trajectory defined by the railway tracks, making the trains always in close proximity to the infrastructure during operations; infrastructure-based localization offers finer granularity than GPS; trains can forgo many of the known threats on GPS [6], [13]; train systems do not need to rely on a third-party that provides the satellite service and can keep the system reliance within its system; and importantly localization through GPS does not work with underground trains.

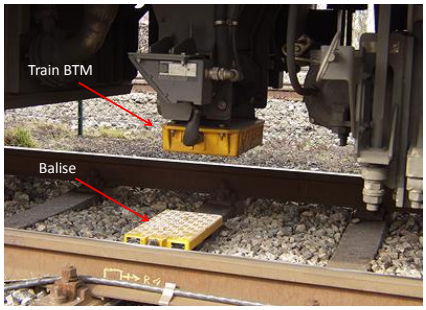

Fig. 1. Balises are laid on the sleepers of the track at regular intervals. Picture credit - [1].

A. Problem Statement

In contemporary train control systems, localization of trains is accomplished through passive transponders called balises, which are regularly placed on the sleepers of the track as shown in Fig. 1. When the train BTM (Balise Transmission Module) crosses the balise, it receives an acknowledgement from the balise thereby learning its current location on the track. Subsequently, this location information will be forwarded to the control center, which gathers such details from all the trains for global knowledge of the trains' positions. In particular, each train is initially loaded with a map of its rail-route, indicating the (approximate) location of balises. Then, while the train is moving, it ticks the reference positions after receiving the signals from the relevant balises. To highlight the importance of these *check-marks*, [7] discusses the implications of degraded detection of balises. Operation-wise, the balises are energized by transmitting an unmodulated

978-1-5386-0683-4/17/$31.00 ©2017 IEEE

978-1-5386-0683-4/17/$31.00 ©2017 IEEE

radio-frequency (RF) signal from the train BTM. In response, the balises backscatter fixed location information and other details. Since train's positioning information is crucial in preventing accidental collisions, countermeasures to cyber-physical attacks on train's localization are of utmost priority. A relevant cyber-physical attack is impersonation of balises by an attacker; in particular at different positions on the track. For an attacker to impersonate a balise at a different location, the attacker has to first listen to the balise's signals either during regular communication, or by separately energizing them using the RF signal. Since no data is transmitted in the downlink in existing systems, i.e., from the train BTM to the balise, the attacker just needs a signal generator of appropriate power and frequency to energize a balise. Furthermore, if the attacker replays the balise's signal (see Fig. 2) next to the legitimate balise, then the train BTM will receive multiple telegrams from the same balise-ID, however each carrying possibly different messages. In such a situation, a relevant question is *which signal should the train BTM trust?* After receiving the same balise-ID at different locations, the train will have to identify the legitimate one among them. If the train identifies the impersonated telegram as the legitimate one, then this may lead to safety issues as vital information related to permissible load, maximum velocity on the track and elevation might have been modified by the attacker. On the other hand, if the train decides to stop due to uncertainty about its location, then this may lead to unnecessary delays in the travel time. Current train systems do not have a mechanism to distinguish between the legitimate signals and their replayed versions.

Overall, the following assumptions are made in our attack model: (i) The attacker is external, and can be mobile, and (ii) The attacker has the necessary equipment to energize the balises, record its telegrams, and then replay the telegram by modifying its contents. Fig. 3 depicts a sophisticated version of the replay attack, wherein the attacker can eavesdrop on the downlink signal, energize the balise, record the telegram, and then replay it, all within a segment between two balises. It is straightforward to verify that the train cannot distinguish the legitimate telegram even after protecting the uplink messages by a message authentication code (MAC) at the balise.

B. Prior Work

In the context of balise-based control systems, replay attacks refer to impersonation of either the balise or the train. Replay attacks on balise's telegrams have been discussed in [14], wherein the authors have proposed a MAC based solution which assumes time-synchronization between balises and the train. Such a strong assumption therefore necessitates its applicability to only a special class of *controlled balises* that are physically connected to LEU (Lineside Electronic Unit). Although powerful, the authentication mechanism in [14] cannot be applied to balises that do not support clock-synchronization with the train.

An independent mechanism to detect replay attacks is by accurately estimating the position of the next balise, through train's on-board odometer system [12], [5], [15]. Since the distance between balises is fixed and *a priori* known, the

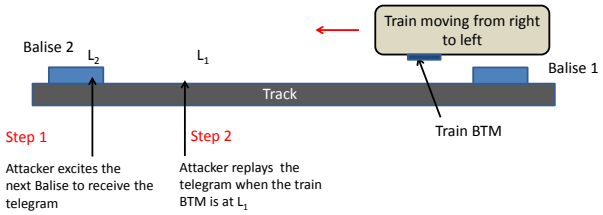

Fig. 2. Depicting a replay attack on the uplink channel: The attacker is impersonating a balise by transmitting the telegram at a different position on the track.

train's odometer system can roughly estimate the location of the next balise. In such a case, if an attacker replays the uplink telegrams at multiple locations, the train can reject those telegrams received at unexpected positions. However, a limitation of this method is its dependency on the accuracy of the odometer system. It has been shown in [14] that odometer measurements could suffer from random variations mostly due to physical conditions such as rain, snow, skidding and sliding. Therefore, if the attacker executes a replay attack within the odometer's accuracy, then the train will still have to solve the ambiguity issue.

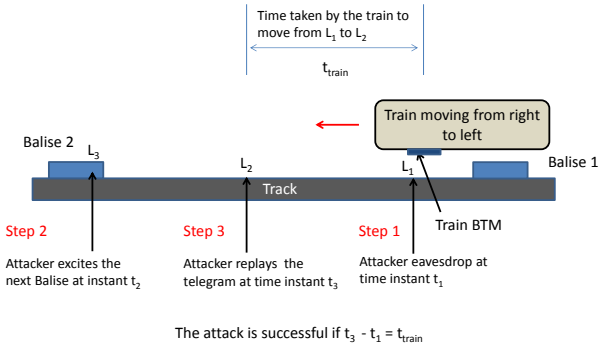

Fig. 3. Depiction of a sophisticated replay attack: The downlink signals are eavesdropped by the attacker at location L_1, and then the same signal is used to excite the next balise, which is Balise 2. Subsequently, the telegram generated from Balise 2 is used to execute an uplink replay-attack at a different position (location L_2) between the two balises. Note that this attacker can deceive the train even if the uplink messages are protected by a MAC.

C. Contributions

We address a new threat model in train control systems, wherein the communication between balises and the train can be subject to replay attacks (see Fig. 2). This threat is specifically applicable to driver-less trains (e.g., Circle-Line in Singapore MRT) that rely on uplink signals

from balises as check-marks for localization. Citing non-applicability of standard authentication methods, we propose a new authentication scheme called Cryptographic Random Fountains (CRF) in order to mitigate replay attacks in balise-to-train communication. We also discuss the implementation aspects of CRF, namely: computational power and latency of communication at the balises. Recent advances in application of cryptographic tools to RFID systems [8] have shown that encryption/decryption methods are possible in power-passive devices, thus leaving us to study the other important challenge, which is latency.

II. Cryptographic Random Fountains

We propose a new scheme for communication between balises and the train BTM, by using balises as random fountains along the track. Our primary objective is to prevent an attacker from recording the telegrams for subsequent replay attacks. Although the radio-access details such as modulation scheme, carrier frequency, and the structure of the telegram are described in the publicly available specification documents [3], explicit excitation will further assist the attackers to learn undisclosed information such as balise identification number, and to also understand the RF characteristics of the uplink signal. To mitigate unauthorized excitation of balises, we propose the feature of cryptographic random fountains (CRF) for the balises, wherein each balise, instead of transmitting telegrams with fixed information, transmits telegrams which also include random signals. A salient feature of our scheme is that the balises incorporate the downlink signals into the random signal generation for uplink. Therefore, unlike the state-of-art schemes, the train BTM does not transmit a plain RF signal, instead it modulates the RF signal by a nonce in order to mitigate uplink replay attack. In a nutshell, when the train crosses the balise, the balise harvests energy from the downlink signal, and subsequently decodes the downlink telegram, and then initiates routines for downlink authentication and uplink signal generation. Later, the *freshness* of the uplink signal is verified by the train to detect a replay attack. Each functionality will be explained in detail in the following subsections.

A. Ingredients for CRF

Suppose that the rail-route has B number of balises along the track. Also, suppose that the CRF scheme allows L trains to cross each balise. Here, L is only a logical number as the number of physical trains (fleet size) can be much smaller than L.[1] Henceforth, throughout the paper, we use the indices $i \in \{1, 2, \ldots, B\}$ and $j \in \{1, 2, \ldots, L\}$ to represent the balises and trains, respectively. The control center securely programs the key k_i in Balise-i, for each $i \in \{1, 2, \ldots, B\}$. The control center also distributes the set of encrypted messages $\{E_{k_i}(j)\ \forall i\}$ to Train-j, for $j \in \{1, 2, \ldots, L\}$, where $E(\cdot)$ is a block cipher that encrypts the index of the train using an appropriate key k_i. Since L can be larger than the fleet size, some trains can get several such encrypted messages. The following ingredients are stored at the balises and the trains:

[1]Suppose that we have 10 distinct trains that operate on a track moving unidirectional from Point A to Point B. If each train operates the route 100 times, then L in this case will be 1000.

1: **procedure** TRAIN-SIDE TRANSMITTER(BALISE DETECT, $E_{k_i}(j)$)
2: **WHILE** BALISE DETECT == 0
3: $k_{TB} \leftarrow nonce$
4: $DL^{(1)} \leftarrow E_{k_i}(j)$
5: $DL^{(2)} \leftarrow k_{TB}$
6: Transmit the message $[DL^{(1)}\ DL^{(2)}]$
7: **END**
8: **end procedure**

Fig. 4. Downlink transmission from train's transmitter block: Until the train detects a balise it continuously transmits $[E_{k_i}(j)\ k_{TB}]$, wherein the second part is nonce, while the first part is a function of the expected balise. The flag BALISE DETECT is continoulsy sampled from the train-side receiver block. Logical '0' on BALISE DETECT implies that the train is yet to detect the balise.

Ingredients at Balises:

1) Balise-i is loaded with the key k_i, which is used to decrypt the message from the train.
2) A message authentication code $MAC(\cdot)$ to generate a random signal for the uplink message. It has two inputs: (i) nonce received from the train, and (ii) the key k_i.

Ingredients at the Trains:

1) Train-j has the set of encrypted messages $\{E_{k_i}(j)\ \forall i\}$
2) Each train has an algorithm to generate nonce value for each downlink frame (how exactly the nonce is generated is not addressed in this work).
3) Each train should have the same message authentication code $MAC(\cdot)$ as that in the balise for uplink authentication.

B. Downlink Signal Generation

Under the framework of CRF, Balise-i does not transmit deterministic signals in the uplink channel, instead, it transmits random signals which are generated using the train-based nonce and the locally stored key k_i. After crossing Balise-$(i-1)$ successfully, Train-j regularly transmits messages of the form

$$DL = [DL^{(1)}\ DL^{(2)}] = [E_{k_i}(j)\ k_{TB}], \tag{1}$$

by modulating them on the tele-powering signal. In (1), the first part of the message $DL^{(1)}$ is the encrypted version of the train's index j by the key k_i, whereas the second part $DL^{(2)}$ is the nonce (which changes randomly with every downlink transmission). A pseudo-code for downlink signal generation is given in Fig. 4.

We assume that the downlink message DL is encoded using an appropriate error correction code (as shown in [3]) to shield DL from channel-induced errors in the air-gap. Assuming that DL is received at the balises without errors, Balise-i first authenticates the downlink signal using the key k_i (as described in Section II-C), and then uses the nonce K_{TB} to generate the uplink signal (as described in Section II-D).

1: **procedure** BALISE-SIDE ALGORITHM($DL^{(1)}, DL^{(2)}$, c)
2: Compute $r \leftarrow D_{k_i}(DL^{(1)})$;
3: **IF** $(r == c+1)$ || $(r == c)$
4: $k_{BT} \leftarrow MAC_{k_i}(DL^{(2)})$
5: Transmit k_{BT} in the uplink telegram
6: $c = r$
7: **ELSE**
8: Discard the received telegram
9: **END**
10: **end procedure**

Fig. 5. Algorithm for downlink authentication and telegram generation at the balise. Among the inputs to the procedure, $DL^{(1)}$ (corresponds to the encrypted message from the train) and $DL^{(2)}$ (corresponds to the nonce k_{TB}) are received from the train, while c is retrieved from the balise. Note that the balise permits repreated excitation by the same encrypted message.

C. Authentication at Balises

Suppose the trains $\{\text{Train-1}, \text{Train-2}, \ldots, \text{Train-}L\}$, are scheduled to cross Balise-i in the ascending order of the index. Then the control center distributes the message $E_{k_i}(j)$ to Train-j for $1 \leq j \leq L$, where k_i is the key used to encrypt the index j. For this distributed allocation, we assume that the track allows one-directional train movement, and also that the order of the trains remain the same each round.[2]

Balise-i is programmed with the variable c, that initially stores 0. When Train-1 crosses Balise-i for the first time, the train transmits the message $E_{k_i}(1)$ in the first portion of the downlink message as shown in (1). Upon receiving the downlink message, denoted by $[DL^{(1)}\ DL^{(2)}]$, Balise-i decrypts the received message using the key k_i to recover the index of the train as $r = D_{k_i}(DL^{(1)})$. The balise checks if the received index value r is one more than the stored value c. If so, then the balise passes the downlink authentication and then generates the uplink message UL (which will be discussed in Section II-D). After the uplink transmission, the balise updates c by the received index r. Any subsequent excitation of the balise with the same message $DL^{(1)} = E_{k_i}(1)$ (before Train-2 cross the balise) is also considered legitimate in our model. When Train-2 crosses Balise-i, it sends the downlink message $DL^{(1)} = E_{k_i}(2)$ to the balise. Thereafter the authentication process advances as explained earlier. The above authentication procedure is depicted as an algorithm in Fig. 5, where Line 3 checks for the index of the train.

D. Uplink Signal Generation at Balises

Once the downlink authentication passes, Balise-i uses the second part of the downlink signal, i.e., $DL^{(2)}$, to generate a random uplink signal as

$$UL = k_{BT} = MAC_{k_i}(DL^{(2)}), \tag{2}$$

where $MAC(\cdot)$ is an appropriate message authentication code used to generate a random signal, and k_i is the secret key. The signal UL is encoded as a packet in the uplink telegram. The random signal k_{BT} transmitted from the balise is such that given a particular signal in the sequence, the next random signal is practically infeasible for the attacker to determine. The random signal generation operation is captured in Line 4 of the algorithm in Fig. 5.

[2]For tracks with bidirectional train movement and/or when there are unexpected changes in the train schedule, the secret keys have to be appropriately distributed.

1: **procedure** TRAIN-SIDE RECEIVER(k_{BT}, k_i)
2: $c \leftarrow MAC_{k_i}(k_{TB})$
3: **IF** $c == k_{BT}$
4: BALISE DETECT $\leftarrow 1$;
5: Retrieve the encrypted message for the next balise
6: **END**
7: **end procedure**

Fig. 6. Algorithm for uplink authentication at the train BTM. The symmetric secret key k_i is retrieved from the train's EEPROM, while k_{BT} is received as a packet in the uplink telegram.

E. Uplink Authentication at Trains

Train-side authentication algorithm is captured in Fig. 6. The received value UL (given in (2)) is compared with the train-based $MAC_{k_i}(K_{TB})$, where K_{TB} is the latest nonce generated by the train. When the two values match, the train successfully identities the balise. Otherwise, it drops the telegram received in the uplink.

III. SECURITY ANALYSIS OF CRF

We show that CRF can mitigate the sophisticated attack depicted in Fig. 3. This sophisticated attack is executed by three collaborative attackers, namely attacker-1, attacker-2 and attacker-3, in a collaborative fashion. Suppose attacker-1 listens to the downlink frame sent from the Train-j at the t_1-th time instant (right after the train has successfully crossed the previous balise). Let us denote that downlink frame as $\mathcal{F}_{t_1}$. The signal eavesdropped by attacker-1 is $[E_{k_i}(j)\ K_{TB}(\mathcal{F}_{t_1})]$, where $E_{k_i}(j)$ denotes the encrypted message from Train-j to Balise-i as discussed in Section II, and $K_{TB}(\mathcal{F}_{t_1})$ is the nonce generated for that frame. After attacker-1 passes this signal to attacker-2, attacker-2 replays the signal $[E_{k_i}(j)\ K_{TB}(\mathcal{F}_{t_1})]$ on Balise-i, say at the t_2-th time instant. Subsequently, Balise-i successfully authenticates the excitation (as the authentication is only based on $E_{k_i}(j)$ and not on the nonce values), and then transmits the telegram along with the signal $MAC_{k_i}(K_{TB}(\mathcal{F}_{t_1}))$. This uplink signal is then received by attacker-2, who forwards that to attacker-3. Then the forwarded signal is replayed by attacker-3 at a different location other than the balise's position (say in the t_3-th time instant). Upon such an uplink transmission, the received value $MAC_{k_i}(K_{TB}(\mathcal{F}_{t_1}))$ at the train is compared with the train-based $MAC_{k_i}(K_{TB}(\mathcal{F}_{t'}))$, where $\mathcal{F}_{t'}$ is the latest downlink frame transmitted from the train. Since the nonce values generated for $\mathcal{F}_{t'}$ are $\mathcal{F}_{t_1}$ are different with high probability, the train BTM can identify this attack, and therefore drops the telegram replayed by the attacker.

IV. ROBUSTNESS OF CRF TO CHANNEL INDUCED ERRORS

We assume that error correction schemes on the uplink and downlink channels ensure error-free reception of the secret

TABLE I
DECISIONS TAKEN BY BALISE-i AND TRAIN-j WHEN ERRORS OCCUR IN UPLINK AND DOWNLINK. CASE 3 AND CASE 1 CAN BE DISTINGUISHED BY THE TRAIN AS IT RECEIVES SIGNALS CLOSE TO NOISE-FLOOR IN THE FORMER CASE.

Index	Downlink	Uplink	Action by Balise-i	Action by trains
Case 1	no error	no error	Balise increments c;	No additional action by the trains (follow the steps in the CRF protocol)
Case 2	no error	error	Balise increments c;	Train-p passes its encrypted message $E_{k_i}(p)$ to Train-$(p+1)$ scheduled after it, for all $p \geq j$
Case 3	error	no error	Balise maintains existing c; does not transmit uplink telegram	Train-p passes its encrypted message $E_{k_i}(p)$ to Train-$(p+1)$ scheduled after it, for all $p \geq j$
Case 4	error	error	Balise maintains existing c; does not transmit the uplink telegram	Train-p passes its encrypted message $E_{k_i}(p)$ to Train-$(p+1)$ scheduled after it, for all $p \geq j$

keys and the random signals. Despite using error-correction schemes, uplink and downlink telegrams may not satisfy the check-sum criterion (akin to cyclic redundancy checks) at either side due to channel induced errors. Therefore, it is paramount to validate the correctness of our algorithm under errors in uplink and downlink communication. Upon errors, a complete summary of decisions taken by balises and trains in our scheme are listed in Table I. In a nutshell, if the balise receives an erroneous downlink telegram, then the state of the balise remains the same, i.e., the variable c continues to store the index of the previous train, while the uplink telegram is not transmitted. On the other hand, if uplink telegrams are received with errors, then starting from the current train, each train in the chain passes its message $E_{k_i}(\cdot)$ to the train scheduled after it. In a real-world implementation of CRF, either the trains themselves can transmit these encrypted secret keys to the next one through a secure link, or the control center can facilitate this update process through a secure link. With this modification, our protocol continues to work consistently amidst channel noise. Case 3 and Case 1 can be distinguished by the train as it receives signals close to noise-floor in the former case. It is important to note that our idea of allowing successive excitation of the balise by the same encrypted message is crucial for providing resilience against channel introduced errors in uplink and downlink. For example, Case 2 and Case 4 take advantage of the fact that multiple successive excitation of the balise is permitted using the same encrypted message. In this section, we have shown that the trains will have to forward their encrypted messages due to random errors in the channel. However, if the error correction mechanism is carefully chosen, then the probability of receiving erroneous telegram in downlink/uplink can be very small, and hence the overhead of forwarding the encrypted messages between the trains is negligible. One way of reducing the overhead of forwarding the encrypted messages is to modify the distribution method of Section II-A. Instead of distributing the encrypted indices to the trains, the control center can provide the keys $\{k_i \mid \forall i\}$ to each train along with an index. Then the train can generate the encrypted messages on-board dynamically. With such a modification, under channel errors, the control center can direct some trains to reduce their index values by broadcasting 1 bit of information.

V. IMPLEMENTATION ASPECTS OF CRF

One of the fundamental aspects to address in a challenge-response based strategy in transportation systems is latency. For example, when a train traveling at 120 kmph, crosses a 0.5-metre wide balise, the two radio devices (the train BTM and the balise) lie within coverage for about 15 milliseconds (referred to as coverage time) [10]. Within those tens of milliseconds, any bi-directional authentication protocol must ensure sufficient time for the authentication process in addition to facilitating accurate reception of downlink and uplink signals. In our application, downlink frames are shorter in length than the uplink ones; the former carries only nonce values and the encrypted version of the counter values, whereas the latter contains packet information, such as balise group identity, individual balise identity (balise-ID), location information, permissible load of the track, and so on.

We now discuss the processing time available at the balises in order to execute the proposed challenge-response strategy. We assume that 128 bits are allocated for $E_{k_i}(j)$ and k_{TB}, which together contributes 256 bits. After incorporating some additional bits for parity checks, and replicating this frame 3 times (for error correction purpose), the downlink frame can reach upto 1000 bits in size. With 1 MHz bandwidth, downlink communication takes roughly about 1 ms. Since the air-gap is 30 centimetres, the propagation delay is about 1 nanosecond, which we henceforth neglect in our calculations. On the other side of the link, since the uplink packet size is about 1200 bits (1023 bits for the packet in long format and additional bits for MAC), after replicating the frame 3 times (for error correction purpose), uplink communication consumes about 3.6 ms. Assuming that data processing at the balise is not performed in a pipelined manner (which is a conservative assumption), the combination of downlink and uplink communication contributes about 4.6 ms delay. For the specific example of 15 ms coverage time, this leaves 11.4 ms for decoding, authentication, encryption and encoding operations at the balise. With these numbers, we believe that ef-

TABLE II
PROCESSING TIME AVAILABLE AT THE 0.5-METRE WIDE BALISE. THE SIZES OF DOWNLINK AND UPLINK FRAMES ARE 1000 AND 6000 BITS, RESPECTIVELY.

Train velocity	Coverage time	Communication bandwidth	Available processing time
50 kmph	36.2 ms	1 MHz	31.6 ms
		5 MHz	35.28 ms
		10 MHz	35.74 ms
100 kmph	18.1 ms	1 MHz	13.5 ms
		5 MHz	17.18 ms
		10 MHz	17.64 ms
200 kmph	9 ms	1 MHz	4.4 ms
		5 MHz	8 ms
		10 MHz	8.5 ms

ficient implementation of standard and lightweight encryption algorithms [9], [11] on moderate-speed processors can help us achieve the latency requirement. In [2], the authors have compared the performance of various encryption algorithms on a wide range of processors, and those results show that our constraints on the available processing time can be met provided the processors are suitably chosen. In general, Table II lists the available processing times with different values of communication bandwidth, and train velocity. The numbers listed in the table show that by increasing the communication bandwidth, it is possible to authenticate high-speed trains as well.

VI. SUMMARY

Replay attack is a relevant threat to the current-day balise-based train control systems, wherein an attacker attempts to replay the telegrams of the balises at different positions on the track so as to misguide the train about its position. To mitigate such attacks, we have proposed a challenge-response based strategy using cryptographic random fountains. Our scheme forbids unnecessary excitation of balises from attackers especially on balises mounted in remote places of the rail-route.

ACKNOWLEDGEMENTS

This work was supported by the National Research Foundation (NRF), Prime Ministers Office, Singapore, under its National Cybersecurity R&D Programme (Award No. NRF2014NCR-NCR001-31) and administered by the National Cybersecurity R&D Directorate and by the Human-Centered Cyber-physical Systems Programme at the Advanced Digital Sciences Center from Singapore's Agency for Science, Technology and Research (A*STAR).

REFERENCES

[1] Balise picture. www.siemens.com/press/photo/soimo201110-03e.
[2] Encryption performance. http://csrc.nist.gov/archive/aes/round1/conf2/Schneier.pdf.
[3] Ertms/etcsclass 1, fffis for eurobalise, ref. subset-036, ver 2.4.1. http://www.era.europa.eu/document-register/documents/set-1-index009-subset-036%20v241.pdf. Accessed: Sept. 2007.
[4] Shenzhen signalling issse. http://www.scmp.com/news/china/article/1078165/passenger-wi-fi-freezes-third-shenzhen\\-metro-train-week. Accessed: Nov. 2012.
[5] B. Allotta, V. Colla, and M. Malvezzi. Train position and speed estimation using wheel velocity measurements. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 216(3):207–225, 2002.
[6] A. Costin and A. Francillon. Ghost in the air (traffic): On insecurity of ads-b protocol and practical attacks on ads-b devices. *Black Hat USA*, pages 1–12, 2012.
[7] S. Dhahbi, A. Abbas-Turki, S. Hayat, and A. El Moudni. Study of the high-speed trains positioning system: European signaling system ertms/etcs. In *2011 4th International Conference on Logistics*, pages 468–473. IEEE, 2011.
[8] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer. Strong authentication for rfid systems using the aes algorithm. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 357–370. Springer, 2004.
[9] P. Hamalainen, T. Alho, M. Hannikainen, and T. D. Hamalainen. Design and implementation of low-area and low-power aes encryption hardware core. In *9th EUROMICRO Conference on Digital System Design (DSD'06)*, pages 577–583. IEEE, 2006.
[10] R. Hornstein, M. Pottendorfer, and H. Schweinzer. Critical demands of data transmission between trains and trackside infrastructure. *IFAC Proceedings Volumes*, 38(2):99–106, 2005.
[11] M. Katagi and S. Moriai. Lightweight cryptography for the internet of things. *Sony Corporation*, pages 7–10, 2008.
[12] M. Malvezzi, G. Vettori, B. Allotta, L. Pugi, A. Ridolfi, F. Cuppini, and F. Salotti. Train position and speed estimation by integration of odometers and imus. In *9th World Congress on Railway Research, Lille, France*, pages 22–26, 2011.
[13] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun. On the requirements for successful gps spoofing attacks. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 75–86. ACM, 2011.
[14] Y. Wu, J. Weng, Z. Tang, X. Li, and R. H. Deng. Vulnerabilities, attacks, and countermeasures in balise-based train control systems. *IEEE Transactions on Intelligent Transportation Systems*.
[15] Z. Xu, W. Wang, and Y. Sun. Performance degradation monitoring for onboard speed sensors of trains. *IEEE Transactions on Intelligent Transportation Systems*, 13(3):1287–1297, 2012.