A Random Access Procedure Based on Tunable Puzzles

Mats Näslund*, Elena Dubrova†, Göran Selander*, Fredrik Lindqvist*
†Ericsson Research, 164 80 Stockholm, Sweden
{mats.naslund,goran.selander,fredrik.lindqvist}@ericsson.com
*Royal Institute of Technology, 164 40 Stockholm, Sweden
dubrova@kth.se

***Abstract*—In a radio network, a denial-of-service attack or an attach storm after a temporary outage may cause severe access network overload. Unavailability of radio network services for its subscribing users causes dissatisfaction among the users and should be prevented. The problem is likely to become even more acute with the growth of Internet-of-Things applications that are expected to support critical infrastructure. In this paper, we present a new random access procedure based on tunable puzzles. Tunable puzzles provide the means to balance the load on the access network, prioritize certain devices, and localize radio resources for subsequent transmissions. By tuning the difficulty of puzzles, a base station can control the period of time before a device can send its next message. The prioritization by means of puzzles creates considerably less extra load on the base station compared to other alternatives, e.g. by using authentication. Encoding of radio resources in the puzzle solution enables a more efficient use of communication and processing resources. In addition, it gives malicious devices no incentive to guess the solution, since any solution other than the intended one fails to convey the information enabling the device to proceed further.**

***Index Terms*—random access; denial-of-service; overload; computational puzzle.**

I. Introduction

In a radio network, a Random Access CHannel (RACH) is used as the common entrance for all devices which request a dedicated communication channel. This includes devices which are accessing the network for the first time as well as those which have already been using the radio network but have temporarily lost their synchronization towards it. A first message exchange between the device and a base station is carried out over the physical RACH where the bandwidth of an uplink is very limited. In an "attach storm" scenario, this may cause heavy load on the RACH as well as on the base station itself. A malicious or malfunctioning device could bypass existing procedures and cause a denial-of-service attack by exhausting channel or processing resources, by just overusing the existing attachment procedure. Such attach storms may also occur naturally, e.g. when thousands of devices located in a dense "hot spot" attempt to reconnect after a temporary outage.

Unavailability of radio network services for its subscribing users obviously causes dissatisfaction among the users and should be prevented. The problem is likely to become even more acute with the growth of Internet-of-Things applications that are expected to support critical infrastructure such as smart power grid, water supply systems, or traffic control. In such applications, unavailability of radio network services may have catastrophic consequences for public health or safety, national economy, or national security. Increasing the radio resources is not a good solution considering their scarce nature, nor is adding processing capacity to the base stations as it is expensive.

In an overload situation, it might be desirable to allow devices that have been connected to the radio network but temporarily lost synchronization a faster access to the radio network compared to the devices which have not yet connected to or established network services. The former type of devices may have ongoing conversation or data transfer and therefore a fast reconnection might reduce user dissatisfaction.

In this paper, we present a new method for providing a device access to the network which helps preventing overload and also gives a base station a possibility to prioritize different devices. The method consists of three steps. In order to access a network, a device sends to a base station an access request containing a preamble which is selected according to the device priority type. In response to the access request, the base station constructs a computational puzzle based on the received preamble and sends the puzzle to the device. The devices solves the puzzle and its solution gives the device information on communication resources to use in subsequent signaling to the base station. Solving the puzzle requires a computational effort, which introduces a delay for the device in the access procedure during which the solution is computed. The difficulty of the puzzle may be controlled by the base station, thereby adjusting the delay and thus smoothening network load over time.

A. Related work

Computational puzzles, also referred to as *client puzzles*, has been proposed in the context of Internet Protocol (IP)-based networks with the purpose of preventing abuse of server resources [1], [2], [3], [4], [5]. A puzzle aims at causing a small additional computational load on a client, thereby creating a period of "idle time" for a server. The client is required to present the puzzle solution before it is allowed to proceed. The puzzles are created in a computationally

”asymmetric” fashion: it is easy for the server to create new puzzles, but somewhat hard for the client to solve them. The hardness can usually be tuned.

The predominant way to implement puzzles is via some cryptographic hash function, F. The server chooses a value s, sends the puzzle $p = F(s)$ to the client and asks the client to find the solution s. If s is random and n bits in size, finding the solution requires of the order of 2^n hash function computations for the client. On the other hand, it requires only one hash function computation for the server to verify.

A drawback of existing puzzle mechanisms is that the client can simply guess a solution without actually performing any computation and respond with this guess. While the guess is most likely wrong, it still causes load on the server and the network, since the server must receive the solution and verify its correctness. Transporting and verifying the solution of a puzzle creates an overhead on both, the communication and the processing.

Another drawback is that the difficulty of solving a puzzle is made the same for all clients, i.e. existing puzzle mechanisms provide no means to prioritize certain clients. This is because the puzzle is created *before* the identity of a client has been established. Therefore nothing prevents a client from falsely claiming that it belongs to a set of prioritized clients that should be assigned an ”easy” puzzle.

Finally, existing puzzle mechanisms do not assign any information to the puzzle solution, besides possibly a single binary bit informing that ”the solution is valid”, which is wasteful.

B. *Our contribution*

The paper has three contributions. First, we demonstrate usability of puzzles in the context of radio networks, as a way to balance the load on a base station. This is a new application of puzzles.

Second, we generalize puzzles in a way which enables a base station to give priority to a certain types devices. These improvements are applicable for both, radio networks as well as wired networks, e.g. IP-networks. The prioritization by means of puzzles creates considerably less extra load on a base station compared to other alternatives for implementing prioritization, e.g. by using authentication.

Third, we introduce a way to encode auxiliary information into a puzzle solution without making the puzzle harder to solve generate. As a result, we can control the difficulty of a puzzle independently of the number of encoded auxiliary bits and also we do not put extra load on the party generating the puzzles. In the context of radio networks, the puzzle solution specifies on what radio resource the next message should be sent. The device requesting access thus needs to determine the solution to the puzzle in order to continue the access procedure. An advantage of the presented method is that malicious devices has no incentive to guess the solution, since any solution to the puzzle other than the intended one fails to convey the information enabling the device to proceed further.

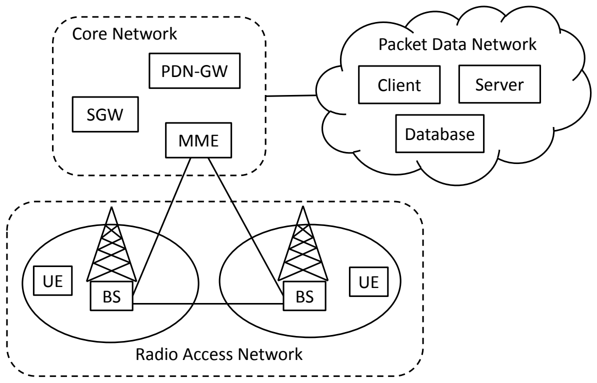


Fig. 1: General structure of an LTE radio access network.

We describe the new random access method in the context of a state-of- the-art cellular radio access network, namely Long Term Evolution (LTE). However, the method is not limited to LTE only. Its can be applied in other wireless access networks as well as to non-wireless access networks.

The rest of the paper is organized as follows. In Section II, we describe the structure of an LTE radio access network and the conventional random access procedure. In Section III, we present new random access procedure based on tunable puzzles. In Section IV we show how tunable puzzles can be constructed. In Section V we evaluate the probability of occurrence of multiple solutions. Section VI concludes the paper and discusses open problems.

II. Background

In this section, we describe the structure of an LTE radio access network and the conventional random access procedure [6]. A reader familiar with LTE can skip this section.

A. *Structure of a cellular wireless network*

Figure 1 show the structure of an LTE cellular wireless network in which the presented method can be implemented.

An LTE radio access network consists of a Radio Access Network (RAN) and a core network. The RAN contains of a number of Base Stations (BS) communicating with the mobile terminals and with each other over an interface. Each base station covers one or more geographical areas, called *cells*, within which the wireless communication is provided to the mobile terminals residing in this cell. A communication link from a terminal to a base station is called *uplink* (UP), and the reverse link from the base station to the terminal is called *downlink* (DL). Throughout the paper, we refer to mobile terminals as User Equipment (UE), or simply devices.

The Core Network (CN) consists of nodes such as Mobility Management Entity (MME), Serving Gateway (SGW) and Packet Data Network Gateway (PDN-GW). These nodes are connected to the base stations of the RAN by an interface. The core network provides terminals connectivity to an external Packet Data Network (PDN) which contains servers, databases or other entities.

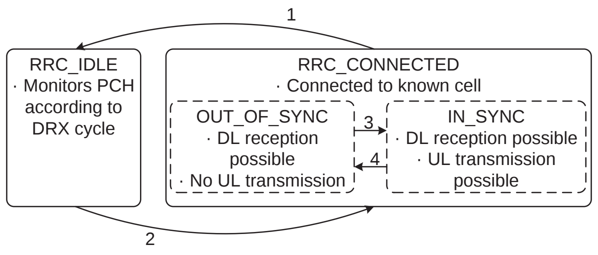


Fig. 2: The state machine of LTE.

Note that the presented method can also be applied in a wired network settings. For example, within the PDN a client device may seek a wired access to a server and the presented method can be used also for supporting such a communication.

B. *LTE state machine*

In the LTE, two main states of devices are RRC_IDLE and RRC_CONNECTED, where RRC stands for Radio Resource Control (see Figure 2). The role and characteristics of states can be summarized as follows.

In RRC_IDLE state, the device does not belong to any cell. No RRC context has been established yet. The device is out of uplink synchronization with the base station. No uplink data transmission is possible (with the exception in RACH). The device monitors a paging channel (PCH) according to a discontinuous reception (DRX) cycle.

In RRC_CONNECTED state, the cell to which the device belongs is known. RRC context has been established. Necessary parameters for communication are known to both, the device and the network. The Cell Radio Network Temporary Identifier (C-RNTI), which is an identity of the device used for signaling purposes between device and network, has been configured.

In RRC_CONNECTED state, the device can be either synchronized with the base station (IN_SYNCH), or be out of synchronization (OUT_OF_SYNCH). When IN_SYNCH, the device is able to receive in downlink and transmit in uplink. When OUT_OF_SYNCH, the device is able to receive in downlink, but cannot transmit in uplink. The device can move between these to states as indicated by arrows in Figure 2.

In order to move a device from RRC_IDLE to RRC_CONNECTED the RACH state, the random access procedure described in the next section has to be carried out. From the energy saving perspective, as well as for saving network memory capacity that keeps the context of the devices, it is advantageous to let the device move back to the RRC_IDLE state after desired transmission has been finalized in the RRC_CONNECTED state.

C. *Conventional random access procedure*

There are several reasons for performing a random access in a cellular wireless network. For the case of LTE, they are:

- To obtain initial access, i.e. move from state RRC_IDLE to RRC_CONNECTED.
- To re-establish a radio link after a radio link failure.

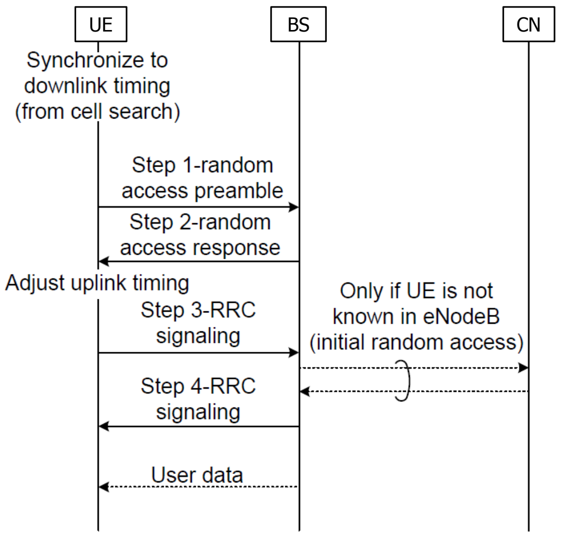

UE
BS
CN
Synchronize to downlink timing (from cell search)
Step 1-random access preamble
Step 2-random access response
Adjust uplink timing
Step 3-RRC signaling
Only if UE is not known in eNodeB (initial random access)
Step 4-RRC signaling
User data

Fig. 3: Steps of the conventional random access procedure.

- To establish uplink synchronization if uplink/downlink data arrives when the device is in RRC_CONNECTED state and uplink not synchronized.
- For handover when uplink synchronization needs to be established to the new cell.
- To enable a scheduling request channel if no dedicated resources have been configured for the device on the uplink control channel.

Figure 3 illustrates different steps involved in the conventional LTE random access procedure.

Step 1: In step 1, the device randomly selects one access preamble from a known set of preambles transmitted by the network through the broadcast channel. The purpose is to avoid collisions by separating the preamble signals in the code domain. In LTE there are typically 64 different available preambles in each cell which in turn may be divided into two groups. The grouping allows the device to signal with one bit whether it needs radio resources for a small or large message (data package). That is, a randomly selected preamble from one group indicates that the device has a small amount of data to send, while a preamble selected from another group indicates that resources for a larger amount of data are needed.

The random access preamble is transmitted by the device only on certain time/frequency resources which are made known to all devices via the broadcast channel. Upon reception, the base station detects all non-colliding preambles and estimates the roundtrip time for each device. The latter is of great importance because LTE is an Orthogonal Frequency Division Multiplexing (OFDM)-based system which requires time and frequency synchronization in both downlink and uplink.

Step 2: In Step 2, the random access response from the base station to the device carries the following information:

- the roundtrip time,
- temporary device identity,
- uplink resources for device to use in Step 3.

The random access response is scheduled on downlink shared channel and is indicated on a downlink control channel using an identity reserved for random access responses. The received roundtrip time allows the device to adjust its transmission window in order to be synchronized in the uplink, as required by OFDM transmission. All devices that have transmitted a preamble monitor downlink control channels for a random access response within a configurable time window. This time window is configured by the base station. If the device does not detect a random access response within the time window, it declares the attempt as failed, and repeats Step 1 using an increased transmitting power.

The received uplink resource assignment to be used in Step 3 is essentially a pointer to the time/frequency resource grid that informs the device exactly which subframes (time) to transmit on and what resource blocks (frequency) to use. A particular example of such pointer may comprise pointing to resource block number 10 four subframes later relative the time instance of the received random access response. The 3GPP LTE specification [7] describes in detail the format of this message and how it shall be interpreted. The message is designed to minimize the number of bits required to convey the resource assignment, but at the same time to provide some flexibility for a scheduler of the base station when deciding upon the resource assignment. The length of the message depends on the system bandwidth. For example, for a 5 MHz system, the entire random access response message is in the order of 80-160 bits, out of which 80-160 bites are used to indicate the radio resource to use in Step 3. The number of bits can be increased by increasing time or modulation order. The number of radio resources can be further expanded by considering fractions of timeslots.

Step 3: Upon correct reception of the random access response in Step 2, the device is now time synchronized with the base station. Before any transmission can take place, a unique identity C-RNTI is assigned. The device transmission in this step uses the uplink channel on the radio resources assigned in Step 2. Additional message exchange might also be needed depending on the device state, as indicated in Figure 3 by the arrows drawn with dashed lines. In particular, if the device is not known in the base station, then some signaling is needed between the base station and the core network.

Step 4: A main purpose of this step is contention resolution, i.e. to resolve any random access response collisions that were not detected before. This step is not relevant for the present method and will not be described further.

III. Random Access Procedure Based on Tunable Puzzles

In this section, we present the new random access procedure based on tunable puzzles. The construction of tunable puzzles in described in detail in Section IV.

We propose to regulate the load on a base station by giving devices requesting to be connected a computational task to perform before they are allowed to continue. We replace uplink resource assignment in Step 2 of the access procedure in Figure 3 by a puzzle that requires some computational effort to be solved and thus delays the exchange of messages.

Puzzles can be designed to be of different degree of difficulty. Thus, we can tune the computational effort required by a device depending on the load of the base station or the number of detected preambles. This allows us to control the period of time before the device can send the message in Step 3 of the access procedure.

A central idea of the presented method is to use the puzzle solution as a carrier of information about the radio resource to be used by the device in Step 3 of the access procedure. Without solving the puzzle, the device does not know when and how to send subsequent messages. Apart from more efficient use of communication and processing resources, this also provides an additional level of protection for the base station since now malicious devices cannot easily disturb the communication for one particular device without jamming all radio resources that are used in Step 3. Thus, we can avoid previous work problem with devices that continue to cause load to the base station by forcing the base station to receive erroneous puzzle solutions. In our case, the device must solve the puzzle correctly in order to be able to continue communicating with the base station.

An addition, the presented method enables a base station to give priority to a certain types devices. In an overload situation, it is desirable to prioritize the devices in RRC_CONNECTED state and to make the devices in RRC_IDLE state to back-off for a certain time period. This enables out-of-synch RRC_CONNECTED devices to repeat step 1 with a smaller likelihood of collision, assuming that the back-off time is larger than the time to the next physical RACH opportunity.

In the next section we describe how the prioritization can be implemented in the context of radio networks.

A. Implementation of prioritization

We propose to implement the prioritization by partitioning the set of access preambles into two subsets, P_C and P_{NC}:

1) P_P is the set of prioritized preambles intended for RRC_CONNECTED devices,
2) P_{NP} is the set of non-prioritized preambles intended RRC_IDLE devices.

In step 1 of the access procedure, a device selects a preamble in the relevant set and sends it to a base station. Devices which possess a key are expected to select a preamble from the prioritized set of preambles. Devices without a key are expected to select a non-prioritized preamble. Obviously, it is possible for a malicious device to attempt to falsely claim that it is in RRC_CONNECTED state, aiming to quickly get access. To mitigate such attempts, we propose to use two different kinds of puzzles:

1) Z_k is the set of tunable puzzles which can be solved with an access to a key only.
2) Z_{nk} is the set of tunable puzzles which can be solved without a key.

We explain how tunable puzzles are constructed in Section IV.

If in step 1 of the access procedure, a base station receives a preamble from the set P_P and replies with a puzzle from the set Z_k. The puzzle key is shared with the device when the device is in RRC_CONNECTED state. If the base station receives a preamble from the set P_{NP}, it replies with a puzzle from the set Z_{nk}.

If a malicious device in RRC_IDLE state selects a preamble from the group P_P and sends to the base station in step 1, it will receive in Step 2 a puzzle from the set Z_k and will not be able to solve it without a key. So, making such false claim does not give any advantage to the device, on the contrary, it prevents the device from getting an easier puzzle from the set P_{NP}.

The tunable puzzles enable the base station to control the amount of radio resources allocated to devices in RRC_CONNECTED state and non-RRC_CONNECTED state. For example, the base station may choose to give access to devices in RRC_CONNECTED state only.

Note that the presented prioritization method can also be applied in settings which do not use preambles. In this case, instead of preambles, devices may provide some other type of identifiers or use some special message format which allows the network to distinguish their priority type.

In the next section, we describe how puzzle keys can be shared with devices in the RRC_CONNECTED state.

B. Establishing of puzzle keys

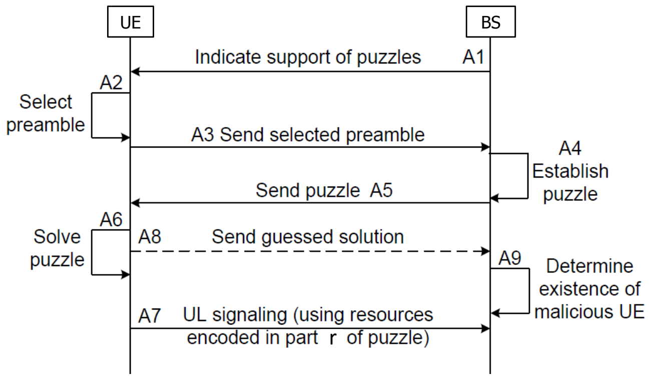
According to current 3GPP standards [8], as a part of attaining the RRC_CONNECTED state, the device establishes a secure (encrypted and integrity protected) communication channel with the network. This secure channel can be used for transporting and establishing puzzle keys.

The puzzle key can be transported after establishing that a device is in RRC_CONNECTED state and configured via the RRC signaling using RRC messages on a physical layer transported on a downlink control channel. This can be implemented as a new RRC message or as a piggyback or extension of an existing RRC message. That is, the base station may first establish that the device is in the RRC_CONNECTED state, e.g. by means of the base station having access to the device context. The base station may then send the puzzle key to the device e.g. in a new RRC message or as an extension of an existing RRC message.

Puzzle keys should be replaced on a regular basis to ensure that devices that have not been connected for a long time are not given priority, since the corresponding key is then revoked.

C. Integration of puzzles into signalling

Figure 4 shows steps of the proposed random access procedure based on computational puzzles.



UE
BS
Indicate support of puzzles
A1
A2
Select preamble
A3 Send selected preamble
A4 Establish puzzle
Send puzzle A5
A6
Solve puzzle
A8
Send guessed solution
A9
Determine existence of malicious UE
A7
UL signaling (using resources encoded in part r of puzzle)

Fig. 4: Steps of the presented random access procedure.

At arrow A1, the base station indicates its ability to support puzzles, e.g. by means of broadcasting this information. In relation to signaling of Figure 3, this step occurs before Step 1 in Figure 3, i.e. before sending of random access preamble.

At arrow A2, the device listens to the broadcast and sends (arrow A3) a preamble from the set P_P or P_{NP}, depending on its state.

At arrow A4, the base station creates a puzzle p according to the received preamble.

At arrow A5, the puzzle p is transported in the random access response. So, the radio resource specification in Step 2 of Figure 3 is replaced by a puzzle whose solution encodes the radio resource. The size of complete random access response in arrow A5 is about 80 bits for a 5MHz system and larger for systems with larger bandwidth. If a system is in an overload situation, it may be acceptable to use more bits for random access responses and thus include larger puzzles.

At arrow A6, the device solves the puzzle p. The solution give the device information about radio resources required to proceed further.

At arrow A7, the device that has correctly solved the puzzle p is able to transmit in uplink.

A malicious device may try to guess which resources to use rather than solve the puzzle (arrow A8). The base station may implement ways to determine existence of malicious devices (arrow A9), e.g. by scanning incorrect resources and interpreting the repetitive use of such resources as an indication that the device is guessing the solution.

If the presented method is implemented in a system which also serve legacy devices, some consideration of backwards compatibility will be necessary. That is, legacy devices which do not support the puzzle mechanism should still be allowed to access the network. One way to achieve this is to use bits reserved for future use to specify reserved frequencies for devices supporting the puzzle mechanism. This may be complemented with extending the group of preambles, such that new devices will support two additional categories of radio resources or preambles, whereas legacy devices will only use the previously defined preambles and radio resources without interfering with the presented method. A broadcast message can be used to announce that puzzles are supported.

IV. Construction of Tunable Puzzle

As we mentioned in Section I-A, the use of client puzzles for denial-of-service mitigation in Internet communication is known. However, the use of puzzles as a means of:

1) Prioritizing certain devices, and
2) Localizing a physical destination address in general, and radio resource in particular,

is to our best knowledge new. To enable these two features, we generalize puzzles to the form

$$p = F(k \,\|\, d \,\|\, r), \tag{1}$$

where $k \,\|\, d \,\|\, r$ is the solution in which k is a secret key possessed by prioritized devices only, d is the bit string which controls difficulty of the puzzle, and r is the bit string which encodes the auxiliary information.

The set Z_k of tunable puzzles which can be solved only with a key contains puzzles of type (1) in which the bit string k is non-empty. The set Z_{nk} of tunable puzzles which can be solved without a key contains puzzles of type (1) in which the bit string k is empty.

The basic requirements for the function F are:

- It should be easy for the access network to generate or feasible to pre-calculate p.
- The puzzle size should be set according to the available bandwidth in Step 2.
- The function F should be easy to compute, but hard to invert.

The standard cryptographic hash functions [9] usually have a pre-defined number of input and output bits in the order of 100-500 bits, i.e. for them the solution space is infeasible to search exhaustively. Similarly to previous work, to adjust the number of input/output bits to the desired difficulty of a puzzle, we can truncate the size of the puzzle p and the solution $s = (k \,\|\, d \,\|\, r)$ as follows:

$$a \,\|\, p = F(b \,\|\, s) \tag{2}$$

where a and b are fixed bit strings. For example, the task can be to find s such that $F(s) = 00\ldots0 \,\|\, p$. Alternatively, the task could be to find s such that $F(b \,\|\, s) = p$, where b is a fixed bit string known to the device.

The constants a and b may be fixed for all random access requests to constants defined by a standard (e.g. $a = 000...0$), or specified by the standard to have a certain value which is dependent on the actual radio setup or communication setup during the message exchange between a device and a base station. For example, the constants may be dependent on the actual random access request, on data that both the device and the base station know, e.g. related to the preamble, or other radio or network parameter. Common to a and b is that they are known by the device, base station, and network and thus need not be guessed or calculated.

A. *AES-based tunable puzzles*

In this section we show how tunable puzzles can be constructed on the base of AES encryption algorithm [10].

AES takes a key K of length n and a message M of length m and transforms them into a ciphertext C of length m:

$$\mathrm{AES}(K, M) = C.$$

Key sizes $k =$128, 192, and 256-bit can be used.

For a given message M, the function

$$F_M(K) = \mathrm{AES}(K, M) \tag{3}$$

behaves like a hash function. One one hand, to create a meaningful puzzle based on AES, the whole AES key K cannot be revealed since the devices knowing the AES key will hardly need to perform any computations at all, just a single AES decryption. On the other hand, in order to make the puzzle possible to solve in a reasonable time, the whole AES key cannot be completely unknown either. In addition, we want to give the bast station a possibility to control the computational effort required to solve a puzzle depending on the device prioritization type.

As a solution, we propose to split the AES key K into three parts:

$$K = b \,\|\, k \,\|\, d \tag{4}$$

where b is a fixed bit string known to all devices, k is the puzzle key known to prioritized devices only, and d is the bit string whose length determines the difficulty of the puzzle.

The message M is used as a carrier of auxiliary information which may be padded with a constant, if needed:

$$M = c \,\|\, r.$$

where c is a fixed bit string and r is the bit string representing the auxiliary information.

The resulting AES-based tunable puzzle is:

$$a \,\|\, p = \mathrm{AES}(b \,\|\, k \,\|\, d, c \,\|\, r) \tag{5}$$

where a is a fixed bit string. Any of the bit strings a, b, c may be empty, but not all at the same time.

The solution to the puzzle (5) is any bit string $s = (k \,\|\, d \,\|\, r)$ such that $a \,\|\, p$ decrypts to $c \,\|\, r$ with the AES key $b \,\|\, k \,\|\, d$. To solve the puzzle, prioritized devices have to determine the bit string d only, while malicious non-prioritized devices which pretend to be prioritized (by selecting the preamble from the the set P_P) have to determine $k \,\|\, d$. Note that the difficulty of the puzzle is independent on the number of bits in r.

The base station encodes the relevant auxiliary information in r, chooses the length of d depending on the network load, and then substitutes k, d and r into equation (5) to generate the puzzle. Puzzles of type (5) are easy to generate efficiently, since they are just the encryption of $c \,\|\, r$ using the AES key $b \,\|\, k \,\|\, d$. Note that the presented way of encoding auxiliary information r into the puzzle solution does not increase the difficulty of generating a puzzle.

As an example, suppose the puzzle solution is supposed to encode uplink grant information in a radio access network. In the LTE standard as defined in 3GPP TS 36.213, the uplink grant is a 20-bit binary string consisting of:

- Hopping flag, 1 bit
- Fixed Size Resource Block (FSRB) assignment, 10 bits
- Truncated modulation and coding scheme, 4 bits
- TPC command for scheduled PUSCH, 3 bits
- uplink delay, 1 bit
- CSI request, 1 bit

The FSRB defines the frequency which the device should use in subsequent signalling to the base station. Note that no time slot information is present in the uplink grant. This is because in the current LTE standard this information is implicit and already known to the device. Therefore, in order to obtain the desired off-load of the base station, a time slot information has to be appended to FSPS. It can be represented by an additional 10-bit string denoted TS. The resulting 20-bit binary string FSRB $\|$ TS encodes the information about radio resource and can be used as r in the puzzle solution. Next, the base station can append a suitable constant c, e.g. a fixed bit string such as $00\ \ldots 0$, and use $c \| r$ as one of the input parameters to create the puzzle. Alternatively, the whole 20-bit uplink grant value concatenated with with TS can be used as r.

Next, the second input parameter defining the puzzle, $b \| k \| d$, is constructed. For prioritized devices, the base station will choose d randomly and concatenate it to the puzzle key k and a fixed or publicly known string b, e.g. $b = 00\ldots0$. For non-prioritized devices, the base station will choose $k \| d$ randomly and concatenate it to a fixed or publicly known string b. In both cases, the number of bits in b will be chosen depending on the desired puzzle difficulty.

Finally, the base station computes the puzzle by encrypting $c \| r$ using the AES key $b \| k \| d$.

The auxiliary information bit string r of the puzzle solution can be used not only for localizing a radio resource, but also for localizing any physical destination address in general. For example, we can use r to encode a Uniform Resource Locator (URL) to which the device should direct a subsequent request for streaming some content such as a movie or music file, say URL = "www.example.com". Or we can use r to encode a password. The base station will choose a suitable representation of the URL/password as a binary string, e.g. it can be the corresponding sequence of ASCII characters, and pad it with a constant c. The resulting bit string $c \| r$ will be used one of the input parameters to create the puzzle. The second parameter $b \| k \| d$ will be selected in the same way as described above.

V. Analysis of Uniqueness of Solution

In tunable puzzles of type (5), and more generally of type (2), the hash function F is typically not one-to-one. Several (b,s) values can give the same (a,p) output.

In known previous work puzzles, multiple solutions are acceptable because a solution does not carry any semantic meaning, so it is not used after it has been verified. In the presented method, however, multiple solutions are undesirable because the bit string r provides information required to access a physical medium. For the case of a wireless network, a device which finds a wrong solution will transmit on a wrong radio resource and will not be heard by the base station. In addition, transmission on a wrong radio resource may potentially cause interference with other devices.

One possibility to solve this problem is to impose some order on solutions. For example, if it is easy to generate the lexicographically smallest solution, then we could define such solution to be the canonical solution to the puzzle. If puzzles are based on a modular arithmetic function, the canonical solution can be defined as having some specific number theoretic properties, e.g. belonging to an interval, or being a quadratic residue, etc.

However, canonical solutions put a burden on the party generating puzzles, since to check canonicity, an exhaustive search needs to be performed. For example, if we define the lexicographically smallest solution to be canonical, then, for any solution s of a puzzle p, we need to check if there is no $s' < s$ such that s' is a solution to p.

To avoid putting extra burden on the party generating puzzles, instead of avoiding multiple solutions, we propose to make the probability of their occurrence negligibly low. As we show next, this can be done by selecting the parameters of the puzzle appropriately.

Let $|x|$ denote the number of bits of the bit sting x. First, consider the case of $|r| = 0$. Then, there are $2^{|k|+|d|}$ different solutions to a puzzle of type (2). Assuming F behaves randomly, the probability that any specific solution maps to $a \| p$ is $1/2^{|p|}$ for any fixed a. Hence, for any fixed a,b, the expected number of collision pairs, i.e. pairs $(k \| d, k' \| d')$ such that $F(b \| k' \| d') = F(b \| k' \| d')$, is approximately

$$E(\#\text{collision pairs}) \approx 2^{2(|k|+|d|)-|p|}.$$

If $|r| \neq 0$, then we get $|r|$ more bits to choose. Since now the solution $s = (k \| d \| r)$ contains $|k| + |d| + |r|$ bits, the expected number of collision pairs is approximately

$$E(\#\text{collision pairs}) \approx 2^{2(|k|+|d|+|r|)-|p|}. \tag{6}$$

The equation (6) gives us a bound on the probability that a device finds a wrong solution and thereby transmits on the wrong radio resource. By solving the equation (6) for $s = (k \| d \| r)$, we get

$$|k| + |d| + |r| = \frac{|p| - E(\#\text{collision pairs})}{2}.$$

Now we can estimate how long should be the solution in order to get a negligible collision probability for a given puzzle size. For example, if the puzzle size is $|p| = 128$ bits and we require $E(\#\text{collision pairs}) = 2^{-40}$, then the solution should be $|k| + |d| + |r| = 44$ bits. These bits have to be partitioned between $|k| + |d|$ and $|r|$ so that, on one hand, $|k| + |d|$ is large enough to make a puzzle not too easy and, on the other hand, $|r|$ is large enough to fit the auxiliary information. For example, if we choose $|k| + |d| = 16$ bits, then we get $|r| = 28$ bits for the auxiliary information in this example.

Note also that if AES is used as basis for the function F, then the size of the bit string $b \,||\, k \,||\, d$ is limited to a few fixed values, namely 128, 192, and 256 bits. However, using the well-known Luby-Rackoff construction [11], it is possible to construct AES-based puzzles for any (even) $|b| + |k| + |d| < 128$.

A. Effect on legacy error probabilities

The presented method may potentially introduce two new types of errors: (1) failing to access the network and (2) disturbing other devices. To evaluate the effect of these errors on legacy error probabilities in LTE, we analyze the following two cases:

1) The probability of a device failing to access the network using the presented scheme is compared to the probability of a device failing to complete a legacy random access procedure.
2) The probability of a device using the presented scheme causing another device to fail is compared to the probability of a device becoming disturbed in the legacy solution.

Case 1: The block error rate target for the message sent at random access response step (Step 2 in Figure 2) is not standardized, but assumed to be of the order of 1%. There is no hybrid automatic repeat request because there is a risk of collision with message sent at random access preamble step (Step 1), meaning that a colliding device would then send a disturbing negative acknowledgement signal on the uplink, which must be prevented. The block error rate target for message sent at RRC signaling step (Step 3) is typically 10% but it is sent with hybrid automatic repeat request, so there are re-transmissions if message fails.

Case 2: When a device attempts to follow the presented method, but does not arrive at the correct radio resource, the device may send an erroneous message in RRC signaling step (Step 3) on a specific but incorrect time and/or frequency. The erroneous message will disturb a part of the sub-frame which may be dedicated for another device, having impact depending on the size of message. This will result in an increase of uplink intra-cell interference, which in turn may result in a re-transmission. However, considering that this disturbance is fairly short, it can be handled as a "normal" disturbance, and need not result in any failed messages at all.

In both cases, if the probability of using the wrong resource due to multiple solutions is set to 2^{-40}, the additional errors will be negligible.

VI. Conclusion

We introduced a new type of puzzles, called tunable puzzles, and proposed a novel random access procedure utilizing them. Tunable puzzles are used to:

1) balance the load on the access network,
2) prioritize devices that have been connected to the network but temporarily lost synchronization, and
3) localize radio resources on which the device should transmit to get access.

By tuning the difficulty of puzzles, the base station can control the period of time before a device can send its next message and smooth the network load over time. The prioritization by means of puzzles creates much less extra load on the base station compared to other alternatives for implementing prioritization, e.g. by using authentication. Encoding radio resources in the puzzle solution enables more efficient use of communication and processing resources and provides an additional level of protection from the malicious devices which cause load on the base station by sending guessed, erroneous puzzle solutions. In the proposed random access procedure, a device must solve the puzzle correctly in order to continue the communication with the base station.

Our idea of using puzzle solution as a carrier of information about a physical medium might find applications beyond radio networks, e.g. in the context of cyber-physical systems.

Acknowledgement

The second author was supported by the research grant No SM14-0016 from the Swedish Foundation for Strategic Research.

References

[1] A. Juels and J. Brainard, "New client puzzle outsourcing techniques for DoS resistance," in *Proceedings of Networks and Distributed Security Systems Symposium (NDSS'99)*. The Internet Society, 1999, pp. 151–165.

[2] T. Aura, P. Nikander, and J. Leiwo, "DoS-resistant authentication with client puzzles," in *Security Protocols*, ser. Lecture Notes in Computer Science, B. Christianson, J. A. Malcolm, B. Crispo, and M. Roe, Eds. Springer Berlin Heidelberg, 2001, vol. 2133, pp. 170–177.

[3] D. Dean and A. Stubblefield, "Using client puzzles to protect TLS," in *Proceedings of the 10th Conference on USENIX Security Symposium - Volume 10*, ser. SSYM'01. Berkeley, CA, USA: USENIX Association, 2001.

[4] B. Waters, A. Juels, J. A. Halderman, and E. W. Felten, "New client puzzle outsourcing techniques for DoS resistance," in *Proceedings of the 11th ACM Conference on Computer and Communications Security*, ser. CCS'04. New York, NY, USA: ACM, 2004, pp. 246–256.

[5] N. Borisov, "Computational puzzles as Sybil defenses," in *Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing*, ser. P2P'06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 171–176.

[6] E. D. P. Skold, *4G: LTE/LTE-Advanced for Mobile Broadband*, 2nd ed. Academic Press, Elsevier, 2013.

[7] 3GPP TS 36.213 The Mobile Broadband Standard, "Evolved universal terrestrial radio access (E-UTRA); Physical layer procedures," http://www.3gpp.org/dynareport/36213.htm.

[8] A. R. Prasad and S.-W. Seo, *Security in Next Generation Mobile Networks: SAE/LTE and WiMAX.*

[9] B. Preneel, *Analysis and Design of Cryptographic Hash Functions*. Ph.D. Thesis, KU Leuven, 1993.

[10] J. Daemen and V. Rijmen, "AES proposal: Rijndael," April 2003, National Institute of Standards and Technology.

[11] M. Luby and C. Rackoff, "How to construct pseudorandom permutations from pseudorandom functions," *SIAM Journal of Computing*, vol. 17, no. 2, pp. 373–386, 1988.