Personalized Mobile Services with Lightweight Security in a Sports Association

Jan Vossaert[1], Jorn Lapon[1], Bart De Decker[2], and Vincent Naessens[1]

[1] Katholieke Hogeschool Sint-Lieven, Department of Industrial Engineering
Gebroeders Desmetstraat 1, 9000 Ghent, Belgium
firstname.lastname@kahosl.be

[2] Katholieke Universiteit Leuven, Department of Computer Science,
Celestijnenlaan 200A, 3001 Heverlee, Belgium
firstname.lastname@cs.kuleuven.be

Abstract. This paper presents an attractive solution to integrate multiple services in the context of sports associations. The mobile solution is tailored to youngsters and makes use of a contactless RFID chip embedded in a bracelet. It realizes a reasonable trade-off between multiple (often conflicting) requirements, namely low-cost and low-power, security, privacy and flexibility to allow for easily adding new services.

Keywords: lightweight, security, privacy, mobility.

1 Introduction

Many sports associations already have their own Web site on which relevant information is publicly available. Examples are training and contest schedules, results, events, etc. Some of them also provide personalized Web pages. Password-based solutions have some major disadvantages. First, passwords offer weak authentication. Second, in many sports clubs, a majority of the members are children who often start at a very young age; hence, passwords are not appropriate for them. PKI based solutions offer a higher level of security but often complicate mobility. Many other services offered by sports associations could be made more user-friendly, more reliable and more secure by converting them into their digital versions. Examples are entrance control, access to lockers, consumption services, etc.

This paper presents an attractive solution to integrate multiple electronic services in the context of a sports association. The solution is tailored to youngsters and makes use of a contactless RFID chip embedded in a bracelet. It realizes a reasonable trade-off between multiple (often conflicting) requirements. First, the bracelet must be low-cost and low-power. Therefore, it should not require a battery. On the other hand, security and privacy are major concerns. For instance, locker and consumption services involve important security requirements. As sensitive personal data (such as contact information) is kept in the bracelet, privacy is also a crucial concern. Hence, cryptographic protocols will be required. However, to achieve a reasonable performance, the crypto primitives must be lightweight. Customized access control policies should be supported. The proposed solution must be flexible enough to allow for easily adding new services.

The rest of this paper is structured as follows. Section 2 points to related work. Section 3 lists the roles and initial services that are implemented. Moreover, the major non-functional requirements are described. The key infrastructure and initialization procedures are presented in section 4. Thereafter, three services are worked out in detail (see section 5). Section 6 evaluates the design and conclusions are given in Section 7.

2 Related Work

RFID tags are small, wireless devices to identify objects and people. They are proliferating in trillions due to the the dropping costs. A. Juels [1] gives an overview of existing RFID tag designs and their applications, and evaluates their security and privacy properties. They are classified according to two categories: *basic RFID tags* and *symmetric-key tags*. The former cannot perform cryptographic operations. Although the lack of cryptography is a big impediment to the security design, a few lightweight technical approaches can address certain security and privacy concerns. Some basic tags can be killed, put to sleep, blocked, etc. Moreover, distance measurement may prevent that RFID readers interrogate tags from large distance. Also, certain tags are protected with a PIN code. However, PIN codes can be eavesdropped and the distribution of the PIN might be problematic. Symmetric-key tags have richer security capabilities which are typically used for (mutual) authentication, and for the prevention of tracking/tracing and cloning attacks. Multiple security designs are presented in the literature [2,3,4]. However, major key management problems arise. Moreover, many solutions require a powerful back-end (i.e. a heavy-weight RF reader).

Current (wireless) smart cards support public-key operations. They are applicable in many domains with strong security and privacy requirements. For instance, many countries are rolling-out eID smart cards [5]. However, for the sports bracelet, low energy consumption is a key concern. Whereas ECC is already well-suited for resource-constrained devices, extremely low power optimizations exist for lightweight symmetric key operations. Moreover, the bracelet will be used in a closed environment, namely services within a sports association. Whereas a major advantage of a PKI is its applicability in open environments, symmetric-key solutions are acceptable to tackle advanced security and privacy concerns in closed environments.

Many lightweight cryptographic libraries are already available [6,7]. They provide implementations of lightweight cryptographic blocks (such as GRAIN, PRESENT, Trivium ...) or optimized implementations of traditional cryptographic blocks (such as DES, AES, ...). For instance, the IAIK crypto libraries [8] contain several implementations of the AES algorithm. They are often optimized for specific processor types, and designed for minimal energy consumption and/or high speed encryption.

3 Requirements

This section first lists the roles and services that need to be supported in the sports association. Thereafter, the infrastructural, security and privacy requirements are discussed.

Roles. An *athlete* (A) is a member of a sports club. After registration, each athlete receives a bracelet (br). The athlete can benefit from multiple services using this mobile companion. A *coach* (C) is responsible for a group of athletes (typically with the same age, grade, or level). An *administrator* (Ad) initializes and issues bracelets to athletes, and issues tokens to coaches. The administrator is responsible for maintaining the IT infrastructure.

Services. The services that are accessible using the bracelet are listed below:

– *Personalized Web service.* Athletes can log in on a personalized Web service by means of security tokens stored in the bracelet. The personalized Web pages provide detailed information about training and contest schedules (time and type), personal achievements, contest results, etc. Moreover, the athlete can submit personal information (personal physician, phone numbers, whereabouts, ...). Also, after login, the user can update information in the bracelet (personal achievements, phone numbers, ...) and download tokens to access lockers and to pay for drinks at the sports canteen.
– *Information retrieval service.* Athletes can specify an access policy and download it into the bracelet (using the Web service). Three levels are defined in the access policy. Level 0 defines which information may be read by any NFC (or at least RF) enabled device. No security tokens are required to access the information. For instance, some athletes are willing to release personal achievements to anyone. Level 1 defines which information may be released to certain roles of the club. For instance, phone numbers of relatives and/or physicians need to be read by coaches in case of injuries (especially in case of very young athletes). Coaches must have appropriate tokens on their mobile phone or PDA to be able to read the information (i.e. they must prove their role). Level 2 defines information that may only be released to certain individuals (i.e. more restrictive than roles).
– *Locker service.* Registered athletes have access to one or more free lockers. The maximum number of lockers that can be used simultaneously can depend on several parameters (registration fee, sport's type, and/or age to the athlete, ...).
– *Consumption service.* Each bracelet can contain a number of free or prepaid tickets for consumptions in the sport club's canteen. Those tokens can be downloaded from the Web site into the bracelet at regular times. In the rest of this paper, the consumption service is not described in detail. The protocols, however, are similar to the ones used to realize the locker service.

The major *non-functional requirements* are listed below:

- **Infrastructural requirements:**
  - *Related to constrained devices. Bracelets* and *lockers* must be low cost. Although they have a tamper-proof resistant part and have RF communication capabilities, they have limited processing power. Public key operations cannot be performed by the bracelets (i.e. they do not run efficiently). Each bracelet can also be (de)activated.
  - *Related to powerful devices.* Each coach must have a *mobile device* with NFC capabilities (or at least have an RFID reader) to read out information that is stored in the bracelet. Only few mobile devices currently support NFC[1]. Moreover, to log in with the bracelet on the Web server, athletes must have a *workstation* with an RFID reader (or NFC transceiver). The current generation of workstations often have these capabilities.
- **Security and privacy requirements:**
  - *Access control.* Athletes must be able to control access to the data stored in the bracelet.
  - *Location privacy.* It must not be possible for outsiders to track athletes.
  - *Reliability.* Appropriate back-up and revocation mechanisms must be implemented (in case of lost or stolen bracelets and mobiles).
  - *Controlled release of personal information.* Only a subset of information is released when the bracelet interacts with other components. The data that are released depend on the type of object and the privacy preferences of the user.
  - *Confidentiality.* Data cannot be eavesdropped by external attackers.

Security and trust assumptions. Redundancy is added to protect the integrity of messages. Constrained devices (e.g. bracelets and lockers) are trusted to release only genuine information to authenticated devices. Moreover, they only perform the protocols presented in section 5 (e.g. they do not participate in man-in-the-middle attacks,...).

Notation. $\mathsf{prf}_{\mathsf{K}}(\mathit{seed})$ defines a lightweight keyed pseudo-random number generator. It is used for generation of random numbers and symmetric keys on constrained devices. $\mathsf{MAC}(\mathit{key, message})$ defines a message authentication code algorithm.

4 Key Infrastructure and Initialization

Each sports association (see also Figure 1) owns a *trusted module (TM)*, and a set of lockers and bracelets. They all keep K_M. The latter is stored in a tamperproof part of each device. A secret key K_{Ad} and a capability CAP_{Ad} are also issued to *an administrator (Ad).* These are – amongst others – required to run certain applications on the trusted module. The trusted module is used for:

[1] The Nokia 6212 was selected for the prototype.

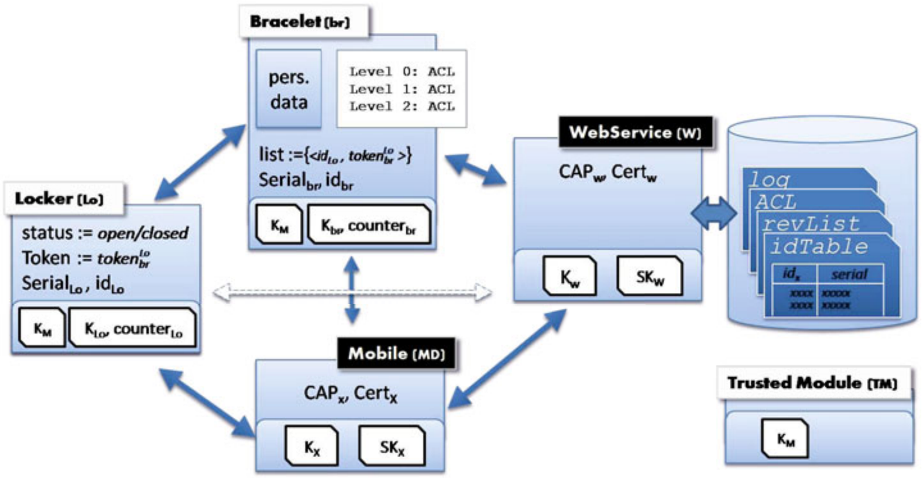

Fig. 1. Overview of information storage, key infrastructure and capabilities

– *Initialization of bracelets and lockers.* The administrator assigns a unique identifier to each bracelet (i.e. id_{br}) and locker (i.e. id_{Lo}). Each bracelet is assigned to one athlete. The administrator inputs the identifier and a newly generated serial number (i.e. $serial_{br}$ or $serial_{Lo}$) to the trusted module. The trusted module (TM) then generates a secret device key (i.e. K_{br} or K_{Lo}). The secret key will later be used to generate pseudo-random numbers during authentication. TM then sets up a secure connection with the embedded device (i.e. the bracelet or locker) using K_M and passes the identifier, the serial and the secret key to the device. Note that only administrators can run this application.
– *Issuance of secret keys and capabilities.* An administrator can issue *secret keys* K_X and *capabilities* CAP_X to Web servers, coaches and possibly to other administrators using the trusted module. A coach typically stores them in his mobile device (i.e. MD) and can use them to read out data from bracelets and to open lockers. K_X and CAP_X are defined as follows:
  - $K_X := \mathsf{prf}_{K_M}(serial_X)$
  - $CAP_X := \mathsf{symEncrypt}([id_X || role_X || serial_X], K_M)$.

Each *bracelet (br)* further keeps a counter to generate pseudo-random numbers, an access control list (ACL) and a set of personal data. The ACL and personal data are initialized by the administrator and can later partially be modified/updated by the athlete using the Web service (W). *br* Also keeps a list of locker identifiers id_{Lo} that were closed and the necessary tokens $token_{br}^{Lo}$ to reopen them. Note that each bracelet has a mechanism to (de)activate it. A sliding switch on the bracelet connects or disconnects the antenna from the chip. Hence, the athlete keeps full control over when the bracelet will respond to RF readers. A green/red LED shortly lights up when a protocol has been successful.

Each locker has an `open` and `close` button. If the `open` (or `close`) button is pressed, the locker initiates an open (or close) locker protocol (see section 5).

Mobiles (MD) and *Web services (W)* also keep a certificate (and corresponding private key) issued by the administrator. They are used for mutual authentication. Moreover, W maintains an *idTable*, a *revList* and an ACL. The *idTable* maps each id_X to its current $serial_X$. Except for lockers, each id_X is also mapped to an individual (either athlete or coach). The *revList* lists the invalid (revoked) serial numbers. The ACL defines access control privileges for each individual. Finally, certain actions are logged. For instance, if a locker is opened by a coach or administrator, the Web server logs certain data (see section 5).

5 Protocols

This section first describes a mutually authenticated key agreement protocol between resource constrained devices (e.g. bracelets, lockers,...) and powerful devices (e.g. Web servers, mobiles,...), respectively denoted as *cd* and *pd*. Thereafter, some services are discussed. We thereby mainly focus on services in which *constrained devices* are involved. They show how the security architecture is used. Some services rely on the key agreement protocol.

5.1 Mutually Authenticated Key Agreement

The powerful device *pd* generates a challenge and sends it together with its capability CAP_{pd} to the constrained device *cd* (1-2). The latter decrypts CAP_{pd}, generates a new random (based on its secret key K_{cd} and a counter), and calculates K_{pd} and K_s (3-5). The random is then sent to *pd*. The latter can now calculate the session key K_s (6-7).

Table 1. Mutually authenticated key agreement between *cd* and *pd*

			keyAgreement():
(1)	pd	:	$rand_1 := \mathsf{genRandom}()$
(2)	$cd \leftarrow pd$	:	$CAP_{pd},\ rand_1$
(3)	cd	:	$[id_{pd} \| \texttt{"Role"} \| serial_{pd}] := \mathsf{symDecrypt}(CAP_{pd},\ K_M)$
(4)	cd	:	$rand_2 := \mathsf{prf}_{K_{cd}}(counter_{cd}{+}{+})$
(5)	cd	:	$K_{pd} := \mathsf{prf}_{K_M}(serial_{pd});\ K_s := \mathsf{prf}_{K_{pd}}(rand_1,\ rand_2)$
(6)	$cd \rightarrow pd$	:	$rand_2$
(7)	pd	:	$K_s := \mathsf{prf}_{K_{pd}}(rand_1,\ rand_2)$

The random numbers generated by both parties ensure the freshness of the session key. Authentication can now be realized through key confirmation. Hereby, *pd* reveals uniquely identifying information to *cd* when authenticating (i.e. CAP_{pd} contains id_X and $role_X$). Key confirmation realizes authentication of constrained devices (i.e. only genuine devices know K_M and can therefore construct K_s). After authentication, *cd* can release a subset of attributes.

5.2 Personalized Web Service

To access a personalized Web page (see also Table 2), the athlete activates the bracelet (1) and browses to the right *url* (2). Next, an `https` session is set up: the Web server authenticates to the client browser (B) using $Cert_W$ (3). The Web server and bracelet then agree on an mutually authenticated session key (4). Next, the bracelet encrypts his serial number $serial_{br}$ with K_s and sends the result to the Web server where it is decrypted (5-7). If $serial_{br}$ is still valid, it is mapped to the corresponding identity and a customized page is sent to the client browser (8-10). The athlete can now view personalized information (such as training schedules) and update some data (such as contact information). Similarly, he can securely update information stored in the bracelet using K_s. He finally deactivates the bracelet (11).

Table 2. Retrieving personalized website after mutual authentication

			<u>accessPersonalizedWebPage():</u>
(1)	A	:	activate_bracelet()
(2)	$A \rightarrow B \rightarrow W$	:	requestService(*url*, `"personalized_website"`)
(3)	$B \rightleftarrows W$	:	setUpSSL(*server_auth*, [$Cert_W$, SK_W])
(4)	$br \rightleftarrows B \rightleftarrows W$	:	K_s:= keyAgreement()
(5)	br	:	E_{msg} := symEncrypt($serial_{br}$,K_s)
(6)	$br \rightarrow B \rightarrow W$	:	E_{msg}
(7)	W	:	$serial_{br}$:= symDecrypt(E_{msg}, K_s)
(8)	W	:	**if** (stillValid($serial_{br}$) == **false**) abort()
(9)	W	:	*personal_page* := genPersonalPage($serial_{br}$)
(10)	$B \leftarrow W$	:	*personal_page*
(11)	A	:	deactivate_bracelet()

5.3 Controlled Release of Data

An athlete can specify its own privacy policy. It consists of three levels. Level 0 defines a subset of personal data that will be released to any RF reader that queries the bracelet. Level 2 and 3 define a subset of data that will only be released to privileged individuals, based on their $role_X$ and id_X respectively. For instance, a coach must be able to read out certain phone numbers in case of injuries. Table 3 shows the protocol for releasing privacy-sensitive data. After activating the bracelet, the coach instructs his mobile device (*MD*) to read out data from the bracelet (1-2). The mobile and bracelet then agree on an authenticated session key (3). Next, the bracelet selects the data (i.e. $data_{br}$) that may be released to id_X with $role_X$ (4). The bracelet encrypts $data_{br}$ with K_s, and sends the result to *MD* (5-6). Next, *MD* decrypts the data and displays it to the coach (7-8). Finally, the bracelet is deactivated (9).

5.4 Locker Service

The bracelet contains a number `MAX` that defines the maximum number of lockers that the owner of a bracelet may close simultaneously, and also a counter

Table 3. Releasing personal information to coach

		releasePersonalInformation():
(1)	A	: activate_bracelet()
(2)	$C \rightarrow MD$	: request($read\_data$)
(3)	$MD \leftrightarrows br$	: K_s:= keyAgreement()
(4)	br	: $data_{br}$:= privilegedData(id_C, $role_C$)
(5)	br	: E_{msg} := symEncrypt($data_{br}$,K_s)
(6)	$MD \leftarrow br$	: E_{msg}
(7)	MD	: $data_{br}$:= symDecrypt(E_{msg}, K_s)
(8)	MD	: display($data_{br}$)
(9)	A	: deactivate_bracelet()

Table 4. Closing locker with bracelet

		closeLocker():
(1)	A	: activate_bracelet()
(2)	$A \rightarrow Lo$	: push `close` button
(3)	Lo	: $rand$:= prf$_{K_{Lo}}$(id_{Lo}||"challenge"||$counter_{Lo}$++)
(4)	$br \leftarrow Lo$	: "close", id_{Lo}, $rand$
(5)	br	: **if** (numberOfLockersClosed() == MAX) abort()
(6)	br	: $token_{br}^{Lo}$:= prf$_{K_{br}}$(id_{br}||"ID"||$counter_{br}$++)
(7)	br	: mac_{br} := MAC(K_M, [$token_{br}^{Lo}$||$rand$||"Close"])
(8)	$br \rightarrow Lo$	: mac_{br}, $token_{br}^{Lo}$
(9)	Lo	: **if** ($mac_{br} \neq$ MAC(K_M, [$token_{br}^{Lo}$||$rand$||"close"])) abort()
(10)	Lo	: store($token_{br}^{Lo}$); $status_{Lo}$:= "closed"
(11)	Lo	: mac_{Lo} := MAC(K_M, [$token_{br}^{Lo}$||id_{Lo}||"FBClose"])
(12)	Lo	: close()
(13)	$br \leftarrow Lo$	: mac_{Lo}
(14)	br	: **if** ($mac_{Lo} \neq$ MAC(K_M, [$token_{br}^{Lo}$||id_{Lo}||"FBClose"])) abort()
(15)	br	: storeTuple(id_{Lo}, $token_{br}^{Lo}$); $nb\_closed++$
(16)	A	: deactivate_bracelet()

`nb_closed` that counts how many lockers are currently in use by the bracelet's owner. Moreover, the bracelet keeps a table with the identifiers (id_{Lo}) of the lockers in use and the corresponding required tokens $token_{br}^{Lo}$ to reopen them.

Closing a locker. This protocol is shown in Table 4. The athlete activates the bracelet and pushes the `close` button of the locker. The locker then checks its status (1-2). If the locker is still *open*, it generates a challenge based on some (secret and unique) parameters and sends his identifier id_{Lo}, the challenge and a `"close"` command to the bracelet (3-4). The bracelet checks if he can still close a locker (i.e. $nb\_closed <$ `MAX`) (5). If so, it generates a $token_{br}^{Lo}$ based on (secret and unique) parameters (6). It then calculates the message authentication code mac_{br} of $token_{br}^{Lo}$, the challenge and the `"close"` command with the master secret $K_M\dot{}$The bracelet then sends the mac_{br} and $token_{br}^{Lo}$ to the locker (7-8). The locker verifies the authenticity and integrity, stores $token_{br}^{Lo}$, modifies its status (i.e. `closed`) and closes the locker (9-12). It also sends a feedback message to the bracelet (13). The bracelet updates its locker table and stores the tuple (id_{Lo}, $token_{br}^{Lo}$) (14-16). To open the locker, a bracelet has to prove knowledge of both $token_{br}^{Lo}$ and K_M. The protocol is similar to the one listed above and is, therefore, omitted.

Recovery procedure. A mechanism is implemented to ensure that lockers can be opened if bracelets are lost. A trivial solution uses a physical key. The key can be used to open the locker. After opening, the embedded device can be reset. A more advanced solution gives privileges to (certain) coaches (and/or administrators) to open lockers. We assume that the coach can set up an NFC connection between his mobile device and the locker if he is close to the locker. To open a locker, the coach goes to the locker and logs in with his mobile device at the Web server (e.g. using GPRS). He then submits a request to the Web service to open a locker. The Web server verifies the identity and role of the requesting entity. Next, a secure connection is established between the locker and the Web server (using the mobile phone of the coach to forward data). The Web Server then sends an open request to the locker, the locker is opened and the connection is closed. Finally, the Web server logs the request (i.e. it stores the identity of the coach and the lockers identifier).

6 Discussion

This section first evaluates the initial requirements followed by a security analysis. Next, the extensibility and universality of the design is discussed.

6.1 Infrastructural Analysis

No public key crypto operations are performed on any low power device (i.e. bracelets and lockers). Moreover, only pseudo-random functions are required. Random numbers are generated based on a counter and a private key that are kept in a tamperproof part of the embedded device. The used pseudo-random function should be lightweight but sufficiently strong to generate symmetric keys. The encryption functionality of the cryptographic module can be exploited to build the random function. This realizes reasonable throughput while minimizing the required hardware logic (i.e. no other dedicated random function is required). Currently, there exists an AES-128 design with only 3k4 gates, which uses 3μA at 100 kHz and 1.5V in 0.35μ technology [9]. This is also compliant with the ECRYPT report on light-weight cryptography [10] that states that low power implementations of block ciphers like AES should be favored above hash functions as the cryptographic building blocks for secure RFID protocols. Hence, the message authentication code algorithm can also be based on encryption.

6.2 Security and Privacy Analysis

We analyze how the presented protocols satisfy the requirements presented in section 3.

Access control. During mutual authentication, constrained devices gain knowledge of $role_X$ and id_X. This enables users to specify a fine-grained access control policy. No trust is required in the other party since the link between K_X and CAP_X is guaranteed by the system administrator (i.e. only the administrator has access to the trusted module that generates K_X and CAP_X based on K_M).

Location privacy. Constrained devices generate new random values when queried, resulting in new session keys. Therefore, even when identical information is released it appears as random to eavesdroppers, providing location privacy. Powerful devices do transmit recurring data (i.e. CAP_X). This has, however, little impact since these devices are typically explicitly turned on by the user (i.e. they cannot be queried without user interaction).

Reliability. A set of valid serial numbers (whitelists) or revoked serial numbers (blacklists) can be kept in bracelets. This allows to check of the validity of capabilities locally. The most recent whitelist/blacklist can be retrieved when accessing the Web service. Note that the whitelist in a bracelet can be very short. In many cases, only the serial number of the coach and some friends need to be stored. Similarly, mobile devices have to download the whitelist/blacklist regularly. Although lockers also use revocation lists, updating them regularly may be time-consuming for the administrator. However, this only implies that a thief can close a limited set of lockers afterwards. To increase the security level, a registration desk may be added at the entrance of the sports club. The revocation lists are then kept at the registration desk. A dedicated application can enforce bracelets to disclose their serial number before an athlete enters the building. Bracelets with a revoked serial number can be killed immediately. A vulnerability interval still exists when a bracelet is stolen. During that interval, the attacker can log in on the Web service, and retrieve/modify personal data. A solution consists of an additional account (i.e. login and password) to retrieve personal pages. Moreover, multiple passwords can be linked to the same bracelet. Access rights can be based on the login. For instance, youngsters and their parents can have a separate login. They both get restricted privileges. For instance, the youngster can display all information that is kept in the bracelet whereas their parents can update contact information or access services for which payment is required.

Controlled release of personal information. During authentication, a constrained device only proves to be genuine (i.e. a constrainted device proves possession of the session key). Powerful devices do release personal information during authentication (i.e. $role_X$, id_X and $serial_X$). However, this information is only released to constrained devices and used for access control (and possibly accountability). The constrained devices cannot transfer the data to other devices or actors.

Confidentiality. During authentication, a new session key (i.e. K_s) is agreed. This key provides an authenticated confidential channel between the two parties.

We show how the presented protocols can resist to major attacks. A weakened version of the Dolev-Yao threat model [11] (i.e. see *security and trust assumptions* in section 3) is applied to model the capabilities of the attacker:

- **Replay attacks.** Replay attacks are prevented since both parties have control over the session key. Therefore, a new key will be established when the protocol is executed, preventing replay attacks on either of the involved parties.
- **Man-in-the-middle attacks.** Since K_M is only known to genuine devices and K_X is only known by the owner of the corresponding CAP_X, only these parties can generate the session key K_s, ensuring a secure end-to-end channel. Constrained device are trusted not to play man-in-the-middle.
- **Denial-of-service attacks.** It is possible that a secure session is set-up with another (i.e. not the intended target) nearby (genuine) device. This can lead to denial-of-service attacks where an attacker reroutes communication to other devices. This can be mitigated by distance measurement techniques and by providing visual feedback to the user (e.g. led display on lockers).
- **Relay attacks.** The (de)activation switch on bracelets partially defends against host-and-leech attacks. However, the bracelet currently does not contain other dedicated mechanisms to prevent relay attacks. Some mechanisms may, however, be added afterwards (such as built-in accelerometers [12]).
- **Cloning attacks.** Attackers cannot guess the challenges generated by both the constrained and powerful device and K_M is required to generate the session key. Since K_M not known by the attacker, it is not possible to create fake constrained devices.

6.3 Extensibility and Universality of the Design

Several other services can be added without a big impact on the design. *Consumable services* are a typical example. For instance, a fitness center can give visitors access to their infrastructure for a limited period of time (linear to the amount of money that is paid). Similarly, a bracelet can keep credits to consume drinks. The credits can be downloaded to the bracelet after payment using the Web service. A dedicated device (e.g. an exit barrier or cash desk) can decrease the credit. Multiple strategies exist to implement credits. Either a counter can be increased/decreased or signed tokens can be downloaded. The former strategy is more flexible whereas the latter is more secure. Alternatively, the credits can be kept centrally. This implies that access barriers and cash desks must be online to check and update the credit.

7 Conclusion

This paper presented a mobile solution to integrate multiple services in a sports association. Lightweight cryptographic primitives are used in low-power devices (such as bracelets and lockers) to enforce security and privacy requirements. In the current implementation, three types of services are already supported, namely a personalized Web service, an information retrieval service and a locker service. However, we showed that the design is flexible enough to add new services. Typical examples are entrance control and consumption services.

References

1. Juels, A.: RFID security and privacy: A research survey. IEEE Journal on Selected Areas in Communication 24(2) (2006)
2. Weis, S., Sarma, S., Rivest, R., Engels, D.: Security and privacy aspects of Low-Cost radio frequency identification systems. In: Hutter, D., Müller, G., Stephan, W., Ullmann, M. (eds.) Security in Pervasive Computing. LNCS, vol. 2802, pp. 210–212. Springer, Heidelberg (2004)
3. Ohkubo, M., Suzuki, K., Kinoshita, S.: Efficient Hash-Chain based RFID privacy protection scheme. In: International Conference on Ubiquitous Computing Ubicomp, Workshop Privacy: Current Status and Future Directions, Nottingham, England (September 2004)
4. Molnar, D., Wagner, D.: Privacy and security in library RFID: issues, practices, and architectures. In: CCS 2004: Proceedings of the 11th ACM Conference on Computer and Communications Security, pp. 210–219. ACM Press, New York (2004)
5. Hogben, G., Naumann, I.: Privacy features of european eID card specifications. Tech. rep., ENISA (2009)
6. Institute for Applied Information Processing and Tu Graz Communications: IAIK: JCE-ME, `http://jce.iaik.tugraz.at/`
7. Legion of the bouncy castle: The bouncy castle crypto APIs, `http://www.bouncycastle.org/`
8. Institute for Applied Information Processing and Tu Graz Communications: Crypto software for microcontrollers, `http://jce.iaik.tugraz.at/sic/products/crypto_software_for_microcontrollers`
9. Hoepman, J.-H., Joosten, R.: Practical schemes for privacy & security enhanced rfid. CoRR, vol. abs/0909.1257 (2009)
10. Pu Public, X., Oswald, E.: Suggested algorithms for light-weight cryptography editor (2006)
11. Dolev, D., Yao, A.C.: On the security of public key protocols. In: SFCS 1981: Proceedings of the 22nd Annual Symposium on Foundations of Computer Science, Washington, DC, USA, pp. 350–357. IEEE Computer Society, Los Alamitos (1981)
12. Czeskis, A., Koscher, K., Smith, J.R., Kohno, T.: RFIDs and secret handshakes: defending against ghost-and-leech attacks and unauthorized reads with context-aware communications. In: Proceedings of the 15th ACM Conference on Computer and Communications Security, Alexandria, Virginia, USA, pp. 479–490. ACM, New York (2008)