Towards Secure TMIS Protocols

David Gérault, Pascal Lafourcade

Towards Secure TMIS Protocols

David Gerault[1] and Pascal Lafourcade[2]

[1] Nanyang Technological University
[2] LIMOS, Universite Clermont Auvergne

Abstract. Telecare Medicine Information Systems (TMIS) protocols aim at authenticating a patient in a telecare context, and permitting information exchange between the patient and a distant server through a verifier. In 2019, Safkhani and Vasilakos [10] showed that several protocols of the litterature were insecure, and proposed a new protocol. In this paper, we show that their proposal is insecure, mainly due to incorrect use of distance bounding countermeasures, and propose a secure version, resistant to distance bounding related threats.

1 INTRODUCTION

RFID technologies can be used in a wide range of applications, from access management to tracking of people and goods or contactless payments. An RFID interraction is performed between a reader (also called verifier) and a tag (also called prover), typically withing a close distance from each other, without contact. Over a few years, the use of RFID technologies has become prominent in medical technologies [8], helping to solve various problem. The most notorious one is baby theft or misplacement [9, 5], which have dramatic consequences on families when they occur. Other application include drugs tracking, or patient identification. In this paper, we focus on Telecare Medicine Information Systems (TMIS) in which a patient's RFID wristband or implant interracts with a distant server through an RFID reader. In these systems, the distant server sends data from the Electronic Medical Record (EMR) of the patients to the reader.

TMIS Protocols Research for generic secure RFID protocols is very active: for instance, a recent survey compares 38 protocols [1]. As for protocols specifically designed for TMIS systems, Masdari et al presented a complete classification in [8], and show that no protocol in the litterature satisfies all the security requirements. More recently, Safkhani et al. presented a cryptanalysis of four recent protocols (Li et al. [7], Zheng et al. [12], Zhou et al. [13], and Benssalah et al. [2]) in [10], and showed that they were insecure. They proposed a new protocol, SV, which they proved secure using the automated framework Scyther [4]. The SV protocol makes use of time measurements, a typical feature of distance bounding protocol.

Distance Bounding Protocols Distance bounding protocols were introduced by Brands and Chaum in 1993 [3] to counter relay attacks. In a distance bounding protocol, a verifier estimates an upper bound on the distance of a prover by measuring the time elapsed during challenge response rounds. Distance bounding protocols must be resistant to *mafia fraud*, in which a man-in-the-middle adversary attempts to impersonate a distant prover to the verifier, as well as attacks where a dishonest, distant prover tries to appear close to the verifier. These distance-related attacks can be declined in different forms: a single distant prover (distance fraud), a distant prover using the proximity of a honnest prover near the verifier (distance hijacking), or a distant prover helped by an accomplice located near the verifier (terrorist fraud). Distance bounding protocols, as authentication mechanisms, have similarities with TMIS systems. They however differ, in that no authentication is performed between the verifier and the prover (they do not share keys) in TMIS, as the authentication is delegated to a distant server. Additionally, in TMIS schemes, the privacy of the prover is crucial, wheareas few distance bounding protocols consider it as a requirement. Moreover, in TMIS protocols that include time measurement, the prover typically replies to the challenge of the verifier with a hash. This practice was ruled out by the distance bounding community for years, because the time taken to compute a hash was deemed too long and unpredictible. Recently, Gerault [6] made arguments in favour of allowing complex responses during the timed exchanges, based on the general improvements of hardware, and the cryptography community effort to provide lightweight and zero-latency primitives. These differences between distance bounding and RMIS protocols make using off-the-shelf distance bounding protocols for TMIS non trivial.

Relay attacks are a very serious concern for TMIS: an adversary impersonating a distant patient potentially gains access to prescription drugs, and sensitive medical information about that patient. Attacks by distant dishonnest provers are also relevant: for instance, some systems use RFID to prevent baby theft. A reader periodically checks whether the rfid tag of the baby is within range. If the used protocol is vulnerable to a distance fraud, a criminal can walk away with the baby, and pass the checks from a distance, so that he is far away by the time someone notices the baby is missing. Distance hijacking attacks can also be harmful: a distant dishonest prover making the authentication of a honest patient appear as his own could cause serious damage, as the honest patient may then receive treatment based on the identity of the adversary, which could lead to severe complications in case of allergy.

To adress distant prover authentication issues, we make the following contributions:

1. We show that the time measurement performed in the SV protocol are not sufficient to counter relay attacks.
2. We propose a new TMIS protocol similar to SV, that is secure against distance bounding related attacks.

2 Protocol of Safkhani and Vasilakos

2.1 The protocol

In [10], Safkhani and Vasilakos show that previous RMIS protocols are insecure, and propose the SV protocol (Fig. 1).
The authors do not include time measurements on the verifier side, in order to resist adversaries that can control the verifier's clock.

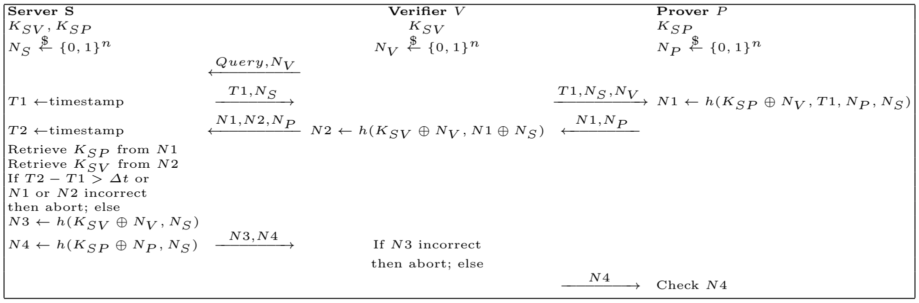
Server S
K_{SV}, K_{SP}
$N_S \xleftarrow{\$} \{0,1\}^n$
Verifier V
K_{SV}
$N_V \xleftarrow{\$} \{0,1\}^n$
Prover P
K_{SP}
$N_P \xleftarrow{\$} \{0,1\}^n$
$\xleftarrow{Query, N_V}$
$T1 \leftarrow$timestamp
$\xrightarrow{T1, N_S}$
$\xrightarrow{T1, N_S, N_V}$
$N1 \leftarrow h(K_{SP} \oplus N_V, T1, N_P, N_S)$
$T2 \leftarrow$timestamp
$\xleftarrow{N1, N2, N_P}$
$N2 \leftarrow h(K_{SV} \oplus N_V, N1 \oplus N_S)$
$\xleftarrow{N1, N_P}$
Retrieve K_{SP} from $N1$
Retrieve K_{SV} from $N2$
If $T2 - T1 > \Delta t$ or
$N1$ or $N2$ incorrect
then abort; else
$N3 \leftarrow h(K_{SV} \oplus N_V, N_S)$
$N4 \leftarrow h(K_{SP} \oplus N_P, N_S)$
$\xrightarrow{N3, N4}$
If $N3$ incorrect
then abort; else
$\xrightarrow{N4}$
Check $N4$

Fig. 1. The TMIS protocol proposed by Safkhani and Vasilakos [10]. Comas denote concatenation, and h is a hash function.

In the next section, we show that the SV protocol is actually not resistant to relay attacks, and exhibit other limitations on its design.

2.2 Flaws in the Protocol of Safkhani and Vasilakos

The SV protocol [10], presented in the previous section, does not grant resistance to relay attacks.
The SV protocol performs a time check to prevent relay attacks, but this check is done by the distant server, over an internet access. This time measurement is therefore highly unreliable: depending on the network traffic, the bandwidth, and the location of the verifier, combined to the possible use of a wireless network by the verifier, the measured time can vary significantly. In a communication over the internet, a message passes through several routers, which essentially perform a relay between the server and the verifier: therefore, relay attack resistance over the internet is by essence vain.
Under the assumption that the provers do not have an internal clock, and that only a prover, a verifier and a server are involved, over a classical TCP/IP connection, reliable relay attack protection can only be done by the verifier. This, in turn, would expose to attacks in which the time of the verifier is altered, as mentioned in [10]. In this paper, we make the common assumption that verifiers are honest, so that their clock can be trusted. If, on the other hand, a stronger guarantee is needed, one can include the methods of [11] to enforce reliable time measurement on the verifier side.

3 Our protocol

The SV protocol successfully fixes security issues of previous protocols, but remains insecure, due to its lack of relay attack resistance. We therefore propose a fixed protocol. Our protocol is very similar to SV, the main differences being that the verifier performs the time checks, and N1, N2, N3, N4 do not contain XORs of messages anymore. Additionally, the timestamps are not sent to the prover anymore (as the unicity of nonces makes them redundant, and we assume honest verifiers) Finally, to account for distance cheating attacks, the verifier's nonce is not sent to the server beforehand: otherwise, a dishonest prover could compute $N1$ in advance after eavesdropping N_S and N_V over the channel.

Assumptions We make the following assumptions:

1. The verifiers and servers are honest
2. The verifier's clock cannot be manipulated
3. The prover does not have a reliable clock
4. The computation time for the hash function is small compared to the RTT of messages

Our protocol is depicted on Fig. 2. It starts with the verifier sending a query to the server, as well as its identifier ID_V, and the server replying with his nonce N_S. The verifier sends his nonce N_V, ID_V and N_S to the prover, and starts a timer. The prover replies with his nonce N_P, as well as $N1 = h(K_{SP}, ID_V, N_V, N_P, N_S)$. The verfier stops its timer and compares the time elapsed with a predefined bound Δt. If the time taken is greater than Δt, it aborts the protocol. Otherwise, it computes $N2 = h(K_{SV}, N_V, N_P, N_S, N1)$ and sends $N1, N2, N_P, N_V$ to the server. The server retrieves K_{SP} by computing $h(K'_{SP}, ID_V, N_V, N_P, N_S))$ with the prover keys K'_{SP} of its database, until it finds the one that matches $N1$. If no match is found for $N1$, or if $N2$ is incorrect, then it aborts the protocol. Otherwise, it sends $N3 = SE_{K_{SV}}(N1, N2, Data)$ and $N4 = h(K_{SP}, N1)$ to the verifier, who checks the correctness of $N3$. It deciphers $N3$ as $(N1', N2', Data')$. If $N1' \neq N1$ or $N2' \neq N2$, it aborts. Otherwise, it retrieves $Data$ and sends $N4$ to the prover. Finally, the prover checks the correctness of $N4$, and aborts if it is incorrect.

4 Security Analysis

Notations We consider that a TMIS protocol has a *security parameter* λ, such that the nonce and key sizes are polynomial in λ. The adversaries we consider are polynomially bounded in λ, and we denote by *negligible* any function that is negligible in λ. In this section, we respectively denote a prover, a verifier, a server and an adversary by $P, V, S, \mathcal{A}$. A *dishonest prover* $\mathcal{A}^*$ refers to an adversary who legitimately registered his secret key into the system. We sometimes write *party* to denote a honest party, either prover, verifier or server. We denote by *session* the three-party execution of a TMIS protocol, involving a prover, a verifier, and a server. The *transcript* of such a session contains all the messages sent by the

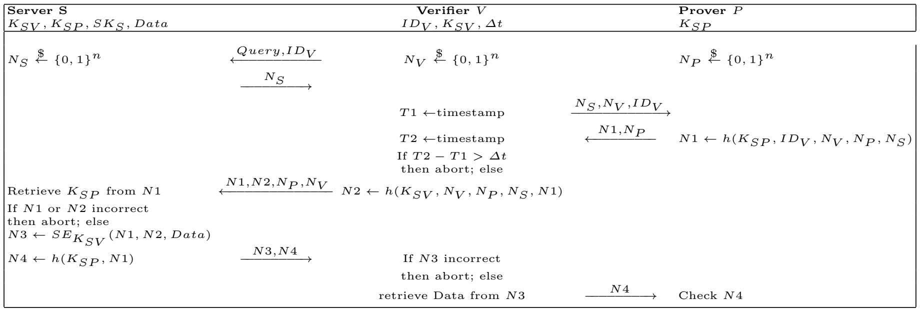
Server S
$K_{SV}, K_{SP}, SK_S, Data$
Verifier V
$ID_V, K_{SV}, \Delta t$
Prover P
K_{SP}
$N_S \overset{\$}{\leftarrow} \{0,1\}^n$
$\xleftarrow{Query, ID_V}$
$\xrightarrow{N_S}$
$N_V \overset{\$}{\leftarrow} \{0,1\}^n$
$N_P \overset{\$}{\leftarrow} \{0,1\}^n$
$T1 \leftarrow$timestamp
$\xrightarrow{N_S, N_V, ID_V}$
$T2 \leftarrow$timestamp
$\xleftarrow{N1, N_P}$
$N1 \leftarrow h(K_{SP}, ID_V, N_V, N_P, N_S)$
If $T2 - T1 > \Delta t$
then abort; else
Retrieve K_{SP} from $N1$
$\xleftarrow{N1, N2, N_P, N_V}$
$N2 \leftarrow h(K_{SV}, N_V, N_P, N_S, N1)$
If $N1$ or $N2$ incorrect
then abort; else
$N3 \leftarrow SE_{K_{SV}}(N1, N2, Data)$
$N4 \leftarrow h(K_{SP}, N1)$
$\xrightarrow{N3, N4}$
If $N3$ incorrect
then abort; else
retrieve Data from $N3$
$\xrightarrow{N4}$
Check $N4$

Fig. 2. Our TMIS protocol. Comas denote concatenation, h a cryptographic hash function, and SE a symmetric key encryption algorithm.

3 involved parties. Finally, we say that a prover is *distant* if he is at a distance grater than the maximum distance allowed in the protocol, defined by Δt. Otherwise, the prover is said to be *close*.

Security Properties Our protocol has similar security goals to the SV protocol, additionally considering distance fraud and distance hijacking attacks. We do, however, not consider adversaries who can control the clock of the verifier. Our protocol is resistant to: impersonation and replay attacks, traceability attacks, relay attacks, and distance fraud and distance hijacking.

Definition 1. *Resistance against impersonnation attacks Let Π be a TMIS protocol. Π is resistant against impersonnation attacks if no polynomially bounded adversary $\mathcal{A}$, given oracle access to provers, verifiers and servers, can be successfully authenticated with a non-negligible probability either:*

- *as a legitimate prover to a verifier or a server; or*
- *as a legitimate verifier to a server or a prover ; or*
- *as a legitimate server to a verifier or a prover*

For the attack to be valid, at least one of the protocol messages must be produced by the adversary: otherwise, this would constitute a relay attack, which are treated separately.

The next property is relay attacks: here, we only consider relay between a prover and a verifier, as relay between servers and other parties over a TCP/IP connection hardly makes sense.

Definition 2. *Resistance against relay attacks Let Π be a TMIS protocol. Π is resistant against relay attacks if no polynomially bounded adversary $\mathcal{A}$, given oracle access to provers, verifiers and servers, can be successfully authenticated with a non-negligible probability as a legitimate prover which is not in range of the verifier.*

Traceability deals with the ability of an adversary to link two sessions by a prover.

Definition 3. *Resistance against traceability attacks Let Π be a TMIS protocol with $n_P > 1$ different provers. Consider the following security game: the adversary $\mathcal{A}$ is given the transcripts of two different sessions, and must determine whether the two sessions were ran by the same prover or not. Let $p_{\mathcal{A}}$ denote the success probability of $\mathcal{A}$ in this game, and $adv(\mathcal{A})$ denote the advantage of the adversary $\mathcal{A}$, computed as $|\frac{1}{2} - p_{\mathcal{A}}|$. Π is resistant against traceability attacks if, for all polynomially bounded $\mathcal{A}$, $adv(\mathcal{A})$ is negligible.*

Finally, distance attacks are attacks in which a distant dishonest prover, holding a valid secret key, is authenticated by a verifier. The difference with impersonnation attacks where a prover is impersonated is that, in distance attacks, the adversary knows the the secret key of the distant prover he is authenticating as.

Definition 4. *Resistance to distance attacks Let Π be a TMIS protocol. Π is distance attack resistant if no polynomially bounded dishonest prover $\mathcal{A}^*$, given oracle access to servers, verifiers, and honest provers (both close and distant) can be successfully authenticated with a non-negligible probability.*

Security Statement Our security claims hold in the Random Oracle Model (ROM), where hash functions are modeled as random oracles, returning a truely random bitstring $O(x)$ when called on any bitstring x, such that later queries to O on the same x return the same value $O(x)$.

Resistance against impersonnation attack In the following, we assume that the symmetric encryption scheme SE is a secure Pseudo-Random Permutation (PRP). This implies that no polynomial adversary can, with non-negligible probability, recover the secret key used to encrypt a message, nor guess the encryption of a message of his choice.

Impersonnation of the server: In our protocol, the verifier authenticates the server based on the message N_3, and the prover authenticates the server based on the message N_4. First, under the assumption that h is a random oracle, and that SE is a secure PRP, $\mathcal{A}$ cannot extract the secret keys of the server from these messages, nor generate a valid N_3 or N_4 for valid ones on his own. Moreover, the probability for N_3 or N_4 from a previous session to be valid in a new session is negligible: it would require either the same nonces being used twice, or a collision in the hash function. Therefore, our protocol is secure against server impersonnation attacks.

Impersonnation of the verifier: Using a similar argument with N_2, our protocol is secure against impersonnation of verifiers to servers. The authentication of the verifier to the prover is indirect: the prover accepts the verifier as legitimate if it receives a valid N_4 from the server, meaning that the server authenticated a verifier with identity ID_V. As shown in the server impersonnation sketch of proof, no polynomial adversary can forge a legitimate fresh N_4: hence, our protocol is secure against impersonnation of verifiers to provers.

Resistance against relay attacks The time measurement prevents the adversary from sending the correct nonces to a distant prover, receiving the response, and forwarding it to the verifier in time to be accepted. Furthermore, the probability of the adversary to guess the either the correct nonces to send to the prover or the response of the prover in advance is negligible. Hence, our protocol is secure against relay attacks.

Resistance against traceability attacks In our protocol, the key identifying the prover only appears within hashes, in the random oracle model: it is therefore completely hidden to the adversary. Therefore, it may only leak if the same N_1 appears, in two different sessions, which occurs with negligible probability, due to N_P being chosen by the prover. Therefore, our protocol is resistant against traceability attacks.

Resistance against distance attacks For a prover to be accepted by the verifier, it needs to send its response within the time bound during the timed exchange. Let $\mathcal{A}^*$ be a dishonest, distant prover. If $\mathcal{A}^*$ waits to receive the nonces from the verifier to send his response, it will arrive too late, by definition of Δt. Hence, for $\mathcal{A}^*$ to be accepted, the verifier must receive a valid response by the server within the time bound. This occurs if either (a) $\mathcal{A}$ sends a response in advance by guessing correct nonce values, or (b) a honest prover near V sends a response correct for $\mathcal{A}^*$, and both occur only with negligible probability. Finally, $\mathcal{A}^*$ may send an incorrect response, and attempt to replace it with a valid one in the message sent by the verifier to the server. This strategy would however fail, since the initial N_1 and N_P values are authenticated by the verifier through a hash that is not falsifiable by $\mathcal{A}^*$, as it constains the shared key between V and S.

5 Conclusions and Discussion

In this paper, we exhibit a flaw in the relay attack resistance of the protocol of Safkhani and Vasilakos, and propose a new secure protocol. To the best of our knowledge, our protocol is the first TMIS protocol to be secure against distance bounding related attacks, as well as the classical threats. It does, however, not provide forward security, which is an interesting property left for future work.

References

1. Baashirah, R., Abuzneid, A.: Survey on prominent rfid authentication protocols for passive tags. In: Sensors (2018)
2. Benssalah, M., Djeddou, M., Drouiche, K.: Security analysis and enhancement of the most recent rfid authentication protocol for telecare medicine information system. Wireless Personal Communications **96**(4), 6221–6238 (Oct 2017). https://doi.org/10.1007/s11277-017-4474-y, https://doi.org/10.1007/s11277-017-4474-y

3. Brands, S., Chaum, D.: Distance-bounding protocols. In: '93. vol. 765, pp. 344–359 (1993)
4. Cremers, C.J.F.: The scyther tool: Verification, falsification, and analysis of security protocols. In: Gupta, A., Malik, S. (eds.) Computer Aided Verification. pp. 414–418. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
5. Dunbar, P.: 300,000 babies stolen from their parents-and sold for adoption: haunting bbc documentary exposes 50-year scandal of baby trafficking by the catholic church in spain. Daily Mail (2011)
6. Gerault, D., Boureanu, I.: Distance bounding under different assumptions: Opinion. In: Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks. pp. 245–248. WiSec '19, ACM, New York, NY, USA (2019). https://doi.org/10.1145/3317549.3319729, http://doi.acm.org/10.1145/3317549.3319729
7. Li, C.T., Weng, C.Y., Lee, C.C.: A secure rfid tag authentication protocol with privacy preserving in telecare medicine information system. Journal of medical systems **39**, 260 (08 2015). https://doi.org/10.1007/s10916-015-0260-0
8. Masdari, M., Ahmadzadeh, S.: A survey and taxonomy of the authentication schemes in telecare medicine information systems. Journal of Network and Computer Applications **87**, 1 – 19 (2017). https://doi.org/https://doi.org/10.1016/j.jnca.2017.03.003, http://www.sciencedirect.com/science/article/pii/S1084804517300978
9. Osaimi, A.A.A., Kadi, K.A., Saddik, B.: Role of radio frequency identification in improving infant safety and the extent of nursing staff acceptance of rfid at king abdulaziz medical city in riyadh. 2017 International Conference on Informatics, Health and Technology (ICIHT) pp. 1–7 (2017)
10. Safkhani, M., Vasilakos, A.: A new secure authentication protocol for telecare medicine information system and smart campus. IEEE Access **PP**, 1–1 (02 2019). https://doi.org/10.1109/ACCESS.2019.2896641
11. Tom Chothia, Ioana Boureanu, L.C.: Making contactless emv payments robust against rogue readers colluding with relay attackers. In: the 23rd International Conference on Financial Cryptography and Data Security (Financial Crypto 2019) (2019, to appear)
12. Zheng, L., Song, C., Cao, N., Li, Z., Zhou, W., Chen, J., Meng, L.: A new mutual authentication protocol in mobile rfid for smart campus. IEEE Access **PP**, 1–1 (10 2018). https://doi.org/10.1109/ACCESS.2018.2875973
13. Zhou, Z., Wang, P., Li, Z.: A quadratic residue-based rfid authentication protocol with enhanced security for tmis. Journal of Ambient Intelligence and Humanized Computing (Oct 2018). https://doi.org/10.1007/s12652-018-1088-5, https://doi.org/10.1007/s12652-018-1088-5