A Practical Implementation of Physical Layer Security in Wireless Networks

Sayed Amir Hoseini
School of Engineering and IT
University of New South Wales
Canberra, Australia
https://orcid.org/0000-0002-3105-4218

Fayçal Bouhafs
School of Engineering and IT
University of New South Wales
Canberra, Australia
https://orcid.org/0000-0001-6626-7881

Frank den Hartog
School of Engineering and IT
University of New South Wales
Canberra, Australia
https://orcid.org/0000-0001-5293-6140

Abstract—**Physical Layer Security (PLS) is widely recognized as a promising approach to secure wireless communications through the exploitation of the physical properties of the wireless channel. However, until today, PLS has mostly been a concept from information theory without practical implementations. In this paper, we present an actual implementation of PLS in a wireless network. We achieved this by leveraging the flexibility and control granularity offered by the relatively new concept of spectrum programming, by which we were able to control and degrade the quality of the eavesdropper's channel by virtually manipulating the connectivity of the legitimate station. The key feature of our design is that we make intelligent use of the architecture of a wireless network instead of trying to control every link individually on the physical layer. The PLS implementation is built using off-the-shelf hardware and open-source software which makes it cost-effective and easy to replicate.**

Keywords—Mobile and wireless security, physical layer security, wireless systems security

I. Introduction

We are often reliant on wireless communication technologies to exchange personal and sometimes confidential data. The broadcasting nature of the wireless medium makes exposure to eavesdroppers a potential threat. So far, this threat has mostly been mitigated by encrypting the wireless link and the information transmitted. Such solutions assume that eavesdroppers lack the computational resources and knowledge of the network parameters to break the encryption. While this assumption is still applicable in many scenarios, the capabilities to decode intercepted traffic given sufficient time are rapidly becoming more potent. In addition, many new devices on the market have limited resources and cannot support resource-intensive cryptographic solutions [1].

Physical Layer Security (PLS) [2] has been widely recognized as a complementary and sometimes alternative approach to encryption. PLS limits the amount of information that can be intercepted by an eavesdropper at the bit level by making it impossible for them to decode any data and thus, if executed well, can provide perfect secrecy. For that, PLS takes advantage of the imperfections of the communication channel due to its inherent randomness and the presence of noise. It achieves secrecy by improving the quality of the channel at the legitimate station while degrading the quality of the channel at the eavesdropper.

Several techniques have been proposed in the literature to implement PLS, falling in the categories of channel coding, channel control and power control. Channel coding techniques introduce robust coding schemes and randomization in the transmitted signal making it difficult for eavesdroppers to decode the intercepted signal [3, 4]. Channel control focuses on manipulating the radio channel parameter and monitoring the channel to detect the presence of eavesdroppers [5, 6]. Power control techniques try to control the power and direction of the signal transmitted to increase the capacity at the legitimate station and degrade the capacity at the eavesdropper [7], for instance by using multiple antennas [8].

So far, these techniques remain limited to the information theory domain, without practical implementations. More importantly, they are all based on a link-level design approach, focusing on the individual wireless connections between sender and receiver. And, apparently, it is rather difficult to realize PLS techniques in this way, particularly if they need to be replicable and extendable for future research and applications. A good example (and, to our knowledge, the closest anybody has got to an actual implementation of PLS) is the work presented in [9] where the author proposed an implementation of two PLS techniques, namely Phase-Enciphered Alamouti Coding (PEAC) and Artificial Noise [10]. Both implementations are based on GNU Radio, an open-source, low level Software Defined Radio (SDR) development toolkit designed for experimental work. The resulting testbed is very complex though, as well as difficult to replicate and even to validate [9]. Besides, the implementation is only applicable to a situation where the Shannon capacity of the eavesdropper is exactly half the Shannon capacity of the benign station.

In this paper we demonstrate that it is much easier to realize PLS, using off-the-shelf equipment, by tackling the problem at the network-level instead of the link-level. For this we make use of a relatively novel technique called spectrum programming [11], which enables centralized and fine-grained management and control of wireless networks. After briefly describing the threat model, we present our PLS technique in section III. Then we describe our Proof-of-Concept (PoC) implementation and provide a simple but effective experimental validation. We then discuss the modelling and experimentation efforts we are currently undertaking to solidify the work, and finish with conclusions.

II. Threat model

Here, we take Wi-Fi networks as an example. However, the concept can easily be generalized to other wireless and mobile networks. We first consider the classic system model as shown in Fig. 1: a legitimate station STA_m receiving information from the Access Point AP_n, while eavesdropped by STA_e. PLS can, in theory, be achieved when the Shannon capacity of STA_e is smaller than the Shannon capacity of STA_m (under a range of conditions as laid out in [2]). For now, we assume that traffic to be secured only goes downstream, i.e., from AP_n to STA_m. This is reasonable for situations where confidential information is provided by servers in the network and is only offered for consumption to client devices. If we then assume that the STAe has similar physical layer capabilities as STA$_m$, PLS can be achieved when STA$_m$ is closer to the AP than the STA$_e$. The STA$_e$ may be a passive eavesdropper. There may me multiple STA$_e$'s, but we assume that they do not collaborate.

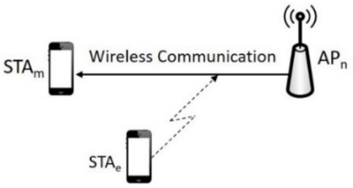

Fig. 1. Classic system model for PLS a station STA$_e$ trying to eavesdrop on the traffic sent to benign station STA$_m$.

III. Network-level PLS

While designing their solutions for achieving PLS, all link-level researchers so far assumed that the AP$_n$ is at a fixed position, i.e., it cannot be moved. This seems reasonable as access points generally do not move. But if we would assume they can, PLS can be achieved by just placing the AP$_n$ at a position where STA$_m$ can still decode the signal, but STA$_e$ cannot. Here, we do not propose to physically move APs (as this is not practical) but, instead, move them *virtually*. This can be achieved by assuming the presence of a network with not just one AP but many APs to which an STA$_m$ could possibly connect. Such network could, for instance, be an enterprise network or a centrally managed consumer network e.g. in apartment blocks [12].

To which AP the STA$_m$ connects is decided and executed by a central controller with up-to-date global knowledge of the network, including channel state information for STA$_m$ as well as STA$_e$ (about which later), as well as fine-grained control capability of the APs. Spectrum programming architectures offer such capability [11]. Spectrum programming makes OSI layer-2 and layer-1 settings in APs programmable by means of open APIs southbound and northbound of a central controller. It can be seen as an extension of Software-Defined Networking (SDN) to the lower OSI layers, recognizing that wireless networks require traffic to be controlled differently from wired, routed networks. Here, we make use of the architecture's ability to handover STA$_m$ quickly between physical APs by programming, deleting, and reprogramming Basic Service Set Identifiers (BSSIDs): an STA$_m$ does not notice it is being handed over as it stays connected to the same BSSID. In [13] this function is called Light Virtual Access Point (LVAP). The new system model is presented in Fig. 2. The dashed lines represent the different Shannon capacities. The model has been proposed before in [14], where the possible benefits were investigated by means of simulation. Here, however, we prove the applicability of the concept by a real and low-cost implementation.

To achieve PLS, we first consider a very simple algorithm. Assume a wireless network that consists of N APs serving M STA$_m$'s, in the presence of E STA$_e$'s. By principles of PLS, STA$_m$ can communicate securely with an access point AP$_n$ in the presence of an eavesdropping station STA$_e$ if $C_{n,m}>C_{n,e}$, where $C_{n,m}$ is the capacity of the channel between AP$_n$ and STA$_m$, and $C_{n,e}$ is the Shannon capacity of the channel between AP$_n$ and STA$_e$: $C_{n,i} = B \log_2(1+S_{n,i})$, with $i = m$ or e, B the bandwidth of the channel in Hz and $S_{n,i}$ the Signal to Interference plus Noise Ratio (SINR) experienced by STA$_i$ from AP$_n$. The algorithm works as follows:

1. STA$_m$ is trying to connect to the network in the presence of STA$_e$, and is requesting data.
2. The algorithm selects all APs that provide the lowest $C_{n,e}$.
3. Of those APs, the algorithm selects the AP for which STA$_m$ receives the strongest $C_{n,m}$ (highest SINR).

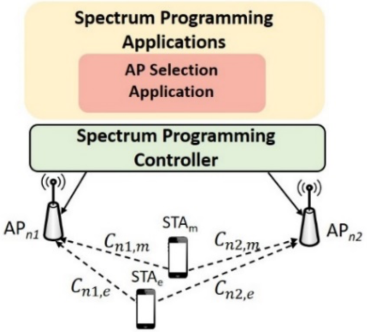

Fig. 2. Network-level PLS: handover the STA$_m$ to the AP$_n$ that is too far away for STA$_e$ to effectively eavedrop.

One idea behind the algorithm is that an STA can always find an AP to connect to such that its channel capacity is larger than the channel capacity of the closest eavesdropper, and thus secrecy can in theory be achieved. This is a reasonable assumption in large networks densely populated with APs and STA$_m$ being not too far outside the network. In other networks there is a non-zero chance that no AP can be found to which an STA can connect and still be able to communicate effectively as well as securely.

However, secrecy being achievable in theory is not enough. Practical realization would then still require the hard-to-achieve implementation of a link-layer mechanism like described above. This is where step 2 of the algorithm comes in: We first find APs for which $C_{n,e}$ is *lowest*, i.e. not just lower than $C_{n,m}$, which is hopefully so low that it falls below the threshold for the eavesdropper to be able to decode the signal, even without applying additional link-layer measures. Secrecy is then realized by a simple network-level approach, and the AP is selected which provides maximum secrecy capacity under that condition (step 3). In case the minimum $C_{n,e}$ is still so high that the signal can be decoded, it may still be so low that the additional measures to be taken can be relatively simple. This could, for instance, include the addition of a noise generator somewhere at the periphery of the network.

Also note that step 2 above assumes that the location of each eavesdropper is known, and that its channel capacity can be measured. This is a well-known limitation of PLS, and various solutions can be proposed and have been proposed in the literature, which we will discuss in section VI below.

IV. Proof of Concept

Our PoC is schematically drawn in Fig. 3. We use two TP-Link ARCHER C7 AC1750 v5 APs, deployed 15 meters apart and connected to a PC that acts as a spectrum programming controller. The controller runs a modified version of Floodlight Controller and hosts the AP selection application. To make the APs programmable, we installed OpenWRT 18.06.9 on both APs, as well as a set of open-source software packages including Click-modular-Router v2.1 [15], Odin agent v01 [13], and OpenvSwitch v2.85 [16]. Both APs are configured to operate on IEEE 802.11g at 20 MHz bandwidth. We chose channel 9 as it was the least crowded channel in the environment where the PoC was deployed. Both APs are connected to another PC that acts as a data traffic server. For the sake of simplicity, the control and data planes are run in-band and we use a basic ethernet switch to connect the APs and the PCs.

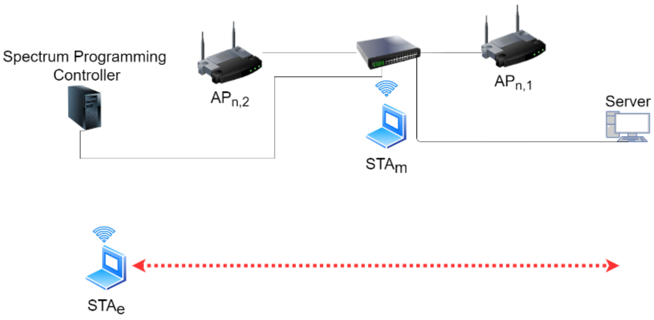

Fig. 3. Schematic diagram of the Proof of Concept

We deployed two laptops using TP-Link wireless USB Adaptors (TL-WN722N v3), one acting as a legitimate station (STA$_m$), and the other as an eavesdropping station (STA$_e$). STA$_e$ is equipped with an additional USB adaptor from the same model that allows the laptop to monitor and capture data traffic. The AP selection application runs on the controller and repeatedly executes a simplified version of the algorithm described the previous section. First, the Received Signal Strength Indicators (RSSIs) of STA$_m$ and STA$_e$ are measured at each AP. RSSI is used as the channel capacity indicator, as we assume that noise and interference do not vary significantly during the experiment. This is a reasonable assumption because all equipment remains the same and stationary (besides STA$_e$) in the same environment, and a Wi-Fi channel with minimum interference was selected. Second, the algorithm selects the AP that has the lowest RSSI$_e$ and guarantees a certain (configurable) minimum channel performance for STA$_m$. Finally, if the selected AP is not the AP that currently serves STA$_m$, STA$_m$ is handed over to the selected AP using the LVAP mechanism.

We assessed the performance of our approach by measuring the decodability of the eavesdropper. Decodability indicates how successful STA$_e$ is at listening and capturing data packets sent by the data traffic server to STA$_m$. We consider a packet to be successfully decoded when STA$_e$ successfully captures the entire frame. Decodability D_e is the number of packets η_e that can be captured successfully by STA$_e$ as a percentage of packets η_m captured successfully by STA$_m$: $D_e = \frac{\eta_e}{\eta_m} \times 100$ during a time window of 2 s.

We generate a downlink stream of messages from the data traffic server to the STA$_m$ at a rate of 5 packets per second. STA$_e$ uses Wi-Fi sniffing tools to capture packets while moving forward and backward following the red dotted line in Fig. 3. It moves from left to right along the red line in about 50 s, returns to the left and repeats these movements 2 additional times in course of the experiment. STA$_m$ is at about 8 m from both APs. The STA$_e$ never got logically closer than ~12 m from both APs during its trajectory, with which we mean that because of space limitations, we removed the antenna from STA$_e$'s adaptor and thus could get a bit closer without gaining too much signal strength.

V. Validation

Fig. 4 illustrates the measured RSSIs for STA$_m$ and STA$_e$ at AP$_{n,1}$ and AP$_{n,2}$. STA$_m$ is stationary, and its signal strength is almost flat for both APs. Both RSSI$_e$'s of STA$_e$ vary significantly while STA$_e$ moves forward and backward. The green arrows indicate a handover of STA$_m$ to AP$_{n,1}$, and the red arrows indicate a handover to AP$_{n,2}$. We can observe that handovers take place shortly after the system has detected that one RSSI$_e$ is smaller than the other.

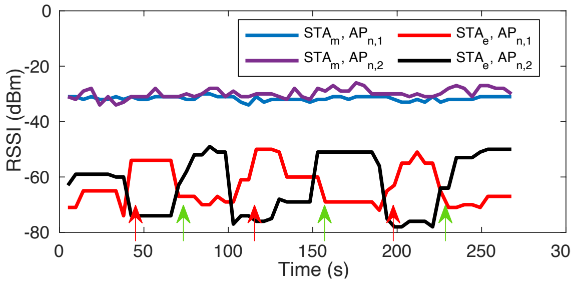

Fig. 4. The signal strength (RSSI) of the legitimate STA$_m$ and eavesdropper STA$_e$ stations at AP$_{n,1}$ and AP$_{n,2}$. Green arrows indicate a handover of STA$_m$ to AP$_{n,1}$, and red arrows indicate a handover to AP$_{n,2}$.

Fig. 5 presents the decodability of STA$_e$. PLS is achieved when none of the transmitted frames can be successfully eavesdropped, i.e. when the decodability is zero. STA$_m$ is initially associated with AP$_{n,1}$, and STA$_e$ is too far from AP$_{n,1}$ to be able to decode a signal. STA$_e$ then moves toward AP$_{n,1}$ and can capture some of the messages between t = 45 s and t = 49 s. But the controller detects the change in RSSI$_e$ and hands STA$_m$ over to AP$_{n,2}$. Consequently, STA$_e$'s decodability drops back to zero as STA$_e$ is not in AP$_{n,2}$ coverage, and thus PLS is reinstated. This happens multiple times as STA$_e$ continues to change its position. In this experiment, PLS cannot be achieved when a) a handover has not happened yet because of earlier mentioned delays in the detection system and handover algorithm, b) STA$_e$ is close enough to both APs such that, regardless of the AP to which STA$_m$ is connected, the STA$_e$ can decode signal. The latter is for instance the case between t = 135 s and t = 155 s. The current algorithm only guarantees PLS if always an AP can be found for STA$_m$ to connect to such that RSSI$_e$ < RSSI$_m$ and preferably so low that no intercepted messages can be decoded without additional measures. And even if PLS is not achieved, a lower RSSI$_e$ makes it harder anyway for STA$_e$ to capture packets. We have uploaded a short video of our validation to YouTube [17].

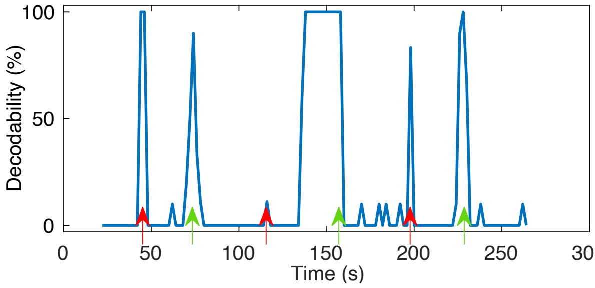

Fig. 5. Decodability of packets received by the eavesdropper STA$_e$. Green arrows indicate a handover of STA$_m$ to AP$_{n,1,}$ and red arrows indicate a handover to AP$_{n,2}$.Where decodability is zero, PLS is achieved.

Our results are not compared empirically with alternative proposals, as such proposals do not exist. The implementation of [9] only works when STA$_e$'s Shannon capacity is *exactly* half STA$_m$'s Shannon capacity which only happened at negligibly small durations of time during our experiments.

VI. Discussion and Future Work

It is obvious from Fig. 5 that PLS is achieved most of the time, but not always. For instance, around t = 150 s the STA$_e$

is close enough to both APs such that the $\mathrm{STA_e}$ can decode either signal. This can only be solved by extending the network with more APs (or by adding link-layer PLS mechanisms). We intend to investigate this further by creating and simulating a detailed model as suggested before, i.e., of a spatially bound network with N APs serving M $\mathrm{STA_m}$'s, in the presence of E $\mathrm{STA_e}$'s, find the conditions when PLS can be achieved and when not, and validate this model with more measurements. Such a model would also allow us to investigate the inclusion of other techniques such as multiple antennae, power control, beamforming and network coding.

We will also migrate to a more spacious and less noisy experimentation environment, where removal of the $\mathrm{STA_e}$'s external antenna will not be needed, and where larger networks can be deployed. The simulations and measurements will also tell us under which conditions additional security measures are needed (such as link-level procedures, network-level noise generation, and light-weight encryption) and how packet decodability may translate to symbol decodability, which is the actual key performance indicator for PLS.

Around the actual handovers, eavesdroppers have seconds in which they can effectively decode packets. This is likely because of non-structural hardware, software, and configuration issues which we should be able to solve in the future. For instance, the delay of ~10 s between detection and actual handover in Fig. 4 may happen because of delays in measuring the RSSI. This finds its roots in us still having to use the now deprecated ath9k wireless driver, which is not well supported by modern hardware and versions of OpenWRT. With the same spectrum programming architecture but with older hardware, it has been shown that handovers can be achieved within 50 ms [18]. We are currently migrating our system to other drivers, software, and hardware, and invite industry partners to collaborate.

Another part of the ~10 s delay is a hysteresis guard interval of 4 s between handovers to protect system stability: it should be avoided that handovers happen too frequently when the choice between APs is not totally clear. This guard interval needs further optimizing.

Another important consideration is the requirement that the eavesdropper's channel condition is known, and therefore its location. This is particularly hard when the eavesdropper is passive. For this, we will be looking at the following solutions:

- Consider the physical boundaries of the network, for instance an enterprise building or apartment block, and distinguish insider threats from external eavesdroppers. Then make statistical estimates for the likely locations of the external eavesdroppers.
- Investigate the use of effective eavesdropper detection tools such as Ghostbuster [19], which make use of the fact that even passive receivers leak RF signals on the wireless medium, and which can be integrated into the spectral programming architecture.

We will also be looking at cases where upstream traffic needs to be confidential too. The relevant eavesdropper's Shannon capacity is then between the $\mathrm{STA_m}$ and $\mathrm{STA_e}$. This is outside the control of the spectrum programming architecture, unless $\mathrm{STA_m}$ is also a controlled entity. PLS can then in theory be achieved by moving the AP closer to the $\mathrm{STA_m}$. However, the $\mathrm{STA_e}$ may still be able to decode the signal unless other, link-level measures are being applied too.

VII. Conclusion

We showed that by using a network-level approach, PLS can be achieved relatively easy: we were able to create and validate a PoC using off-the-shelf hardware, protecting the downlink communication of a legitimate station against a moving eavesdropper. To our knowledge, this is the first practical implementation of PLS in wireless networks in existence, and we believe that it can also be replicated. The key to this success lies in the use of the relatively novel spectrum programming concept, which allows centralization of the necessary measurements and controls.

Acknowledgment

This work is supported by the UNSW Institute for Cyber Security (IFCYBER).

References

[1] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "SIMON and SPECK: Block Ciphers for the Internet of Things," IACR Cryptol. ePrint Arch, vol. 2015, p. 585, 2015.

[2] V. Poor and R. Schaefer, "Wireless physical layer security," Proc. Natl. Acad. Sci. U.S.A., vol. 114, no. 1, pp. 19-26, 2017.

[3] W. Harrison, J. Almeida, M. Bloch, S. McLaughlin, and J. Barros, "Coding for secrecy: An overview of error-control coding techniques for physical-layer security," IEEE Signal Process. Mag., vol. 30, no. 5, pp. 41-50, 2013.

[4] W. Harrison and S. McLaughlin, "Physical-layer security: Combining error control coding and cryptography," in 2009 IEEE International Conference on Communications, 2009, pp. 1-5.

[5] X. Li and P. Ratazzi, "MIMO transmissions with information-theoretic secrecy for secret-key agreement in wireless networks," in 2005 IEEE Military Communications Conference, 2005, pp. 1353-1359.

[6] C. Sperandio and P. Flikkema, "Wireless physical-layer security via transmit precoding over dispersive channels: optimum linear eavesdropping," in 2002 IEEE Military Communications Conference, 2002, vol. 2, pp. 1113-1117.

[7] W. Wang, K. C. Teh, and K. H. Li, "Artificial noise aided physical layer security in multi-antenna small-cell networks," IEEE Trans. Inf. Forensics Secur., vol. 12, no. 6, pp. 1470-1482, 2017.

[8] X. Chen, D. W. K. Ng, W. Gerstacker, and H.-H. Chen, "A survey on multiple-antenna techniques for physical layer security," IEEE Commun. Surv. Tutor., vol. 19, no. 2, pp. 1027-1053, 2016.

[9] K. Ryland, "Software-Defined Radio Implementation of Two Physical Layer Security Techniques," Virginia Tech, 2018.

[10] R. Negi and S. Goel, "Secret communication using artificial noise," in IEEE vehicular technology conference, 2005, vol. 62, no. 3, p. 1906.

[11] F. Bouhafs et al., "Wi-5: A programming architecture for unlicensed frequency bands," IEEE Commun. Mag., vol. 56, no. 12, pp. 178-185, 2018.

[12] F. den Hartog, A. Raschella, F. Bouhafs, P. Kempker, B. Boltjes, and M. Seyedebrahimi, "A pathway to solving the Wi-Fi Tragedy of the Commons in apartment blocks," in 2017 International Telecommunication Networks and Applications Conference, 2017, pp. 1-6.

[13] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, "Towards programmable enterprise WLANS with Odin," in 1st workshop on hot topics in software defined networks, 2012, pp. 115-120.

[14] F. Bouhafs, F. den Hartog, A. Raschella, M. Mackay, Q. Shi, and S. Sinanovic, "Realizing Physical Layer Security in Large Wireless Networks using Spectrum Programmability," in 2020 IEEE Globecom Workshops, Taipei, Taiwan, 2020.

[15] E. Kohler, R. Morris, B. Chen, J. Jannotti, and F. Kaashoek, "The Click modular router," ACM Trans. Comput. Syst., vol. 18, no. 3, pp. 263-297, 2000.

[16] B. Pfaff et al., "The design and implementation of open vswitch," in 12th USENIX Symposium on Networked Systems Design and Implementation, 2015, pp. 117-130.

[17] https://www.youtube.com/watch?v=F0eOkUbaqCc

[18] J. Saldana et al., "Unsticking the Wi-Fi client: Smarter decisions using a software defined wireless solution," IEEE Access, vol. 6, pp. 30917-30931, 2018.

[19] A. Chaman, J. Wang, J. Sun, H. Hassanieh, and R. R. Choudhury, "Ghostbuster: Detecting the presence of hidden eavesdroppers," in Proc. of the 24th Annual International Conference on Mobile Computing and Networking, 2018, pp. 337-351.