MPKIX: Towards More Accountable and Secure Internet Application Services via Mobile Networked Systems

Tian Xie*, Sihan Wang*, Xinyu Lei, Jingwen Shi, Guan-Hua Tu†, Chi-Yu Li

Abstract—Nowadays, both Internet Application Service (IAS) providers and users face various security threats and legal issues. Due to the lack of reliable user information verification mechanisms, adversaries can abuse IASs to launch various cyberattacks, such as misinformation distributing and phishing, by using fake user accounts. IAS providers may thus inadvertently offer inappropriate content to restricted users, thereby suffering a serious risk of prosecution under local or international laws. Also, IAS users may suffer from nefarious ID theft attacks. In this paper, we proposed a novel security framework, MPKIX, designated as Mobile-assisted PKIX (Public-Key Infrastructure X.509). MPKIX secures both IAS providers and users by leveraging the broadly used PKIX services and mobile networked systems. It not only provides IAS providers with a reliable user verification mechanism while simultaneously enabling cross-IAS user privacy protection, but also largely mitigates the possibility of ID theft attacks and benefits other involved parties, such as cellular network operators and PKIX service providers. We further conduct a security analysis of MPKIX and implement an MPKIX prototype. The evaluation results based on the prototype confirm the effectiveness and efficiency of MPKIX with low overhead.

Index Terms—Personal certificate, PKIX, security and privacy, and cellular network.

1 Introduction

With the rapid deployment of high-speed Internet infrastructure, more and more applications (e.g., email, messaging and games) are offered by Internet Application Service (IAS) providers (e.g., Google and Facebook). However, both IAS providers and users are faced with a wide variety of security problems and the legal issues derived from them. Three of the most common security problems are as follows: (1) IAS providers may inadvertently offer inappropriate goods and services to restricted users due to the lack of a practical means for the validation of user information; (2) IASs are abused by unaccountable users or adversaries to launch cyberattacks or other disruptive activities; and (3) IAS users suffer from ID theft attacks.

Specifically, the Communications Decency Act (CDA) [1] in the U.S. prohibits indecent materials from being disseminated to children over computer networks. However, it is technically challenging for IAS providers to fulfill this prohibition in practice. For example, Google allows only users over 18 years old to create user accounts and verifies user ages only based on the information provided by users; however, the information may be fake. In this case, adversaries can benefit from filing malicious lawsuits against those companies violating the CDA, e.g., claiming that their children have a psychologically negative impact while accessing these IASs; IAS providers thus suffer from millions/billions of dollars in punitive damages [2], [3].

The root cause lies in that there are no reliable means for IAS providers to verify user information. Adversaries can thus register bogus accounts easily using fake user information [4]. The bogus accounts can be further exploited to disseminate fake news and disinformation [5], [6], which can result in an annual economic loss with tens of billions of dollars [7], [8], and launch a variety of cyberattacks, such as phishing [9], Distributed Denial-of-Service (DDoS) [10], [11], and user identity theft/fraud attacks.

On the other hand, the user identity theft/fraud is one of major security threats against IAS users; it is known as the unauthorized use of another person's information in creating IAS user accounts and achieving illicit financial gain (e.g., asking for financial aids by impersonating the victim [12]). According to a recent report [13], it takes 13%, 14%, 20%, and 29% of all consumer complaints in 2017, 2018, 2019, and 2020, respectively. Its derived attacks may not only lead to financial loss (e.g., 56 billion in 2020 [14]) but also cause emotional and physical health damages to victims [15].

At first glance, real-name registration systems [16] can be used to tackle the issue of user information verification. However, the situation is complex in practice due to privacy concerns. Benign users may provide IAS providers with false user information [17] while creating user accounts. As a result, their real identities may be impersonated or stolen for malicious usage, so they still expect that IAS providers can protect their real identities even when they are not given at the registration. We thus believe that there is a pressing need for developing a novel solution to make IASs more accountable and secure while preserving user privacy.

Existing Technologies and Limitations: We examine current solutions and limitations thereof for securing IAS providers and users from three aspects. (1) *Preventing the provision of goods/services to restricted users*: the most common approach is to request each user to sign a legal agreement that assures provided information to be correct during account registration. However, this approach

- *T. Xie, S. Wang, J. Shi, and G.-H. Tu, are with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, 48825. E-mail:{xietian1, wangsih3, shijingw, ghtu}@msu.edu. *: The first two authors contribute equally to this work. †: Corresponding author.*
- *X. Lei is with the Department of Computer Science, Michigan Technological University. Email:xinyulei@mtu.edu*
- *C.-Y. Li is with the Department of Computer Science, National Yangming Jiaotong University. E-mail:chiyuli@cs.nctu.edu.tw*

may not certainly free IAS providers from possible lawsuits. (2) *Defending against cyberattacks launched from fake IAS user accounts*: the existing solutions can be broadly classified into detection-based and traceback-based methods. The detection-based methods [18], [19] identify abnormal account activities mainly based on feature-based detection mechanisms. They are based on known patterns of abnormal activities and thus hardly detect new attacks. The traceback-based methods (e.g., IP/ICMP/entropy traceback [20], [21]) aim to trace back to adversaries and hit back at them. However, they may not always work. For example, adversaries can simply use anonymity networks such as Tor [22] to disguise their locations and then bypass the traceback. (3) *Defending against ID theft attacks*: current solutions can be classified into two categories, namely *ID theft detection* and *impersonated ID revocation*. There have been many ID theft detection services (e.g., Identity Guard and myFICO) in the U.S. These service providers collaborate with major credit reporting companies (e.g., Experian) to monitor all the credit queries belonging to their customers based on social security numbers (SSNs) and then detect ID theft cases. However, these services cannot protect IAS users who are unwilling to share SSNs with IAS providers. For the revocation of an impersonated ID, most IAS providers request the user to provide his/her ID proof [23]. However, this process is time-consuming and may take several days or even longer time [24].

Proposed Approach: We aim to develop a new solution based on PKIX (Public-Key Infrastructure (X.509) [25]) to support accountable and secure IASs.The reason is that the PKIX-based authentications have been broadly supported by mainstream security protocols (e.g., HTTPS, SSL/TLS, and IPSec). IAS users and providers can use any of the security protocols to exchange PKIX certificates to authenticate each other. Also, PKIX has obtained great success in validating the authenticity of IAS websites. Specifically, more than 67% of websites use HTTPS as their default protocol [26] and our study shows that mainstream browsers, such as Chrome and Edge, mainly authenticate websites using PKIX.

With PKIX-based user authentication, IAS providers can easily authenticate a user based on his/her certificate and then obtain verified user information. Given that user information can be correctly verified, the aforementioned attacks can be thus avoided.

Technical Challenges: However, it is far from trivial to develop a PKIX-based solution in practice. The development involves four major technical challenges. (**C1**): Issuing PKIX certificates to users is too time-consuming to be scalable for billions of users. Specifically, for a certificate application, it may take 3-5 business days [27], [28] for CA to certify the correctness of subject (i.e., the certificate owner) information by carefully inspecting the applicant's official ID document. (**C2**): Using PKIX requires IAS users to additionally provide CA with their personal information, which has been given to IAS providers, but the users may be reluctant. (**C3**): IAS users may be only willing to disclose partial user information which is necessary to IAS providers, due to privacy concerns. However, PKIX does not support the provision of partial user information. (**C4**): When only partial user information is given, IAS providers may not protect IAS users from possible ID theft attacks. Once an ID theft attack occurs, the revocation/claim of an impersonated ID requires its real owner to provide more proofs; it is time-consuming and inevitably discloses more user information.

Proposed Solutions: We thus propose to develop MPKIX, designated as **M**obile-assisted **PKIX**, to tackle the above challenges by leveraging embedded intelligence of mobile networked systems. MPKIX is composed of three novel approaches, namely *carrier-endorsed PKIX user certificate issuance (**ceIssuance**)*, *cross-IAS privacy-preserving user information querying (**ppQuery**)*, and *privacy-aware ID claim/revocation arbitration (**paClaim**)*. The first one addresses C1 and C2, whereas the others resolve C3 and C4, respectively.

- **ceIssuance** leverages the cellular network to facilitate the PKIX user certificate issuance. Since it has had authentic identities of billions of mobile users, it can be a good anchor for PKIX to enable scalable user credential issuance without inspecting user IDs at CAs. The authentic identities are collected at user account registration, during which carriers verify user identities based on government-issued photo IDs. Such security policy is required by the law in many areas, such as China, Taiwan, and Thailand. It also has been a common practice in the U.S.
- **ppQuery** not only allows IAS servers to verify the correctness of user information by querying the cellular network using the GSMA OneAPI [40] interface[1], but also provides IAS users with cross-IAS privacy protection. The latter privacy protection guarantees that (1) the users can choose which pieces of user information are revealed to IAS providers, and (2) the real identity of an IAS user cannot be discovered or narrowed down to a small group of possible candidates even when adversaries collect all the information that the user ever reveals to different IAS providers.
- **paClaim** enables an efficient ID claim/revocation arbitration mechanism on a disputed user ID for the current owner and the claimer without revealing unnecessary user information to IAS providers.

People may wonder if cellular network operators are trusted; they may leak user information to other parties without user consent. However, it shall not happen. The reason is that cellular networks are considered as critical national infrastructures in most countries. So, the telecommunication service is mostly franchised and supervised by a particular government organization (e.g., Federal Communications Commission (FCC) in the U.S. and Ministry of Industry and Information Technology (MIIT) in China). Without abiding by the law about user privacy (e.g., information privacy law [41]), operators may lose the franchises. More advantages of using cellular networks, compared with other potential institutions/companies/organizations, are elaborated in Section 8.

Comparison with state-of-the-art approaches: We next compare MPKIX with state-of-the-art schemes from both academia and industry in two main categories, namely reliable IAS user information verification and IAS user ID claim/revocation, from four aspects: (1) functionality (i.e.,

[1]OneAPI is a set of APIs commonly supported by cellular networks to enable external application servers to securely access cellular network services and user profiles.

Approaches	Reliable IAS User Information Verification					Efficient IAS User ID Claim/Revocation			
	Supported?	Only provide IAS with required info?	Support Cross-IAS privacy protection?	Communcation cost	Computation cost[2]	Supported?	No additional user info is required?	Communication cost	Computation cost
Google/Facebook[1] [29], [23]	◐#	○	○	Seconds	$O(1)$	●	○	Days	$O(1)$
Side-channel Scheme [30], [31], [32]	◐	○	○	–	–	○	N/A	N/A	N/A
Let's Encrypt [33]	◐#	○	○	Seconds	$O(m^2)+O(\log n)$	◐	●	Seconds	$2O(m^2)+O(m^3)+O(\log n)$
FIDO-based Scheme [34]	◐	○	○	Seconds/Days	$2O(m^2)+O(\log n)$	◐	○	Seconds/Days	$2O(m^2)+O(m^3)+O(\log n)$
GSMA Mobile Connect [35]	●	●	○	Seconds	$2O(m^2)+O(\log n)$	○	N/A	N/A	N/A
MPKIX	●	●	●	Seconds	$2O(m^2)+O(k\log n)$	●	●	Seconds	$4O(m^2)+2O(m^3)+O(k\log n)$

●: yes, ◐: partial, ○: no
1: IAS providers such as Google and Facbook always trust the information provided by users.
2: We compare the computation cost of each method by accumulating the time complexity of encryption operations using RSA $O(m^2)$, decryption operations using RSA $O(m^3)$, hash operations using SHA256 $O(m)$, digital signature operations using SHA256withRSA $O(m^2)$, and database searching operations, where m is the length of the message, n is the number of users' records in the database, and k is the number of features for each record in the database [36], [37].
#: Can only verify users' phone numbers or emails. Can not prove the phone number indeed owned by the user who may use temperory phone number [38] and email [39].

TABLE 1: Comparison between MPKIX and the state-of-the-art approaches.

being supported or not), (2) security and privacy features, (3) communication cost, and (4) computation cost, if applicable. The comparisons are summarized in Table 1.

- **Google and Facebook [23], [29]:** The current practice of these IAS providers is to trust the information provided by users. They can verify only whether the provided phone numbers and email addresses are currently controlled by the users, but not whether they are indeed owned by the users or whether they are just temporary information [38], [39], as reported in [4]. Regarding the ID claim/revocation, although the IAS providers allow legitimate IAS users to claim their IDs that are created by adversaries using unauthorized user information, they usually request the users to provide more proof documents (e.g., driver licenses and utility bills) for manual inspection. It is not only time-consuming with taking several days or weeks, but also inevitably leaking more user information.
- **Side-channel User Information Verification [30], [31], [32]:** Several methodologies have been proposed to infer user demographics using side-channel information. Specifically, some studies infer user demographics by analyzing data collected from WiFi access points, such as network traffic [30] and user activity records [32]. Li *et al.* [30] infer user gender and education level by analyzing campus WiFi traffic, and Neal *et al.* [31] derive user gender based on usage records of Bluetooth and WiFi. However, these schemes have two common issues: (1) the error rates are non-negligible (e.g., 22% error in estimating gender using WiFi traffic [30]); the erroneous inference results for IAS users may lead to unnecessary suspension or mistaken operations of IAS services; (2) the above inference methods can only be applied to registered users, so they do not prevent IAS providers from various ID abusing attacks.
- **Let's Encrypt [33]:** It enables IAS providers to obtain CA-signed PKIX server certificates within seconds, and can be possibly extended to support the issuance of PKIX user certificates and the verification of IAS user information. However, it can verify only if a requester's email is legally associated with the owner of a domain name, whereas MPKIX does not have such limitation.
- **FIDO (Fast Identity Online) [34]:** During the registration with a FIDO-supported IAS service, FIDO allows a new IAS user to create a new key pair and register his/her public key with the IAS provider. It then does user authentication based on the signature of a challenge, which is generated and verified by the IAS user and provider, respectively. According to a claim from the FIDO working group, FIDO can be possibly extended to support identity verification and binding. However, compared with MPKIX, FIDO has three disadvantages: (1) additional FIDO authentication protocols need to be supported by IAS providers, whereas MPKIX leverages the standardized security protocols; (2) FIDO does not address the scalability and efficiency issues of user information verification; (3) FIDO does not provide privacy-aware ID claim/revocation service for IAS users.
- **GSMA Mobile Connect [35]:** This approach allows mobile users to log onto IAS services using mobile phones and specify what user information is shared with particular IAS providers. However, it has three major limitations: (1) it does not work without cellular signals; (2) it does not provide IAS users with cross-IAS privacy protection, thereby being vulnerable to cross-IAS user information inference attacks [42]; (3) it does not provide IAS users with privacy-aware ID claim/revocation arbitration service, which is supported by MPKIX.

In summary, MPKIX is the only mechanism that provides both reliable IAS user information verification and efficient IAS user ID claim/revocation while largely preserving user privacy. Moreover, its communication and computation costs are comparable to the others.

Contributions: This paper makes four contributions.

- MPKIX provides IAS providers with a reliable verification mechanism of user information while providing IAS users with cross-IAS privacy protection via the developed ppQuery mechanism. It can prevent various cyberattacks launched by false user accounts and distribution of improper content. Moreover, MPKIX secures IAS users from nefarious ID theft attacks without revealing unnecessary user information to IAS providers. By conforming to existing PKIX and cellular network standards, MPKIX has a small deployment cost. It can facilitate the delivery of accountable and secure online application services.
- The effectiveness of the proposed MPKIX framework is demonstrated experimentally. First, the MPKIX testbed is capable of processing up to 130,000 CSRs (Certificate Signing Requests) per minute and producing the corresponding CA-signed PKIX user certificates. Second, the terminal-side prototype of MPKIX is evaluated on both phones and computers. It is shown that MPKIX works well even on low/medium-end phone models. Third, MPKIX enables IAS providers to effectively verify the correctness of user information within less than 1 second without compromising user privacy. Fourth, the decision of the arbitration of

a disputed IAS ID revocation/claim can be made within 4 seconds, whereas the current practice takes several business days or weeks.

- A security analysis of the MPKIX framework is conducted. It shows that MPKIX not only offers desirable security guarantees, such as data integrity, non-repudiation, user privacy, and accountability, but also defends against various attacks.
- MPKIX benefits all the involved parties. Specifically, *CAs* can expand their enterprise-based PKIX credential services to billions of mobile users. *cellular network operators* can make profit by answering the queries about user information from IAS providers. *IAS providers* can ensure the correctness of user information so that the risk of improper content distribution and cyberattacks can be minimized. *IAS users* have an efficient privacy-aware mechanism to claim/revoke impersonated IDs without revealing additional user information to IAS providers. More details will be discussed in §8.

Paper organization: The rest of this paper is organized as follows: §2 introduces the background of PKIX services and 4G LTE cellular networks. §3 describes the threat model, assumptions, and offered security guarantees. §4 introduces the design of MPKIX. §5 gives the security analysis of MPKIX. §6 and §7 present the MPKIX implementation and performance evaluation, respectively. §8 discusses some remaining issues of MPKIX. §9 presents the related work and §10 concludes the paper.

2 Preliminaries

PKIX[25]: PKIX is built based on the asymmetric cryptography, in which the data encrypted by a public key can only be decrypted by its paired private key and vice versa. The public key is disseminated to the public, whereas the private key is known only by its owner. The PKIX certificate is usually formed in the format of X.509, which is an ITU-T (International Telecommunications Union) standard. The certificate contains three main elements, namely (1) the subject (owner) information (e.g., name, residence and age), (2) the owner's public key, and (3) the digital signature of the CA that issued the certificate. In practice, to obtain a CA-signed PKIX user certificate, the applicant needs to provide the CA with a government-issued photo ID and a CSR [43] request containing the applicant's subject information, public key, and digital signature. The CA confirms the applicant's identity by validating his/her digital signature using the public key and verifying the subject information by inspecting the photo ID. After the confirmation, the CA generates a PKIX user certificate and attaches a digital signature generated for the certificate.

4G LTE Cellular Network: Figure 1 illustrates 4G LTE cellular network architecture that consists of UE (User Equipment), RAN (Radio Access Network), and CN (Core Network). The RAN contains eNodeB (evolved Node B) base stations providing UE with radio access, whereas the CN offers UE with Internet access. The CN comprises three main network elements: (1) MME (Mobility Management Entity), which manages UE mobility and takes care of user authentication; (2) HSS (Home Subscriber Server), which stores subscriber information; and (3) 4G gateways, which route data packets between eNodeB and the Internet. The UE has two components, namely USIM (Universal Subscriber Identity Module) and ME (Mobile Equipment), such as smartphone. To carry out user authentication [44], the ME must collaborate with the USIM, which maintains a secret key shared with the HSS.

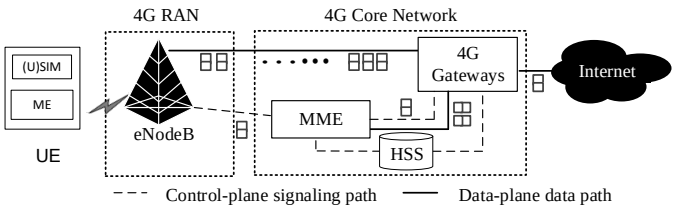


Fig. 1: 4G LTE cellular network architecture.

3 Threat Model, Assumptions, and Security Guarantees

Threat Model: In this study, adversaries are people or organizations who aim to impersonate IAS users, abuse IASs with false user information, or infer undisclosed information of IAS users. Two different types of adversaries are considered, namely semi-trusted IAS providers, which are interested in disclosing user identity and information, and network adversaries. The adversary capabilities are assumed to be the same as the Dolev-Yao model [45]; that is, adversaries can overhear, intercept, and synthesize any messages, but are constrained by the cryptographic methods in use (e.g., adversaries cannot decrypt ciphered messages without corresponding cipher keys). Moreover, we bear in mind to conduct this study in a responsible manner. All experiments and evaluations were conducted conforming to the IRB policy; no human subjects were involved.

Assumptions: MPKIX makes the following assumptions: (1) cellular network operators follow local/international information privacy laws (e.g., Code of Federal Regulations: Title 47 [41]) to protect user information from being leaked to other parties without user consent; and (2) the adversaries adhere to all cryptographic assumptions; e.g., they cannot restore an original message from its hashed value or decrypt an encrypted message without its decryption key.

Security Guarantees: MPKIX offers four security guarantees: (1) *data integrity*, which guarantees accuracy and consistency of the user information provided by an IAS user to an IAS provider; (2) *non-repudiation*, which guarantees that an IAS user cannot dispute authorship of the information revealed by the user to an IAS provider; (3) *user privacy*, which guarantees that an IAS user can reveal only partial user information to an IAS provider while accessing the IAS and initiating ID claim/revocation arbitration, and moreover, the undisclosed user information cannot be inferred (e.g., adversaries cannot correlate any IAS user with a particular individual or a small group based the user's information; (4) *accountability*, which guarantees that, given an IAS-based cyberattack, the law enforcement authority can discover real identities of the IAS provider and user.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3141694, IEEE Transactions on Mobile Computing

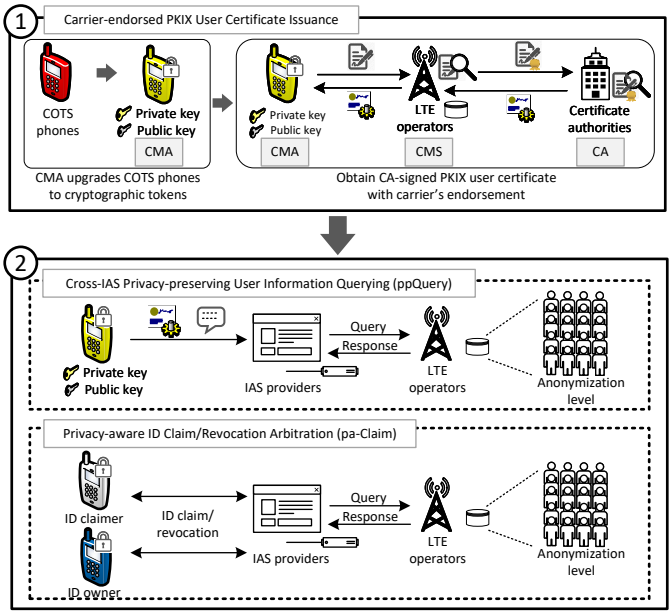


Fig. 2: The overview of MPKIX.

Category	Symbol	Description
ceIssuance	CMA	Certificate Management Application
	CMS	Certificate Management Server
	ppCSR	Privacy-preserving Certificate Signing Request
	ppCert	Privacy-preserving User Certificate
	K_{enc}	An encryption key used to encrypt data.
	K_{aut}	An authentication key to calculate message authentication code for integrity protection.
ppQuery	$\mathbb{IA}$	Anonymization factor.
	S	A subject attribute (e.g., name).
	f_S	The anonymization function of a given S.
	V_S	The value of a given S (e.g., Smith).
	m	The number of subject attributes.
	$\|DB\|$	The number of users in the database.
	$H_{u,i}$	The highest anonymization level used by user u for the value of S_i.
paClaim	$V_i^{claimer}$	ID claimer's value for S_i.
	V_i^{owner}	*Owner*'s value for S_i.
	W_i	The weight of the Levenshtein distance for S_i

TABLE 2: Summary of abbreviations, symbols, and parameters in MPKIX.

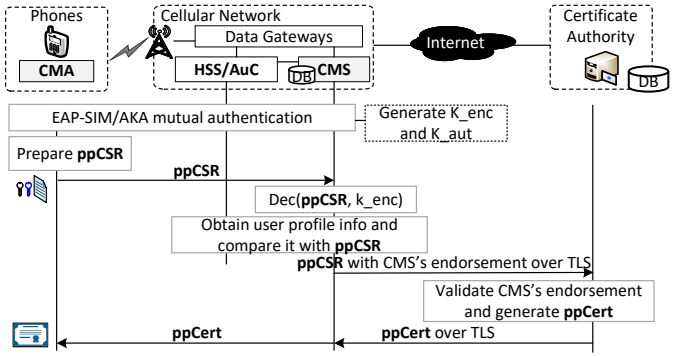


Fig. 3: MPKIX carrier-endorsed user certificate issuance.

4 MPKIX Design

MPKIX enables a mobile user to securely access IAS services while preserving user privacy from semi-trusted IAS providers and providing the IAS providers with a reliable means to verify the user information essential to IASs. Figure 2 shows an overview of MPKIX containing three major service components, namely carrier-endorsed PKIX user certificate issuance (**ceIssuance**), cross-IAS privacy-preserving user information querying (**ppQuery**), and privacy-aware ID claim/revocation arbitration (**paClaim**). To enable the MPKIX service, a mobile user must apply to **ceIssuance** for an MPKIX user credential including a CA-signed PKIX user certificate, where user information is encrypted and has been verified by a cellular carrier, and a key pair of public and private keys. With the MPKIX user credential, the user can securely access IAS servers with the support of PKIX-based mutual authentication, which is supported by most mainstream security protocols (e.g., HTTPS, SSL/TLS, and IPSec). To preserve user privacy, **ppQuery** enables IAS providers to verify the user certificate through a cellular carrier for user authentication without decrypting user information in the certificate. **paClaim** enables MPKIX users to claim/revoke an IAS ID that an adversary forges from IAS providers without disclosing any additional user information.

We next elaborate on each of the three service components, where abbreviations, symbols, and parameters are summarized in Table 2.

4.1 ceIssuance: Carrier-endorsed PKIX User Certificate Issuance

The ceIssuance mechanism was developed to facilitate the issuance process of PKIX-based user certificates while satisfying diverse demands of privacy protection from IAS users. It leverages mobile user information that has been verified by cellular network operators during mobile service activation[2], and introduces privacy-preserving certificate signing request (ppCSR) and certificate (ppCert).

Figure 3 presents an overview of the proposed mechanism involving four key parties: (1) Certificate Management Application (**CMA**), which is an MPKIX application running on the applicant's mobile phone; (2) HSS/AuC (Authentication Center), where HSS stores verified user information (e.g., names, ages) and subscriptions (e.g., service plans) of mobile users, and AuC is a subsect of the HSS that maintains secret keys shared with mobile users and generates a pair of challenge and expected response to HSS for user authentication; (3) Certificate Management Server (**CMS**), which is an application server (AS) [47] deployed in the cellular network and can obtain user information from the HSS over the cellular-specific Sh interface [48] with secure communications based on the 3GPP-stipulated Diameter protocol [49] over TLS; and (4) MPKIX-supported CA, which collaborates with cellular network operators to issue PKIX user certificates. Notably, the CMS is a standard-compliant AS accessing the HSS based on the 3GPP-stipulated interface and secure communication protocol, so its deployment does not cause new security threats to cellular networks.

The **ceIssuance** service comprises three parts: (1) secure mutual authentication between CMA and CMS; (2) ppCSR preparation, validation, and endorsement; and (3) ppCert

[2]Verifying mobile user information has been required by the law in many areas (e.g., China and Thailand) and is becoming a mandatory policy [46].

issuance. We describe them in detail below.

4.1.1 Secure Mutual Authentication

We deployed a mechanism of secure mutual authentication between CMA and CMS to defend against the attacks of certificate applicant masquerading and rogue infrastructure. It is based on mobile Extensible Authentication Protocol (EAP), which relies on cellular-specific symmetric cryptography with a secret key K shared between UE (in the (U)SIM card) and HSS. It has two methods, namely EAP-SIM [50] and EAP-AKA [51], which are used by 2G and 3G/4G/5G networks, respectively. They were adopted to enable the secure mutual authentication in MPKIX, and two 128-bit security keys were thus derived and shared between CMA and CMS: (1) K_{aut}, an authentication key used to calculate message authentication code for integrity protection; and (2) K_{enc}, an encryption key used to encrypt data.

In particular, CMA and CMS authenticate each other and derive the above two keys as follows:

Step 1: CMA provides CMS with the user's subscriber identity, i.e., international mobile subscriber identity (IMSI), through an exchange of EAP-Request and EAP-Response identity messages.

Step 2: As an EAP authenticator, CMS obtains a user authentication vector from HSS for the authentication purpose of CMA. The authentication vector contains a random number serving as a challenge, an expected challenge response, a transient master secret key, and a network authentication token, which consists of an ownership proof of the secret key K and a configuration of 3GPP authentication and key generation functions [52]. Note that all the above functions require the secret key K.

Step 3: After receiving the user authentication vector, CMS sends an EAP-Request message carrying the challenge and the network authentication token to CMA.

Step 4: After receiving the EAP-Request message, CMA first validates the ownership proof of the secret key K to authenticate CMS, then generates an answer to the challenge, and finally produces a shared transient master secret key using the configured security functions and the shared secret key K within the (U)SIM card. The transient master secret key is then fed as a seed to an EAP-defined pseudo-random number function [53], and then the function generates a pair of the K_{enc} and K_{aut} security keys. Afterwards, CMA replies an EAP-Response message to CMS with the answer to the challenge.

Step 5: On receipt of the EAP-Response message, CMS verifies the answer and then generates the pair of the K_{enc} and K_{aut} security keys based on the transient master secret key shared with CMA. Note that the K_{enc} and K_{aut} will be generated once when applying for the MPKIX user credential via ceIssuance service.

We further use the K_{enc} and K_{aut} security keys to generate the ppCSR, as described below.

4.1.2 ppCSR preparation, validation, and endorsement

To request an MPKIX certificate, i.e., ppCert, CMA prepares a certificate request, ppCSR, and sends it to CMS for validation and endorsement. For the ppCSR preparation, CMA first generates a pair of private and public keys, and then produce four major elements: (1) subject: containing

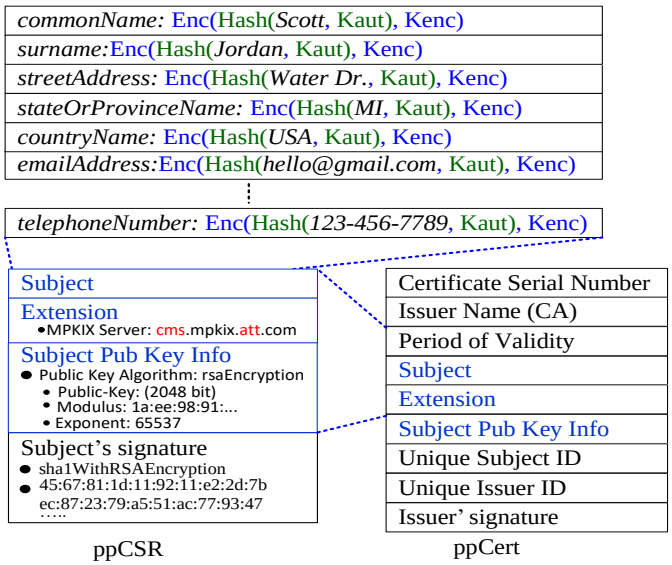


Fig. 4: The formats of ppCSR and ppCert.

user information attributes such as name, address, and phone number; (2) subject extension: domain name of the MPKIX CMS server (e.g., cms.mpkix.att.com); (3) public key information: the generated pubic key and key algorithm; and (4) digital signature. For each attribute, a hash value of the attribute value is generated based on the SHA-1 algorithm and the authentication key (K_{aut}), and then the hash value is encrypted by the AES encryption algorithm and the encryption key (K_{enc}), as illustrated in the upper part of Figure 4.

After receiving the ppCSR from CMA, CMS first verifies the digital signature and then validates the encrypted hash value of each attribute by using the same keys and algorithms shared with CMA and checking authentic user information from HSS. If any error occurs, CMS rejects the ppCSR; otherwise, it endorses the ppCSR by attaching its digital signature and then sends the endorsed ppCSR to an MPKIX-supported CA over a secure channel (e.g., TLS connection).

4.1.3 ppCert Issuance

The MPKIX-supported CA issues a privacy-preserving PKIX user certificate (ppCert) with its digital signature for each valid carrier-endorsed ppCSR from the CMS, as shown in the lower right part of Figure 4. It validates each ppCSR by verifying the digital signatures of both the CMS and the applicant in the ppCSR. The ppCert is then issued to the CMA via the CMS. Note that once the ppCert issuance succeeds, those two security keys (K_{aut} and K_{enc}) associated with the ppCert are recorded in the CMS. They are further used to answer queries from IAS providers when the ppCert is used to access IASs, as described in §4.2.

4.1.4 Compared with Conventional PKIX User Certificates

ppCert has two key advantages over conventional PKIX user certificates. First, the conventional certificate application process requires applicants to provide the CA with their user information, but the ppCert applicants do not need to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3141694, IEEE Transactions on Mobile Computing

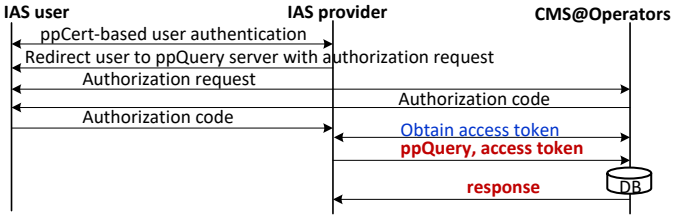


Fig. 5: MPKIX: privacy-preserving query and verification of the user information.

take this action. The reason is that MPKIX leverages the user information that has been verified in the serving cellular network. Second, the conventional certificates, which carry user information in plain text, are delivered without the protection of secure channels[54], so the user information may be leaked; however, only the hash values of encrypted user information are given in the ppCert.

4.2 ppQuery: Privacy-preserving User Information Querying

ppQuery is a carrier-certified service that not only allows IAS providers to query/verify IAS user information but also protects IAS users from the leakage of user information. The ppQuery service comprises three parts: ppCert-based user acquisition, ppQuery access token acquisition, and carrier-certified user information querying, as shown in Figure 5. We elaborate on each of them below.

4.2.1 ppCert-based User Acquisition

An IAS user can send his/her ppCert to an IAS provider and the provider verifies the ppCert based on the CA signature. This ppCert-based user acquisition between the IAS user and the IAS provider can be protected based on one of mainstream security protocols (e.g., HTTPS and SSL/TLS), since ppCert conforms to the PKIX standard, the authentication mechanism of which has been broadly supported in the mainstream security protocols. If the verification fails, the IAS provider may still offer the IAS user anonymous or unrestricted services.

4.2.2 ppQuery Access Token Acquisition

The ppQuery service is provided based on the common OAuth [55] framework. To consume the service, the IAS provider needs to obtain an access token from CMS through the IAS user. As shown in Figure 5, the acquisition procedure of the access token is described below. First, the IAS provider obtains the IAS user's serving CMS server address (e.g., cms.mpkix.att.com) and a unique subject ID from its received ppCert, generates an authorization request including user information for a query, and then redirects the IAS user to the CMS with the authorization request. Second, upon the redirection, the IAS user logs onto the CMS server, reviews the authorization request, and decides if the authorization is granted. Third, given a granted authorization request, the IAS user obtains an authorization code from the CMS server and then forwards it to the IAS provider. Fourth, the IAS provider can receive a ppQuery access token for the IAS user from the CMS by presenting the authorization code to the CMS.

4.2.3 Carrier-certified User Information Query

For each IAS user with a granted authorization request, the IAS provider can use the corresponding access token to query the CMS about the user's information via GSMA OneAPI [40], which is a set of standard APIs designed for external service providers to access cellular network services and user profiles. The CMS responds to the query in accordance with the policy of user-specific privacy protection. The key idea of the privacy protection is to allow an IAS user to specify an anonymization degree of user information in terms of which attributes (e.g., age) can be disclosed.

Moreover, a **minimum individual anonymization level** ($\mathbb{IA}_{min}$) is adopted for each IAS user to guarantee that the user's real identity cannot be discovered or narrowed down to a small group of possible candidates, even though adversaries collect all the user information that the user ever revealed to different IAS providers. Specifically, an IAS user's $\mathbb{IA}_{min}$ represents the minimum percentage of the users with the same disclosed user information as the user in the database of the cellular operator. Thus, for example, if $\mathbb{IA}_{min}$ is set to 20% for an IAS user, adversaries cannot discover the user's real identity but can only narrow down the user identity to a group of possible candidates that take a percentage no smaller than 20% of all the users. Notably, any modification on an anonymization degree that violates the desirable $\mathbb{IA}_{min}$ is denied.

In the following, we first introduce how to anonymize a given subject attribute in the ppCert certificate for a user and then present how an individual privacy protection, i.e., minimum individual anonymization level, spans multiple subject attributes.

Anonymization of a Subject Attribute: MPKIX anonymizes attribute data using the Domain Generalization Hierarchy (DGH) approach [56]. Given a subject attribute, S, and its value, V_S, there is an anonymization function $f_S : (V_S, n) \rightarrow V_S^n$, where n lies in the range between 0 and $L_S - 1$, and L_S indicates the number of anonymization levels for S. S has L_S different attribute values, namely $V_S^0, V_S^1, ..., V_S^{L_S-1}$. V_S^0 is equivalent to V_S and indicates the complete attribute value, whereas $V_S^{L_S-1}$ provides only a minimum detail. Notably, the number of anonymization levels can vary with subject attributes. In some cases, there are only two anonymization levels: disclosed and undisclosed. Each IAS user is allowed to set their preferred number on the anonymization level of each subject attribute.

Consider two examples on the anonymization of subject attributes. The first example attribute is user address. Given $L_S = 4$, there are four different attribute values: V_{Addr}^0 = {State-City-Street-StreetNumber}, V_{Addr}^1 = {State-City-Street-***}, V_{Addr}^2 = {State-City-****-***}, and V_{Addr}^3 = {State-****-****-***}. The second one is cell number. Given $L_S = 3$, three different attribute values are generated as V_{Phone}^0 = 323-111-2222, V_{Phone}^1 = 323-111-****, and V_{Phone}^2 = 323-***-****.

Minimum Individual Anonymization Level ($\mathbb{IA}_{min}$): Although the anonymization level of each subject attribute can be customized by an IAS user, the user may not know which level is sufficiently secure. Moreover, the secure degree of each level depends on the disclosed information itself. For example, if an IAS user's first or last name is rarely used,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3141694, IEEE Transactions on Mobile Computing

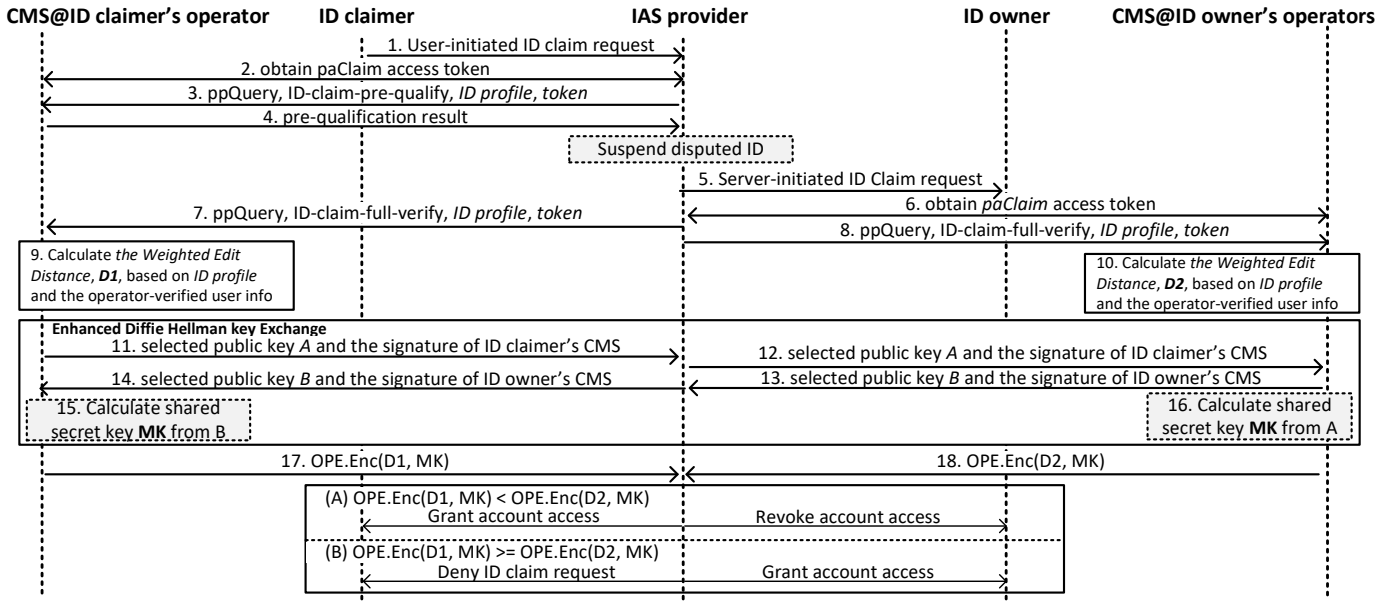


Fig. 6: Privacy-aware ID claim/revocation arbitration mechanism.

adversaries may be able to narrow down the user's identity to a small group of candidates. Once more information is given from other attributes, the user's identity may be further inferred. As a result, the IAS user can be more interested in the anonymization levels that can have at least a certain percentage of the users with the same disclosed user information as the user so that adversaries cannot tell the user's identity among those users, which makes MPKIX less useful in practice.

To address the above concern, we propose a minimum individual anonymization level, $\mathbb{IA}_{min}$, to provide individuals with cross-attributes user privacy protection, thereby preventing adversaries from identifying the real user identities by analyzing all user information attributes that (s)he ever disclosed (partially or fully) to IAS providers. Specifically, $\mathbb{IA}_{min}$ is a configurable parameter indicating the minimum level of $\mathbb{IA}$ for each user; the $\mathbb{IA}$ is an individual anonymization factor representing the current anonymization degree of permitted information disclosure across attributes for individuals. The $\mathbb{IA}$ factor of a user u is defined as:

$$\mathbb{IA}_u = \frac{\sum_{j=1}^{|DB|} \bigcap_{i=1}^{m} (V_{j,S_i}^{H_{u,i}} == V_{u,S_i}^{H_{u,i}})}{|DB|}$$

, where m and $|DB|$ are the number of subject attributes and the number of users, respectively, in the database, and $H_{u,i}$ is the lowest anonymization level ever used by the user for the value of attribute S_i in response to the queries of IAS providers. In other words, the numerator in the above equation indicates the number of users with the same disclosed values of all the attributes as the user u. Intuitively, the higher value the $\mathbb{IA}_u$ has, the more difficult it is for adversaries to identify the user's identity.

Consider an example to calculate $\mathbb{IA}$ for a user u whose first name is John and birth year is 1951 in the Michigan Voter database with 121,489 qualified voters (see more details in §5). For the two subject attributes, first name and age, two anonymization levels are adopted; the former has the undisclosed and fully disclosed levels, whereas the latter is with the undisclosed level and a disclosed level on whether the user age is over 21. Assume that the user is willing to disclose both first name and age, the $\mathbb{IA}_u$ is calculated as $\frac{\text{3,766 (\#users whose first names are John} \bigcap \text{ages over 21)}}{\text{121,489 (\#voters in database)}} = 3.1\%$, which indicates that 3.1% of users in the database or more than 3,760 users have the same values of both subject attributes as the user u.

The $\mathbb{IA}_u$ is calculated for each query from the IAS provider or when any modifications are made to anonymization levels of subject attributes. Whenever $\mathbb{IA}_u$ is smaller than $\mathbb{IA}_{min,u}$, an alert is sent to the user and his/her approval is required. In this study, the default value of $\mathbb{IA}$ is set to 1.6% (see details in §6). To increase the diversity of applicable use scenarios, the current MPKIX prototype is designed to maximize the number of subject attributes without the highest anonymization level, i.e., the least information disclosure, for each user while satisfying his/her desirable $\mathbb{IA}_{min}$.

4.2.4 Compared with Conventional User Information Verification

ppQuery not only offers IAS providers a reliable means to verify user information but also protects user privacy for the access of different IASs. It differs from conventional approaches of user information verification from two aspects. First, ppQuery allows IAS users to disclose verified user information based on different degrees of data anonymization. For example, it is unnecessary for a user to reveal his/her full birthday to Google during account registration since Google only needs to verify if the user is over 18 years old. Second, ppQuery allows IAS users to control information disclosure based on the $\mathbb{IA}$ factor so that the leakage of user identity can be prevented. With conventional approaches, an IAS user may inadvertently reveal different kinds of user information while accessing different IASs; it may allow an

adversary to discover the user's identity and then keep track of his/her activities.

Note that we admit that ppQuery may fail to prevent the identity leakage in some cases, e.g., a user reveals an attribute value which is unique or ignores a privacy leakage alert and agrees to reveal critical information to IAS providers. Some data perturbation techniques may be adopted to address this problem. We leave this improvement to our future work.

4.3 paClaim: Privacy-aware ID Claim/Revocation Arbitration

We developed the paClaim mechanism to improve the efficiency of ID dispute resolution based on the ppQuery service. Figure 6 shows an overview of this mechanism involving two main procedures: (1) ID claimer pre-qualification and (2) order-preserving-encryption (OPE)-enabled ID Levenshtein Distance [57] comparison.

4.3.1 ID Claimer Pre-qualification

The ID claimer needs to pass the ID claimer pre-qualification before initiating an ID claim/revocation request to the IAS provider. It can filter out unnecessary or malicious ID claim/revocation requests by examining whether the carrier-verified user information of the ID claimer is equivalent to those of the disputed ID to some extent. The IAS provider first selects some subject attributes (e.g., the first and last names) for pre-qualification and the ID claimer then needs to prove that his/her name values are similar enough to those of the disputed ID.

In this study, we use the ID Levenshtein Distance (*IDLevDist*) to quantify the similarity; the Levenshtein Distance is the minimum number of single-character edits required to change one word into the other (e.g., the Levenshtein distance between "Alex" and "Alexa" is 1). Notably, for certain subject attributes (e.g., address), different values may still represent the same information (e.g., HK and Hong Kong), and additional formatting functions (e.g., translating a user-entered address to a USPS-suggested address) for attribute values are thus required (more details will be discussed in §6).

Specifically, the pre-qualification process works as follows. First, the ID claimer provides the IAS provider with the access token of a ppQuery service. Second, the IAS provider sends a query to the CMS server of the ID claimer using the access token. The query message comprises three key elements: (1) a subset of provider-selected subject attributes and values for the disputed ID, $Owner = \{S_1, V_1, S_2, V_2, ..., S_n, V_n\}$, where S_i is the ith subject attribute and V_i is the value of S_i; (2) a set of Levenshtein distance weights, $W = \{W_1, W_2, ..., W_n\}$, where W_j is the weight of the Levenshtein distance between V_j and the ID claimer's value for S_j; (3) the maximum of the *IDLevDist* values that are allowed to pass the ID pre-qualification. *IDLevDist* is calculated as $\sum W_i * LevDist(V_i^{claimer}, V_i^{owner})$. Note that, to prevent the IAS provider from inferring the ID claimer's user information, it is suggested that the maximum number of the compared attributes is set to 3. Moreover, the recommended $Owner$ contains the first name, the last name, and an additional provider-selected subject attribute (e.g., address).

Third, the CMS first checks if the computed *IDLevDist* exceeds the maximum value and then sends back the pre-qualification result to the provider. If the ID claimer passes the pre-qualification, the IAS provider initiates the ID claim/revocation arbitration and may temporarily suspend the disputed ID accordingly.

4.3.2 OPE-enabled IDLevDist Comparison

After the ID claimer is pre-qualified for the ID claim/revocation arbitration, the IAS provider initiates it by sending a ppQuery message for full ID verification to the ID claimer's CMS server and the ID owner's (Steps 5-8). Then, each of them computes its own *IDLevDist* (Steps 9-10). Similar to the ppQuery message previously introduced in the pre-qualification, the ppQuery message comprises $Owner$ and W. But, there are two major differences. First, the number of subject attributes specified in $Owner$ is not limited. Second, the maximal *IDLevDist* that is allowed to pass the verification is not specified.

Given those two *IDLevDist* values, the IAS provider can easily determine which of the ID claimer and the ID owner has more operator-verified user information corresponding to the disputed ID. The one with a shorter distance (i.e., smaller *IDLevDist* value) wins and is allowed to access or revoke it. However, the *IDLevDist* value in plain-text may allow the IAS provider to infer additional user information of the ID claimer and the ID owner. For example, the *IDLevDist* given by the ID owner's CMS indicates how close the user information that the ID owner left on the IAS provider is to the operator-verified information of the disputed ID.

To prevent this inference attack, the ID Levenshtein distances computed by the CMSs are not directly returned to the IAS provider; instead, only the distances encrypted by the OPE (Order Preserving Encoding) method [58], which is an encryption algorithm ensuring the order of plain-text numbers to be equal to that of encrypted numbers, are delivered for the comparison. For the secure distribution of the encryption key shared between the CMSs, the Diffie Hellman Key Exchange (DHKE) protocol [59] was adopted; DHKE is a method of enabling the secure exchange of cryptographic keys over public channels. By exchanging security parameters (e.g., two DHKE public keys A and B), DHKE enables the CMSs of ID claimer and ID owner to derive a shared secret key MK using their DHKE private keys for the further OPE-based ID Levenshtein distance encryption (Steps 15-18). With OPE-based ID Levenshtein distance comparison, the IAS provider can identify the one with a shorter distance while preserving the privacy of the ID owner.

Note that current paClaim service only supports one ID claimer in each ID claim/revocation arbitration; if there is more than one user claiming the same IAS ID, multiple arbitrations are required. For example, by assuming that IAS users A and B both claim the ownership of a disputed ID, whose owner is user C currently, and the IAS provider receives A's request first, the IAS provider arranges the first arbitration between users A and C, and then does the second arbitration between user B and the winner of the first arbitration.

4.3.3 Compared with Conventional ID Claim/revocation Mechanisms

The *paClaim* has two key advantages. First, the *paClaim*-based ID claim arbitration can be done in seconds, but the existing mechanisms may take several days or even longer. Second, the *paClaim* does not require current ID owners or claimers to disclose additional operator-verified user information to the IAS provider, whereas current mechanisms (e.g., uploading government-issued ID documents) can inevitably cause an excessive information disclosure.

5 Security Analysis

In this section, we analyze the desirable security guarantees provided by MPKIX and the common attacks against which MPKIX can defend.

5.1 Security Guarantees

Integrity and non-repudiation: MPKIX leverages the merits of current PKIX practice and cellular network security to achieve both data integrity and non-repudiation of the ppCert certificate. To obtain a ppCert certificate, an IAS user needs to create a ppCSR request and attach his/her digital signature. After validating the user's information and digital signature, the serving operator endorses the ppCSR with its digital signature. The operator's signature allows MPKIX-supported CAs to validate the user's ppCSR and digitally sign it. Thus, the accuracy of the user information is guaranteed by the serving operator, and the data integrity is then guaranteed by both the cellular symmetric cryptography with the key K_{aut} (see §4.1) and PKIX asymmetric cryptography with the CA's private key; these two keys are hardly to be stolen. Regarding the non-repudiation property, in many countries/areas, e.g., the European Union and the U.S., previously described digital signatures have legal significance [60]. Therefore, IAS users and operators cannot dispute the authorship/validity of their digital signatures.

Privacy: MPKIX provides IAS users with a multitude of privacy protection. First, MPKIX allows a user to freely determine which subject attributes in the ppCert are disclosed to the IAS provider through the ppQuery service. Since the attribute values in the ppCert are hashed and encrypted, neither the IAS provider nor adversaries can infer the values without the encryption and integrity keys (i.e., K_{aut} and K_{enc}). Second, MPKIX guarantees that adversaries cannot infer the identity of an IAS user or narrow it down to a small group of possible candidates. Third, MPKIX allows an IAS user to create his/her IAS user account with false information due to privacy concerns; however, if the IAS user suffers from ID theft attacks, where an adversary impersonates the user's identity, MPKIX empowers the IAS user to claim/revoke the impersonated ID without revealing more verified user information to the IAS provider.

Accountability: MPKIX allows law enforcement authorities to discover the real identity of an IAS provider or an IAS user, when an IAS-based cyber attack/crime occurs. The IAS provider's identity can be revealed from its CA-signed PKIX server certificate, whereas although the IAS user's identity may not be disclosed in his/her ppCert, the law enforcement authorities can discover it from the user's serving operator, which can be identified through the CA.

5.2 MPKIX's Resilience Against Possible Attacks

We next analyze the resilience of those three MPKIX services against various cyberattacks and discuss how MPKIX deals with other possible attacks (e.g., stealing mobile phones) beyond the adversary model of this study, where the Dolev-Yao model [45] is considered (see details in §3).

Assumptions: Two assumptions are made. First, we assume that the cellular infrastructure is secure, and operators deploy security patches timely. Second, we assume that all security and service protocols (e.g., TLS and OAuth) used by MPKIX are properly configured and with recommended security patches (e.g., eliminating obsolete TLS configurations, such as ECDHE with custom curves [61]).

Notations: We denote an IAS user with a mobile phone having Certificate Management Application (CMA) installed by $\mathbb{A}$, the Certificate Management Server (CMS) by $\mathbb{S}$, the MPKIX-supported certificate authority by $\mathbb{CA}$, the IAS provider by $\mathbb{I}$, the encryption function by Enc, the decryption function by Dec, the function producing message authentication code by Mac, the signature function by Sig, the private key by Pri, and the public key by Pub.

ceIssuance Analysis: We model the ceIssuance service and analyze it in terms of security as follows:

1) $\mathbb{A}$ and $\mathbb{S}$ conduct EAP-SIM/AKA-based mutual authentication and obtain two shared security keys, K_{enc} and K_{aut}.
2) $\mathbb{A}$ sends $Enc(ppCSR|Mac(ppCSR, K_{aut}), K_{enc})$ to $\mathbb{S}$, where '|' is a concatenation operator.
3) $\mathbb{S}$ decrypts the encrypted $ppCSR$ and verifies the MAC using K_{enc} and K_{aut}, respectively. Given a valid $ppCSR$, $\mathbb{S}$ obtains the verified user information from the HSS through Diameter over TLS and compares it to the user information in $ppCSR$.
4) $\mathbb{S}$ sends $Enc(ppCSR|Sig(ppCSR, Pri_S), Pub_{CA})$ to $\mathbb{CA}$.
5) $\mathbb{CA}$ verifies $\mathbb{S}$'s signature using Pub_S. If valid, $\mathbb{CA}$ generates a CA-signed ppCert with its signature $Sig(ppCert, Pri_{CA})$ and sends $Enc(ppCert), Pub_S)$ to $\mathbb{S}$.
6) $\mathbb{S}$ sends the CA-signed $Enc(ppCert, Pub_A)$ to $\mathbb{A}$.

At Step 1, the messages exchanged between $\mathbb{A}$ and $\mathbb{S}$ are plain-text. Adversaries can thus intercept and synthesize those messages to launch Man-in-the-Middle (MitM) attacks. However, Alt *et al.* [62] have proven that the cellular AKA protocol with unique server identifiers attains the properties (e.g., state-confidentiality and soundness) that can defend against MitM attacks even in the presence of corrupted servers. Thus, adversaries cannot compromise the mutual authentication, and further infer K_{enc} and K_{aut}.

At Steps 2-3, adversaries may apply for a CA-signed PKIX user credential on behalf of $\mathbb{A}$ by launching an impersonation attack. However, without K_{enc} and K_{aut}, the adversaries cannot generate a valid request message, $Enc_(ppCSR|Mac(ppCSR, K_{aut}), K_{enc})$.

At Steps 4-6, all the message exchanges of $ppCSR$ and $ppCert$ are provided with confidentiality and integrity protection. Thus, without the private keys of $\mathbb{CA}$ and $\mathbb{S}$, adversaries cannot decrypt any intercepted ciphertext messages or fabricate digital signatures of $\mathbb{CA}$ and $\mathbb{S}$.

ppQuery Analysis: We model the ppQuery service and do security analysis on it below.

1) $\mathbb{A}$ sends *Client Hello* to $\mathbb{I}$.

2) $\mathbb{I}$ sends *Server Hello*, *Certificate*, *Server Key Exchange*, *Certificate Request*, and *Server Hello Done* to $\mathbb{A}$.
3) $\mathbb{A}$ sends **ppCert**, *Server Key Exchange*, *Certificate Verify*, *Change Cipher Spec*, and *Finished* to $\mathbb{I}$.
4) $\mathbb{I}$ sends *Change Cipher Spec* and *Finished* to $\mathbb{A}$.
5) $\mathbb{I}$ obtains the CMS address (i.e., $\mathbb{S}$) and the subject ID of $\mathbb{A}$ from **ppCert** for further ppQuery operations, which verify correctness of the user information provided by $\mathbb{A}$.
6) $\mathbb{I}$ initiates an OAuth-based ppQuery access token acquisition with $\mathbb{S}$ over TLS.
7) $\mathbb{I}$ sends an encrypted ppQuery message, $Enc(ppQuery|Token|Mac(ppQuery|Token, K_{int_{TLS}}), K_{enc_{TLS}})$, to $\mathbb{S}$, where $K_{enc_{TLS}}$ and $K_{int_{TLS}}$ are the encryption key and integrity key derived from the establishment of TLS connection between $\mathbb{I}$ and $\mathbb{S}$.
8) $\mathbb{S}$ sends an encrypted response, $Enc(Response|Mac(Response, K_{int_{TLS}}), K_{enc_{TLS}})$ to $\mathbb{I}$.

At Steps 1-4, $\mathbb{A}$ and $\mathbb{I}$ establish a TLS connection while authenticating each other. In particular, $\mathbb{A}$ provides $\mathbb{I}$ with *ppCert* during the TLS connection establishment. Adversaries may intercept and synthesize those handshake messages including *ppCert* to launch MitM attacks, infer the verified user information, or conduct long-term user tracking attacks. However, MPKIX is immune to these attacks due to the following three reasons. First, according to a recent NSA (National Security Agency) report [61], an established TLS connection is considered as a secure communication channel against various MitM attacks (e.g., MitMProxy and SSLSplit attacks) when obsolete TLS configurations are avoided. Second, the values of subject attributes in *ppCert* are encrypted hashed values (see Figure 4), and the used keys, K_{enc} and K_{aut}, are hardly obtained from the ceIssuance service. Third, the real-world risk of ppCert-based user tracking attacks is limited since MPKIX guarantees that adversaries cannot discover the real identity of a ppCert owner or narrow it down to several possible individuals.

At Steps 5-6, adversaries may attempt to launch various attacks against token acquisition and usage, but Fett *et al* [63] have proven that the OAuth protocol establishes strong authorization, authentication, and session integrity guarantees, which can well defend potential attacks.

At Steps 7-8, $\mathbb{I}$ sends a ppQuery message with the granted access token to $\mathbb{S}$, and $\mathbb{S}$ replies a response to $\mathbb{I}$ based on $\mathbb{A}$'s privacy protection setting. To defend against possible cyberattacks, the ppQuery request and response messages are protected with confidentiality and integrity using those two keys, $K_{enc_{TLS}}$ and $K_{int_{TLS}}$.

paClaim Analysis: The paClaim service is comprised of three ppQuery request-response transactions over TLS for the qualification examination of the ID claimer, the collection of the OPE-encoded ID Levenshtein distance from the ID claimer's CMS, and that from the ID owner's CMS, respectively. Since the user information verification and message exchange in the ppQuery service have been analyzed, we here focus on the security analysis of deriving the shared OPE security keys at CMSs (i.e., Steps 11-18 in Figure 6).

1) $\mathbb{S}_{\text{Claimer}}$ selects a DHKE (Diffie Hellman Key Exchange) public key, X_{S1} and a DHKE private key, Y_{S1}, generate a signature, $Sig(X_{S1}, Pri_{S_{\text{Claimer}}})$ for X_{S1} using its $Pri_{S_{\text{Claimer}}}$, and sends $X_{S1}|Sig(X_{S1}, Pri_{S_{\text{Claimer}}})$ to $\mathbb{I}$.
2) $\mathbb{I}$ forwards $X_{S1}|Sig(X_{S1}, Pri_{S_{\text{Claimer}}})$ to $\mathbb{S}_{\text{Owner}}$.
3) $\mathbb{S}_{\text{Owner}}$ selects a DHKE public key, X_{S2} and a DHKE private key Y_{S2}, generate a signature, $Sig(X_{S2}, Pri_{S_{\text{Owner}}})$, for X_{S2} using its $Pri_{S_{\text{Owner}}}$, and sends $X_{S2}|Sig(X_{S2}, Pri_{S_{\text{Owner}}})$ to $\mathbb{I}$.
4) $\mathbb{I}$ forwards $X_{S2}|Sig(X_{S2}, Pri_{S_{\text{Owner}}})$ to $\mathbb{S}_{\text{Claimer}}$.
5) $\mathbb{S}_{\text{Claimer}}$ calculates the shared OPE security key using DHKE algorithm[3] as: $(X_{S2})^{Y_{S1}}$ mod q, where q is a prime number shared by all CMSs MPKIX.
6) $\mathbb{S}_{\text{Owner}}$ calculates the shared OPE security key using DHKE algorithm as: $(X_{S1})^{Y_{S2}}$ mod q.

Different from the ppQuery service, where outside adversaries are considered, the paClaim service may suffer from an inside adversary, the IAS provider (i.e., $\mathbb{I}$), which may be interested in discovering the plain-text ID Levenshtein distances from $\mathbb{S}_{\text{Claimer}}$ and $\mathbb{S}_{\text{Owner}}$ to infer more user information. Thus, it can motivate $\mathbb{I}$ to compromise the procedure of the OPE security key exchange by launching an MitM attack [64]. To this end, $\mathbb{I}$ first selects two DHKE key pairs: (1) $X_{I\leftrightarrow S_{\text{Claimer}}}$ and $Y_{I\leftrightarrow S_{\text{Claimer}}}$ and (2) $X_{I\leftrightarrow S_{\text{Owner}}}$ and $Y_{I\leftrightarrow S_{\text{Owner}}}$, intercepts X_{S1} and X_{S2}, and then sends $X_{I\leftrightarrow S_{\text{Claimer}}}$ and $X_{I\leftrightarrow S_{\text{Owner}}}$ to $\mathbb{S}_{\text{Claimer}}$ and $\mathbb{S}_{\text{Owner}}$, respectively. In the unmodified DHKE protocol, $\mathbb{I}$ can obtain two shared OPE security keys: one is for $\mathbb{I}$ and $\mathbb{S}_{\text{Claimer}}$ (i.e., $(X_{S1})^{Y_{I\leftrightarrow S_{\text{Claimer}}}}$ mod q), and the other is for $\mathbb{I}$ and $\mathbb{S}_{\text{Owner}}$ (i.e., $(X_{S2})^{Y_{I\leftrightarrow S_{\text{Owner}}}}$ mod q), and further discover the plain-text ID Levenshtein distances.

However, the paClaim service is immune to the above MitM attack. This is because S_{Claimer} and S_{Owner} attach their digital signatures while transmitting X_{S1} and X_{S2} to $\mathbb{I}$ at Steps 1 and 3, respectively. Without their private keys, $Pri_{S_{\text{Claimer}}}$ and $Pri_{S_{\text{Owner}}}$, $\mathbb{I}$ cannot produce the digital signatures and have them accepted the fabricated DHKE keys.

Other potential attacks: We next discuss how MPKIX defends against several potential attacks beyond the Dolev-Yao adversary model.

- *(U)SIM Card Compromising Attacks:* By compromising mobile users' (U)SIM cards, adversaries can apply for ppCerts on behalf of them. There have been several SIM-based attacks, which include inferring the secret key K_i by abusing A3 algorithm COMP128v1 [65], rooting SIM cards via insecure OTA [66], and launching a SIM swap attack [67]. The root causes mainly lie in improper configurations of the cellular network [65], security flaws from SIM card manufacturers [66], and social engineering attacks [67]. Most of these attacks can be addressed with proper configurations and timely security patches.
- *Mobile Phone Compromising Attacks:* An adversary may infer user information from a pre-compromised mobile phone by eavesdropping on the issuance of carrier-endorsed PKIX user certificate. However, MPKIX is immune to this attack since no plain-text user information is sent over the air. Moreover, such attack requires root privilege of the compromised phone, which has been shown with a significant technical challenge [68], [69], [70].

[3]The DHKE algorithm is based on the discrete logarithm problem; given α and a, find b so that $\alpha^b = a$.

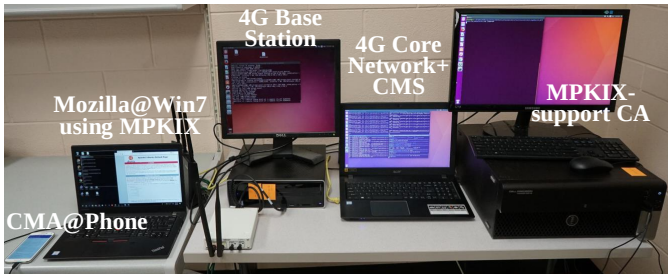


Fig. 7: MPKIX prototype.

- *Stealing Phones:* If an IAS user's phone is stolen, his/her PKIX user credential may be abused. However, this problem can be largely mitigated by an action that the user promptly updates the public CRL (Certificate Revoke List) to revoke his/her credential. Furthermore, compared with traditional cryptographic tokens (e.g., YubiKey), the MPKIX phone-based cryptographic tokens provide better security of user credentials. Specifically, modern smartphones support a variety of bio-based security mechanisms, such as fingerprint and facial recognition, which can prevent adversaries from abusing user credentials on stolen/lost phones.

6 Implementation of MPKIX

Figure 7 illustrates three key entities of the MPKIX prototype: CMA on a mobile phone, an MPKIX-enabled 4G LTE infrastructure with CMS, and an MPKIX-supported CA. Each of them is elaborated below. Notably, the secure communications between MPKIX-supported network elements were enabled by TLSv1.2 using the ciphersuite of ECDHE-RSA-AES256-SHA, and the cryptographic key operations were implemented using the OpenSSL [71] library.

CMA was written in Java and implemented on four low/mid-end smartphones including Samsung S2 (2011), Samsung S5 (2014), Sony Xperia Z (2013), and Google Pixel XL (2016). Notably, successfully deploying CMA on these old phone models with fewer computing resources than modern ones indicates that CMA works for most phone models. CMA uses the credential services provided by an IsoApplet [72] to generate public and private keys, prepare ppCSR, and obtain/maintain CA-issued ppCert. Moreover, to increase the applicability of the MPKIX credential service, the PKCS#11 (Public Key Cryptography Standards [73]) interface, which is a standard platform-independent API to access diversified cryptographic tokens and has been broadly supported by many operating systems, was implemented on CMA. It enables CMA to transform an IAS user's phone to a cryptographic token with the PKCS#11 interface.

MPKIX-enabled 4G LTE infrastructure was set up using the SDR(software-defined radio)-based OpenAirInterface (OAI) platform, which comprises a 4G LTE core network and a base station. The core network was deployed on a Lenovo desktop with Intel i7-9700k and 16GB RAM, whereas the base station was built on a PowerSpec desktop with Intel i7-9700K and 16GB RAM connecting to an Ettus USRP B210. In the core network, a CMS server supporting *ceIssuance*, *ppQuery*, and *paClaim* services was deployed. We next introduce implementation details about CMS.

- **CMS services.** Three MPKIX services were implemented: (1) the *ceIssuance* service used OpenSSL [71] to implement cryptographic key operations, and employed Node-diameter [74], a diameter protocol over TLS, to enable secure communications with HSS; (2) the *ppQuery* service was implemented on top of oauth2-server [75] and enabled to support OAuth (using scribejava-6.9.0 [76]) and OneAPI (using an open GSMA OneAPI library [40]); and (3) the *paClaim* service used the Boost Algorithm [77] and Fast OPE [78] to calculate the ID Levenshtein distances and perform the order-preserving encryption, respectively.
- **CMS database.** The SQL database was built on top of the CryptoDB library [79] to store and anonymize user information obtain from HSS. To emulate real mobile user data, the HSS's database contained the information of 120,531 users, which was purchased from a voter registration database [80] with 120,531 voters. The user attributes included name, gender, birthday, address, and phone number. In the current prototype, the data anonymization levels for each attribute are as follows (the information specified at each level was disclosed): (1) *name*: two levels (L0: full name; L1: none); (2) *gender*: two levels (L0: gender; L1: none); (3) *birthday*: six levels (L0: year, month, and day; L1: year and month; L2: year; L3: small age ranges ({0-17, 18-40, 41-60, 61-80, >80}); L4: large age ranges ({0-40, 40-80, >80}); L5: none); (4) *phone number*: four levels (L0: phone number; L1: last seven digits; L2: last four digits; L3: none). (5) *addresses*: five levels (L0: street number, street name, city, and state; L1: street name, city, and state; L2: city and state; L3: state; and L4: none). Notably, to tackle the different addresses that have the same legal semantics (e.g., HK and Hong Kong), we will reformat all address inputs to unified ones using Google Geocoding API [81] prior to the data processing.

Moreover, the value of $\mathbb{IA}_{min}$ was set as $\frac{2{,}000}{120{,}531} \simeq 1.66\%$ for all the users; it indicates that the number of mobile users who have the same disclosed user information cannot be smaller than 2,000. With the given $\mathbb{IA}_{min}$, the MPKIX prototype can automatically adjust the number of subject attributes and the anonymization level of each attribute, if needed, for each user. In particular, to improve the diversity of applicable use scenarios, the current MPKIX prototype is designed to maximize the number of a user' subject attributes that do NOT apply the highest anonymization levels while ensuring desirable $\mathbb{IA}_{min}$

MPKIX-supported CA was written in Java and used the bouncycastle-v1.6 [82] library, a lightweight cryptography library, to validate ppCSR and generate ppCert. It was deployed on a Dell 5810 precision tower.

7 Evaluation of MPKIX

In this section, we evaluated the effectiveness and performance of the three key MPKIX services.

7.1 ceIssuance

We evaluated the ceIssuance service by two metrics: (1) certificate issuance time, which is the time required by an

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3141694, IEEE Transactions on Mobile Computing

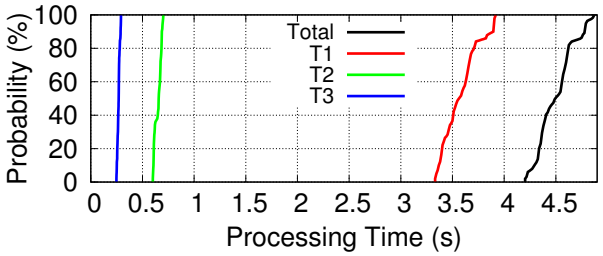


Fig. 8: PKIX user certificate issuance time. *T1* is the time between the generation of a public/private key pair and that of ppCSR; *T2* is the time between when ppCSR is sent by the phone and when the carrier-endorsed ppCSR is received by the CA; *T3* is the period from the time right after the end of T2 to that ppCert is received by the phone.

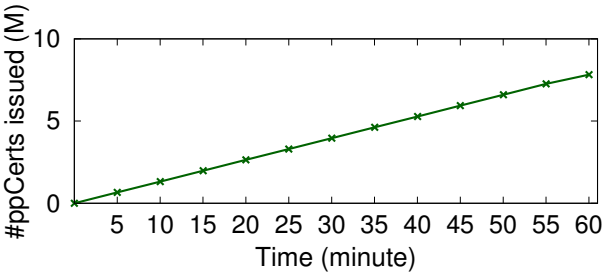


Fig. 9: #ppCerts issued by the MPKIX infrastructure.

IAS user to generate a pair of public and private keys, prepare a ppCSR request, and obtain a ppCert certificate on a mobile phone, but does not include the time required by the user to input user data; and (2) certificate issuance rate, which indicates the maximum number of PKIX user certificates issued by the MPKIX prototype per unit of time. Finally, the overhead of supporting varied anonymization levels was evaluated.

Experimental settings: For the issuance time, the experiment was intentionally conducted on a low-end mobile phone, Samsung Galaxy S5 equipped with Qualcomm Snapdragon801 CPU and 2GB RAM, and had 20 runs. For the issuance rate, a program was developed to keep sending ppCSR to the MPKIX infrastructure. The experiment lasted for one hour, where each new ppCSR was sent right after the ppCert of the last ppCSR, if there was any, was received.

Experimental results: Figure 8 plots the CDF of the time spent on the overall issuance and its three events including ppCSR preparation (T1), ppCSR validation (T2), and ppCert generation (T3). We have three observations. First, an IAS user can obtain a ppCert certificate within 5 s even on a low/medium-end smartphone, but typical CAs, e.g., GlobalSign and DigiCert [27], [28], require several days for a certificate application. Second, T1 ranges from 3.3 s to 3.8 s, whereas T2 and T3 take only 1.2-1.7 s. The main reason is that the IsoApplet used by the current CMA prototype required more actions to carry out the credential service functions because of its data length limitation, no larger than 256 bytes, for communicating with external applications. The usage of the IsoApplet, a lightweight Java applet offering credential services, is to support low/medium-end resource-constrained phones. Notably, the maximum values of the observed instant RAM and CPU usages in the experiment for the CMA are 57 MB and 27%, respectively. Our experiment results show that even on a low/medium-end smartphone, a user is still able to obtain his/her CA-issued PKIX user certificate within less than 5 seconds, whereas the typical CAs, e.g., GlobalSign and DigiCert, require several days [27], [28].

Figure 9 plots the issuance rate of the number of ppCerts issued per minute. It is observed that the MPKIX infrastructure issued around 130,000 ppCerts per minute and issued a total of 7.82 million ppCerts without any significant variance within an hour. It shows that the MPKIX infrastructure has a stable issuance performance.

For the overhead of supporting varied anonymization levels, CMS produces the values of all the subject attributes based on given anonymization levels for each ppCSR request before sending the carrier-endorsed ppCSR to CA. For example, four values are produced for the *phone number* attribute with four anonymization levels. In this experiment, we used a global variable α as the maximum anonymization level for all the subject attributes and then examined whether varied anonymization levels would affect T2 (ppCSR validation time) by varying α. The experiment was conducted with 20 runs for each level.

The result shows that T2 was increased by 42 ms, 48 ms, 54 ms, and 59 ms when α was set to 1, 2, 3, and 4, respectively, compared with the case with $\alpha = 0$. Although T2 is observed to increase with anonymization level, producing values for all the anonymization levels is conducted only once at CMS for each carrier-endorsed ppCSR, and does not affect the subsequent ppQuery response times regardless of user privacy settings.

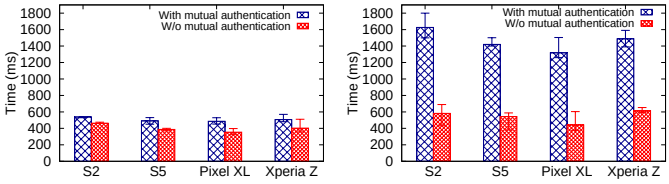
7.2 ppQuery

We evaluated the ppQuery service based on not only correctness, but also two metrics: (1) IAS access time, which is the time required by an IAS user to establish a secure TLS connection with an IAS server using his/her CA-issued ppCert; (2) IAS query time, which is the time spent by the IAS server on receiving a query response after submitting the query.

Experimental settings: We randomly selected two IAS users from the user database at HSS: one user is older than 18 years old, whereas the other is not. After obtaining their CA-issued ppCerts through MPKIX, these two selected users attempted to connect with an IAS server, which allowed only users older than 18 years old, using browsers on phones and computers. During the connection of each user, the IAS server examined the age eligibility of the user by querying the CMS server and then determined whether the user is allowed to have the access. The experiment was conducted with 20 runs.

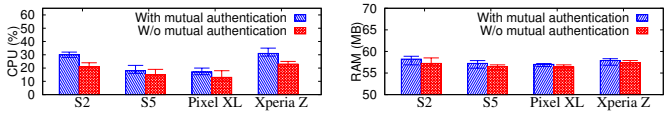
Experimental results: Figure 10 plots the statistics of the IAS access time for the WebKit browser on different MPKIX-enabled phones and the Mozilla browser on a Windows computer connecting to those phones for the MPKIX service. It is observed that the Mozilla browser requires 1.5s on average for the IAS access time On the contrary, the Webkit browser takes only 0.5s on average. The reason is that the WebKit can access the CMA locally on the phones, whereas the Mozilla cannot. Moreover, compared to the case without

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3141694, IEEE Transactions on Mobile Computing



(a) WebKit on phones (b) Mozilla on computer

Fig. 10: The IAS access time (max/med/min) involving the establishment of a TLS connection with or without mutual authentication varies with MPKIX-enabled phones and a computer connecting to those phones.



(a) CPU usage (b) RAM usage

Fig. 11: The peak CPU and RAM usage statistics (max/med/min) of the MPKIX-enabled phones for the ppQuery service.

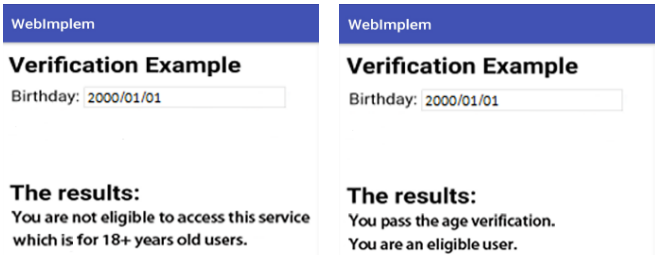
TLS mutual authentication as shown in Figure 11, the peak CPU and RAM usages are increased by 2-10% and 1-2 MB, respectively.

Figure 12 confirms that the IAS server successfully verified the ages of the IAS users using the ppQuery service. The result showed that the IAS took 1.2s averagely, where 0.5s IAS access time and 0.7s IAS query time, on the query of a single attribute. Notably, it was observed that no full birthday information was returned to the IAS server in the experiment.

7.3 paClaim

We finally evaluated the effectiveness and performance of the *paClaim* service. Three performance metrics were used: (1) T_{pre}, the time required to perform the ID pre-qualification (Steps 1-4 in Figure 6); (2) T_{dis}, the time required to calculate the full ID Levenshtein distances (Steps 5-10 in Figure 6); (3) T_{ope}, the time required to perform the order-preserving encryption with key exchange (Steps 11-18 in Figure 6).

Experimental settings: We randomly selected 11 users from the user database at HSS; the first 10 users were assumed to be the victims of an ID theft attack and denoted as benign users, whereas the last user was an attacker of the ID theft. We obtained ppCerts for all the users through MPKIX. On the IAS server, the attacker created 10 accounts



(a) User: no older than 18. (b) User: older than 18.

Fig. 12: The IAS connection results based on the ppQuery.

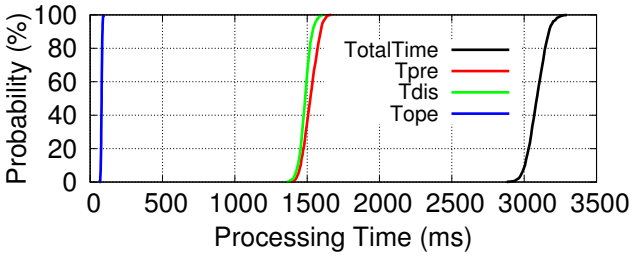


Fig. 13: The CDF of the ID claim/revocation arbitration time.

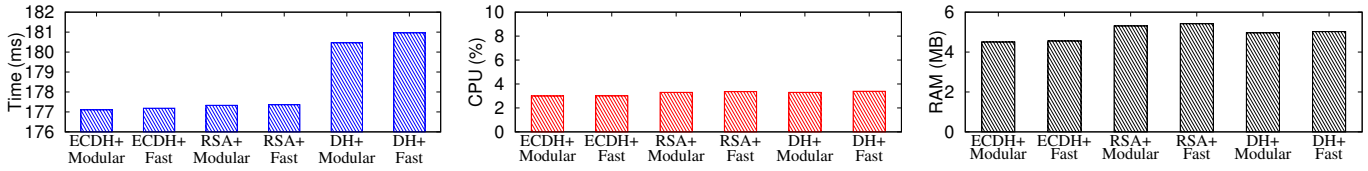
impersonating those 10 benign users, respectively, with their first and last names. An ID claim/revocation arbitration was performed for each benign user. For the ID claim pre-qualification at the IAS server, $Owner$ comprised two subject attributes {first name, last name}, $W = \{1, 1\}$ indicated the IDLevDist weights of those two attributes, and the maximal *IDLevDist* value was set to 5. For the ID claim full verification, $Owner$ and W were set to {first name, last name, year of birth, state of address} and $\{3, 3, 2, 1\}$, respectively. The experiment was conducted with 100 runs.

Experimental results: It was observed that each of those 10 benign users passed the ID claim pre-qualification and won the arbitration. The statistics of the total arbitration time and the time spent on each stage, T_{pre}, T_{dis} and T_{ope}, are plotted in Figure 13. We have two observations. First, the overall ID claim arbitration process could be finished within 3.4s under the condition that the ID owner can timely respond to the arbitration request. Second, the 90th percentile values of T_{pre}, T_{dis}, and T_{ope} are less than 1.63s, 1.58s, and 0.18s, respectively. The results have confirmed the effectiveness and efficiency of the paClaim service.

We further studied the impact of different cryptographic schemes on the performance of the paClaim service, especially for the OPE key exchange and encryption mechanisms (Steps 11-18 in Figure 6). Specifically, we considered three key exchange schemes, Diffie-Hellman (DH), Elliptic Curve Diffie-Hellman (ECDH), and RSA, and two OPE algorithms, Modular OPE [83] and Fast OPE [78]. For each combination of a key exchange scheme and an OPE algorithm, we initiated ID claim requests and measured T_{ope}, CPU usage, and RAM usage on average at CMS; there are totally six combinations of the cryptographic schemes. We make three observations from the experimental results, as shown in Figure 14. First, the ECDH-based schemes are faster than the others; the EDCH scheme plus modular OPE achieves the smallest T_{ope}, whereas the DH scheme plus fast OPE leads to the largest T_{ope}. Second, all the cryptographic schemes have comparable CPU usages, but the CPU usages of the RSA-based schemes are slightly higher than the others. Third, the ECDH-based schemes consume about 500 KB RAM less than the others.

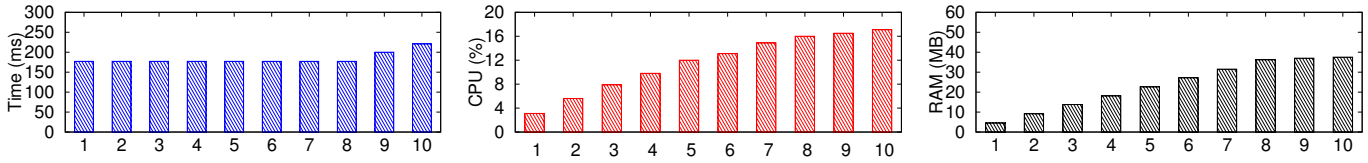
We finally studied the performance and overhead of the paClaim service with a varying number of concurrent ID claim requests. The above experiment was repeated with two modifications: (1) the most efficient combination of cryptographic schemes, ECDH plus modular OPE, was used for MPKIX; and (2) the number of concurrent ID claim requests initiated ranges from 2 to 10. The results show that T_{ope}, CPU usage, and RAM usage are increased from 180 ms

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3141694, IEEE Transactions on Mobile Computing



(a) T_{ope}: order-preserving encryption time (b) CPU usage (c) RAM usage

Fig. 14: The statistics of T_{ope}, CPU usage and RAM usage at CMS for different cryptographic schemes.



(a) T_{ope}: order-preserving encryption time (b) CPU usage (c) RAM usage

Fig. 15: The statistics of T_{ope}, CPU usage and RAM usage at CMS for different numbers of concurrent ID claim requests.

to 210 ms, from 3% to 18%, from 4 MB to 38 MB, respectively, when the number of concurrent requests increases from 2 to 10, as shown in Figure 15.

In summary, our experimental results confirm not only the effectiveness of the paClaim service, where concurrent ID claim requests can be processed efficiently, but also its merit that largely reduces the time of ID claim/revocation arbitration, compared with current technologies, while preserving IAS user privacy.

8 Discussion

In this section, we discuss potential concerns about deploying and using MPKIX, as well as its limitations.

Incentives for MPKIX deployment. We believe that all the involved parties can benefit from the MPKIX deployment. The reasons are four-fold. First, *CAs* can expand their enterprise-based PKIX credential services to billions of mobile users, and the cost of user information verification can be greatly reduced without the need for photo ID inspection. Second, *cellular network operators* can make profit on new services including offering MPKIX to IAS providers and providing mobile users with emerging services (e.g., facilitating the aircraft boarding process and short-term keyless apartment rental) based on user certificate. Third, *IAS providers* can ensure the correctness of user information so that the risk of cyberattacks with potential legal issues and complaints can be reduced, e.g., those from governments [84] and online advertisers [8], where there is a total of $1.3 billion loss due to fake followers. Fourth, *IAS users* can transform their phones to PKCS#11-supported cryptographic tokens supporting a variety of PKIX credential services, with an efficient privacy-aware mechanism of ID claim/revocation arbitration.

Note that an IAS user may not need to pay the serving CA for the PKIX user certificate issuance if a reciprocal agreement between the connected cellular operator and the CA is signed. This business model is commonly observed in practice. For example, Google provides users with free cloud services but makes profit from online advertisers.

Enforcing users to disclose more information? People may think that MPKIX enforces IAS users to disclose more user information to cellular network operators, CAs, and IAS providers, compared with traditional mechanisms of user certificate. However, it is not true due to three major reasons. First, MPKIX leverages only the existing user information that has been verified by the operators. Second, MPKIX prevents IAS users from revealing user information to the CAs by encrypting data in carrier-endorsed ppCSRs. Third, MPKIX provides the service of ID claim/revocation arbitration to IAS users without the need of disclosing additional information to IAS providers. This privacy protection mechanism does not exist in current solutions [23].

Why use cellular network infrastructure? People may wonder why build MPKIX with cellular network operators but not the other institutions (e.g., banks and insurance companies) that also have verified user information. The reasons are two-fold. First, the cellular network infrastructures are built based on GSMA and 3GPP standards with a unified framework, but those institutions have diversified network systems. The standardized and unified cellular framework allows MPKIX to be developed on top of the GSMA OneAPI [40], which is generally supported by cellular operators, so that MPKIX can be easily deployed in operational cellular networks. Second, the CMS server is deployed as a 3GPP-defined application server (AS) [47], which can securely access HSS via the standardized Sh interface [48], but the other institutions may be afraid that deploying a new server in their network infrastructures may cause new security threats, especially for its access of their user information databases.

Why not use email certificates? Several CAs can issue a user with an X.509 email certificate within a few minutes. However, this kind of certificates can prove only the access of a particular email account for the user, but not other user information such as age and address.

How about family-plan users? In some countries, operators offer mobile services with family plans that contain more than one user. Some of them verify only the ID of the primary user; such a case is currently not supported by MPKIX. We leave it to our future work.

How about photo-based ID theft attack? An ID thief may impersonate an IAS user by using only the user's personal photo without other user information such as name and birthday. MPKIX can be extended to effectively defend against this attack due to three reasons. First, cellular

network operators can easily obtain verified user photos while verifying each user's government-issued photo ID for service activation. Second, ppCert can carry any type of octet data including an encrypted user photo by using X.509 certificate extensions [25]. Third, given the encrypted user photo in ppCert, MPKIX can verify a provided user photo based on face recognition techniques, and thus prevent the photo-based ID theft attack.

Is the paClaim service better than current solutions? The common approach against ID theft attacks is to do the manual inspection on government-issued photo IDs or other proof documents provided by ID claimers. Seemingly, by involving the investigators of IAS providers with more user information, this approach is error-free and more rigorous than MPKIX. However, it may not be the case due to three reasons. First, this approach is not considered to proceed in a scientific way. The efficiency and accuracy of the ID claim process highly depend on the investigators, and they may delay or have a bias due to human factors. For example, one police's personal information was abused to create a Facebook ID by an adversary; however, the disputed ID still remained valid for a long time after a revocation request corresponding to the ID was submitted to Facebook [24]. Second, the current approach may be still vulnerable to ID owner masquerading attacks. With a stolen photo ID or a utility bill from a benign IAS user (e.g., accessing mailbox), an adversary can submit a request to an IAS provider by masquerading as the ID owner, and successfully claim the ownership of the benign IAS user account. However, with MPKIX, an IAS user cannot submit any ID claim/revocation request without passing the pre-qualification based on verified user information from CMS. For example, Bob cannot claim the ownership of Alice's account, even when Bob possesses Alice's driver license. Third, the photo-IDs and proof documents provided by ID claimers may compromise user privacy by leaking more user information. On the contrary, the proposed paClaim service is a scientific, rigorous and privacy-protected approach.

9 Related Work

Side-channel inference/verification: Several methods have been proposed to infer/verify user demographics (e.g., age, gender and education level) using side-channel information (e.g., HTTPS packets and social network activities). Specifically, Wang *et al.* [32] developed a tensor factorization based method, *Dinfer*, for inferring user demographic attributes from WiFi AP trajectories; Li *et al.* [30] applied machine learning to analyzing campus WiFi traffic and inferred the user's gender and education level; Neal *et al.* [31] devised a multimodal-based approach to predict user gender based on usage records of Bluetooth and Wi-Fi. However, these schemes have several common issues. First, the error rates are not negligible (e.g., 22% in [30] and 9% [31]). The erroneous inference results for IAS users may lead to unnecessary suspension or mistaken operations of IAS services. Second, the above inference methods can only be applied to registered users, so they do not protect IAS providers from numerous ID-related attacks during user registration.

Public-key infrastructure: The PKI has been widely developed and studied in recent years. Specifically, two studies [85], [86] conducted a large-scale analysis of current PKI-based certificate ecosystem, whereas another study [87] used practical symbolic execution to expose noncompliance in X.509 certificates. Moreover, Aas *et al.* [33] introduced an automated certificate authority, *Let's Encrypt*, for free issuance of HTTPS certificates. Wang *et al.* [88] distributed the trust for certificate authenticity between the corresponding CA and the certificate owner by letting them co-sign the certificate. Wang *et al.* [89] employed cache spaces on IoT devices as a large pool to store validated certificates. Hoglund *et al.* [90] introduced a lightweight profile for X.509 digital certificates for resource-constrained IoT devices. Rashid *et al.* [91] and Papageorgiou *et al.* [92] developed a blockchain-based public key infrastructure for the decentralized issuance and management of digital certificates.

Different from the above studies, MPKIX aims to leverage cellular networked systems to provide IAS providers with authentic user information and protect IAS users from nefarious ID theft attacks while preserving user privacy. Notably, this problem has not been addressed.

Mobile Connect: Mobile Connect[35] enables mobile users to log onto IAS services using mobile phones. Specifically, when an IAS user accesses an IAS service, a cellular-network-initiated user authentication is conducted on the user's mobile phone. The authentication result is then returned to the IAS server. The IAS user can choose to provide the IAS provider with nothing, Mobile Connect identity (i.e., phone number) only, or Mobile Connect identity and other user information (e.g., birthday).

Compared with MPKIX, Mobile Connect has the following limitations. First, an IAS user using Mobile Connect is required to use his/her mobile phone and have cellular network connectivity on it while accessing an IAS service; this requirement may decrease the applicability of Mobile Connect. However, MPKIX does not have this limitation, since it supports not only computers connecting to mobile phones with the MPKIX service but also an offline mode in which a CA-issued ppCert and its corresponding private key are exported to other cryptographic tokens (e.g., Yubikey). Second, Mobile Connect does not provide users with a cross-IAS querying mechanism with fine-grained privacy-preserving configuration. It allows IAS users to disclose only least information for user verification. Third, Mobile Connect does not support a privacy-aware ID claim/revocation mechanism, which prevents users from disclosing additional information to IAS providers for the ID claim or revocation. However, the above two mechanisms are supported by MPKIX.

10 Conclusion

Both IAS providers and users face various security threats nowadays. IAS providers are abused by adversaries based on fake user accounts, since they have no reliable means to verify correctness of user information. IAS users suffer from nefarious ID theft attacks, which lead to both financial losses and emotional/physical health damages. To address these security threats, we proposed the MPKIX framework to improve the security and accountability of IAS. MPKIX offers IAS providers with a general, reliable mechanism of

user information verification, which makes IAS users accountable while largely preserving user privacy. Specifically, three novel mechanisms with privacy protection are introduced for user verification, namely carrier-endorsed PKIX user certificate issuance, privacy-preserving user information querying, and privacy-aware ID claim/revocation arbitration. Our evaluation results have shown that MPKIX is an effective and scalable approach. MPKIX not only provides a potent solution to secure the present-day IAS, but also benefits all the involved parties.

ACKNOWLEDGMENTS

This work is supported in part by the National Science Foundation under Grants No. CNS-1814551 and CNS-1815636. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors only and do not necessarily reflect those of the National Science Foundation.

REFERENCES

[1] "Communications decency act of 1995," https://www.congress.gov/bill/104th-congress/senate-bill/314/, 1995.
[2] CBS News, "Google to pay 170 million for violating kids' privacy on youtube," 2019.
[3] Inquirer.com, "Google faces 3 billion lawsuit over use of children's data," 2020.
[4] T. Xie, S. Wang, G.-H. Tu, C.-Y. Li, and X. Lei, "Exploring the insecurity of google account registration protocol via model checking," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2019.
[5] D.-M. Ordway, "Fake news and the spread of misinformation," https://journalistsresource.org/studies/society/internet/fake-news-conspiracy-theories-journalism-research/, 2018.
[6] K. Foundation, "Disinformation, 'fake news' and influence campaigns on twitter," https://www.knightfoundation.org/reports/disinformation-fake-news-and-influence-campaigns-on-twitter, 2018.
[7] J. Gingerich, "The cost of fake news: $78 billion," https://www.odwyerpr.com/story/public/13448/2019-11-26/cost-fake-news-78-billion.html, 2019.
[8] M. Graham, "Cnbc: Fake followers in influencer marketing will cost brands $1.3 billion this year, report says," 2019.
[9] L. Abrams, "The nasty list phishing scam is sweeping through instagram," 2019. [Online]. Available: https://www.bleepingcomputer.com/news/security/the-nasty-list-phishing-scam-is-sweeping-through-instagram/
[10] "5 most famous ddos attacks," https://www.a10networks.com/resources/articles/5-most-famous-ddos-attacks/, 2018.
[11] "Ddos attacks in q4 2018," https://securelist.com/ddos-attacks-in-q4-2018/89565/, 2019.
[12] S. Sheth, "Former president barack obama's twitter account appears to have been hacked as part of a cryptocurrency scam," https://www.businessinsider.com/barack-obama-twitter-account-hacked-in-cryptocurrency-scam-2020-7, 2020.
[13] F. T. Commission, "Consumer sentinel network data book 2020," https://www.ftc.gov/reports/consumer-sentinel-network-data-book-2020, 2021.
[14] T. K. John Buzzard, "2021 identity fraud study: Shifting angles," https://www.javelinstrategy.com/content/Javelin-2021-Identity-Fraud-Study, 2021.
[15] K. Golladay and K. Holtfreter, "The consequences of identity theft victimization: An examination of emotional and physical health outcomes," *Victims & Offenders*, vol. 12, no. 5, 2017.
[16] M. Cyrill, "New curfew rules, real-name registration for china's young online gamers," https://www.china-briefing.com/news/china-online-gaming-curfew-minors-real-name-registration-system/, 2019.
[17] B. Bulgurcu, H. Cavusoglu, and I. Benbasat, "Understanding emergence and outcomes of information privacy concerns: a case of facebook." in *ICIS*, 2010.
[18] J. Jia, B. Wang, and N. Z. Gong, "Random walk based fake account detection in online social networks," in *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2017.
[19] B. Erşahin, . Aktaş, D. Kılınç, and C. Akyol, "Twitter fake account detection," in *2017 International Conference on Computer Science and Engineering (UBMK)*, 2017.
[20] S. M. Bellovin, M. Leech, and T. Taylor, "Icmp traceback messages," 2003.
[21] S. Yu, W. Zhou, R. Doss, and W. Jia, "Traceback of ddos attacks using entropy variations," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 3, 2011.
[22] "Tor project," https://www.torproject.org/, 2019.
[23] Facebook, "Report an impostor account," https://www.facebook.com/help/contact/295309487309948, 2021.
[24] M. C. Jo Ling Kent, "Nbc news: Fake facebook profiles cause heartbreak for families and colleagues."
[25] "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," 2008, RFC 5280.
[26] W3Techs, "Usage statistics of default protocol https for websites," https://w3techs.com/technologies/details/ce-httpsdefault, 2020.
[27] "Globalsign authentication," https://www.globalsign.com/en/authentication/, 2021.
[28] "Digicert client certificate," https://www.digicert.com/client-certificates/, 2021.
[29] "Google account help," https://support.google.com/accounts, 2021.
[30] H. Li, Z. Xu, H. Zhu, D. Ma, S. Li, and K. Xing, "Demographics inference through wi-fi network traffic analysis," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*.
[31] T. J. Neal and D. L. Woodard, "A gender-specific behavioral analysis of mobile device usage data," in *2018 IEEE 4th International Conference on Identity, Security, and Behavior Analysis (ISBA)*.
[32] P. Wang, F. Sun, D. Wang, J. Tao, X. Guan, and A. Bifet, "Inferring demographics and social networks of mobile device users on campus from ap-trajectories," in *Proceedings of the 26th International Conference on World Wide Web Companion*.
[33] J. Aas, R. Barnes, B. Case, Z. Durumeric, P. Eckersley, A. Flores-López, J. A. Halderman, J. Hoffman-Andrews, J. Kasten, E. Rescorla *et al.*, "Let's encrypt: an automated certificate authority to encrypt the entire web," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019.
[34] FIDO, "Fido alliance," https://fidoalliance.org, 2018.
[35] GSMA, "Gsma mobile connect," https://mobileconnect.io/, 2020.
[36] A. Kahate, *Cryptography and network security*. Tata McGraw-Hill Education, 2013.
[37] D. Rachmawati, J. Tarigan, and A. Ginting, "A comparative study of message digest 5 (md5) and sha256 algorithm," in *Journal of Physics: Conference Series*, vol. 978, no. 1, 2018.
[38] "Receive sms online," https://freephonenum.com/, 2021.
[39] "Tempmail," https://temp-mail.org, 2021.
[40] "Gsma oneapi," https://github.com/GSMADeveloper/GSMA-OneAPI/wiki, 2019.
[41] F. C. Commission *et al.*, "Code of federal regulations: Title 47–telecommunications: Universal service," 2008.
[42] H. Li, Q. Chen, H. Zhu, D. Ma, H. Wen, and X. S. Shen, "Privacy leakage via de-anonymization and aggregation in heterogeneous social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 2.
[43] "Trusted digital signatures," https://www.globalsign.com/en/digital-signatures/, 2021.
[44] 3GPP, "TS33.401: 3GPP SAE; Security architecture," Sep. 2013.
[45] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on information theory*, vol. 29, no. 2, 1983.
[46] GSMA, "Mandatory registration of prepaid sim cards," https://www.gsma.com/publicpolicy/wp-content/uploads/2016/04/Mandatory-SIM-Registration.pdf, 2016.
[47] 3GPP, "TS23.002: Network architecture," Mar. 2017.
[48] ——, "TS29.329: Sh interface based on the Diameter protocol; Protocol details," Jul. 2020.
[49] P. Calhoun, "Diameter framework document," *Internet draft, draft-ietf-aaa-diameter-framework-01. tex*, 2001.
[50] H. Haverinen and J. Salowey, "Rfc 4186: Extensible authentication protocol method for global system for mobile communications (gsm) subscriber identity modules (eap-sim)," 2006.

[51] J. Arkko, V. Lehtovirta, Eronen *et al.*, "RFC 4187: Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)."

[52] 3GPP.

[53] NISF, "FIPS Publication 186-2: Digital Signature Standard (DSS)," January 2000.

[54] T. Dierks and E. Rescorla, "RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2."

[55] "Oauth 2.0," https://oauth.net/2/, 2021.

[56] U. Hengartner and P. Steenkiste, "Exploiting hierarchical identity-based encryption for access control to pervasive computing information," in *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*, 2005.

[57] F. P. Miller, A. F. Vandome, and J. McBrewster, *Levenshtein Distance: Information Theory, Computer Science, String (Computer Science), String Metric, Damerau?Levenshtein Distance, Spell Checker, Hamming Distance*. Alpha Press, 2009.

[58] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, 2004.

[59] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE transactions on Information Theory*, vol. 22, no. 6, 1976.

[60] "Digital signature," https://en.wikipedia.org/wiki/Digital_signature, 2019.

[61] NSA, "Eliminating obsolete transport layer security (tls) protocol configurations."

[62] S. Alt, P.-A. Fouque, G. Macario-rat, C. Onete, and B. Richard, "A cryptographic analysis of umts/lte aka," in *Applied Cryptography and Network Security*, 2016.

[63] D. Fett, R. Küsters, and G. Schmitz, "A comprehensive formal security analysis of oauth 2.0," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[64] W. Stallings and L. Brown, *Computer Security: Principles and Practice*, 3rd ed. USA: Prentice Hall Press, 2014.

[65] "Sim cards: attack of the clones," https://www.kaspersky.com/blog/sim-card-history-clone-wars/11091/, 2016.

[66] K. Nohl, "Rooting sim cards," *BlackHat Briefings*, 2013.

[67] "Sim swap attack (sim intercept attack)," https://whatis.techtarget.com/definition/SIM-swap-attack-SIM-intercept-attack, 2018.

[68] Y. J. Jia, Q. A. Chen, Y. Lin, C. Kong, and Z. M. Mao, "Open doors for bob and mallory: open port usage in android apps and security implications," in *Security and Privacy (EuroS&P), 2017 IEEE European Symposium on*, 2017.

[69] L. Armasu, "How google improved android security in 2017?" https://www.tomshardware.com/news/google-android-security-improvements-2017,36673.html, 2018.

[70] Y. Acar, M. Backes, S. Bugiel, S. Fahl, P. McDaniel, and M. Smith, "Sok: Lessons learned from android security research for appified software platforms," in *2016 IEEE Symposium on Security and Privacy (SP)*, 2016.

[71] OpenSSL, "OpenSSL: Cryptography and SSL/TLS Toolkit," 2021, https://www.openssl.org/.

[72] P. Wendland, "A java card pki applet aiming to be iso 7816 compliant," https://github.com/philipWendland/IsoApplet, 2015.

[73] OASIS, "Pkcs #11 cryptographic token interface base specification version 2.40," http://docs.oasis-open.org/pkcs11/pkcs11-base/v2.40/os/pkcs11-base-v2.40-os.html, 2015.

[74] node diameter, "node-diameter," 2021, https://github.com/node-diameter/node-diameter/.

[75] "Oauth 2.0 server," https://github.com/thephpleague/oauth2-server, 2021.

[76] ScribeJava, "faker.js," https://github.com/scribejava/scribejava, 2019.

[77] boost, "Boost c++ libraries," http://erikerlandson.github.io/algorithm/index.html, 2020.

[78] Y. H. Hwang, S. Kim, and J. W. Seo, "Fast order-preserving encryption from uniform distribution sampling," in *Proceedings of the 2015 ACM Workshop on Cloud Computing Security Workshop*.

[79] M. C. reseach group, "Cryptdb," http://css.csail.mit.edu/cryptdb/, 2013.

[80] "Voter registration records," https://raidforums.com/Announcement-Database-Index-CLICK-ME, 2019.

[81] "Geocoding api," https://developers.google.com/maps/documentation/geocoding/overview, 2021.

[82] B. Castle, "The legion of the bouncy castle," https://www.bouncycastle.org/java.html, 2020.

[83] A. Boldyreva, N. Chenette, Y. Lee, and A. O'neill, "Order-preserving symmetric encryption," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2009.

[84] H. M. Elanine Moore, "Facebook's massive fake numbers problem," https://www.latimes.com/business/technology/story/2019-11-18/facebooks-massive-fake-numbers-problem, 2019.

[85] B. VanderSloot, J. Amann, M. Bernhard, Z. Durumeric, M. Bailey, and J. A. Halderman, "Towards a complete view of the certificate ecosystem," in *Proceedings of the 2016 Internet Measurement Conference*.

[86] Y. Zhang, B. Liu, C. Lu, Z. Li, H. Duan, J. Li, and Z. Zhang, "Rusted anchors: A national client-side view of hidden root cas in the web pki ecosystem," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021.

[87] S. Y. Chau, O. Chowdhury, E. Hoque, H. Ge, A. Kate, C. Nita-Rotaru, and N. Li, "Symcerts: Practical symbolic execution for exposing noncompliance in x. 509 certificate validation implementations," in *Security and Privacy (SP), 2017 IEEE Symposium on*.

[88] X. Wang and M. El-Said, "Domainpki: Domain aware certificate management," in *Proceedings of the 21st Annual Conference on Information Technology Education*, 2020.

[89] M. Wang, C. Qian, X. Li, and S. Shi, "Collaborative validation of public-key certificates for iot by distributed caching," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, 2019.

[90] J. Höglund, S. Lindemer, M. Furuhed, and S. Raza, "Pki4iot: Towards public key infrastructure for the internet of things," *Computers & Security*, vol. 89, 2020.

[91] A. Rashid, A. Masood, H. Abbas, and Y. Zhang, "Blockchain-based public key infrastructure: A transparent digital certification mechanism for secure communication," *IEEE Network*, vol. 35, no. 5, 2021.

[92] A. Papageorgiou, A. Mygiakis, K. Loupos, and T. Krousarlis, "Dpki: a blockchain-based decentralized public key infrastructure system," in *2020 Global Internet of Things Summit (GIoTS)*, 2020.