Pers Ubiquit Comput (2012) 16:677–685
DOI 10.1007/s00779-011-0434-9

ORIGINAL ARTICLE

Secure and efficient public key management in next generation mobile networks

Kyusuk Han · Hyeran Mun · Taeshik Shon · Chan Yeob Yeun · James J. (Jong Hyuk) Park

Received: 10 January 2011 / Accepted: 30 May 2011 / Published online: 4 August 2011

Abstract Employing public key-based security architecture is inevitable for the advanced security applications in the mobile networks. However, key storage management problems have arisen, because the public key computation is still the large overhead to USIM, and the mobile equipment has potential threats of the key leakage or loss. In order to solve such shortcomings, we improve the key-insulated models and propose "Trust Delegation" model that the overall security computations are operated in ME, while the initial private key still remains in the secure storage in USIM. Our model is resilient against not only key exposure but also key loss. Finally, we show that the overall transactions can be reduced to one-third than current 3GPP Generic Authentication Architecture.

Keywords Mobile network · Security architecture · ID-based cryptosystem · Generic authentication architecture · 3GPP

K. Han
KAIST, Daejeon, South Korea

H. Mun
ETRI, Daejeon, South Korea

T. Shon
Ajou University, Suwon, South Korea

C. Y. Yeun
KUSTAR University, Sharjah, UAE

J. J. (Jong Hyuk) Park (✉)
SeoulTech, Seoul, South Korea
e-mail: jhpark1@seoultech.ac.kr

1 Introduction

Widely deployed security architectures of 2G/3G mobile networks are based on symmetric key-based security architecture [14, 15]. In this architecture, the master seed key is securely stored in the universal subscriber identity module (USIM) and generates the session keys such as the cipher key and the integrity key for mobile equipments (ME) to use in the secure communications and applications.

Recent developments of mobile communication technologies [13] request the deployment of the public key-based security architecture for more advanced applications. However, the large computational overhead of PKI brings the public key management issue. Storing public keys and security computation in low-cost USIM could be the bottleneck in advanced security services [6]. Since the public key-based security operations such as signature generations occur much larger computational overhead, the operations depend on the computational power of USIM. Currently, even more advanced USIM technologies with high capability are shown by telecommunication manufacturers [9], and additional replacement is required to use it and the user identity should be still stored in the USIM weaker than ME. Storing public key pairs in ME weaken the strength of key storage as the claim in [16], while storing keys in USIM has the computational overhead problem due to the security computations operated in USIM.

Recently, several studies such as [2] focused on deploying ID-based cryptosystem (IDBC) [3, 8] that does not require the public key management; in IP Multimedia System, they still have the potential threat of key leakage.

Therefore, our motivation is to overcome key management problems that are mentioned above. Although several researches such as "key-insulated" encryption [4] and signature scheme [5, 10] are proposed to be resilient against the key leakage, their designs did not consider mobile environments that the key losses occasionally happen. Since the communication is operated via wireless environments, they have wider applications than previous environment. Also, the mobile devices are always carried by users and can be lost. Moreover, their designs are related to the specific protocols and insufficient to support the various applications in mobile network.

Our contribution is to improve the “key-insulated” model and show “Trust Delegation” model resilient against not only the key exposure but also the key loss and to provide the secure and efficient public key management for the next generation mobile networks. Our trust delegation model is based on IDBC and achieves the great benefit regarding the efficiency of public key management. Compared with the current architecture [16], our model does not require the involvement of symmetric key base architecture [14] and has only one-third of transaction that helps the resilience against DoS attacks to mobile networks [17].

The last of our paper is organized as follows: Sect. 2 briefly describes the current mobile network security architecture. Section 3 argues the key management issues in mobile networks and introduces the trust delegation model. We propose our novel trusted delegated key management designs in Sect. 4. In Sect. 5, we show the analysis of our design. Finally, we conclude our paper in Sect. 6.

2 Security architecture of mobile network

2.1 Current security architecture

Basically, a mobile device consists of two main components: One is mobile equipment (ME) and the other is a universal subscriber identity module (USIM). ME is a device such as mobile phones, those attaches USIM that contains the unique identity of user such as phone number and the master seed key. Since USIM is considered as a tamper-resistant security hardware module and secret information such as user identity and secret key are stored in USIM. Therefore, 3GPP specifies the security architecture that the seed key is securely stored in USIM, and the session keys (the cipher key and the integrity key) are generated from the seed key using the UMTS-AKA algorithm and transferred to ME [15]. Then, ME uses the session keys for the secure communication.

3GPP also specifies the generic authentication architecture (GAA) to support the third-party network applications such as mobile banking and various multimedia services. Currently, widely employed GAA is the symmetric key-based generic bootstrapping architecture (GBA) [14] as in Fig. 1. The architecture consists of four essential entities such as the bootstrapping function (BSF), the network application function (NAF), home subscriber server (HSS), and ME. For the third-party service, ME can be communicated with NAF that can be used with any specific application protocol necessary. HSS has the initial key shared with USIM and sends the authentication information to BSF. BSFs are located in each domain and send the received authentication information to NAF. GBA employs the UMTS-AKA algorithm for mutual authentication between the mobile equipment and BSF in the network.

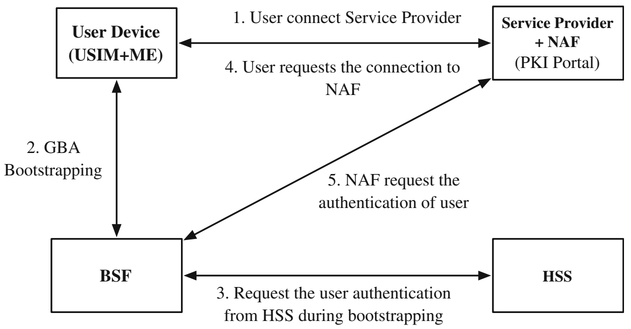
User Device (USIM+ME)
1. User connect Service Provider
4. User requests the connection to NAF
Service Provider + NAF (PKI Portal)
2. GBA Bootstrapping
5. NAF request the authentication of user
BSF
3. Request the user authentication from HSS during bootstrapping
HSS

Fig. 1 The process of generic authentication architecture. 3GPP TS 33.221 specifies that NAF has the role of the PKI portal

2.2 PKI support for advanced security service

The advance of mobile network brings the request for the deployment of PKI that enables the more various security applications such as the digital signature.

Thus, 3GPP also specifies the asymmetric key-based security architecture [16] to support the certificate service. In the architecture, a NAF acts as the PKI portal that issues the certificates of the ME as in Fig. 1. In order to establish the secure channel between the PKI portal and the ME, the BSF should have the shared secret key with NAF and ME. That means that the PKI support in [16] is only available along with GBA introduced in the previous section.

However, deploying the PKI occurs the key management problem. Storing the certificate and computing the security operations in the USIM will be significant overhead to USIM. Instead, storing the certificate and computing the security operations in the ME are better for the overall performance. Nevertheless, the storing the certificate in ME increases the potential threats of the leakage of the secret keys. Such problem is also argued in [16], and we discuss more detail in the following section.

3 Trust delegation

3.1 Public key management issues in mobile networks

One of the important issues on deploying public key-based cryptosystem is key management problem. Because only the key owner should know the private key, and the certificate are securely stored. There can be two cases on storing the certificate: One is storing the certificate in the USIM as in Fig. 2a and the other is storing the certificate in ME as in Fig. 2b.

Although storing the certificate in the USIM provides enhanced security, the USIM has weaker resources and the performance.

Commercially used typical hardware configuration of the USIM is about 5–40 MHz clock speed, a few kilobytes of memories. Also, the communication between USIM and the ME is based on ISO 7816-3 based I/O interface (T = 0, 1) provides half-duplex communication between USIM and ME with 9600 baud—115 kbps communication speed. Even the recent developments [9] are prepared to provide the full-duplex I/O interface, multi-application service, flash memory, and browser-based service, more time are need for such technologies are deployed, and mobile devices will be more powerful at that time. Recent Java card platform [12] is designed to support the better hardware that has a 32-bit processor, 128 kilobytes of EE-PROM, 512 kb of ROM, and 24 kb of RAM, while the performance of the recent smart phone is equivalent to entry-level mobile computer. In order to compare the performance between USIM and ME, we refer [11] that the computation time for encryption with RSA 1024 is 5–25 ms in USIM with 5–40 MHz clock speed and 1 ms in Intel Celeron 450 MHz. Nowadays, the performance of smart phones shows the better performance than Celeron 450 MHz. Thus, we can consider the ME significantly outperforms the USIM.

While storing the certificate in the ME could overcome such constraints, it increases the potential security threat of key leakage instead. Alternatively, we can use only short-lived certificates for enrolling subscriber. Even if new user may access the old user's private key, he/she should fail to masquerade as the old user in authorization transactions when the subscriber certificates expired. However, the use of the short-lived certificates requires the more frequent communication between PKI portal and the user for update the certificates. Also, the risk lives until the expiration of the certificates. Key pair generation should protect disclosure/cloning of private key, because the key pair generation is important for the secrecy of the private key.

3.2 Trust delegation

3.2.1 Trust delegation concept

There were the several ideas against the key exposure problem. In 2002, Dodis et al. [4] proposed the "key-insulated public key cryptosystem" for the encryption and "key-insulated signature" (KIS) scheme for the signature generation [5]. Later, Ohtake et al. showed more efficient KIS scheme and showed the application that a large-scale multi-receiver authentication system in which a signer communicates with a huge number of receivers [10]. In such KIS schemes, the "master" private key remains in the secure storage and the "temporary" private key generated from the master key is actually used for the security applications.

However, such KIS schemes have no consideration for the mobile network. In the mobile network, the losses of data including keys in ME occasionally happen, while the KIS schemes are focusing on resilience against the leakage of the master key. Since the old temporary key is required to generate the updated temporary key [5, 10], the loss of key in ME disables the key update. For example, TSK_3 and later keys cannot be generated in case TSK_2 is lost in Fig. 3a. Also, KIS schemes use N number of the temporary private keys, and each key is used during a constant time period t. After $t \times N$ times later, the large overhead for reconstructing the temporary key set is required [5].

Moreover, KIS schemes are deeply related to specific security protocol. For instance, Ohtake et al.'s KIS scheme [10] is based on Abe-Okamoto proxy signature scheme [1]. Thus, simultaneous deployment of both KIS scheme and the encryption scheme requests separated process to

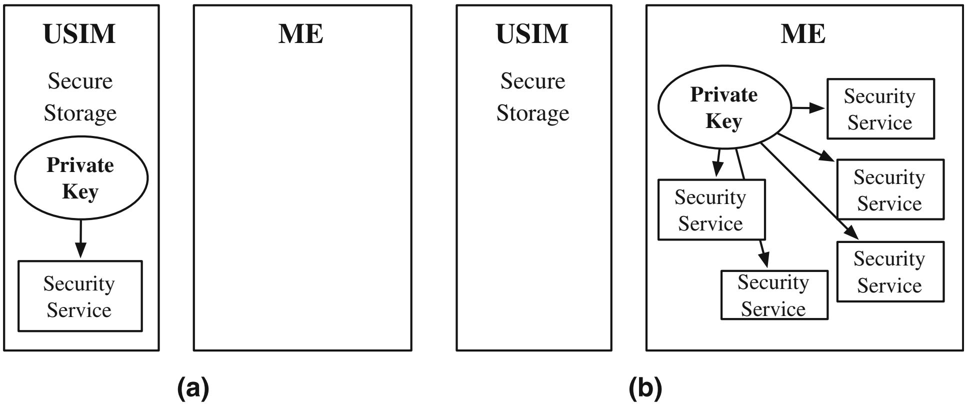

Fig. 2 **a** Storing public key in USIM gives security strength. **b** Storing public key in ME enables many applications

generate the temporary keys, which can be the potential security threat.

Instead KIS schemes, our design is to provide the "common" architecture that supports practical mobile networks. Since the design criterion is rather different to the KIS schemes, we introduce the alternative model of "Trust Delegation" (TD) employing the ID-based cryptosystem (IDBC) that the user's identity is used as the public key and private key as shown in initial setup phase of proposed model. Because the old temporary key is not required to update the new temporary key as in Fig. 3b, our TD model is not only resilient against the loss of key, but also provides simultaneous invocation of multiple distinct temporary private keys. Also, TD model enables the various security services are computed in the mobile device while the private key is securely stored in USIM. Figure 4 depicts the brief TD model.

3.2.2 *Brief overview of ID-based cryptosystem*

IDBC is based on properties of pairing [3] and the cryptographic problem [7] as following.

Elliptic curve discrete logarithm problem With given P, $P' \in G_1$, find an integer n that satisfies $P = n \cdot P'$ where an additive group G_1 over q, and P as the generator of G_1.

For the brief of IDBC, let us consider an additive group G_1 and a multiplicative group G_2 of the same order q. Assume that the discrete logarithm problem is hard in both groups. Let p be a generator of G_1 and $e{:}G_1 \times G_1 \rightarrow G_2$ a bilinear map satisfying the following properties:

- **Bilinearity** $\mathrm{e(aP,bQ)} = \mathrm{e(P,Q)}^{\mathrm{ab}}$ for all $P,Q \in G_1$ and all $a,b \in Z$.
- **Non-degeneracy** If $e(P,Q) = 1$ for all $Q \in G_1$, then $P = O$.
- **Computability** There exists an efficient algorithm to compute $e(P,Q)$ for and $P,Q \in G_1$.

Using the properties of IDBC, we can easily construct key exchange or signature protocol. Our TD model is based on the property of bilinearity, and the security of temporary private key is based on the computational hardness of elliptic curve discrete logarithm problem (ECDLP). The details of ID-based cryptography are explained in [8]. Due to the characteristic of IDBC that does not require the public key managements, there are studies such as [2] that deploy ID-based cryptosystem (IDBC) [3, 8] for IP Multimedia System.

4 Proposed design

In this section, we explain our proposed trust delegation model for the mobile networks. Section 4.1 shows the private key distribution in initial setup that users obtain their private key. Section 4.2 shows the session key setup between peer users. We propose the enhanced generic authentication architecture in Sect. 4.3. We define notations in Table 1 and the message format to request to USIM in Table 2.

4.1 Initial setup: private key distribution

Let the Key Generation Center (KGC) that is a trusted entity that distributes the private keys to users. KGC generates a random integer $s \in Z_p{}^*$, which will be the master secret of KGC. Each subscriber owns the unique identity ID. KGC distributes the private key $sk_{ID} = s \cdot H(ID)$ for each subscriber, where the public key knows the hash function $H{:}Z_p{}^* \rightarrow G_1$. The symbol "·" denotes the point multiplication over Elliptic curve.

The private key sk_{ID} is initially distributed in off-line environment. In practical application, users obtain sk_{ID} stored in USIM when they subscribe mobile services.

4.2 Session key establishment between peer entities

Assume two users A and B try to establish their secure communication. Each entity has a device, ME with USIM, U. Then, A initiates the session key establishment in ME_A and proceeds following steps:

P.1. ME_A generates a timestamp TS_A. And then, ME_A sends $REQ1$, ID of B, and TS_A to $U_{A.}$

P.2. U_A generates a random nonce r_A and generates e_A and sig_A and return them to ME_A, where $e_A = e_{pkB}(r_A)$ and $sig_A = sign_{skA}(e_A \| TS_A)$.

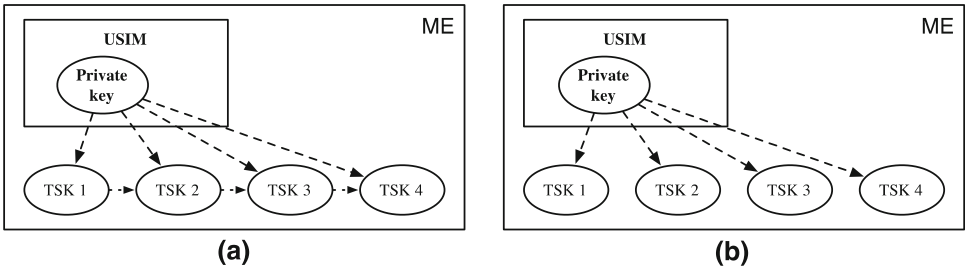

Fig. 3 **a** Temporary private keys are linked in KIS model. **b** Each temporary key has no link in "Trust Delegation" model

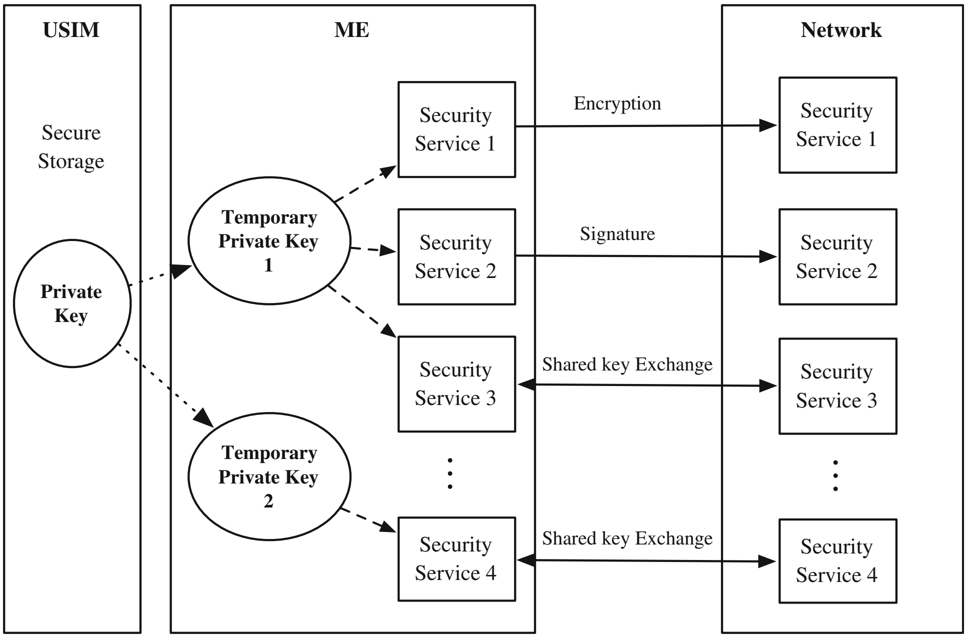

Fig. 4 Trust delegation model enables the various security applications such as encryption, signature generation, and the shared key exchange using multiple temporary private key

Table 1 Notations

Notation	Description
r_{ID}	Random nonce generated by user *ID*
TS	Timestamp
sk_{ID}	Private key of *ID*, $s \cdot H(ID)$
pk_{ID}	Public key of *ID*, $H(ID)$
tsk_{ID}	Temporal private key of *ID*
$sign_K(m)$	Sign a message *m* using private key *K*
$tsig_{ID}$	Signature of *ID* using *tsk*
sig_{ID}	Signature of *ID*
U_{ID}	USIM of an identity *ID*
ME_{ID}	Mobile equipment of *ID*
REQ	Trust delegated key request
RES	Trust delegated key response

Table 2 USIM request message type

Type	Input			Output	
REQ1	*r, TS*	–	–	*sig*	–
REQ2	*r, TS*	*sig*	r'	*tsk*	*sig*
REQ3	*r*	*sig*	–	*tsk*	–
REQ4	*r, TS*	*sig*	–	*tsk*	–

P.3. ME_A sends *REQ*, e_A, TS_A, and sig_A to ME_B.

P.4. ME_B sends *REQ2*, e_A, TS_A, and sig_A to U_B.

P.5. After verifying sig_A with the *A*'s public key pk_A generated by *A*'s ID, U_B decrypts e_A and obtain r_A. U_B then generates a random nonce r_B and compute e_B, sig_B and tsk_B, where $e_B = e_{pkA}(r_B)$, $sig_B = sign_{skB}(r_B \| TS_A)$, and $tsk_B = t \cdot sk_B$, respectively. We can compute $t = (r_A \oplus r_B)$, where $\oplus$ denotes the arbitrary operation of two inputs.

P.6. U_B returns e_B, sig_B, and tsk_B to ME_B.

P.7. ME_B sends *RES*, e_B, and sig_B to ME_A.

P.8. ME_A sends *REQ3*, e_B, and sig_B to U_A.

P.9. U_A verifies sig_B and decrypts e_B to obtain r_B. U_A then generates $tsk_A = t \cdot sk_A$. After that, U_A returns tsk_A to ME_A.

After that, ME_A stores tsk_A and ME_B stores tsk_B. With tsk_A and tsk_B, ME_A and ME_B can operate secure computation without revealing original sk_A and sk_B. Overall procedures are shown in Fig. 5.

After the authentication procedures are completed, ME_A generates the shared session key $K_A = e(tsk_A, pk_B)$, while ME_B generates $K_B = e(pk_A, tsk_B)$. The correctness of $K_A = K_B$ is as $K_A = e(tsk_A, pk_B) = e(t \cdot sH(A), H(B)) = e(H(A), H(B))^{t \cdot s} = e(H(A), t \cdot sH(B)) = e(pk_A, tsk_B) = K_B$.

For the general mobile communication networks, the subscribers request the communication to the mobile access point that is linked to the servers. For the practical application of trust delegation model, we apply our design to the 3GPP generic authentication architecture [14, 16] in following section.

4.3 Enhanced generic authentication architecture with trust delegation

In this section, we show the enhanced design of GAA that applies PKI. Scheme 3 consists of three phases: *temporary*

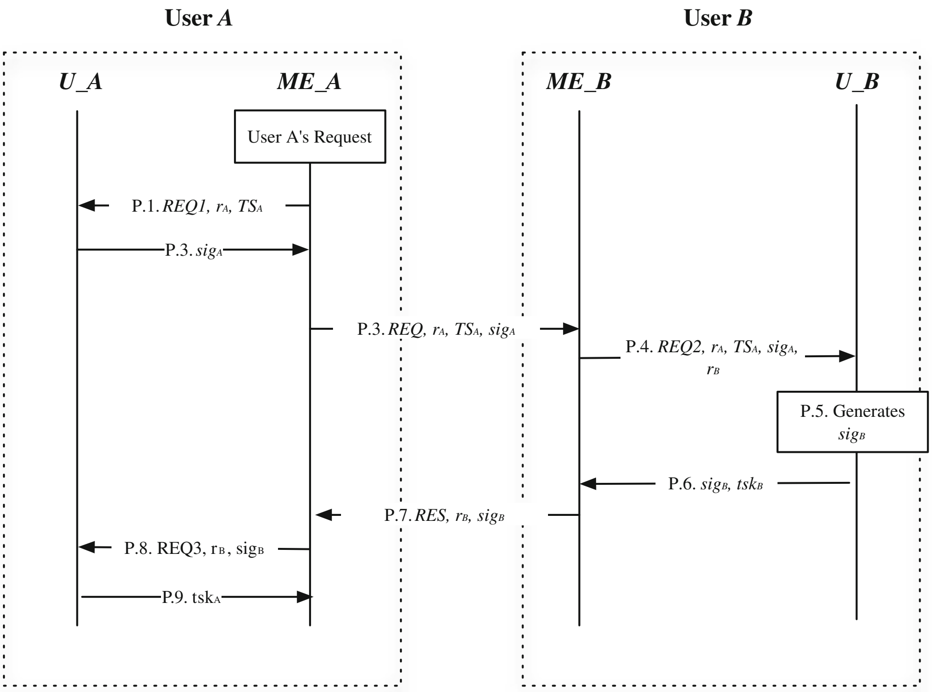

Fig. 5 Session key establishment between peer entities A and B

private key generation, *bootstrapping procedure*, and *service request to NAF.*

4.3.1 *Phase 1: temporary private key generation*

TD.1. ME_A sends ME_REQ, TS_A to U_A, where ME_REQ is the request of tsk_A and TS_A is the timestamp generated by ME_A.

TD.2. U_A generates r_A and computes e_A as in Sect. 4.2. U_A then returns e_A, tsk_A and sig_A to ME_A, where $tsk_A = r_A{\cdot}sk_A$ and $sig_A = sign_{skA}(e_A \| TS_A)$.

After the phase 1 is completed, ME_A stores tsk_A, e_A, TS_A, and sig_A.

4.3.2 *Phase 2: bootstrapping procedure*

If there is no shared information with NAF, ME_A has to contact BSF.

GB.1. ME_A sends ID of A, $tsig_A$, e_A, TS_A, and sig_A with bootstrapping request BSF_REQ to BSF, where $tsig_A = sign_{tskA}(BSF\_REQ)$.

GB.2. BSF generates $pk_A = H(A)$ for the verification of sig_A. After verifying sig_A, BSF can retrieve r_A by decrypting e_A and check the validity of tsk_A. If tsk_A is valid, BSF can verify $tsig_A$. After the successful verification, BSF stores r_A and TS_A with the ID of ME_A and sends the response BSF_RES with corresponding signature back.

4.3.3 *Phase 3: service request to NAF*

After Phase 2, ME_A requests the service to NAF, and then NAF authenticates ME_A as following procedures.

NF.1. ME_A sends NAF_REQ, $APPL\_ID$, and $tsig_A$' to the NAF, where NAF_REQ is the request of application service and $APPL\_ID$ is the application ID. $tsig_A$' is the signature where $tsig_A' = sign_{tskA}\ (NAF\_REQ \| APPL\_ID)$.

NF.2. If NAF has already authorized tsk_A, NAF instantly verifies $tsig_A$'. In other case, NAF requests BSF the authentication information of A. We assume that NAF and BSF have the secure channel.

NF.3. BSF returns r_A and TS_A those are used for NAF to verify $tsig_A$'. NAF stores r_A until TS_A is expired.

NF.4. NAF generates pk_A and verifies $tsig_A$'. When $tsig_A$' is valid, NAF authenticates ME_A and provides its service to ME_A.

Overall procedures are shown in Fig. 6.

4.4 Simplified enhanced generic authentication architecture with trust delegation

Since IDBC does not request the public key management, we can also reduce the BSF involvement for the authentication procedures. Thus, we can simplify the step **S-NF.1** and **S-NF.2** as follows. The overall simplified procedures are shown in Fig. 7.

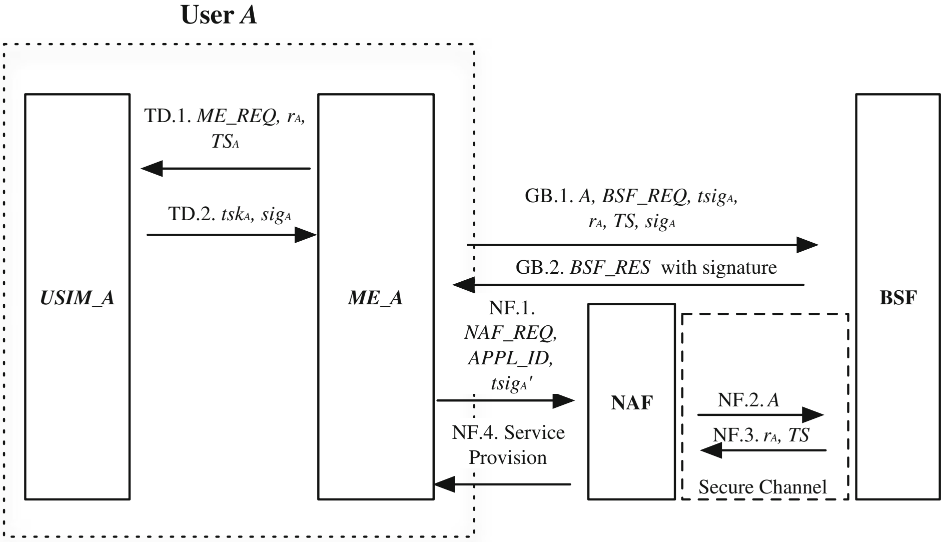

Fig. 6 Public key-based GAA with trust delegation. BSF involved for the compatibility

S-NF.1. ME sends *NAF_REQ*, ID of *A*, e_A, TS_A, sig_A, *APPL_ID*, and $tsig_A$ to NAF for requesting the service *APPL_ID*.

S-NF.3. After generating pk_A, NAF verifies sig_A and compute $tsig_A$ by decrypting e_A in sequence. When $tsig_A$ is valid, NAF authenticates ME_A and provides its service to ME_A.

5 Design analysis

In this section, we briefly analyze our proposed model and compare with 3GPP generic authentication architecture. We show security analysis first and performance analysis in sequence.

5.1 Security analysis

For the analysis of our design, we define the attack scenarios as follows:

- Impersonation by malicious adversaries: An adversary $A*$ tries to impersonate a innocent subscriber *A*. As a result of this attack, $A*$ communicate with mobile service provider pretending *A*.

In order impersonate *A*, $A*$ may resend *REQ*, e_A, TS_A, and sig_A (**P.3**) or *RES*, e_B, and sig_B (**P.7**). However, $A*$ should be able to manipulate the fake sig_A and sig_B without knowing sk_A or sk_B.

- Leakage of private key by a compromised ME: An adversary $A*$ obtain a ME and exposure the original private key from USIM.

In case of A* obtains compromised ME, since the *tsk*, e_A, and *H(A)* are known to ME_A, the compromised ME_A tries to compute *sH(A)* with e_A, *H(A)*, and $r_A \cdot sH(A)$. However, compromised ME_A fails to retrieve the original private key without any information of r_A. Recall elliptic curve discrete logarithm problem that is a well-known computational hard problem, we know such trial has the same success probability of solving DLP. (Breaking DLP is computationally infeasible.) Even though compromised ME_A sends tsk_A to other adversary, the adversary fails to impersonate after *TS* is expired. Thus, even ME is compromised, *sk* is still secure in USIM and the effect on *tsk* in ME is limited. Since we already assume that the private key in the USIM is stored in secure, the security of the USIM is considered as the security of the security storage of the USIM and the out of focus in this thesis.

- And weaken strength from temporary private key: An adversary A* obtain the original private key from the several temporary private keys or session keys.

Comparing the session keys derived from the initial private key *sk* and the temporary private key *tsk*, a session key using *sk* is computed as $e(sk_A, pk_B)$, while a session key using *tsk* is computed as $e(t{\cdot}sk_A, pk_B)$. It is trivial that the both have the same security strength. Finally, if *tsk* is erased in ME, new *tsk* can be simply generated by choosing new random nonce r'_A. Thus, our design is resilient to not only key exposure problem, but also key erase problem.

5.2 Performance analysis

5.2.1 Computational overhead

Our design reduces the overall computational overhead in USIM comparing the case that the private key is stored in USIM (Fig. 2a) that all public key-based security

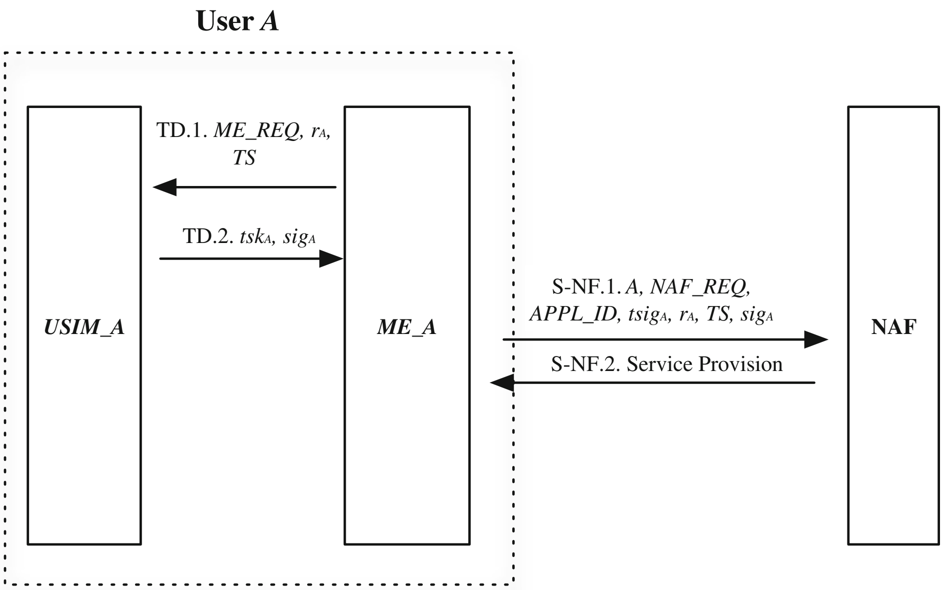
User A
USIM_A
ME_A
NAF
TD.1. ME_REQ, rA, TS
TD.2. tskA, sigA
S-NF.1. A, NAF_REQ, APPL_ID, tsigA, rA, TS, sigA
S-NF.2. Service Provision

Fig. 7 NAF directly authenticates the mobile device for request of services without BSF

computations are operated in USIM. In the proposed model, the computations in the USIM are one hash computation to generate the public key, one point multiplication to generate the temporary private key, and the signature generation of random nonce for the temporary private key. Because USIM is the only trusted entity, signature generation and verification in USIM are inevitable. The computational overhead of hash function generation is negligible.

Finally, our design does not require public key-based security computation after the initial signature generation and verification, while the private key still remains in the secure storage. We do not count the computational overheads in ME with the large computational power.

5.2.2 Transaction overhead

Our model shows about a half number of transactions than current 3GPP security architecture [14, 16], because our design is fully based on asymmetric key cryptosystem. Applying the IDBC, our design reduces the number of transaction to 4 rounds when we let NAF authenticate ME for itself. The design supporting PKI [16] still requires the support of GBA [14] for the certificate management that requires 13 rounds of transaction, while our design does not have the overhead for PKI certificate management that eventually follows the use of the GBA. Thus, our model is resilient to the DoS attack that makes HSS or BSF unavailable [17]. Table 3 shows the comparisons.

Table 3 Comparision with previous model

	Previous model [16]	Proposed model
Transactions	13 rounds	4 rounds
Computation in USIM	For every communication	Only in initial communication

6 Conclusion

In this paper, we described public key management issues in the mobile networks and proposed "Trust Delegation" concept based on IDBC that enables multiple security applications simultaneously and is resilient against not only the key exposure but also the key loss. Reducing the number of transactions as well as involved entities such as HSS and BSF, our design is resilient to the DoS attack targeting HSS or BSF. In conclusion, our novel design is applicable to the next generation mobile networks that the public key-based security architecture is inevitably deployed.

Acknowledgments This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (KRF-2008-331-D00580).

References

1. Abe T, Okamoto M (2002) Delegation chains secure up to constant length. IEICE Trans. Fundamentals E85-A(1):110–116
2. Abid M, Song S, Moustafa H, Afifi H (2009) Integrating identity-based cryptography in IMS service authentication. Int J Netw Secur Appl (IJNSA) 1(3)
3. Boneh D, Franklin MK (2001) Identity-based encryption from the Weil Pairing advances in cryptology. Proceedings of CRYPTO 2001

4. Dodis Y, Katz J, Xu S, Yung M (2002) Key-insulated public key cryptosystems. In: EUROCRYPT ‘02 proceedings of the international conference on the theory and applications of cryptographic techniques: advances in cryptology
5. Dodis Y, Katz J, Xu S, Yung M (2003) Strong key-insulated signature schemes. Proceedings of PKC’03
6. Handschuh H, Paillier P (2000) Smart card crypto-coprocessors for public-key cryptography, CARDIS ‘98. In: Proceedings of the international conference on smart card research and applications, Springer, London, UK, pp 372–379
7. Koblitz N (1987) Elliptic curve cryptosystems. In: Mathematics of Computation 48, p 203–209
8. Martin L (2008) Introduction to identity-based encryption. Number ISBN-13: 978-1-59693-238-8. Artech House, Inc., 685 Canton Street, Norwood, MA 02062
9. Na JC (2008) Next generation USIM technologies. TTA Journal (written in Korean) 116:80–85
10. Ohtake G, Hanaoka G, Ogawa K (2008) An efficient strong key-insulated signature scheme and its application. 5th European PKI Workshop, NTNU, Trondheim, Norway, June 16–17
11. RSA Laboratories (2000) RSAES-OAEP Encryption Scheme—Algorithm specification and supporting documentation
12. Sun Microsystems, Inc. (2009) Runtime Environment Specification, java card platform, version 3.0.1 connected edition
13. Third Generation Partnership (3GPP) (2011) TS 33.401 v 11.0.1 3GPP System Architecture Evolution (SAE); Security Architecture (Release 11)
14. Third Generation Partnership (3GPP) (2010) TS 33.220 v10.0.0 Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (Release 10)
15. Third Generation Partnership (3GPP) (2010) TS 33.102 v10.0.0 3G Security: security architecture (Release 10)
16. Third Generation Partnership (3GPP) (2010) TS 33.221 v10.0.0 Generic Authentication Architecture (GAA); Support for Subscriber Certificates (Release 10)
17. Traynor P, Lin M, Ongtang M, Rao V, Jaeger T, McDaniel P, La Porta V (2009) On cellular botnets: measuring the impact of malicious devices on a cellular network core. In CCS’09: Proceedings of the 16th ACM conference on Computer and communications security, p 223–234, New York, NY, USA, ACM