5G-AKAの考察

辛　星漢[1,a)]

概要：5G-AKA (Authentication and Key Agreement) プロトコルは 5G セキュリティにおいて最も重要な暗号プリミティブです。本稿では、5G-AKA に前方秘匿性を提供する方法を検討します。

キーワード：5G セキュリティ、5G-AKA、前方秘匿性

A Consideration on 5G-AKA

SeongHan Shin[1,a)]

Abstract: The 5G-AKA (Authentication and Key Agreement) protocol is a core cryptographic primitive for 5G security. In this paper, we consider how to provide 5G-AKA with forward secrecy.

Keywords: 5G security, 5G-AKA, forward secrecy

1. Introduction

Currently, the fifth generation (5G) mobile networks and telecommunication standard has been developed to meet the needs of enhanced mobile broadband, massive machine-type communications, and ultra-reliable and low-latency communications. Among several building blocks in this standard, the Authentication and Key Agreement protocol for 5G (5G-AKA) [4], developed by the 3GPP consortium, is of utmost importance which allows a User Equipment (UE) and a Home Network (HN) to authenticate each other and establish key materials (i.e., anchor keys) for protecting subsequent 5G communications. The 5G-AKA protocol is a new version of the AKA variants used for 3G and 4G networks. A distinctive feature of the 5G-AKA protocol is that a SUPI (Subscriber Permanent Identifier) of UE is sent to HN in the form of encrypted by using the ECIES (Elliptic Curve Integrated Encryption Scheme) KEM (Key Encapsulation Mechanism) with HN's public key.

In [15], D. Basin et al. provided a comprehensive formal model of the 5G-AKA protocol, and evaluated the model with respect to the 5G security goals using the security protocol verification tool Tamarin [16]. Then, they found that some critical security goals for the 5G-AKE protocol are not met. In [17], R. Borgaonkar et al. showed a new privacy attack on subscriber privacy against all the AKA variants (including 5G-AKA) by exploiting a logical vulnerability in the protection mechanism of SQN (Sequence Number). Also, A. Koutsos [18] showed that all the known privacy attacks (except the IMSI-catcher attack) are possible in the 5G-AKA protocol, and then proposed a modified 5G-AKA protocol that satisfies the unlinkability property and is proven in the Bana-Comon logic model [19]. After describing that the 5G-AKA protocol is still vulnerable to a series of active linkability attacks, Y. Wang et al. [5] proposed a privacy-preserving 5G-AKA (called, 5G-AKA') that is secure against active linkability attacks by encrypting a random challenge of HN with an ECIES-KEM key, and is compatible with the SIM cards and Serving Networks (SNs). By using Tamarin [16], they also proved that the 5G-AKA' protocol achieves privacy, authentication, and secrecy. Very recently, IETF EMU WG [3] has been working on an EAP-AKA' (EAP-AKA' FS)

[1] 産業技術総合研究所サイバーフィジカルセキュリティ研究センター
Cyber Physical Security Research Center (CPSEC), National Institute of Advanced Industrial Science and Technology (AIST)
[a)] seonghan.shin@aist.go.jp

protocol that provides forward secrecy. The EAP-AKA' FS protocol is a simple combination of the EAP-AKA' [19] and Diffie-Hellman key exchange [6].

1.1 Motivation and Our Contributions

Affected by Snowden's disclosures of pervasive surveillance, TLS 1.3 [20] is designed to provide forward secrecy where a client and a server first execute the Diffie-Hellman key exchange protocol [6], and all the subsequent messages are encrypted with a Diffie-Hellman key. This prevents an attacker, who obtained long-term secrets, from decrypting any past communications. However, most of the 5G-AKA and relevant AKA protocols do not guarantee forward secrecy.

In this paper, we propose a secure AKA (for short, $\mathsf{AKA}^\star$) protocol that provides UE anonymity and forward secrecy for 5G and beyond networks where the first message is for sending a SUCI (Subscriber Concealed Identifier), and the second and third exchanged messages are a challenge/response type of authentication. The main idea of the $\mathsf{AKA}^\star$ protocol is 1) to send a randomized identifier for UE anonymity and 2) to employ the DCR signature scheme in Section 2.3.2 for a challenge/response type of authentication and forward secrecy. Also, we compare the $\mathsf{AKA}^\star$ and relevant protocols (EAP-AKA [1], EAP-AKA' [2], EAP-AKA' FS [3], 5G-AKA [4], and 5G-AKA' [5]) in terms of efficiency, forward secrecy, UE anonymity, and UE unlinkability.

2. Preliminaries

2.1 Notation

Let $k \in \mathbb{N}$ be the security parameter. Let $\{0,1\}^*$ be the set of finite binary strings and $\{0,1\}^k$ be the set of binary strings of length k. Let $A\|B$ be the concatenation of A and B. If U is a set, then $u \xleftarrow{\$} U$ indicates the process of selecting u at random and uniformly over U. If U is a function (whatever it is), then $u = U$ indicates the process of assigning the result to u. Let U and N be the identities of User Equipment (UE) and Home Network (HN), respectively.

Also, let $\lambda \in \mathbb{N}$ be the security parameter. Let $\mathcal{G}$ be the group generation algorithm that takes as input 1^λ and outputs a group description $(\mathbb{G}, q, g)$ where $\mathbb{G}$ is a finite cyclic group of prime order q with g as a generator and its operation is denoted multiplicatively. In the aftermath, all the subsequent arithmetic operations are performed in modulo p unless otherwise stated where p is a prime, s (≥ 2) is a positive integer and $p = sq+1$. Also, $\mathsf{H}_1, \mathsf{H}_2 : \{0,1\}^* \to \mathbb{Z}_q^\star$ and $\mathsf{H}_3, \mathsf{H}_4, \mathsf{H}_5 : \{0,1\}^* \to \{0,1\}^k$ are descriptions of cryptographic hash functions (e.g., SHA-3).

2.2 Computational Assumption

Here, we define the Computational Diffie-Hellman (CDH) problem.

Definition1 (CDH Problem) Let $\mathcal{G}$ be the group generation algorithm described above. A (t,ε)-$\mathsf{CDH}_{\mathbb{G}}$ adversary is a probabilistic polynomial time (PPT) machine $\mathcal{B}$, running in time t, such that its success probability $\mathsf{Succ}^{\mathsf{cdh}}_{\mathbb{G}}(\mathcal{B})$, given random elements g^α and g^β to output $g^{\alpha\beta}$, is greater than ε. We denote by $\mathsf{Succ}^{\mathsf{cdh}}_{\mathbb{G}}(t)$ the maximal success probability over every adversaries, running within time t. The CDH problem states that $\mathsf{Succ}^{\mathsf{cdh}}_{\mathbb{G}}(t) \leq \varepsilon$ for any t/ε not too large.

2.3 Exponential Challenge-Response Signature Schemes

In this subsection, we describe the Exponential Challenge-Response (XCR) and Dual XCR (DCR) signature schemes [7] where a valid signature is not only message-specific but also challenge-specific.

2.3.1 XCR Signature Scheme

A signer $\hat{A}$ has a private key $a \xleftarrow{\$} \mathbb{Z}_q^\star$ and a public key $A \equiv g^a$. A verifier (or challenger) $\hat{B}$ provides a message m together with a challenge Y where $\hat{B}$ chooses a random element $y \xleftarrow{\$} \mathbb{Z}_q^\star$ and computes $Y \equiv g^y$. A signature of $\hat{A}$ on message m using challenge Y is defined as a pair $(X, Y^{x+\mathsf{H}_1(X,m)\cdot a})$ where $\hat{A}$ chooses a random element $x \xleftarrow{\$} \mathbb{Z}_q^\star$ and computes $X \equiv g^x$. The verifier $\hat{B}$ accepts a signature pair (X,σ) as valid (for message m and with respect to challenge $Y \equiv g^y$) if and only if both $X \neq 0$ and $(X \cdot A^{\mathsf{H}_1(X,m)})^y = \sigma$ hold. Hereafter, we denote by $XSIG_{\hat{A}}(X,m,Y) \stackrel{\mathsf{def}}{=} Y^{x+\mathsf{H}_1(X,m)\cdot a}$ the second element of an XCR signature pair. In [7], the XCR signature scheme is proven to be EUF-CMA secure (see below) in the random oracle model [8] under the CDH problem of Definition 1.

Definition2 (Security of XCR) An XCR signature scheme XCR is said to be secure against existential forgery on adaptively chosen message attacks (EUF-CMA secure) if, for any probabilistic polynomial time adversary $\mathcal{F}$, there exists a negligible function $\varepsilon(\cdot)$ in the security parameter λ such that $\Pr[\mathsf{Exp}^{\mathsf{euf\text{-}cma}}_{XCR}(\mathcal{F}) = 1] \leq \varepsilon(\cdot)$ in the experiment $\mathsf{Exp}^{\mathsf{euf\text{-}cma}}_{XCR}(\mathcal{F})$ defined as below:

(1) $\mathcal{G}(1^\lambda)$ outputs $(\mathbb{G}, q, g)$.

(2) Adversary $\mathcal{F}$ is given two random values A, Y_0 where $A, Y_0 \in \mathbb{G}$.

(3) During this experiment, $\mathcal{F}$ has access to a signing oracle $\mathcal{O}_{\mathsf{Sign}}$ (representing a signer $\hat{A}$ with the private key a and public key A) which takes as input a challenge Y and a message m, and returns a signature pair $(X, XSIG_{\hat{A}}(X,m,Y))$ where $\mathcal{O}_{\mathsf{Sign}}$ chooses a random element $x \xleftarrow{\$} \mathbb{Z}_q^\star$ and computes $X \equiv g^x$ afresh with each query. $\mathcal{F}$ is allowed a polynomial number of queries to $\mathcal{O}_{\mathsf{Sign}}$

where the queries (Y,m) are chosen adaptively by $\mathcal{F}$.

(4) Adversary $\mathcal{F}$ outputs a triple (X_0,m_0,σ).

The output of the experiment is defined to be 1 if the following two conditions hold: (a) The pair (X_0,σ) is a valid XCR signature of $\hat{A}$ on message m_0 with respect to challenge Y_0 (i.e., $X_0 \neq 0$ and $\sigma = XSIG_{\hat{A}}(X_0,m_0,Y_0)$); and (b) The pair (X_0,m_0) did not appear in any of the responses of O_{Sign} to $\mathcal{F}$'s queries. Otherwise, the output of the experiment is 0. We denote by $\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{XCR}(\mathcal{F}) = \Pr[\mathsf{Exp}^{\mathsf{euf\text{-}cma}}_{XCR}(\mathcal{F}) = 1]$ the adversary's advantage in attacking the XCR signature scheme XCR.

2.3.2 DCR Signature Scheme

In a DCR signature scheme, any two parties $\hat{A}$ and $\hat{B}$ can interact with each other with the dual role of challenger and signer, and each produces a signature that no third party can forge. A party $\hat{A}$ (resp., $\hat{B}$) has a private key $a \xleftarrow{\$} \mathbb{Z}_q^\star$ (resp., $b \xleftarrow{\$} \mathbb{Z}_q^\star$) and a public key $A \equiv g^a$ (resp., $B \equiv g^b$). Let m_1, m_2 be two messages. A DCR signature of $\hat{A}$ and $\hat{B}$ on messages m_1, m_2 is defined as a triple of values: X, Y and $DSIG_{\hat{A},\hat{B}}(m_1,m_2,X,Y) \stackrel{\mathsf{def}}{=} g^{(x+d\cdot a)(y+e\cdot b)}$ where $X \equiv g^x$ and $Y \equiv g^y$ are challenges chosen by $\hat{A}$ and $\hat{B}$, respectively, and $d = \mathsf{H}_1(X,m_1)$ and $e = \mathsf{H}_2(Y,m_2)$. A fundamental property of the DCR signature is that, after exchanging the values X and Y (with x and y randomly chosen by $\hat{A}$ and $\hat{B}$, respectively), both $\hat{A}$ and $\hat{B}$ can compute and verify the same signature $DSIG_{\hat{A},\hat{B}}(m_1,m_2,X,Y)$ as follows:

$$\begin{aligned} DSIG_{\hat{A},\hat{B}}(m_1,m_2,X,Y) &= g^{(x+d\cdot a)(y+e\cdot b)} = (Y\cdot B^e)^{x+d\cdot a} \\ &= (X\cdot A^d)^{y+e\cdot b}. \end{aligned} \quad (1)$$

Intuitively, a DCR signature is an XCR signature of $\hat{A}$ on message m_1 under challenge $Y\cdot B^e$ and, at the same time, an XCR signature of $\hat{B}$ on message m_2 under challenge $X\cdot A^d$. In [7], the DCR signature scheme (i.e., the DCR signature of $\hat{A}$ with respect to B) is proven to be EUF-CMA secure in the random oracle model [8] under the CDH problem of Definition 1.[*1] For the security of DCR, Definition 2 should be modified with the followings: (1) In step 3, the queries to O_{Sign} are of the form (Y,m,m_2) and the signature by O_{Sign} is the pair $(X, XSIG_{\hat{A}}(X,m,Y\cdot B^e))$ where $e = \mathsf{H}_2(Y,m_2)$; and (2) In step 4, $\mathcal{F}$ outputs a quadruple (X_0,m_0,m_2,σ) where m_2 is an arbitrary message chosen by $\mathcal{F}$. Accordingly, the output of the experiment is defined to be 1 if (a) $X_0 \neq 0$ and $\sigma = XSIG_{\hat{A}}(X_0,m_0,Y_0\cdot B^e)$; and (b) The pair (X_0,m_0) did not appear in any of the responses of O_{Sign} to $\mathcal{F}$'s queries. We denote by $\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{DCR}(\mathcal{F}) = \Pr[\mathsf{Exp}^{\mathsf{euf\text{-}cma}}_{DCR}(\mathcal{F}) = 1]$ the adversary's advantage in attacking the DCR signature scheme *DCR*.

[*1] Actually, the proof in [7] shows that it is EUF-CMA secure even if adversary $\mathcal{F}$ is given the private key b of $\hat{B}$ (but not the private key of $\hat{A}$). For more details, please refer to [7].

3. A Secure AKA Protocol for 5G and Beyond Networks

In this section, we propose a secure AKA (for short, AKA*) protocol that provides UE anonymity and forward secrecy for 5G and beyond networks where the first message from UE to HN is for sending a SUCI (Subscriber Concealed Identifier) or GUTI (Globally Unique Temporary Identity) of UE, and the second and third exchanged messages are a challenge/response type of authentication between UE and HN. The main idea of the AKA* protocol is 1) to send a randomized identifier (to be computed with a DCR signature) for UE anonymity and 2) to employ the DCR signature scheme in Section 2.3.2 for a challenge/response type of authentication and forward secrecy. In the AKA* protocol, we do not assume PKI (Public Key Infrastructure) meaning that raw public keys of UE and HN do not need to be checked (e.g., via CRL (Certificate Revocation List) or OCSP (Online Certificate Status Protocol)). The AKA* protocol consists of **Initialization** and **Authentication and Key Agreement** phases.

3.1 Initialization

First, UE randomly chooses his/her identifier U from $\{0,1\}^k$. Also, UE chooses a private key $a \xleftarrow{\$} \mathbb{Z}_q^\star$ and computes a public key $A \equiv g^a$. Then, UE sends (U,A) to HN along with SUPI (Subscriber Permanent Identifier). After receiving $(\mathsf{U},A,\mathrm{SUPI})$ from UE, HN chooses its private key $b \xleftarrow{\$} \mathbb{Z}_q^\star$ and computes a public key $B \equiv g^b$, and then sends (N,B) to UE where N is HN's identifier. Finally, UE stores $(\mathrm{SUPI},\mathsf{U},(a,A\equiv g^a),\mathsf{N},B)$ secretly and HN holds $(\mathsf{N},(b,B\equiv g^b),\mathsf{U},A,\mathrm{SUPI})$ secretly. Note that A and B are raw public keys of UE and HN, respectively. This initialization phase should be done once and securely between UE and HN.

3.2 Authentication and Key Agreement

In this phase, UE and HN execute the AKA* protocol, whenever needed, over insecure networks in order to share an authenticated session key to be used for protecting subsequent communications. This phase of the AKA* protocol has four steps as below.

Step 1. The UE chooses a random element $x \xleftarrow{\$} \mathbb{Z}_q^\star$ and computes a Diffie-Hellman public value $X \equiv g^x$. Then, UE sends his/her randomly-chosen identifier U to HN.

Step 2. The HN chooses a random element $y \xleftarrow{\$} \mathbb{Z}_q^\star$ and computes a Diffie-Hellman public value $Y \equiv g^y$. After receiving a message U from UE, HN sends back its identifier N and Diffie-Hellman public value Y to UE.

Step 3. After receiving a message (N,Y) from HN, UE com-

putes $d = \mathsf{H}_1(X, \mathsf{N})$ and $e = \mathsf{H}_2(Y, \mathsf{U})$. Using his/her private key a, UE computes a DCR signature $K \equiv (Y \cdot B^e)^{x+d \cdot a}$ (i.e., an XCR signature $XSIG_{\mathrm{UE}}(X, \mathsf{N}, Y \cdot B^e)$ on message N under challenge $Y \cdot B^e$). With a session identifier $sid = \mathsf{U}||\mathsf{N}||Y||X$, UE computes his/her authenticator $V_{\mathsf{U}} = \mathsf{H}_3(sid||A||B||K)$, and then sends (X, V_{U}) to HN. Also, UE computes a session key $SK_{\mathsf{U}} = \mathsf{H}_4(sid||A||B||K)$ and updates his/her identifier as follows: $\mathsf{U} = \mathsf{H}_5(sid||A||B||K)$.

Step 4. After receiving a message (X, V_{U}) from UE, HN computes $d = \mathsf{H}_1(X, \mathsf{N})$ and $e = \mathsf{H}_2(Y, \mathsf{U})$. Using its private key b, HN computes a DCR signature $K \equiv (X \cdot A^d)^{y+e \cdot b}$ (i.e., an XCR signature $XSIG_{\mathrm{HN}}(Y, \mathsf{U}, X \cdot A^d)$ on message U under challenge $X \cdot A^d$). Then, HN checks whether the authenticator V_{U} is valid or not. If $V_{\mathsf{U}} \neq \mathsf{H}_3(sid||A||B||K)$ where a session identifier $sid = \mathsf{U}||\mathsf{N}||Y||X$, HN aborts the protocol. Otherwise, HN computes a session key $SK_{\mathsf{N}} = \mathsf{H}_4(sid||A||B||K)$ and updates UE's identifier as follows: $\mathsf{U} = \mathsf{H}_5(sid||A||B||K)$.

4. Security Model

Here, we extend the security model [9], [10] to be suitable for our setting, in which an adversary $\mathcal{A}$ is additionally allowed to invoke a RevealIRPK-query to obtain raw public keys of UE and HN, and define the semantic security of session keys.

Let $\mathbf{U}$ and $\mathbf{N}$ be sets of UE and HN, respectively. We denote by $\mathsf{U} \in \mathbf{U}$ and $\mathsf{N} \in \mathbf{N}$ two parties that participate in an authenticated key exchange protocol P. Each of them may have several instances called oracles involved in distinct, possibly concurrent, executions of P. We denote U (resp., N) instances by U^ζ (resp., N^η) where $\zeta, \eta \in \mathbb{N}$, or by I in the case of any instance. During the protocol execution, an adversary has the entire control of networks and has access to the raw public keys. Let us show the capability of adversary $\mathcal{A}$ each query captures:

- $\mathsf{Execute}(\mathsf{U}^\zeta, \mathsf{N}^\eta)$: This query models passive attacks, where the adversary gets access to honest executions of P between the instances U^ζ and N^η by eavesdropping.
- $\mathsf{Send}(I, msg)$: This query models active attacks by having $\mathcal{A}$ send a message to instance I. The adversary $\mathcal{A}$ gets back the response I generates in processing the message msg according to the protocol P. A query $\mathsf{Send}(\mathsf{U}^\zeta, \mathtt{Start})$ initializes the protocol, and thus the adversary receives the first flow message.
- $\mathsf{Reveal}(I)$: This query handles misuse of the session key (e.g., use in a weak symmetric-key encryption) by any instance I. The query is only available to $\mathcal{A}$, if the instance actually holds a session key, and the latter is released to $\mathcal{A}$.
- $\mathsf{RevealIRPK}(\mathsf{U}/\mathsf{N})$: This query allows the adversary to obtain the raw public keys of UE and HN.
- $\mathsf{Test}(I)$: This oracle is used to see whether or not the adversary can obtain some information on the session key by giving a hint on the key. The Test-query can be asked at most once by the adversary $\mathcal{A}$ and is only available to $\mathcal{A}$ if the instance I is fresh[*2]. This query is answered as follows: One flips a (private) coin $b \in \{0, 1\}$ and forwards the corresponding session key SK ($\mathsf{Reveal}(I)$ would output) if $b = 1$, or a random value with the same size except the session key if $b = 0$.

The adversary $\mathcal{A}$ is provided with random coin tosses, some oracles and then is allowed to invoke any number of queries as described above, in any order. The aim of the adversary is to break the privacy of the session key (a.k.a., semantic security) in the context of executing P.

Definition3 (**AKE Security**) The AKE security is defined by the game $\mathbf{Game}^{\mathsf{ake}}(\mathcal{A}, P)$, in which the ultimate goal of the adversary is to guess the bit b involved in the Test-query by outputting this guess b'. We denote the AKE advantage, by $\mathsf{Adv}_P^{\mathsf{ake}}(\mathcal{A}) = 2\Pr[b = b'] - 1$, as the probability that $\mathcal{A}$ can correctly guess the value of b. The protocol P is said to be (t, ε)-AKE-secure if $\mathcal{A}$'s advantage is smaller than ε for any adversary $\mathcal{A}$ running time t.

5. Security of AKA*

For the security, we can say that the AKA* protocol is provably secure in the random oracle model [8] under the CDH problem.

Theorem1 Let P be the AKA* protocol. For any adversary $\mathcal{A}$ within a polynomial time t, with less than q_{se} active interactions with the parties (Send-queries) and q_{ex} passive eavesdroppings (Execute-queries), $\mathsf{Adv}_P^{\mathsf{ake}}(\mathcal{A}) \leq \varepsilon$, with ε upper-bounded by

$$\frac{(q_{\mathsf{ex}} + q_{\mathsf{se}})^2}{q} + \frac{4q_{\mathsf{se}}}{2^k} + 12n \cdot q_{\mathsf{se}} \times \mathsf{Adv}_{DCR}^{\mathsf{euf\text{-}cma}}(\mathcal{F}) \,, \qquad (2)$$

where n is the cardinality of $\mathbf{U}$, and k is the output length of H_j, for $j = 3, 4, 5$.

6. Discussions

In this section, we compare the AKA* protocol with relevant protocols (EAP-AKA [1], EAP-AKA' [2], EAP-AKA' FS [3], 5G-AKA [4], and 5G-AKA' [5]) in terms of efficiency, FS (Forward Secrecy), UE anonymity, and UE unlinkability.

[*2] We say that an instance I is fresh unless the $\mathsf{Reveal}(I)$-query is asked by an adversary $\mathcal{A}$.

表 1 Comparison of the $\mathsf{AKA}^\star$ and relevant protocols where R is a random challenge *RAND*, SE is AES-128 ECB mode, and $|l|$ indicates a bit-length of l

Protocols	Computation costs		Comm. costs	FS	UE anonymity / unlinkability
	UE	HN			
EAP-AKA [1], EAP-AKA' [2]			$\|\mathsf{U}\|+\|\mathsf{N}\|$ $+\|R\|+4\|\mathsf{H}\|$	No	Yes / No
EAP-AKA' FS [3]	$2\mathsf{Exp}_{\mathbb{G}}$	$2\mathsf{Exp}_{\mathbb{G}}$	$\|\mathsf{U}\|+\|\mathsf{N}\|+2\|p\|$ $+\|R\|+4\|\mathsf{H}\|$	Yes	Yes / No
5G-AKA [4]	$2\mathsf{Exp}_{\mathbb{G}}$	$1\mathsf{Exp}_{\mathbb{G}}$	$\|p\|+\|\mathsf{SE}\|+\|\mathsf{N}\|$ $+\|R\|+3\|\mathsf{H}\|$	No	Yes / No
5G-AKA' [5]	$2\mathsf{Exp}_{\mathbb{G}}$	$1\mathsf{Exp}_{\mathbb{G}}$	$\|p\|+2\|\mathsf{SE}\|$ $+\|\mathsf{N}\|+3\|\mathsf{H}\|$	No	Yes / Yes
$\mathsf{AKA}^\star$ (Section 3)	$1\mathsf{Exp}_{\mathbb{G}}+$ $1\mathsf{MExp}(2)_{\mathbb{G}}$	$1\mathsf{Exp}_{\mathbb{G}}+$ $1\mathsf{MExp}(2)_{\mathbb{G}}$	$\|\mathsf{U}\|+\|\mathsf{N}\|$ $+2\|p\|+\|\mathsf{H}\|$	Yes	Yes / Yes *1

*1: One round of message exchanges are needed.

6.1 Efficiency

Let $\mathsf{Exp}_{\mathbb{G}}$ and $\mathsf{MExp}(\mathsf{m})_{\mathbb{G}}$ be a modular exponentiation g^x in $\mathbb{G}$ and an m-fold multi-exponentiation $g_1^{x_1}\cdots g_m^{x_m}$ in $\mathbb{G}$, respectively. The computation cost of $\mathsf{MExp}(\mathsf{m})_{\mathbb{G}}$ is 1 exponentiation plus 2^m multiplications, which for small $m=2$ or $m=3$ is essentially the same as 1 exponentiation [22], [23]. Since the DCR signature K needs $1\mathsf{MExp}(2)_{\mathbb{G}}$, the computation costs (i.e., $1\mathsf{Exp}_{\mathbb{G}}+1\mathsf{MExp}(2)_{\mathbb{G}}$ on each side) of the $\mathsf{AKA}^\star$ protocol are almost same as those of the Diffie-Hellman key exchange [6]. If pre-computation (i.e., computing X and Y in advance) is allowed, the computation costs of each side are reduced to $1\mathsf{MExp}(2)_{\mathbb{G}}$. Compared to the Diffie-Hellman key exchange [6], the $\mathsf{AKA}^\star$ protocol requires one hash size of communication costs and one message flow additionally.

6.2 Comparison

Here, we compare the $\mathsf{AKA}^\star$ protocol of Section 3 with relevant protocols (EAP-AKA [1], EAP-AKA' [2], EAP-AKA' FS [3], 5G-AKA [4], and 5G-AKA' [5]) in terms of efficiency, forward secrecy, UE anonymity, and UE unlinkability. For a fair comparison, the following assumptions are applied: (1) We do not consider roaming (i.e., SN (Serving Network)); (2) The computation and communication costs of ECIES-KEM/DEM (in 5G-AKA [4] and 5G-AKA' [5]) and ECDHE (in EAP-AKA' FS [3]) are counted in the group description $(\mathbb{G},q,g)$; and (3) $1\mathsf{Exp}_{\mathbb{G}} \approx 1\mathsf{MExp}(2)_{\mathbb{G}}$ due to [22], [23]. We summarize a comparative result in Table 1. It is clear that only the $\mathsf{AKA}^\star$ and EAP-AKA' FS [3] protocols provide forward secrecy. However, the $\mathsf{AKA}^\star$ protocol is much more efficient than EAP-AKA' FS [3] with respect to communication costs.

参考文献

[1] IETF RFC 4187, "Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)," January, 2006. https://www.rfc-editor.org/rfc/rfc4187

[2] IETF RFC 9048, "Improved Extensible Authentication Protocol Method for 3GPP Mobile Network Authentication and Key Agreement (EAP-AKA')," October, 2021. https://www.rfc-editor.org/rfc/rfc9048.html

[3] IETF Internet-Draft, "Forward Secrecy for the Extensible Authentication Protocol Method for Authentication and Key Agreement (EAP-AKA' FS)," March, 2022. https://www.ietf.org/archive/id/draft-ietf-emu-aka-pfs-06.html

[4] TS 33.501, "Security architecture and procedures for 5G system (Release 16)," July, 2020. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3169

[5] Y. Wang and Z. Zhang and Y. Xie, "Privacy-Preserving and Standard-Compatible AKA Protocol for 5G," USENIX Security 2021, pp. 3595–3612, USENIX Association, 2021.

[6] W. Diffie and M. Hellman, "New Directions in Cryptography," IEEE Transactions on Information Theory, volume 22, number 6, pp. 644–654, IEEE, 1976.

[7] H. Krawczyk, "HMQV: A High-Performance Secure Diffie-Hellman Protocol," CRYPTO 2005, pp. 546–566, Springer, 2005.

[8] M. Bellare and P. Rogaway, "Random Oracles are Practical: A Paradigm for Designing Efficient Protocols," CCS'93, pp. 62–73, ACM, 1993.

[9] M. Bellare and P. Rogaway, "Entity Authentication and Key Distribution," CRYPTO'93, pp. 232–249, Springer, 1993.

[10] M. Bellare and D. Pointcheval and P. Rogaway, "Authenticated Key Exchange Secure against Dictionary Attacks," EUROCRYPT 2000, pp. 139–155, Springer, 2000.

[11] C. H. Lim and P. J. Lee, "A key recovery attack on discrete log-based schemes using a prime order subgroup," CRYPTO' 97, pp 249–263, Springer, 1997.

[12] IETF RFC 2785, "Methods for Avoiding the "Small-Subgroup" Attacks on the Diffie-Hellman Key Agreement Method for S/MIME," March, 2000. https://www.rfc-

editor.org/rfc/rfc2785.html

[13] IETF RFC 6628, "Efficient Augmented Password-Only Authentication and Key Exchange for IKEv2," June 2012. https://www.rfc-editor.org/rfc/rfc6628.html

[14] IETF Internet-Draft, "Hashing to Elliptic Curves," February, 2022. https://www.ietf.org/archive/id/draft-irtf-cfrg-hash-to-curve-14.html

[15] D. Basin and J. Dreier and L. Hirschi and S. Radomirovic and R. Sasse and V. Stettler, "A Formal Analysis of 5G Authentication," CCS 2018, pp. 1383–1396, ACM, 2018.

[16] S. Meier and B. Schmidt and C. Cremers and D. Basin, "The TAMARIN Prover for the Symbolic Analysis of Security Protocols," CAV 2013, pp. 696–701, Springer, 2013.

[17] R. Borgaonkar and L. Hirschi and S. Park and A. Shaik, "New Privacy Threat on 3G, 4G, and Upcoming 5G AKA Protocols," Privacy Enhancing Technologies, pp. 108–127, 2019.

[18] A. Koutsos, "The 5G-AKA Authentication Protocol Privacy," EuroS&P 2019, pp. 464–479, IEEE, 2019.

[19] G. Bana and H. C.-Lundh, "A Computationally Complete Symbolic Attacker for Equivalence Properties," CCS 2014, pp. 609–620, ACM, 2014.

[20] IETF RFC 8446, "The Transport Layer Security (TLS) Protocol Version 1.3," August, 2018. https://www.rfc-editor.org/rfc/rfc8446.html

[21] 5G & Beyond, https://www.nist.gov/programs-projects/5g-beyond

[22] E. G. Straus, "Addition chains of vectors," American Mathematical Monthly, volume 71, number 7, pp. 806–808, 1964.

[23] A. J. Menezes and P. C. van Oorschot and S. A. Vanstone, "Handbook of Applied Cryptography," pp. 617–618, CRC Press, 1996.