APPENDIX A
A COMPARATIVE EVALUATION OF EXISTING TWO-FACTOR AUTHENTICATION SCHEMES

To demonstrate the effectiveness of our framework in practice, we provide a comparative evaluation of 67 two-factor typical schemes by assessing whether the twelve criteria proposed in Section II-B have been met under the security model proposed in Section II-A. Particularly, among these 67 schemes, four were designed before 2004 where the tamper-resistance assumption of the smart cards are generally made. As expected, these early four schemes perform poorly under our new security model (see the bottom of Table A.1), while the recent schemes (e.g., [1]–[6]) generally perform much better. There is a trend that, under our evaluation framework, more recent the scheme is, more desirable the scheme will be.

However, this trend would not be obvious had these 67 schemes been assessed by the existing evaluation frameworks (i.e., [7]–[10]). More specifically, the criteria set in [7], essentially, only includes our criteria C2-C5, and thus one will not see the differences between the schemes proposed in 2010 and the schemes proposed in 2015; The criteria set in [8], [9] concerns "protocol efficiency" and most importantly, no security model is explicitly defined in [8], [9], all this would make these two frameworks virtually impossible to be decidable when assessing a scheme; Using the criteria set in [10] will not reveal the critical "usability-security tension" (discussed later), because it misses the desirable property "freely password change". All in all, the trend revealed by our framework partially demonstrates the soundness of our evaluation framework.

From the microcosmic point of view, one can see that each criterion is *satisfied* by at least 15 schemes and at the same time, it is *unmet* by at least 7 schemes. *This implies the necessity of each of the twelve criteria.* In addition, there is no scheme that can fulfill all the twelve criteria— the only scheme that can achieve eleven criteria is proposed by Odelu et al. in 2015 [4]. *This suggests the comprehensiveness of our criteria set. This also highlights the needs for more research efforts to design a better scheme.* With a careful examination, one can observe that this scheme suffers from the same "usability-security tension" with other latest schemes (e.g., [2], [3], [5]): the criterion C2 (or C9) and C4 cannot be achieved at the same time. *This suggests the necessity of the separation of C4 from C5*, in contrast to the framework in [7].

It is also worth noting that, in selecting a particular two-factor scheme for inclusion in the comparison Table A.1, we do not necessarily endorse it as better than alternatives that are not included in the table—merely because of that it is reasonably representative, or illuminates in some way what category (from a point view of the development tree where a specific scheme lies, see the history tree Fig. 2 in Section II) it belongs to can achieve. In addition, here we mainly focus on schemes for the single-server architecture because: (1) different architectures/environments may involve quite different attacking vectors and security models, and thus fair comparison is virtually impossible under a single security model; and (2) single-server-architecture-based schemes constitute the basis for schemes that are designed for other more complex architectures/environments (e.g., schemes for wireless sensor networks [11] and mobile networks [12]).

TABLE A.1. A COMPARATIVE EVALUATION OF TWO-FACTOR AUTHENTICATION SCHEMES

Scheme	Year	Ref.	No verifier table (C1)	Password friendly (C2)	No password exposure (C3)	No smart card loss problem (C4)	Resistance to known attacks (C5)	Sound repairability (C6)	Provision of key agreement (C7)	No clock synchronization (C8)	Timely typo detection (C9)	Mutual authentication (C10)	User anonymity (C11)	Forward secrecy (C12)
Lu *et al.*	2016	[13]	✓	✓	✓	·	·	✓	✓	✓	✓	✓	·	✓
Xie *et al.*	2016	[14]	✓	✓	✓	·	✓	·	✓	·	·	✓	·	✓
Islam *et al.*	2016	[15]	✓	✓	✓	·	✓	✓	✓	✓	✓	✓	✓	✓
Muhaya	2015	[1]	✓	✓	✓	·	✓	·	✓	✓	✓	✓	✓	·
Xu-Wu	2015	[2]	✓	·	✓	✓	·	·	✓	✓	·	✓	✓	✓
Mishra *et al.*	2015	[3]	✓	✓	✓	·	✓	✓	✓	·	✓	✓	✓	·
Odelu *et al.*	2015	[4]	✓	✓	✓	·	✓	✓	✓	✓	✓	✓	✓	✓
Byun	2015	[5]	·	·	✓	✓	✓	✓	✓	✓	·	✓	✓	✓
Wang *et al.*	2015	[6]	✓	·	✓	✓	✓	✓	✓	·	✓	✓	✓	·
Truong *et al.*	2015	[16]	✓	✓	·	·	✓	·	✓	✓	✓	✓	·	✓
Djellali *et al.*	2015	[17]	✓	✓	·	·	✓	·	✓	✓	·	✓	✓	·
Mishra *et al.*	2015	[18]	✓	·	✓	·	✓	✓	✓	·	✓	✓	·	✓
Wu *et al.*	2015	[19]	·	✓	✓	·	·	·	✓	✓	·	✓	✓	✓
Chaudry *et al.*	2015	[20]	✓	✓	✓	·	·	✓	✓	·	✓	✓	·	·
Kumari *et al.*	2014	[21]	✓	✓	✓	·	✓	·	✓	✓	✓	✓	·	·
Chen *et al.*	2014	[22]	✓	·	·	·	✓	·	✓	·	·	✓	·	·
Tsai *et al.*	2013	[23]	✓	✓	✓	·	✓	·	✓	✓	·	✓	✓	✓
Li *et al.*	2013	[24]	✓	✓	·	·	✓	·	✓	·	✓	✓	·	✓
Kumari-Khan	2013	[25]	✓	·	·	·	✓	·	✓	·	✓	✓	✓	·
Sun-Cao	2013	[26]	✓	·	✓	·	✓	✓	✓	·	·	✓	✓	·
Lee-Liu	2013	[27]	✓	✓	✓	·	✓	·	✓	✓	·	✓	✓	·
Islam-Biswas	2013	[28]	·	·	·	✓	·	✓	✓	✓	·	✓	·	✓
Li-Zhang	2013	[29]	✓	·	·	✓	·	✓	✓	·	·	✓	✓	✓
Chang et al.	2013	[30]	✓	·	·	·	·	·	·	·	·	✓	·	·
Li	2013	[31]	✓	·	✓	·	·	✓	✓	✓	·	✓	✓	✓
Kim-Kim	2012	[32]	✓	·	·	✓	·	✓	✓	✓	·	✓	✓	·
Ramasamy-Muniyandi	2012	[33]	✓	✓	·	✓	·	·	·	·	·	·	·	·
Wu *et al.*	2012	[8]	✓	✓	✓	·	✓	✓	✓	✓	·	✓	✓	✓
Zhu	2012	[34]	✓	✓	✓	·	✓	✓	·	✓	·	✓	·	·
Wang-Ma	2012	[35]	✓	✓	✓	·	✓	✓	✓	·	✓	✓	✓	✓
Hsieh-Leu	2012	[36]	✓	·	·	·	✓	✓	·	·	·	✓	·	·
Wen-Li	2012	[37]	✓	·	·	·	·	✓	✓	·	·	✓	·	·
Wang *et al.*	2012	[38]	✓	✓	✓	·	✓	·	✓	✓	✓	✓	✓	✓
He *et al.*	2012	[39]	✓	·	✓	✓	✓	·	✓	·	·	✓	·	·
Wang *et al.*	2011	[40]	✓	✓	·	·	·	✓	✓	✓	·	✓	✓	✓
Chen *et al.*	2011	[41]	✓	✓	✓	·	·	·	✓	·	·	✓	·	·
Khan *et al.*	2011	[42]	✓	✓	✓	·	✓	✓	✓	·	✓	✓	·	·
Kim *et al.*	2011	[43]	✓	·	·	✓	✓	✓	✓	·	·	✓	·	·
Awasthi *et al.*	2011	[44]	✓	·	·	·	·	·	·	·	·	✓	·	·
Li-Lee	2011	[45]	✓	·	✓	·	·	✓	✓	✓	·	✓	·	·
Sood *et al.*	2011	[46]	✓	✓	·	·	·	✓	✓	·	✓	·	·	·
Li *et al.*	2010	[47]	✓	✓	✓	✓	·	✓	✓	✓	·	✓	✓	·
Tsai *et al.*	2010	[48]	✓	✓	·	·	✓	·	✓	·	·	✓	✓	·
Song	2010	[49]	✓	·	·	·	✓	·	✓	·	·	✓	·	·
Sood *et al.*	2010	[50]	✓	✓	·	✓	✓	·	✓	·	·	·	·	·
Yeh *et al.*	2010	[51]	✓	✓	·	·	✓	✓	✓	·	·	✓	·	·
Horng *et al.*	2010	[52]	✓	·	✓	✓	·	·	✓	✓	·	✓	✓	✓
Sun *et al.*	2009	[53]	✓	✓	✓	·	✓	✓	✓	✓	·	✓	·	✓
Hsiang-Shi	2009	[54]	✓	✓	·	·	✓	✓	·	·	·	✓	·	·
Xu *et al.*	2009	[55]	✓	·	·	·	·	·	✓	·	·	✓	·	✓
Kim-Chung	2009	[56]	✓	✓	·	·	·	·	·	·	✓	✓	·	·
Chung *et al.*	2009	[57]	✓	·	✓	✓	✓	✓	✓	✓	·	✓	·	✓
Ramasamy	2009	[58]	✓	·	·	✓	✓	✓	·	·	·	✓	·	·
Yang *et al.*	2008	[7]	✓	✓	✓	·	✓	·	✓	✓	·	✓	·	✓
Juang *et al.*	2008	[59]	✓	·	✓	✓	·	✓	✓	✓	·	✓	·	·
Chen *et al.*	2008	[60]	✓	·	✓	·	·	·	·	✓	·	✓	·	·
Wang *et al.*	2007	[61]	·	✓	✓	·	·	·	✓	·	✓	✓	·	·
Wang *et al.*	2007	[62]	✓	·	✓	·	·	✓	✓	✓	·	✓	·	·
Liao *et al.*	2006	[9]	✓	✓	✓	·	·	·	✓	·	·	✓	·	✓
Lee et al.	2005	[63]	✓	·	·	·	·	·	·	·	·	✓	·	·
Fan et al.	2005	[64]	✓	·	·	✓	✓	✓	·	✓	·	✓	·	·
Lu-Cao	2005	[65]	✓	·	·	✓	✓	·	·	·	·	·	·	·
Yoon *et al.*	2004	[66]	✓	✓	✓	·	·	✓	·	·	·	✓	·	·
Ku *et al.*	2004	[67]	✓	✓	✓	·	·	✓	·	·	·	✓	·	·
Wu-Chieu	2003	[68]	✓	✓	·	·	·	·	·	·	·	·	·	·
Awasthi-Lal	2003	[69]	·	·	·	·	·	·	·	·	·	·	·	·
Sun	2000	[70]	·	·	·	·	·	·	·	·	·	·	·	·
Hwang-Li	2000	[71]	·	·	·	·	·	·	·	·	·	·	·	·

*The present authors have contributed to the following schemes: Wang et al. [6], Wang et al. [38] and Wang-Ma [35]. Interested readers may verify that we have evaluated them impartially.

Finally, during the past five years the present authors have examined more than three hundred two-factor schemes, including over 170 ones for the single server architecture (see

some of our results [72]–[74]), over 100 ones for the multi-server architecture (see some of our results [75]), over 60 ones for mobile networks (see some of our results [76], [77]) and over 50 for the wireless sensor networks (see some of our results [78], [79]). Based on our past cryptanalysis experience, we believe that (1) whenever a specific scheme is identified by us to be *unable* to achieve some criteria, this is sufficiently definite to be true; and (2) when a specific scheme is identified by us to be able to achieve some criteria, this is highly likely to be true, while there may be some probability that we have missed some attacking modes. For instance, we have found that there are at least 8 vastly different attacking modes (e.g., attacking by returning back the extracted card and attacking by exploiting the first/second/third protocol flow) when investigating whether a scheme can resist offline password guessing attack in case the adversary has got access to victim's card, while this only constitutes part of the total work involved in evaluating just a single criteria C4. *This shows the great difficulty and the mount of manual work entailed when constructing a table like* A.1.

APPENDIX B
WIDE APPLICABILITY OF "FUZZY-VERIFIER" + "HONEYWORDS"

In this Section, we use representative schemes [23], [80], [81] as case studies to demonstrate exactly how our "fuzzy-verifier" and "honeywords" can be integrated into other schemes (and even schemes for other architectures). More specifically, we employ two typical schemes, i.e., Tsai et al.'s scheme [23] and Xue et al.'s scheme [80], to our approach can be integrated into two-factor authentication schemes for the single-server architecture and multi-server environment, respectively. In addition, we also use Odelu et al.'s scheme [81] to briefly show the applicability of our approach to three-factor authentication schemes. After our integration, one can confirm that existing usability-security conflicts in these schemes would be well eliminated. This shed light on how to eliminate the usability-security dilemma in various other schemes (e.g., [7], [8], [53], [82]).

A. Integration into Tsai et al.'s scheme

Tsai et al.'s scheme [23] has two versions: a two-factor one and a three-factor one. Here we mainly focus on the two-factor version. Their scheme consists of five phases, namely, parameter generation, registration, pre-computation, login and password update. Readers are referred to [23] for details of this scheme, but for ease of comprehension, we will follow its notations as closely as possible.

This scheme exhibits many desirable features over existing schemes, such as user anonymity, high efficiency and formal security proofs, yet it was recently found vulnerable to the smart card loss problem (i.e., unable to achieve C4) [72]. The key point is that, this scheme allows users to change their passwords freely and locally (i.e., able to achieve C2), yet there is no verification of the old password before the update of new password. If an attacker $\mathcal{A}$ manages to gain temporary access to the smart card of legitimate user U_i (note that this is a quite realistic assumption in practice), she can easily change the password of user U_i without any obstacle.

As revealed in [72] and further investigated in Sec. V-A, there are some subtleties and tricks in coping with this problem and before this work, there is no existing solution. Fortunately, our proposed "fuzzy-verifier" and "honeywords" can be integrated into this scheme as follows:

(1) In the registration phase, besides computing $V = h(ID_i\|x) \oplus h(PW_i\|b)$, the server further computes a fuzzy-verifier $A_i = h((h(\boldsymbol{ID}_i)\oplus h(PW_i\|b))\ \texttt{mod}\ n_0)$ for U_i, and also stores both A_i and $A_i \oplus a_i$ in U_i's smart card, where $h(\cdot)$ is a one-way hash function and a_i is a random number. S stores $\{\boldsymbol{ID}_i, a_i,$ `Honey_List=` NULL$\}$ in its backend database. The other parts remain the same as in [23].

(2) In the login phase, after the user U_i keys her identity ID_i and password PW_i, the card computes $A_i^* = h((h(\boldsymbol{ID}_i)\oplus h(PW_i\|b))\ \texttt{mod}\ n_0)$ and verifies the validity of A_i^* by checking whether A_i^* equals to the stored A_i. If the verification holds, it implies the input $\boldsymbol{ID}_i$ and PW_i are valid with a probability of $\frac{n_0-1}{n_0}(\approx \frac{99.61}{100}$, when n_0=$2^8)$. Otherwise, the smart card rejects. If A_i^* equals A_i, the smart card computes $a_i = (A_i\oplus a_i)\oplus A_i$ and $C_1 = (ID_i\| a_i\|h(\boldsymbol{ID_i}\|x)\ \|(h(ID_i\|x)\oplus N_{C2}))\oplus h_1(c)$. The other parts remain the same as in [23].

(3) In Step 3 of the login phase, after the server S receives the login request $\{C_1, e\}$ from U_i, S derives $ID_i'\|a_i'\|h(ID_i\|x)'$ from C_1, and checks whether a_i' equals the stored a_i. S rejects if they are not equal. Otherwise, S checks whether the derived $h(ID_i\|x)'$ equals the computed $h(ID_i\|x)$. If they are equal, S proceeds to the next step. If they are unequal, S now knows that $a_i' = a_i$ but $h(ID_i\|x)' \neq h(ID_i\|x)$, implying that there is a $1/2^{n_0}$ probability that U_i's card has been corrupted. Accordingly, S performs either (1) inserts $h(ID_i\|x)'$ into `Honey_List` when there are less than m_0 (e.g., $m_0 = 10$) items in `Honey_List`; or (2) suspends U_i's card (i.e., when there are m_0 items in `Honey_List`) until U_i re-registers. The other parts remain the same as in [23].

(4) In the password-change phase, the user U_i keys *twice* her identity ID_i and password PW_i. If U_i accidentally keys two unmatched (ID_i, PW_i) pairs, which is quite realistic in mobile device use cases, the card rejects. If they are match, the card computes $A_i^* = h((h(\boldsymbol{ID}_i) \oplus h(PW_i\|b))\ \texttt{mod}\ n_0)$ and verifies the validity of A_i^* by checking whether A_i^* equals to the stored A_i. If the verification holds, it implies the input $\boldsymbol{ID}_i$ and PW_i are valid with a probability of $\frac{n_0-1}{n_0}(\approx \frac{99.61}{100}$, when n_0=$2^8)$. Otherwise, the smart card rejects. Then, the card asks the user to submit a new password PW_i^{new} and computes $V^{new} = V \oplus h(PW_i\|b) \oplus h(PW_i^{new}\|b)$, $A_i^{new} = h((h(\boldsymbol{ID}_i) \oplus h(PW_i^{new}\|b))\ \texttt{mod}\ n_0)$ and $a_i^{new} = (A_i\oplus a_i)\oplus A_i\oplus A_i^{new}$. Then, smart card updates the values of V, A_i and a_i with V^{new}, A_i^{new} and a_i^{new}, respectively.

Our above amendments are essentially based on the idea illustrated in Sec. V of the main text: we first force the attacker $\mathcal{A}$ to *has to* launch an online password guessing attack by interacting with S in order to determine the exactly correct password, and then design ways to enable S to *timely* detect the event that the parameters in U_i's card have been extracted and used by $\mathcal{A}$ to perform an online guessing attack. While there may be other ways to conquer this usability-security tension,

we for the first time show the potential and provide a concrete solution to conquer this usability-security tension in [23].

B. Integration into Xue et al.'s scheme

Xue et al.'s scheme [80] is designed for the multi-server architecture, which is suitable for environments where users need to access more than one service server but only maintain one password and one smart card. Their scheme involves three participants: user U_i, service server S_j and control server CS. It consists of three phases (i.e., registration, login and authentication) and two phases (i.e., password update and dynamic identity update). Readers are referred to [80] for details of this scheme, but for ease of comprehension, we will follow its notations as closely as possible.

In U_i's card memory, there are two parameters stored: a random number b and $C_i = h(ID_i\|A_i) = h(ID_i\|h(b\|PW_i))$. These two parameters together can be used to support the property "timely typo detection" (i.e., C9). One can see that, if an adversary $\mathcal{A}$ obtains the card and extracts these two parameters, $\mathcal{A}$ can determine U_i's password PW_i as follows:

Step 1. Guesses the value of ID_i to be ID_i^* from dictionary space $\mathcal{D}_{id}$ and the value of PW_i to be PW_i^* from dictionary space $\mathcal{D}_{pw}$;
Step 2. Computes $C_i^* = h(ID_i^*\|A_i) = h(ID_i^*\|h(b\|PW_i^*))$, where b is revealed from U_i's card;
Step 3. Checks the correctness of (ID_i^*, PW_i^*) by comparing if the computed C_i^* equals the extracted C_i;
Step 4. Repeats Step 1, 2 and 3 of this procedure until the correct value of (ID_i^*, PW_i^*) is found.

The time complexity of the above attacking procedure is $\mathcal{O}(|\mathcal{D}_{id}|*|\mathcal{D}_{pw}|*2T_H$, where T_H is the running time for Hash operation. In reality, the dictionary size is very restricted, e.g., $|\mathcal{D}_{id}| \leq |\mathcal{D}_{pw}| \leq 10^6$ [83], [84]. Further, according to the timings in Table B.1, $\mathcal{A}$ may determine the password in about 17.63 days on a common PC.

TABLE B.1. COMPUTATION EVALUATION OF RELATED OPERATIONS ON COMMON PCS

Experimental platform (common PCs)	Modular Exp. T_E ($\|n\| = 512$)	Hash operation T_H (SHA-1)	Other lightweight oper.(e.g., XOR, $\|$)
Intel T5870 2.00 GHz	2.573 ms	2.437 µs	0.011 µs
Intel i5750 2.66 GHz	2.106 ms	1.980 µs	0.009 µs
Pentium IV 3.06 GHz	1.676 ms	1.523 µs	0.008 µs

In addition, *user ID generally cannot be considered as a secret and actually, it is often publicly available*. Thus, there is a high probability for $\mathcal{A}$ to learn the user's identity ID_i other than guessing it. In this light, the above attack will be more practical. What's more, high performance computers are quite common those days and cheap cloud computing services are also easily available (e.g., Amazon EC2 [85]). All this indicates that the above attack is effective even if $\mathcal{A}$ has to figure out both ID_i and PW_i simultaneously.

To conquer this vulnerability while still preserving C9, the *definite* password verifier $C_i = h(ID_i\|h(b\|PW_i))$ shall be changed to a *fuzzy* verifier $FC_i = h(ID_i\|h(b\|PW_i)\texttt{mod}\ n_0)$. Furthermore, some "honeywords" shall be kept by CS to detect the user card breach event. More specifically, our proposed "fuzzy-verifier" and "honeywords" can be integrated into this scheme as follows:

(1) In the registration phase, U_i does not compute C_i but instead computes a fuzzy-verifier $FC_i = h(h(ID_i\|h(b\|PW_i))\ \texttt{mod}\ n_0)$, and stores FC_i and $FC_i \oplus a_i$ in U_i's smart card, where $h(\cdot)$ is a one-way hash function and a_i is a random number. The control server CS stores $\{ID_i, a_i, \texttt{Honey\_List=NULL}\}$ in its backend database. The other parts remain the same as in [23].
(2) In the login phase, after the user U_i keys her identity ID_i and password PW_i, the card computes $FC_i = h(h(ID_i^*\|h(b\|PW_i^*))\ \texttt{mod}\ n_0))$ and verifies the validity of FC_i^* by checking whether FC_i^* equals to the stored FC_i. If the verification holds, it implies the input ID_i and PW_i are valid with a probability of $\frac{n_0-1}{n_0}(\approx \frac{99.61}{100}$, when n_0=2^8). Otherwise, the smart card rejects. If FC_i^* equals FC_i, the smart card computes $a_i = (FC_i \oplus a_i) \oplus FC_i)$ and $CID_i = (ID_i\|a_i\|B_i) \oplus h(B_i\|N_{i1}\|TS_i\|\text{“00”})$. The other parts remain the same as in [80].
(3) In Step 3 of the login phase, after the control server CS receives the login request $\{F_i, P_{ij}, CID_i, G_i, PID_i, TS_i, J_i, K_i, L_i, M_i, PSID_j\}$ from S_j, CS derives $ID_i'\|a_i'\|B_i'$ from CID_i, and checks whether a_i' equals the stored FC_i. S rejects if they are not equal. Otherwise, CS checks whether the derived B_i' equals the computed $h(PID_i\|x)$. If they are equal, CS proceeds to the next step. If they are unequal, CS now knows that $a_i' = a_i$ but $h(PID_i\|x)' \neq h(PID_i\|x)$, implying that there is a $1/2^{n_0}$ probability that U_i's card has been corrupted. Accordingly, CS performs either (1) inserts $h(PID_i\|x)'$ into $\texttt{Honey\_List}$ when there are less than m_0 (e.g., $m_0 = 10$) items in $\texttt{Honey\_List}$; or (2) suspends U_i's card (i.e., when there are m_0 items in $\texttt{Honey\_List}$) until U_i re-registers. The other parts remain the same as in [80].
(4) In the password-change phase, the user U_i keys *twice* her identity ID_i and password PW_i. If U_i accidentally keys two unmatched (ID_i, PW_i) pairs, which is quite realistic in mobile device use cases, the card rejects. If they are match, the card computes $FC_i^* = h(h(ID_i^*\|h(b\|PW_i^*))\ \texttt{mod}\ n_0)$ and verifies the validity of FC_i^* by checking whether FC_i^* equals to the stored FC_i. If the verification holds, it implies the input ID_i and PW_i are valid with a probability of $\frac{n_0-1}{n_0}(\approx \frac{99.61}{100}$, when n_0=2^8). Otherwise, the smart card rejects. Then, the card asks the user to submit a new password PW_i^{new} and computes $FC_i^{new} = h(h(ID_i\|h(b\|PW_i^{new}))\ \texttt{mod}\ n_0)$, $D_i^{new} = D_i \oplus h(h(ID_i\|b) \oplus h(b\|PW_i) \oplus h(b\|PW_i^{new}))$ and $a_i^{new} = (A_i \oplus a_i) \oplus A_i \oplus A_i^{new}$. Then, smart card updates the values of FC_i, D_i and a_i with FC_i^{new}, D_i^{new} and a_i^{new}, respectively.

C. Integration into Odelu et al.'s scheme

Odelu et al.'s scheme [81] is three-factor scheme designed for the multi-server architecture. As we have said earlier, this kind of scheme is suitable for environments where users need to access more than one service server but only maintain one password and one smart card (as well as her fingerprint). Their scheme involves three participants: user U_i, service server S_j and registration center RC (which serves as the same role of the control server CS in Xue et al.'s scheme [80]). It consists of four phases (i.e., initialization, registration, login and authentication) and two phases (i.e., password update, and Revocation and re-registration). Readers are referred to [81]

for details of this scheme, but for ease of comprehension, we will follow its notations as closely as possible.

In U_i's card memory, there are three parameters stored: an auxiliary string θ_i, $s_i = H(k_i \| ID_i \| H(pw_i \| \sigma_i))$ and $z_i = k_i \oplus H(pw_i \| \sigma_i)$. It is not difficult to see that, Odelu et al.'s scheme [81] cannot provide truly "three-factor security", which is the most essential goal of a three-factor scheme. More specifically, U_i's password factor can be offline guessed by using s_i as a comparison target if the other two authentication factors (i.e., smart card and biometric) have been breached. Odelu et al. have realized this issue and pointed out that their scheme "can achieve the three-factor authentication by removing the hash value s_i from the smart-card" and "In that case, the password change will not be possible locally."

Fortunately, our "fuzzy-verifier" and "honeywords" can be integrated into this scheme to ensure that three-factor security can still be achieved while preserving the property of "local and secure password change". The details are as follows:

(1) In the registration phase, the registration center RC computes s_i as $s_i = H((k_i \| ID_i \| H(pw_i \| \sigma_i)) \texttt{mod } n_0)$, and stores s_i and $s_i \oplus a_i$ in U_i's smart card, where $H(\cdot)$ is a one-way hash functions. RC stores $\{\boldsymbol{ID}_i, a_i, H(ID_i \| k), r_i,$ `Honey_List`=NULL$\}$ in its backend database. The other parts remain the same as in [81].

(2) In the login phase, after the user U_i keys her identity ID_i and password PW_i and imprints her personal biometrics B_i' at the sensor. Then, smart card computes $\sigma_i' = Rep(B_i', \theta_i)$ and $k_i' = z_i' \oplus H(pw_i' \| \sigma_i')$ and checks whether $s_i' = H((k_i' \| ID_i \| H(pw_i' \| \sigma_i')) \texttt{mod } n_0)$ equals the stored s_i. If the verification holds, it implies the input PW_i is valid with a probability of $\frac{n_0-1}{n_0} (\approx \frac{99.61}{100}$, when n_0=2^8). Otherwise, the smart card rejects. If s_i' equals s_i, the smart card computes $C_1 = E_{K_{1x}}[ID_i \| a_i \| k_i \| SID_j \| s_j \| n_1]$. The other parts remain the same as in [81].

(3) In Step AK2 of the Authentication phase, after the the registration center RC receives the login request $\{C_1, X, h_1, C_2, h_2\}$ from S_j, RC derives $ID_i' \| a_i' \| k_i'$ from C_1, and checks whether a_i' equals the stored a_i. RC rejects if they are not equal. Otherwise, RC checks whether the derived k_i' equals the computed $k_i = H(ID_i \| k \| r_i \| H(ID_i \| k))$. If they are equal, RC proceeds to the next step. If they are unequal, RC now knows that $a_i' = a_i$ but $k_i' \neq H(ID_i \| k \| r_i \| H(ID_i \| k))$, implying that there is a $1/2^{n_0}$ probability that U_i's card has been corrupted. Accordingly, RC performs either (1) inserts k_i' into `Honey_List` when there are less than m_0 (e.g., $m_0 = 10$) items in `Honey_List`; or (2) suspends U_i's card (i.e., when there are m_0 items in `Honey_List`) until U_i re-registers. The other parts remain the same as in [81].

(4) In the password-change phase, the user U_i keys *twice* her identity ID_i and password PW_i. If U_i accidentally keys two unmatched (ID_i, PW_i) pairs, which is quite realistic in mobile device use cases, the card rejects. If they are match, U_i inputs her biometric B_i' and the card computes $\sigma_i' = Rep(B_i', \theta_i)$, $s_i^* = H((k_i \| ID_i \| H(pw_i \| \sigma_i')) \texttt{mod } n_0)$ and verifies the validity of s_i^* by checking whether s_i^* equals to the stored s_i. If the verification holds, it implies the input PW_i is valid with a probability of $\frac{n_0-1}{n_0} (\approx \frac{99.61}{100}$, when n_0=2^8). Otherwise, the smart card rejects. Then, the card asks the user to submit a new password PW_i^{new} and computes $s_i^{new} = H((k_i \| ID_i \| H(pw_i^{new} \| \sigma_i)) \texttt{mod } n_0)$, $z_i^{new} = z_i \oplus H(PW_i \| \sigma_i) \oplus H(PW_i^{new} \| \sigma_i)$ and $a_i^{new} = (s_i \oplus a_i) \oplus s_i \oplus s_i^{new}$. Then, smart card updates the values of s_i, z_i and a_i with s_i^{new}, z_i^{new} and a_i^{new}, respectively.

APPENDIX C
FORMAL SECURITY ANALYSIS OF OUR SCHEME

A. Proof of Theorem 1

Proof. In the proof below, we do not consider forward-secrecy for simplicity. We incrementally define a sequence of games starting at the real attack game G_0 and ending up with G_8. For each game G_n (n=0,1, ..., 8), we define the following events:

- $\mathbf{Succ_n}$ occurs if $\mathcal{A}$ correctly guesses the bit c involved in the Test-query.
- $\mathbf{AskPara_n}$ occurs if $\mathcal{A}$ correctly computes the parameter k by asking a hash query $\mathcal{H}_0$ on $b \| PW_i$ or $x \| ID_i \| T_{reg}$.
- $\mathbf{AskAuth_n}$ occurs if $\mathcal{A}$ correctly computes the parameter k and asks a hash query $\mathcal{H}_1$ (or $\mathcal{H}_2$) on $ID_i \| ID_S \| Y_1 \| C_2 \| k \| K$, where K is K_U or K_S.
- $\mathbf{AskH_n}$ occurs if the adversary asks a hash query $\mathcal{H}_i$ $(i = 1, 2, 3)$ on $ID_i \| ID_S \| Y_1 \| C_2 \| k \| K$, where K is K_U or K_S.

Game $\mathbf{G_0}$: This game corresponds to the real attack, in the random oracle model. By definition, we have

$$\mathrm{Adv}_{\mathcal{P}}^{\mathrm{ake}}(\mathcal{A}) = 2\Pr[\mathrm{Succ}_0] - 1. \quad (1)$$

Game $\mathbf{G_1}$: In this game, we simulate the hash oracles $\mathcal{H}_i$(i=0,1,2 and 3, but also four additional hash functions $\mathcal{H}_i'$ that will appear in Game $\mathbf{G_7}$) as usual by maintaining a hash list $\Lambda_{\mathcal{H}}$(and another list $\Lambda_{\mathcal{A}}$ containing the hash-queries asked by the adversary itself). We also simulate all the instances, as the real players would do, for the Send-queries and for the Execute, Reveal, Corrupt and Test-queries (see Figure C.1).

From this simulation, one can easily see that this game is perfectly indistinguishable from the real attack. Hence,

$$\left|\Pr[\mathrm{Succ}_1] - \Pr[\mathrm{Succ}_0]\right| = 0 \quad (2)$$

Game $\mathbf{G_2}$: For an easier analysis, in this game, we simulate all oracles as in game $\mathbf{G_1}$ except that we cancel games in which some (unlikely) collisions appear:

- collisions on the partial transcripts $((C_1, M_i, CAK_i, CID_i), (C_2, C_3), C_4)$. Note that transcripts involve at least one honest party, and thus one of C_1 or C_2 is truly uniformly distributed;
- collisions on the output of hash queries.

Both probabilities are bounded by the birthday paradox:

$$\left|\Pr[\mathrm{Succ}_2] - \Pr[\mathrm{Succ}_1]\right| \leq \frac{(q_{send} + q_{exe})^2}{2p} + \frac{q_h^2}{2^{l+1}} \quad (3)$$

where $l = min\{l_i\}, i = 0, 1, 2, 3$.

Game $\mathbf{G_3}$: We define game $\mathbf{G_3}$ by aborting the game wherein the adversary may have lucky in guessing the correct authenticator C_3 or C_4 (that is, without asking the corresponding hash query $\mathcal{H}_1$ or $\mathcal{H}_2$). Since C_1 and C_2 did not appear in a previous session (since the Game $\mathbf{G_2}$), this happens only if

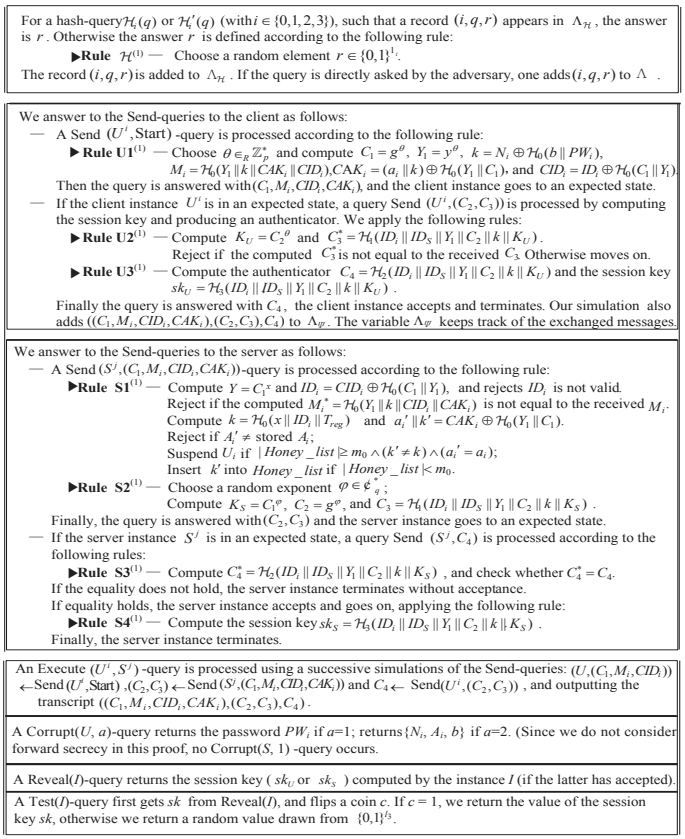

Fig. C.1. Simulation of the queries in our scheme

the authenticator C_3 (or C_4) had been correctly guessed by $\mathcal{A}$ without asking $\mathcal{H}_1$ (or $\mathcal{H}_2$):

$$\left|\Pr[\text{Succ}_3] - \Pr[\text{Succ}_2]\right| \leq \frac{q_{send}}{2^l} \quad (4)$$

Game $\mathbf{G_4}$: We define game $\mathbf{G_4}$ by aborting the game wherein the adversary may have lucky in guessing the correct parameter k (i.e., without asking the corresponding query). We reach this aim by modifying the way the participants process the queries. We use the rule as follows:

- **Rule U3$^{(4)}$** – Look for a record $(0, *\|ID_i\|*, k)$ in $\Lambda_{\mathcal{A}}$. If such a record does not exist, we abort the game. Otherwise, compute $C_4 = \mathcal{H}_2(ID_i\|ID_S\|Y_1\| \ C_2\|k \ \|K_U)$ and the session key $sk_U = \mathcal{H}_3(ID_i\|ID_S\| \ Y_1\|C_2\|k\|K_U)$.
- **Rule S3$^{(4)}$** – computes $C_4^* = \mathcal{H}_2(ID_i\|ID_S\|Y_1\|C_2 \ \|k\|K_S)$ and then checks if C_4^* equals the received value of C_4. If this verification holds, the server looks for a record $(0, *, P_i)$ in $\Lambda_{\mathcal{A}}$ and a record $((C_1, M_i, CAK_i, CID_i), (C_2, \ C_3), C_4)$ in Λ_Ψ. If such records do not exist, we abort the game.

Since C_1 and C_2 did not appear in a previous session (since the Game $\mathbf{G_2}$), this happens only if the parameter k had been correctly guessed by the adversary without asking $\mathcal{H}_0$:

$$\left|\Pr[\text{Succ}_4] - \Pr[\text{Succ}_3]\right| \leq \frac{q_{send}}{2^{l_0}} \quad (5)$$

Game $\mathbf{G_5}$: We define this game by aborting the game wherein the adversary may have computed the correct parameter k and impersonate as a client or server. We reach this aim by modifying the way the participants process the queries. We use the rule as follows:

- **Rule U2$^{(5)}$** – Look for a record $(0, *\|ID_i\|*, k)$ in $\Lambda_{\mathcal{A}}$. If such a record exists, we abort the game. Otherwise, compute $K_U = (C_2)^u \bmod p$, $C_3^* = \mathcal{H}_1(ID_i\|ID_S\|Y_1 \ \|C_2\|k\|K_U)$, and compare C_3^* with the received C_3. If the equality does not hold, terminate without acceptance. Otherwise, move on.
- **Rule S3$^{(5)}$** – computes $C_4^* = \mathcal{H}_2(ID_i\|ID_S\|Y_1\| \ C_2\|k\|K_S)$ and then checks if C_4^* equals the received value of C_4. If this verification holds, the server looks for a record $(0, *, P_i)$ in $\Lambda_{\mathcal{A}}$ and a record $((C_1, M_i, CAK_i, CID_i), (C_2, \ C_3), C_4)$ in Λ_Ψ. If such records exist, we abort the game.

The two games $\mathbf{G_5}$ and $\mathbf{G_4}$ are perfectly indistinguishable unless the event $\mathbf{AskPara_5}$ occurs:

$$\left|\Pr[\text{Succ}_5] - \Pr[\text{Succ}_4]\right| \leq \Pr[\text{AskPara}_5] \quad (6)$$

To upper bound $\Pr[\text{AskPara}_5]$, the parameter k is assumed to be correctly computed by $\mathcal{A}$ in all the ensuing games.

Game $\mathbf{G_6}$: We define this game by aborting the game wherein the adversary may have computed the correct authenticator C_3 or C_4 (that is, by asking the corresponding hash query $\mathcal{H}_1$ or $\mathcal{H}_2$) and impersonate as a client or server. We reach this aim by modifying the way the participants process the queries. We use the rule as follows:

- **Rule U3$^{(6)}$** – Check if $(1, ID_i\|ID_S\|Y_1\|C_2\|k\|*, C_3) \in \Lambda_{\mathcal{A}}$. If it holds, we abort the game. Otherwise, the user goes on to compute C_4 and sk_U.
- **Rule S3$^{(6)}$** – computes $C_4^* = \mathcal{H}_2(ID_i\|ID_S\|Y_1\| \ C_2\|k\|K_S)$ and then checks if C_4^* equals the received value of C_4. If this verification holds, the server looks for a record $(0, *, P_i)$ or $(2, ID_i\|ID_S\|Y_1\|C_2\|k\|*, C_4)$ in $\Lambda_{\mathcal{A}}$, and a record $((C_1, CAK_i, CID_i), (C_2, \ C_3), C_4)$ in Λ_Ψ. If such records exist, we abort the game.

The two games $\mathbf{G_6}$ and $\mathbf{G_5}$ are perfectly indistinguishable unless event $\mathbf{AskAuth_6}$ occurs:

$$\left|\Pr[\text{Succ}_6] - \Pr[\text{Succ}_5]\right| \leq \Pr[\text{AskAuth}_6] \quad (7)$$
$$\left|\Pr[\text{AskPara}_6] - \Pr[\text{AskPara}_5]\right| \leq \Pr[\text{AskAuth}_6] \quad (8)$$

Game $\mathbf{G_7}$: In this game, we replace the random oracles $\mathcal{H}_i$ with the private oracles $\mathcal{H}_i'(i = 1, 2, 3)$:

$$C_3 = \mathcal{H}_1'(ID_i\|ID_S\|C_1\|C_2);$$
$$C_4 = \mathcal{H}_2'(ID_i\|ID_S\|C_1\|C_2);$$
$$sk_U = sk_S = \mathcal{H}_3'(ID_i\|ID_S\|C_1\|C_2)$$

As a result, the values of C_3, C_4, sk_U, sk_S are completely independent from k, K_U and K_S. $\mathbf{G_7}$ and $\mathbf{G_6}$ are indistinguishable unless the event $\mathbf{AskH_7}$ occurs:

$$\left|\Pr[\text{Succ}_7] - \Pr[\text{Succ}_6]\right| \leq \Pr[\text{AskH}_7] \quad (9)$$
$$\left|\Pr[\text{AskPara}_7] - \Pr[\text{AskPara}_6]\right| \leq \Pr[\text{AskH}_7] \quad (10)$$
$$\left|\Pr[\text{AskAuth}_7] - \Pr[\text{AskAuth}_6]\right| \leq \Pr[\text{AskH}_7] \quad (11)$$

Lemma 1: The probabilities of the events $\mathbf{Succ_7}$ and $\mathbf{AskPara_7}$ in this game can be up-bounded by:

$$\big|\Pr[\text{Succ}_7]\big| = \frac{1}{2} \qquad \big|\Pr[\text{AskPara}_7]\big| \leq C' \cdot q_{send}^{s'} + \frac{q_{send}}{2^{l_0}} \quad (12)$$

Proof. In the game $\mathbf{G_7}$, the session keys are computed with private hash oracle unknown to $\mathcal{A}$, and thus $\Pr[\text{Succ}_7] = \frac{1}{2}$.

Let us denote by $R(U)$ the set of (C_2, C_3) received by a client instance, and by $R(S)$ the set of C_4 used by a server instance. Since we have avoided the cases where $\mathcal{A}$ have been lucky in guessing k, $\mathcal{A}$ can correctly compute k with the help of either a Corrupt($I = U^i$, 1)-query or a Corrupt($I = U^i$, 2)-query, the probability of which is denoted by $\Pr[\text{AskPara}_7\text{WithCorr}_1]$ and $\Pr[\text{AskPara}_7\text{WithCorr}_2]$, respectively. As discussed in Sec. III of the main text, it is more desirable (realistic) to assume passwords to be Zipf-distributed [86] than to make the traditional, commonly used (yet unrealistic) assumption that passwords are uniformly distributed (see some notable literature [7], [87]–[90]). From an information theoretical point of view, since we have avoided collisions in the Game $\mathbf{G_2}$,

$$\begin{aligned}&\big|\Pr[\text{AskPara}_7\text{WithCorr}_1]\big| \\ &= \Pr_{pw}[\exists C_3 \in R(U), (1, ID_i \| ID_S \| Y_1 \| C_2 \| k \| *, C_3) \in \Lambda_{\mathcal{A}}] \\ &+ \Pr_{pw}[\exists C_4 \in R(S), (0, *, P_i) \in \Lambda_{\mathcal{A}}] \leq C' \cdot q_{send}^{s'} \quad (13)\end{aligned}$$

$$\Pr[\text{AskPara}_7\text{WithCorr}_2]\big| \leq \frac{q_{send}}{2^{l_0}} \quad (14)$$

Game $\mathbf{G_8}$: In this game, we simulate the executions using the random self-reducibility of the Diffie-Hellman problem [91], given one CDH instance (A, B). Note that, we do not need to know the values of θ and φ, since the values of K_U and K_S are no longer needed to compute the authenticators or session keys:

- **Rule $\mathbf{U1^{(8)}}$** – chooses a random number $\alpha \in Z_p^*$ and computes C_1=A^α mod p, Y_1=y^α mod p, $k = \mathcal{H}_0(x \| ID_i \| T_{reg}) = N_i \oplus \mathcal{H}_0(b \| PW_i)$, $CID_i = ID_i \oplus \mathcal{H}_0(C_1 \| Y_1)$, $CAK_i = (a_i \| k) \oplus \mathcal{H}_0(Y_1 \| C_1)$ and $M_i = \mathcal{H}_0(Y_1 \| k \| CID_i)$. Also add the record (C_1, α) in Λ_A.
- **Rule $\mathbf{S2^{(8)}}$** – chooses a random number $\beta \in Z_p^*$, and computes $C_2 = B^\beta$ and $C_3 = \mathcal{H}_1'(ID_i \| ID_S \| C_1 \| C_2)$. Also adds the record (C_2, β) in Λ_B.

$$\Pr[\text{AskH}_7] = \Pr[\text{AskH}_8] \quad (15)$$

Remember that $\mathbf{AskH_8}$ means that the adversary $\mathcal{A}$ had queried the random oracles $\mathcal{H}_i (i = 1, 2, 3)$ on $(ID_i \| ID_S \| Y_1 \| C_2 \| * \| CDH(C_1, C_2))$. By picking randomly in the $\Lambda_{\mathcal{A}}$-list we can get the Diffie-Hellman secret value with probability $\frac{1}{q_h}$. This is a triple $(C_1, C_2, \text{CDH}(C_1, C_2))$. We can then simply look in the lists Λ_A and Λ_B to find the values α and β such that $C_1 = A^\alpha$ and $C_2 = B^\beta$:

$$\text{CDH}(C_1, C_2) = \text{CDH}(A^\alpha, B^\beta) = \text{CDH}(A, B)^{\alpha\beta}$$

and thus:

$$\big|\Pr[\text{AskH}_8]\big| \leq q_h \text{Adv}_{\mathcal{P}}^{\text{CDH}}(t') \quad (16)$$

where $t' \leq t + (q_{send} + q_{exe} + 1) \cdot \tau_e$.

Conclusion of the proof: By combining above equations, one gets the announced result. Firstly, from Eqs.(1)-(5) we get:

$$\big|\Pr[\text{Succ}_4] - \Pr[\text{Succ}_0]\big| \leq \frac{(q_{send}+q_{exe})^2}{2p} + \frac{q_h^2}{2^{l+1}} + \frac{q_{send}}{2^l}.$$

Secondly, from Eqs.(6)-(9) we get:

$$\begin{aligned}\big|\Pr[\text{Succ}_7] - \Pr[\text{Succ}_4]\big| &\leq \Pr[\text{AskPara}_5] + \Pr[\text{AskAuth}_6] \\ &+ \Pr[\text{AskH}_7].\end{aligned}$$

Thirdly, from the definition we know:

$$\Pr[\text{AskAuth}_6] \leq \Pr[\text{AskH}_6].$$

Finally, based on Eqs.(10)-(16) we get:

$$\begin{aligned}\text{Adv}_{\mathcal{P}}^{\text{ake}}(\mathcal{A}) &= 2\Pr[\text{Succ}_7] - 1 + 2(\Pr[\text{Succ}_0] - \Pr[\text{Succ}_7]) \\ &\leq C' \cdot q_{send}^{s'} + 12 q_h \text{Adv}_{\mathcal{P}}^{\text{CDH}}(t') \\ &+ \frac{q_h^2 + 6q_{send}}{2^l} + \frac{(q_{send} + q_{exe})^2}{p},\end{aligned}$$

where we use the Zipf model of Tianya in [86], where C'= 0.062239 and s'=0.155478; $n_0 = 2^8$; $t' \leq t + (q_{send} + q_{exe} + 1) \cdot \tau_e$ and $l = min\{l_i\}, i = 0, 1, 2, 3$. □

B. Proof of Theorem 2

Proof. The proof is similar to that of Theorem 1.

Firstly, we define an additional event:

- $\mathbf{Auth_n}$ occurs if $\mathcal{A}$ correctly guesses the authenticator C_3 or C_4 that will be accepted by the corresponding party and that has been built by the adversary herself in game $\mathbf{G_n}, n = 0, 1, \ldots, 7$.

Thus, we define

$$\text{Adv}_{\mathcal{P}}^{\text{auth}}(\mathcal{A}) = \Pr[\text{Auth}_0]$$

Secondly, we use the same sequence of games presented in the previous section, and extend Eqs. (1)-(7) to obtain:

$$\begin{aligned}&\big|\Pr[\text{Auth}_1] - \Pr[\text{Auth}_0]\big| = 0 \\ &\big|\Pr[\text{Auth}_2] - \Pr[\text{Auth}_1]\big| \leq \frac{(q_{send} + q_{exe})^2}{2p} + \frac{q_h^2}{2^{l+1}} \\ &\big|\Pr[\text{Auth}_3] - \Pr[\text{Auth}_2]\big| \leq \frac{q_{send}}{2^l} \\ &\big|\Pr[\text{Auth}_4] - \Pr[\text{Auth}_3]\big| \leq \frac{q_{send}}{2^l} \\ &\big|\Pr[\text{Auth}_5] - \Pr[\text{Auth}_4]\big| \leq \Pr[\text{AskPara}_5] \\ &\big|\Pr[\text{Auth}_6] - \Pr[\text{Auth}_5]\big| \leq \Pr[\text{AskAuth}_6] \leq 2\Pr[\text{AskH}_7] \\ &\Pr[\text{Auth}_7] = \Pr[\text{Auth}_6] = 0\end{aligned}$$

Thus, we have

$$\begin{aligned}\text{Adv}_{\mathcal{P}}^{\text{auth}}(\mathcal{A}) \leq C' \cdot q_{send}^{s'} + 5 q_h \text{Adv}_{\mathcal{P}}^{\text{CDH}}(t + (q_{send} + q_{exe} \\ +1) \cdot \tau_e) + \frac{q_h^2 + 6q_{send}}{2^{l+1}} + \frac{(q_{send} + q_{exe})^2}{2p}.\end{aligned}$$ □

REFERENCES FOR APPENDIX

[1] F. T. B. Muhaya, "Cryptanalysis and security enhancement of zhu's authentication scheme for telecare medicine information system," *Secur. Commun. Netw.*, vol. 8, no. 2, pp. 149–158, 2015.

[2] L. Xu and F. Wu, "An improved and provable remote user authentication scheme based on elliptic curve cryptosystem with user anonymity," *Secur. Commun. Netw.*, vol. 8, no. 2, pp. 245–260, 2015.

[3] D. Mishra, A. K. Das, A. Chaturvedi, and S. Mukhopadhyay, "A secure password-based authentication and key agreement scheme using smart cards," *J. Inform. Secur. Appl.*, vol. 23, pp. 28–43, 2015.

[4] V. Odelu, A. K. Das, and A. Goswami, "An effective and robust secure remote user authenticated key agreement scheme using smart cards in wireless communication systems," *Wirel. Pers. Commun.*, vol. 84, no. 4, pp. 2571–2598, 2015.

[5] J. W. Byun, "Privacy preserving smartcard-based authentication system with provable security," *Securi. Commun. Netw.*, vol. 8, no. 17, pp. 3028–3044, 2015.

[6] D. Wang, N. Wang, P. Wang, and S. Qing, "Preserving privacy for free: Efficient and provably secure two-factor authentication scheme with user anonymity," *Info. Sci.*, vol. 321, pp. 162–178, 2015.

[7] G. M. Yang, D. S. Wong, H. X. Wang, and X. T. Deng, "Two-factor mutual authentication based on smart cards and passwords," *J. Comput. Syst. Sci.*, vol. 74, no. 7, pp. 1160–1172, 2008.

[8] S. H. Wu, Y. F. Zhu, and Q. Pu, "Robust smart-cards-based user authentication scheme with user anonymity," *Secur. Commun. Netw.*, vol. 5, no. 2, pp. 236–248, 2012.

[9] I. Liao, C. Lee, and M. Hwang, "A password authentication scheme over insecure networks," *J. Comput. Syst. Sci.*, vol. 72, no. 4, pp. 727–740, 2006.

[10] R. Madhusudhan and R. Mittal, "Dynamic id-based remote user password authentication schemes using smart cards: A review," *J. Netw. Comput. Appl.*, vol. 35, no. 4, pp. 1235–1248, 2012.

[11] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1086–1090, 2009.

[12] P. Gope and T. Hwang, "Lightweight and energy-efficient mutual authentication and key agreement scheme with user anonymity for secure communication in global mobility networks," *IEEE Syst. J.*, 2015, doi:10.1109/JSYST.2015.2416396.

[13] Y. Lu, L. Li, H. Peng, and Y. Yang, "Robust anonymous two-factor authenticated key agreement schemem for mobile client-server environment," *Secur. Commun. Netw.*, 2016, doi: 10.1002/sec.1419.

[14] Q. Xie, N. Dong, D. S. Wong, and B. Hu, "Cryptanalysis and security enhancement of a robust two-factor authentication and key agreement protocol," *Int. J. Commun. Syst.*, vol. 29, no. 3, pp. 478–487, 2016.

[15] S. Islam, "Design and analysis of an improved smartcard-based remote user password authentication scheme," *Int. J. Commun. Syst.*, vol. 29, no. 11, pp. 1708–1719, 2016.

[16] T.-T. Truong, M.-T. Tran, A.-D. Duong, and I. Echizen, "Chaotic chebyshev polynomials based remote user authentication scheme in client-server environment," in *Proc. SEC 2015*, pp. 479–494.

[17] B. Djellali, K. Belarbi, A. Chouarfia, and P. Lorenz, "User authentication scheme preserving anonymity for ubiquitous devices," *Secur. Commun. Netw.*, vol. 8, no. 17, pp. 3131–3141, 2015.

[18] D. Mishra, A. K. Das, A. Chaturvedi, and S. Mukhopadhyay, "A secure password-based authentication and key agreement scheme using smart cards," *J. Inform. Secur. Appl.*, vol. 23, pp. 28–43, 2015.

[19] F. Wu, L. Xu, S. Kumari, X. Li, and A. Alelaiwi, "A new authenticated key agreement scheme based on smart cards providing user anonymity with formal proof," *Secur. Commun. Netw.*, 2015, doi: 10.1002/sec.1305.

[20] S. A. Chaudhry, M. S. Farash, H. Naqvi, S. Kumari, and M. K. Khan, "An enhanced privacy preserving remote user authentication scheme with provable security," *Securi. Commun. Netw.*, vol. 8, no. 18, pp. 3782–3795, 2015.

[21] S. Kumari, M. K. Khan, and X. Li, "An improved remote user authentication scheme with key agreement," *Comput. & Electr. Eng.*, vol. 40, no. 6, pp. 97–112, 2014.

[22] B. Chen and W. Kuo, "Robust smart-card-based remote user password authentication scheme," *Int. J. Commun. Syst.*, vol. 27, no. 2, pp. 377–389, 2014.

[23] J.-L. Tsai, N.-W. Lo, and T.-C. Wu, "Novel anonymous authentication scheme using smart cards," *IEEE Trans. Ind. Inform.*, vol. 9, no. 4, pp. 2004–2013, 2013.

[24] X. Li, J. Niu, M. K. Khan, and J. Liao, "An enhanced smart card based remote user password authentication scheme," *J. Netw. Comput. Appl.*, vol. 36, no. 5, pp. 1365–1371, 2013.

[25] S. Kumari and M. K. Khan, "Cryptanalysis and improvement of 'a robust smart-card-based remote user password authentication scheme'," *Int. J. Commun. Syst.*, vol. 27, no. 12, pp. 3939–3955, 2014.

[26] D.-Z. Sun and Z.-F. Cao, "On the privacy of khan et al.'s dynamic id-based remote authentication scheme with user anonymity," *Cryptologia*, vol. 37, no. 4, pp. 345–355, 2013.

[27] T.-F. Lee and C.-M. Liu, "A secure smart-card based authentication and key agreement scheme for telecare medicine information systems," *J. Med. Syst.*, vol. 37, no. 3, 2013.

[28] S. H. Islam and G. Biswas, "Design of improved password authentication and update scheme based on elliptic curve cryptography," *Math. Comput. Model.*, vol. 57, no. 11-12, pp. 2703–2717, 2013.

[29] X. Li and Y. Zhang, "A simple and robust anonymous two-factor authenticated key exchange protocol," *Secur. Commun. Netw.*, vol. 6, no. 6, pp. 711–722, 2013.

[30] Y.-F. Chang, W.-L. Tai, and H.-C. Chang, "Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update," *Int. J. Commun. Syst.*, vol. 27, no. 11, pp. 3430–3440, 2014.

[31] C.-T. Li, "A new password authentication and user anonymity scheme based on elliptic curve cryptography and smart card," *IET Inform. Secur.*, vol. 7, no. 1, pp. 3–10, 2013.

[32] K.-K. Kim and M.-H. Kim, "An enhanced anonymous authentication and key exchange scheme using smartcard," in *Proc. ICISC 2012*, ser. LNCS, A. Juels and C. Paar, Eds. Springer, vol. 7839, pp. 487–494,.

[33] R. Ramasamy and A. P. Muniyandi, "An efficient password authentication scheme for smart card." *Int. J. Netw. Secur.*, vol. 14, no. 3, pp. 180–186, 2012.

[34] Z. Zhu, "An efficient authentication scheme for telecare medicine information systems," *J. Med. Syst.*, vol. 36, no. 6, pp. 3833–3838, 2012.

[35] D. Wang and C. G. Ma, "Cryptanalysis and security enhancement of a remote user authentication scheme using smart cards," *Elsevier J. China Uni. Posts Telecommun.*, vol. 19, no. 5, pp. 104–114, 2012.

[36] W. Hsieh and J. Leu, "Exploiting hash functions to intensify the remote user authentication scheme," *Comput. Secur.*, vol. 31, no. 6, pp. 791–798, 2012.

[37] F. Wen and X. Li, "An improved dynamic id-based remote user authentication with key agreement scheme," *Comput. & Electr. Eng.*, vol. 38, no. 2, pp. 381–387, 2012.

[38] D. Wang, C. G. Ma, and P. Wu, "Secure password-based remote user authentication scheme with non-tamper resistant smart cards," in *Proc. DBSec 2012*, ser. LNCS. Springer, 2012, vol. 7371, pp. 114–121.

[39] D. He, J. Chen, and J. Hu, "Improvement on a smart card based password authentication scheme," *J. Internet Tech.*, vol. 13, no. 3, pp. 38–42, 2012.

[40] R. C. Wang, W. S. Juang, and C. L. Lei, "Robust authentication and key agreement scheme preserving the privacy of secret key," *Comput. Commun.*, vol. 34, no. 3, pp. 274–280, 2011.

[41] T. Chen, H. Hsiang, and W. Shih, "Security enhancement on an improvement on two remote user authentication schemes using smart cards," *Future. Gener. Comput. Syst.*, vol. 27, no. 4, pp. 377–380, 2011.

[42] M. Khan, S. Kim, and K. Alghathbar, "Cryptanalysis and security enhancement of a more efficient & secure dynamic id-based remote user authentication scheme'," *Comput. Commun.*, vol. 34, no. 3, pp. 305–309, 2011.

[43] J. Kim, H. Choi, and J. Copeland, "Further improved remote user authentication scheme," *IEICE Trans. Fund. Electron. Comm. Comput. Sci.*, vol. 94, no. 6, pp. 1426–1433, 2011.

[44] A. K. Awasthi, K. Srivastava, and R. Mittal, "An improved timestamp-based remote user authentication scheme," *Comput. & Electr. Eng.*, vol. 37, no. 6, pp. 869–874, 2011.

[45] C. T. Li and C. Lee, "A robust remote user authentication scheme using smart card," *Info. Tech. Control*, vol. 40, no. 3, pp. 236–245, 2011.

[46] S. K. Sood, "Secure dynamic identity-based authentication scheme using smart cards," *Inform. Secur. J.*, vol. 20, no. 2, pp. 67–77, 2011.

[47] X. Li, W. Qiu, D. Zheng, K. F. Chen, and J. Li, "Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards," *IEEE Trans. Ind. Elec.*, vol. 57, no. 2, pp. 793–800, 2010.

[48] J. Tsai, T. Wu, and K. Tsai, "New dynamic id authentication scheme using smart cards," *Int. J. Commun. Syst.*, vol. 23, no. 12, pp. 1449–1462, 2010.

[49] R. Song, "Advanced smart card based password authentication protocol," *Comput. Stand. & Inter.*, vol. 32, no. 5, pp. 321–325, 2010.

[50] S. Sood and K. Sarje, K.and Singh, "An improvement of xu et al.'s authentication scheme using smart cards," in *Proc. Compute 2010*. ACM, Jan. 2010, pp. 1–5.

[51] K. H. Yeh, C. Su, and N. W. Lo, "Two robust remote user authentication protocols using smart cards," *J. Syst. Soft.*, vol. 83, no. 12, pp. 2556–2565, 2010.

[52] W. Horng, C. Lee, and J. Peng, "A secure remote authentication scheme preserving user anonymity with non-tamper resistant smart cards," *WSEAS Trans. Inform. Sci. Appl.*, vol. 7, no. 5, pp. 619–628, 2010.

[53] D. Sun, J. Huai, J. Sun, J. Li, and Z. Feng, "Improvements of juang et al.'s password-authenticated key agreement scheme using smart cards," *IEEE Trans. Ind. Elec.*, vol. 56, no. 6, pp. 2284–2291, 2009.

[54] H. C. Hsiang and W. K. Shih, "Weaknesses and improvements of the yoon-ryu-yoo remote user authentication scheme using smart cards," *Comput. Commun.*, vol. 32, no. 4, pp. 649–652, 2009.

[55] J. Xu, W. Zhu, and D. Feng, "An improved smart card based password authentication scheme with provable security," *Comput. Stand. & Inter.*, vol. 31, no. 4, pp. 723–728, 2009.

[56] S.-K. Kim and M. G. Chung, "More secure remote user authentication scheme," *Comput. Commun.*, vol. 32, no. 6, pp. 1018–1021, 2009.

[57] H. Chung, W. Ku, and M. Tsaur, "Weaknesses and improvement of wang et al.'s remote user password authentication scheme for resource-limited environments," *Comput. Stand. & Inter.*, vol. 31, no. 4, pp. 863–868, 2009.

[58] R. Ramasamy and A. P. Muniyandi, "New remote mutual authentication scheme using smart cards," *Trans. Data Privacy*, vol. 2, pp. 141–152, 2009.

[59] W.-S. Juang, S.-T. Chen, and H.-T. Liaw, "Robust and efficient password-authenticated key agreement using smart cards," *IEEE Trans. Ind. Elec.*, vol. 55, no. 6, pp. 2551–2556, 2008.

[60] T.-H. Chen and W.-B. Lee, "A new method for using hash functions to solve remote user authentication," *Comput. & Electr. Eng.*, vol. 34, no. 1, pp. 53–62, 2008.

[61] X. Wang, W. Zhang, J. Zhang, and M. Khan, "Cryptanalysis and improvement on two efficient remote user authentication scheme using smart cards," *Comput. Stand. & Inter.*, vol. 29, no. 5, pp. 507–512, 2007.

[62] R.-C. Wang, W.-S. Juang, and C.-L. Lei, "A simple and efficient key exchange scheme against the smart card loss problem," in *IEEE/IFIP EUC 2007*, ser. LNCS, vol. 4809, pp. 728–744.

[63] S. Lee, H. Kim, and K. Yoo, "Improvement of chien et al.'s remote user authentication scheme using smart cards," *Comput. Stand. & Inter.*, vol. 27, no. 2, pp. 181–183, 2005.

[64] C. Fan, Y. Chan, and Z. Zhang, "Robust remote authentication scheme with smart cards," *Comput. Secur.*, vol. 24, no. 8, pp. 619–628, 2005.

[65] R. Lu and Z. Cao, "Efficient remote user authentication scheme using smart card," *Comput. Netw.*, vol. 49, no. 4, pp. 535–540, 2005.

[66] E. Yoon, E. Ryu, and K. Yoo, "Further improvement of an efficient password based remote user authentication scheme using smart cards," *IEEE Trans. Comsum. Electr.*, vol. 50, no. 2, pp. 612–614, 2004.

[67] W.-C. Ku and S.-M. Chen, "Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards," *IEEE Trans. Comsum. Electr.*, vol. 50, no. 1, pp. 204–207, 2004.

[68] S.-T. Wu and B.-C. Chieu, "A user friendly remote authentication scheme with smart cards," *Comput. Secur.*, vol. 22, no. 6, pp. 547–550, 2003.

[69] A. Awasthi and S. Lal, "A remote user authentication scheme using smart cards with forward secrecy," *IEEE Trans. Comsum. Electr.*, vol. 49, no. 4, pp. 1246–1248, 2003.

[70] H.-M. Sun, "An efficient remote use authentication scheme using smart cards," *IEEE Trans. Consum. Electr.*, vol. 46, no. 4, pp. 958–961, 2000.

[71] M.-S. Hwang and L.-H. Li, "A new remote user authentication scheme using smart cards," *IEEE Trans. Consum. Electr.*, vol. 46, no. 1, pp. 28–30, 2000.

[72] D. Wang, D. He, P. Wang, and C.-H. Chu, "Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment," *IEEE Trans. Depend. Secur. Comput.*, vol. 12, no. 4, pp. 428–442, 2015.

[73] D. Wang and P. Wang, "Offline dictionary attack on password authentication schemes using smart cards," in *Proc. ISC 2013*, ser. LNCS, Y. Desmedt, Ed., vol. 7807, pp. 221–237.

[74] C.-G. Ma, D. Wang, and S.-D. Zhao, "Security flaws in two improved remote user authentication schemes using smart cards," *Int. J. Commun. Syst.*, vol. 27, no. 10, pp. 2215–2227, 2014.

[75] D. Wang, C. G. Ma, S. Zhao, and C. Zhou, "Cryptanalysis of two dynamic id-based remote user authentication schemes for multi-server architecture," in *Proc. NSS 2012*, ser. LNCS, vol. 7645, pp. 462–475.

[76] D. Wang, P. Wang, and J. Liu, "Improved privacy-preserving authentication scheme for roaming service in mobile networks," in *Proc. IEEE WCNC 2014*, pp. 3178–3183.

[77] D. Wang and C. Ma, "Cryptanalysis of a remote user authentication scheme for mobile client-server environment based on ECC," *Inform. Fusion,*, vol. 14, no. 4, pp. 498–503, 2013.

[78] D. Wang and P. Wang, "On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions," *Comput. Netw.*, vol. 73, pp. 41–57, 2014.

[79] ——, "Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks," *Ad Hoc Netw.*, vol. 20, pp. 1–15, 2014.

[80] K. Xue, P. Hong, and C. Ma, "A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture," *J. Comput. Syst. Sci.*, vol. 80, no. 1, pp. 195–206, 2014.

[81] V. Odelu, A. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Trans. Inform. Foren. Secur.*, vol. 10, no. 9, pp. 1953–1966, 2015.

[82] J. Yu, G. Wang, Y. Mu, and W. Gao, "An efficient generic framework for three-factor authentication with provably secure instantiation," *IEEE Trans. Inform. Foren. Secur.*, vol. 9, no. 12, pp. 2302–2313, 2014.

[83] J. Bonneau, M. Just, and G. Matthews, "Whats in a name?" in *Proc. FC 2010*, ser. LNCS, R. Sion, Ed. Springer, 2010, vol. 6052, pp. 98–113.

[84] J. Bonneau, "The science of guessing: Analyzing an anonymized corpus of 70 million passwords," in *Proc. IEEE S&P 2012*, pp. 538–552.

[85] "Amazon elastic compute cloud (Amazon EC2)," Amazon Web Services Inc., 2015, https://aws.amazon.com/ec2/pricing/.

[86] D. Wang and P. Wang, "On the implications of Zipf's law in passwords," in *Proc. ESORICS 2016*, ser. LNCS, vol. 9878. Springer, pp. 1–21.

[87] M. Abdalla, F. Benhamouda, and P. MacKenzie, "Security of the j-pake password-authenticated key exchange protocol," in *Proc. IEEE S&P 2015*. IEEE Computer Society, 2015, pp. 571–587.

[88] J. Katz, R. Ostrovsky, and M. Yung, "Efficient and secure authenticated key exchange using weak passwords," *J. ACM*, vol. 57, no. 1, pp. 1–41, 2009.

[89] X. Yi, F. Hao, L. Chen, and J. Liu, "Practical threshold password-authenticated secret sharing protocol," in *Proc. ESORICS 2015*, ser. LNCS, vol. 9326, pp. 347–365.

[90] M. Shirvanian, S. Jarecki, N. Saxena, and N. Nathan, "Two-factor authentication resilient to server compromise using mix-bandwidth devices," in *Proc. NDSS 2014*. The Internet Society, pp. 1–16.

[91] N. Fazio, R. Gennaro, I. M. Perera, and W. E. Skeith III, "Hard-core predicates for a diffie-hellman problem over finite fields," in *Proc. CRYPTO 2013*, pp. 148–165.