SECURE HUMAN COMMUNICATIONS BASED ON BIOMETRICS SIGNALS

Yongdong Wu[1], Feng Bao[1] and Robert H. Deng[2]
[1]*Instititue for Infocomm Research* (**I²R**), *Singapore, {wydong,baofeng}@i2r.a-star.edu.sg;*
[2]*School of information Systems, Singapore Management Univeristy, robertdeng@smu.edu.sg*

Abstract: User authentication is the first and probably the most challenging step in achieving secure person-to-person communications. Most of the existing authentication schemes require communicating parties either share a secret/password or know each other's public key. In this paper we suggest a novel user authentication scheme that is easy to use and overcomes the requirements of sharing password or public keys. Our scheme allows two human users to perform mutual authentication and have secure communications over an open channel by exchanging biometrics signals (e. g., voice or video signals). In addition to user authentication, our scheme establishes a secret session key between two users by cryptographically binding biometrics signals with users's Diffie-Hellman public values. Under the assumption that the two communicating persons are familiar with each other's biometrics signals, we show that the scheme is secure against various attacks, including the man-in-the-middle attack. The proposed scheme is highly suitable for applications such as Voice-over-IP.

1. INTRODUCTION

The explosive growth of computer systems and their applications has considerably increased the dependence of both organizations and individuals on the information communicated using the Internet. However, the Internet is an interconnection of open public networks. Without security measures, communications over the Internet, such as Voice-over-IP (VOIP) and video conferences, can be eavesdropped without much difficulty. This in turn has led to a heightened effort to protect data from disclosure and to guarantee the

integrity of data and messages communicated over open networks. User authentication is the first and probably the most challenging step in achieving secure communications in the Internet.

To date, the most pervasive user authentication schemes are based on cryptographic techniques which require that the parties either share a secret key (e.g., a password)[1] or know each other's public key[2]. Although password based authentication protocols are widely used, there are many potential difficulties for a human user to share passwords with a large number of remote users. First of all, establishing a shared password between two users requires a secure secret distribution mechanism to be in place. This is very challenging. Second and more importantly, human users are not good at remember passwords of good quality, not to mention remembering multiple passwords shared with many remote users. Public key based authentication protocols require users to know each other's public key in authenticated manners in the form of public key certificates. This turn requires the existence of a public key infrastructure in the Internet, an impossible task at least in the near to medium terms[3].

In this paper our focus is on human user authentication in person-to-person communications in an open environment such as the Internet. In this case, it is much more convenient and natural for human users to authenticate each other using biometrics techniques.

Most of the existing research on biometrics based user authentication techniques allows a human user to authenticate himself or herself to a local machine. Little effort has been spent to study biometrics based methods which perform authentication between two remote human users. To our knowledge, the only work related to our effort is the Pretty Good Privacy Phone or PGPfone[4]. PGPfone implements an authentication protocol based on the exchange of voice signals. However, PGPfone is vulnerable to replay attack. If an attacker is able to collect sound samples of all the 256 octets by, for example, eavesdropping on someone's phone calls, the attacker is able to impersonate the victim at will.

As in PGPfone, our scheme requires that communicating users be able to identify each other based on the other party's biometrics signals (such as acoustic waves or face expression). Based on the exchange of biometrics signals, the proposed scheme not only authenticates remote human users but also enables them to have secure communications over open channels. Specifically, to achieve authentication and agreement of a secret session key, the Diffie-Hellman public key values are cryptographically committed or bound with biometrics signals such that the trust on the biometric information is extended to the Diffie-Hellman public values. The trusted Diffie-Hellman public values are then used to perform the Diffie-Hellman Key Exchange Protocol so as to defeat the man-in-the-middle attack. Since

our scheme does not require users to share any password or know each other's public key in advance, it is attractive for applications such as secure VOIP or secure video conferences.

The reminder of the paper is organized as follows. Section 2 addresses the primaries for clarity. Section 3 elaborates the proposed scheme and its variant. Section 4 discusses the availability and security. Section 5 contains our concluding remarks.

2. PRILIMINARIES

2.1 Notations

A: shorthand notation for Alice (or her communication device) who initiates the communication unless stated otherwise. Preminatary

B: shorthand notation for Bob (or his communication device) who responses to Alice's communication request.

C: shorthand notation for Clark who tries to attack the communications between Alice and Bob.

C_X: a challenge biometrics signal. Without loss of generality, we will use voice signals as the representative biometrics signals throughout the paper. Thus, C_X is the acoustic wave or digital representation of a challenge statement spoken by user **X** (either **A** or **B**); whether it is the acoustic wave or the digital representation should be clear from the context of discussion.

R_Y: an acoustic wave or digital representation of a response statement spoken by user **Y** in reply to C_X.

$R_Y \sim C_X$: The response R_Y matches challenge C_X. For instance, the content of R_Y is the same/similar to that of C_X, or R_Y is a correct answer to C_X.

$|C_X|$: the time duration of C_X.

$|R_Y|$: the time duration of R_Y.

$e(K, m)$: encryption of message m with a symmetric key cryptosystem (e. g., AES) using a secret key K.

$d(K, c)$: decryption of a ciphertext c with a symmetric key cryptosystem using a secret key K.

$h(\cdot)$: a one-way hash function (e.g., SHA-1).

T: the required minimum time duration (e. g., 10 seconds) of any statement spoken by a user.

δ: a threshold value which is much less than T, (e.g. $\delta=0.1T$). The value of δ (or equivalently that of T) plays an important role in deciding the security strength of the protocol (refer to Eq.(1)).

To keep our notation compact, only residue modulo is shown in the following. That is, we will write $g^x \bmod p$, $g^y \bmod p$ and $g^{xy} \bmod p$ simply as g^x, g^y and g^{xy} respectively, where p is a predefined large prime.

2.2 System Architecture

The system architect for person-to-person communications between two remote users, Alice and Bob, is depicted in Figure 1. We assume that Alice and Bob are aware that they will have an authenticated and confidential communication session and Alice will start the present secure protocol. This awareness assumption can be satisfied easily via any non-secure channel. The transmission channel includes but is not limited to any communication systems or media such as computer networks, public telephone switching networks and radio links.

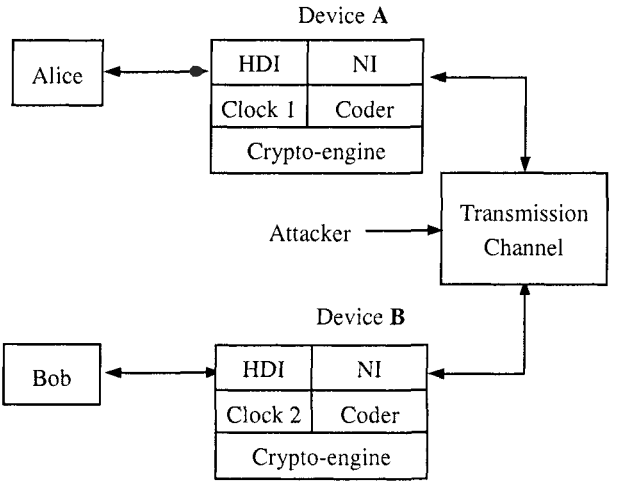

Figure 1. The communication system architecture.

Alice and Bob communicate with each other by interfacing with Device **A** and Device **B**, respectively. Device **A** (or Device **B**) accepts audio input from Alice (Bob) and outputs Bob's (Alice's) audio signal to Alice (Bob). The signals are sent and received via the Network Interface (NI). Each device has a clock for timing purpose, a coder performing audio encoding/decoding operations, and a crypto-engine executing the Diffie-Hellman and symmetric key cryptosystem operations. We assume that the Diffie-Hellman parameters, g and p, are negotiated on-line or hard coded in the software. Without loss of generality, Alice is assumed to be the initiator and Bob is the responder of a communication session.

2.3 Assumptions

The attacker Clark sits in the middle of the channel between Alice and Bob. He is able to perform both passive (eavesdropping) and active (message tampering, delay, replay). He may know biometrics data of Alice and Bob recorded from their past conversations. Clark may have much more powerful resources (e.g. super-computers and large storage devices) than Alice and Bob. The only restriction is that Clark is not able to mimic the natural speech of Alice or Bob in real time.

Alice and Bob neither share any secret data (e.g., password) nor have each other's public key. In order to achieve user authentication, we make the following assumptions:

S1: Alice and Bob are familiar with each other's voice (biometrics characteristics in general) and able to recognize each other by listening each other's speech. This assumption is reasonable and practical since there are generally no confidential topics between two strangers unless there is the involvement of a trusted third party.

S2: It is difficult for a human being to mimic the dynamic biometrics features of others in real time without being detected.

S3 It is difficult for a machine to mimic the dynamic biometrics features of a human being without being detected. Text-To-Speech (TTS) technology targets for creation of audible speech from computer readable text. A high quality TTS has to select text units from large speech databases in an optimum way[5]. To make use of TTS, an attacker needs to organize a database of large samples. On the other hand, although speech syntheses technology has made significant advancement in minimizing audible signal discontinuities between two successive concatenated units, and prosodic variation, it is still not satisfactory to mimic natural speech[6]. For example, in the TTS demo[7] of Microsoft Research, the speech is not nature although each word or short phrase is pronounced accurately, such that it is easy to distinguish the voice of a machine from that of a natural human. Similarly, the concatenation artifacts of TTS from AT&T[8] can be detected easily. In other words, presently, synthesized speech is still distinguishable from human speech after many years of research and development.

S4: Each participant can speak fresh sentences whose durations are sufficient long (e.g. at least T).

S5: The *RTT* (round-trip-time) of the communication channel can be estimated (e.g., command `ping www.yahoo.com`). It is required that $RTT \ll T$. This requirement must be met in order for the conversations between the communicating parties to be audible.

3. AUTHENTICATION PROTOCOLS BASED ON BIOMETRICS SIGNALS

In this section, we present authentication protocols based on the exchange of users' biometrics signals. The protocols are designed to perform mutual authentication between two parties called Alice and Bob and at the same time allow them to share a secret session key for securing their subsequent communications.

3.1 Basic Idea

To start a secure communication session with our proposed scheme, Alice initiates the session by speaking a challenge statement, such as

"This is Alice! The time is 21 minutes passed 9am. How was your mid-term examination, Bob? "

Bob receives and listens to Alice's challenge, and makes sure that the message is indeed spoken by Alice. He then speakes a response statement, such as

"Hi, Alice! Bob's here. My mid-term exam was not very good. But thank God, it was over! ".

Upon hearing Bob's response, Alice decides whether the response is spoken by Bob and whether it is related to her challenge. If the answer is positive, Alice authenticates Bob. Bob can authenticate Alice in the same way.

In order to establish a secret session key during the above authentication process, we incorporate the Diffie-Hellman key exchange into our scheme. By cryptographically binding biometric signals with Diffie-Hellman public values, the proposed scheme is protected against the man-in-the-middle attack. The above conceptual description seems very simple, the scheme is more complicated. To demonstrate the above concept, we present two protocols, a sequential protocol and a parallel protocol in the following.

3.2 A Sequential Protocol

The authentication protocol consists of three phases: Authentication of Bob, Authentication of Alice. Additionally, a Key Confirmation will be executed so as to guarantee that both share the same session key.

3.2.1 Authentication of Bob

This phase, shown in Figure 2, allows Alice to authenticate Bob and

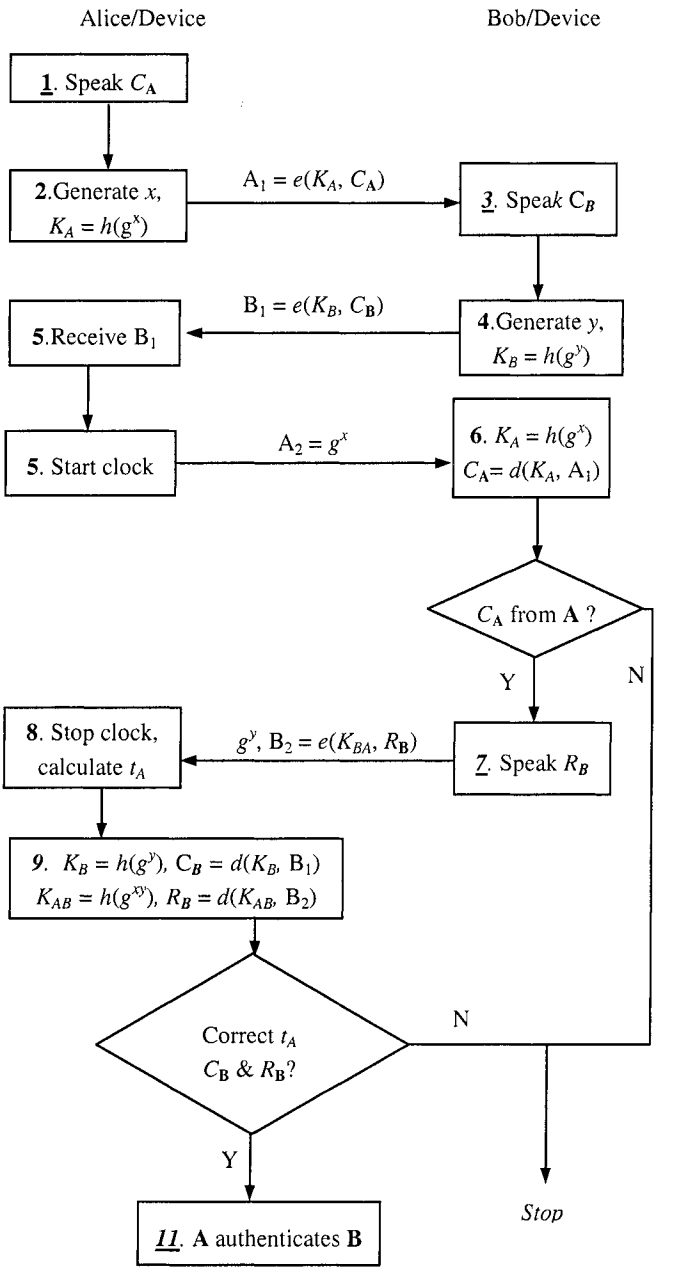

Figure 2. Authentication of Bob. An underlined step is performed by Alice or Bob, while other steps are executed by devices. $R_B \sim C_A$ means that the reply R_B matches the challenge C_A. For instance, the content of R_B is the same/similar to that of C_A, or RB is a correct answer to C_A.

obtain Bob's Diffie-Hellman public value g^y in an authenticated manner.

- Alice's Challenge. Here Alice sends a challenge statement to Bob. This biometrics signal is cryptographically bound to Alice's Diffle-Hellman public value.
  (1) Alice speaks a challenge statement $C_{\mathbf{A}}$ which is input to Device **A**. It is highly preferred that $C_{\mathbf{A}}$ contains some "freshness" elements such as the date and time, news headlines of the day.
  (2) Device **A** generates a random number x, computes g^x and a key $K_A=h(g^x)$. Next, Device **A** encrypts $C_{\mathbf{A}}$ using K_A with a symmetric key cryptosystem (e.g. AES) and sends the ciphertext $A_1=e(K_A, C_{\mathbf{A}})$ to Bob over the transmission channel.
- Bob's Commitment. In the next 2 steps, Bob sends commitment to Alice so that Alice discloses her challenge. Bob's commitment contains the encryption of his challenge statement which will be opened by Alice at a later stage.
  (3) Device **B** receives message A_1 and prompts Bob to speak a challenge statement $C_{\mathbf{B}}$.
  (4) Device **B** generates a random number y, computes g^y and a key $K_B=h(g^y)$, encrypts $C_{\mathbf{B}}$ using K_B with a symmetric key cryptosystem and transmits the ciphertext $B_1=e(K_B, C_{\mathbf{B}})$ to Alice.
- Bob's Response. The next 4 steps allow Bob to send his response statement to Alice.
  (5) Device **A** receives B_1, sends $A_2 = g^x$ to Device **B** and starts a clock.
  (6) Device **B** computes $K_A=h(g^x)$, recovers $C_{\mathbf{A}} =d(K_A, A_1)$, and computes a key $K_{BA} = h(g^{xy})$.
  (7) Device **B** plays back $C_{\mathbf{A}}$ to Bob who listens to it and verifies if the voice belongs to Alice. If the verification fails, Bob terminates the session; otherwise, Bob speaks a response statement $R_{\mathbf{B}}$ in reply to $C_{\mathbf{A}}$. Device **B** encrypts $R_{\mathbf{B}}$ with K_{BA} to obtain $B_2 =e(K_{BA}, R_{\mathbf{B}})$, and sends g^y and B_2 to Device **A**.
  (8) Upon receipt of message B_2, Device **A** stops the clock and obtains t_A the elapsed time of the clock.
- Alice's Verification. In the next 3 steps, Alice verifies the originality of the response and checks the elapsed time used in obtaining the response.
  (9) Device A computes $K_{AB} =h(g^{yx})$ and $K_B =h(g^y)$, and then she recovers $C_{\mathbf{B}} =d(K_B, B_1)$ and $R_B =d(K_{AB}, B_2)$, and computes $|C_{\mathbf{A}}|$ (the duration of $C_{\mathbf{A}}$) and $|R_{\mathbf{B}}|$ (the duration of $R_{\mathbf{B}}$).
  (10) Note that within the time t_A, Bob has to listen to $C_{\mathbf{A}}$ and speaks a response $R_{\mathbf{B}}$. Hence, $t_A \geq (|C_{\mathbf{A}}| + |R_{\mathbf{B}}| + \Delta_{\mathbf{B}})$, where $\Delta_{\mathbf{B}}$ is the delay due to transmitting messages A_2 and B_2, and processing time introduced by Device **B** in steps (6) and (7). $\Delta_{\mathbf{B}}$ can be estimated by device A. Then, Device **A** calculates

$$\delta_A = t_A - (|C_A| + |R_B| + \Delta_B) \quad (1)$$

If $\delta_A > \delta$, Device A terminates the session; otherwise, Alice listens and verifies R_B. If Alice recognizes that either R_B is not in Bob's voice or R_B is not a reply to C_A, she stops the session.

(11)Alice concludes that g^y comes from Bob and authenticates Bob.

3.2.2 Authentication of Alice

To provide mutual authentication and key agreement, Bob will proceed to authenticate Alice and obtain Alice's Diffie-Hellman public value g^x in an authenticated manner. The process is similar to that given above with the exception that Bob is the initiator and Alice is the responder. Note that Bob's challenge statement C_B was sent to Alice in step (4). This is done intentionally so as to prevent the man-in-the-middle attack during the process of authenticating Alice.

After both Alice and Bob have obtained the each other's authenticated Diffie-Hellman public key values, they are confident that the agreed Diffie-Hellman key K_{AB} is shared only among them. After mutual authentication, Alice and Bob can confirm their shared key easily.

3.3 A Parallel Protocol

A careful reader might have noticed that certain steps in Figure 2 can be executed in parallel so as to speed up the protocol. Figure 3 depicts the flow chart of the parallel protocol which has the same phases as those of the sequential protocol.

- Challenges
  - (1) Alice starts the session by speaking a challenge statement C_A.
  - (2) Device **A** generates a random x, computes a key $K_A = h(g^x)$, encrypts C_A as $A_1 = e(K_A, C_A)$ and sends the ciphertext A_1 to Bob.
  - (3) After receiving message A_1, Bob speaks a challenge statement C_B.
  - (4) Device **B** generates a random number y, computes a key $K_B = h(g^y)$, encrypts C_B as $B_1 = e(K_B, C_B)$ and sends the ciphertext B_1 to Alice.
  - (5) After receiving B_1, Device **A** sends $A_2 = g^x$ to Bob and starts clock 1.
  - (6) Upon receipt of message A_2, Device **B** starts its clock 2 and sends message $B_2 = g^y$ to Alice.
- Responses
  - (7) After receiving message B_2, Device **A** computes $K_B = h(g^y)$, recovers Bob's challenge message $C_B = d(K_B, B_1)$. On the other hand, Device **B** computes $K_A = h(g^x)$, recovers Alice's challenge as $C_A = d(K_A, A_1)$.
  - (8) Alice listens C_B and stops the protocol if she believes that C_B is not in Bob's voice; Bob listens to C_A and terminates the protocol if he

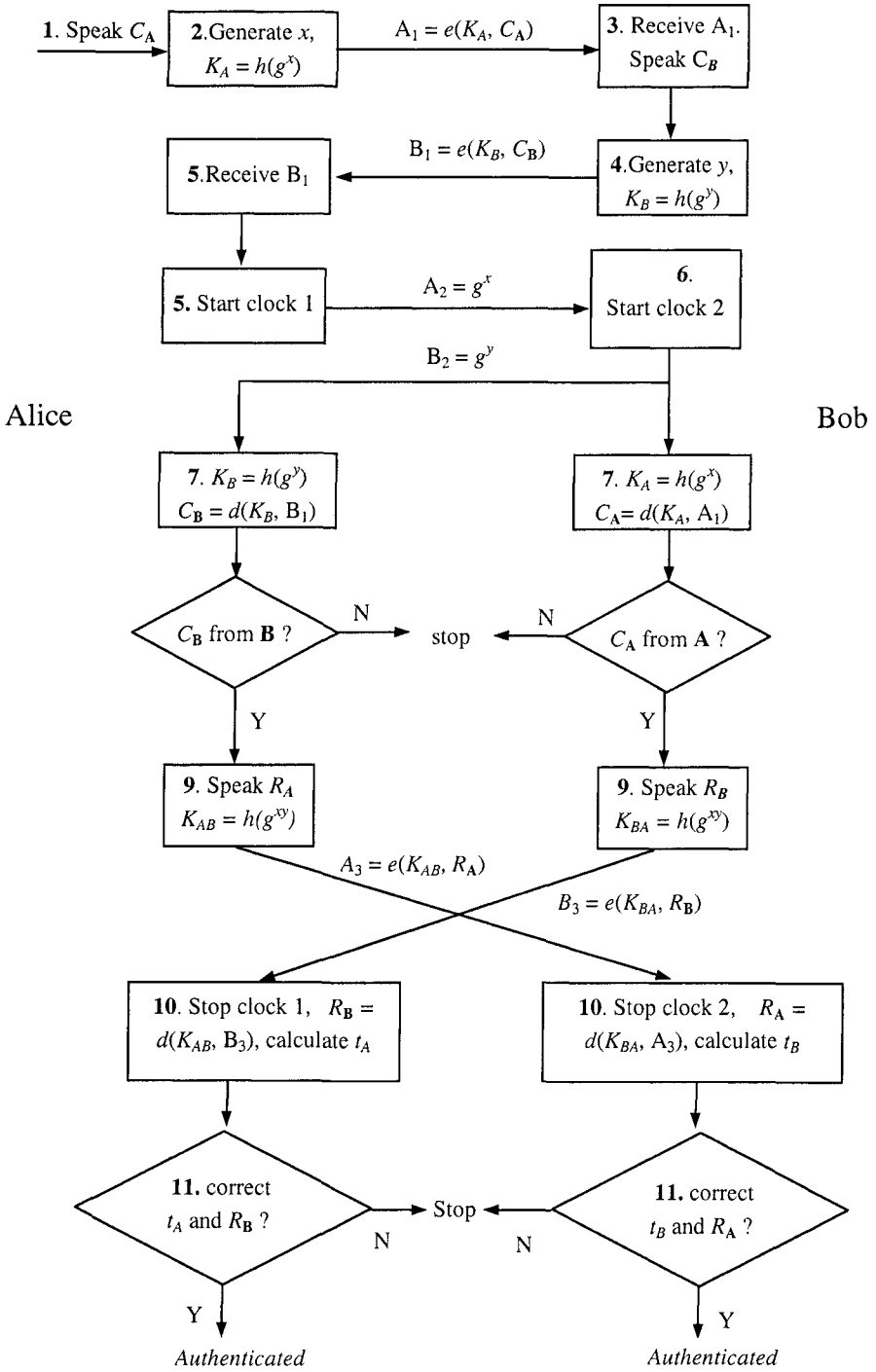

Figure 3. Parallel protocol.

doubts on the originality of C_A.

(9) Alice speaks a response R_A to the challenge C_B. Device **A** computes $K_{AB}=h(g^{yx})$, encrypts R_A and sends the ciphertext $A_3=e(K_{AB}, R_A)$ to Bob. Meanwhile, Bob speaks R_B in reply to C_A. Device **B** computes a key $K_{BA}=h(g^{xy})$, and sends the ciphertext $B_3=e(K_{BA}, R_B)$ to Alice.

(10) After receiving message B_3, Device **A** stops clock 1, recovers Bob's response $R_B=d(K_{AB}, B_3)$ and calculates the elapsed time t_A. After receiving message A_3, Device **B** stops clock 2, recovers Alice's response $R_A=d(K_{BA}, A_3)$, and calculates the elapsed time t_B.

- Verifications

(11) Device **A** calculates δ_A as

$$\delta_A = t_A - (|C_A| + |R_B| + \Delta_B)$$

where Δ_B is the delay due to transmitting messages A_2 and B_3, and processing interval introduced by Device **B** in steps (7)-(9). Device **A** terminates the session if $\delta_A > \delta$.

Simultaneously, Device **B** calculates δ_B as

$$\delta_B = t_B - (|C_B| + |R_A| + \Delta_A),$$

where Δ_A is the delay due to transmitting messages B_2 and A_3, and processing interval introduced by Device **A** in steps (7)-(9). Device **B** terminates the session if $\delta_B > \delta$. Alice listens and verifies R_B. She stops the session if she is not convinced that R_B is Bob's response to C_A. Bob listens and verifies R_A. He stops the session if he is not convinced that R_A is Alice's response to C_B.

3.4 Variant

An alternative approach in the protocol is that the symmetric key cryptosystem for messages C_A and C_B can be replaced by a cryptographic commitment function. For example, the commitment function is using a cryptographic one-way function $h(\cdot)$. To commit to an item m, the committing party computes the commitment $h(k \| m)$, where k is a secret key and $\|$ is the concatenation. To verify the commitment, the verifying party must have k and m, compute $h(k \| m)$ and compare it with the commitment. In other word, A_1 can be replaced with $h(K_A \| C_A)$, then C_A will be transmitted along with A_2. Similarly, the parallel protocol can be implemented with the commitment variant too.

4. DISCUSSION

4.1 Availability

In the present protocols, time restriction plays an important role for the availability. The scheme requires the responsor produce a related response in real time, otherwise, the authentication fails. To relieve this burden, the responsor may merely repeat the challenge in his/her own voice. Here, the challenge can be prepared in advance and has no impact on the availability.

Another factor related to the availability is the variability of the network delay T. An inappropriate parameter T may disable to set up an authenticated channel. Thus, the proposed scheme is applicable such as in VOIP where the quality of the service itself is required to be high.

Despite the proposed protocols may reject some genuine communication, no forgery is possible. In other words, although false rejection ratio FRR≠0 due to network traffic, FAR (false acceptance ratio) is negligible. From the viewpoint of security, FAR is much more important than FRR.

4.2 Impersonating Bob

In the proposed protocols, an important condition for Alice (Bob) to authenticate Bob (Alice) is that Bob's (Alice's) response to her (his) challenge must arrive within a defined time interval. Therefore, if Clark can obtain the correct answer in the voice of Bob (or Alice) in the predefined time interval, he can impersonate Alice (Bob) successfully. To this end, Clark may adopt one of the following three methods to provide the response in the voice of the impersonated party.

- Clark replays recorded speeches of the impersonated party.
- Clark or his device responses to the challenge by emulating the speech of the impersonated party.
- Clark lures the impersonated party to answer the challenge.

The first two methods are not possible based on security assumptions S1-S3. To defend against Clark's attack using the third method, it is crucial to check the lengths of the elapsed time of the clocks.

Assume that an attacker would like to impersonate Bob, Figure 4 illustrates a possible way to lure Bob to respond with R_B. To this end, Clark performs a man-in-the-middle attack shown in Figure 4 so as to obtain C_A and R_B. In this simulated attack, Alice proceeds in the same way as that shown in Figure 2. For the sake of simplicity, we will only show the main steps which are related to the attack.

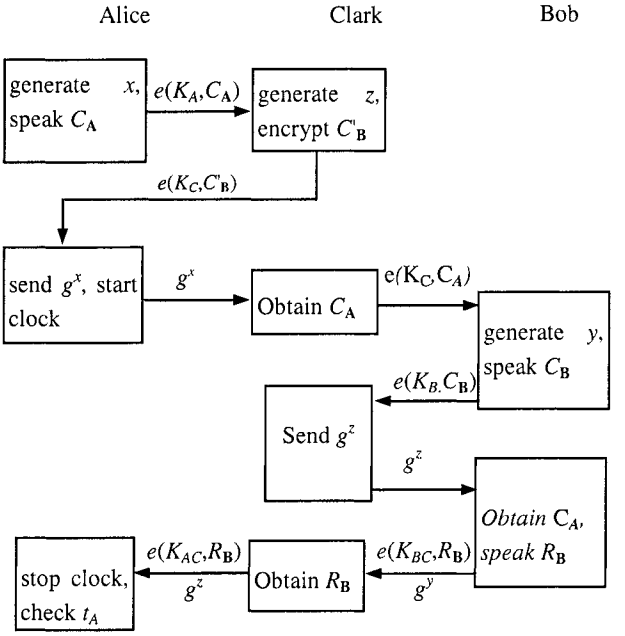

Figure 4. Impersonate Bob. Clark shares a channel with Alice and Bob simultaneously so that he can eavesdrop messages between them.

4.2.1 Obtaining Alice's challenge C_A

In Figure 4, Alice starts a session and sends the ciphertext $A_1=e(K_A, C_A)$ to Bob. After intercepting the ciphertext A_1, Clark generates a number z, computes a key $K_C=h(g^z)$. Based on Assumption S2, Clark can not mimic Bob to speak a challenge, but he may reuse Bob's recorded speech. Clark encrypts an old statement C'_B spoken by Bob, and sends to Alice the ciphertext $e(K_C, C'_B)$.

Alice receives the ciphertext, replies with g^x and starts a clock. Clark derives a key $K_A=h(g^x)$ and recovers $C_A = d(K_A, C_A)$.

4.2.2 Obtaining Bob's response R_B

Because Clark can not mimic Bob's voice to produce an appropriate response R_B, he has to lure Bob to respond to Alice's challenge C_A. To this end, he impersonates Alice and starts a new session with Bob by sending Bob $e(K_C, C_A)$. Next, upon receipt of $e(K_C, C_A)$, Bob generates a random y,

computes a key $K_B = h(g^y)$, speaks a challenge statement C_B, and sends the ciphertext $e(K_B, C_B)$ to Alice. Clark intercepts the ciphertext.

To continue to impersonate Alice, Clark sends g^z to Bob. Bob computes $K_C = h(g^z)$, decrypts $e(K_C, C_A)$, listens to C_A which was indeed spoken by Alice. Bob speaks a response statement R_B, computes a key $K_{BC} = h(g^{zy})$, and transmits g^y and the ciphertext $e(K_{BC}, R_B)$ to Alice. Clark again intercepts the ciphertext, computes $K_{BC} = h(g^{yz})$ to decrypt $e(K_{BC}, R_B)$. Now he gets R_B !

4.2.3 Calculating the elapsed time

Clark computes $K_{AC}=h(g^{xz})$, encrypts R_B with K_{AC}, and sends the ciphertext $e(K_{AC}, R_B)$ to Alice. Alice stops the clock and calculates the elapsed time t_A, decrypts the ciphertext $e(K_C, C'_B)$ and $e(K_{AC}, R_B)$ to recover C'_B and R_B, respectively. Alice makes sure that C'_B and R_B are in Bob's voice. Since R_B is indeed a response to C_A from Bob, Alice will be fooled into believing Clark as Bob! Luckily, our protocol prevents this from happening by checking the clock's elapsed time t_A in the following.

4.2.4 Checking the elapsed time

Consider the man-in-the middle attack shown in Figure 4. Within the time interval t_A, Bob has to speak his challenge statement C_B, listens to C_A, and speaks the response R_B; therefore, $t_A \geq |C_B| + |C_A| + |R_B| + \Delta_b$, where Δ_b is the time used in computation and transmission. With reference to Eq.(1), Alice checks $\delta_A = t_A - (|C_A| + |R_B| + \Delta_B) \geq |C_B| \geq T > \delta$.

Therefore, by checking the value of t_A, Alice detects the man-in-the-middle attack and stops the session.

4.3 Impersonating Alice

The other kind of possible attack is to impersonate the initiator Alice. To this end, Clark has to obtain the original challenge C_B and then Alice's respond R_A to Bob's challenge C_B. Figure 5 shows the second scenario of the man-in-the-middle attack, where Clark impersonates Alice to Bob. Therefore, Clark must start the communication with Bob at first.

4.3.1 Obtaining Bob's challenge C_B

Clark generates z, computes a key $K_C = h(g^z)$, and encrypts C'_A - an old statement from Alice. Clark starts the impersonation by sending the ciphertext $e(K_C, C'_A)$ to Bob.

Upon receipt of Clark' message, Bob generates y, and a key $K_B = h(g^y)$. He then speaks a reply C_B, and transmits the ciphertext $e(K_B, C_B)$ to Alice. Clark sends g^z to Bob. Bob derives $K_C = h(g^z)$, decrypts $e(K_C, C'_A)$ with K_C to recover C'_A. Bob listens to C'_A and believes that C'_A was indeed spoken by Alice. He then speaks a response statement R_B, derives a key $K_{BC}=h(g^{zy})$, and transmits g^y and the ciphertext $e(K_{BC}, R_B)$ to Alice. Bob then starts a clock.

Next, upon interception of $e(K_{BC}, R_B)$ and g^y, Clark derives $K_B=h(g^y)$ and $K_{BC} = h(g^{zy})$, decrypts $e(K_B, C_B)$ to recover C_B!

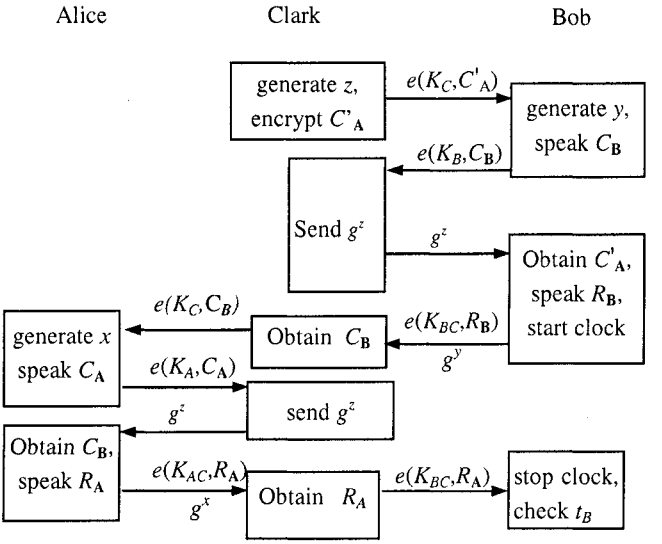

Figure 5. Impersonate Alice.

4.3.2 Obtaining Alice's response R_A

Since Clark is not able to reply C_B in Alice's voice, he starts a session with Alice by sending $e(K_C, C_B)$ to Alice. Upon receipt of $e(K_C, C_B)$, Alice generates x, and computes a key $K_A=h(g^x)$. She speaks a challenge C_A, and sends the ciphertext $e(K_A, C_A)$ to Bob.

Clark intercepts the ciphertext and sends g^z to Alice. Alice derives a key $K_C = h(g^z)$, decrypts $e(K_C, C_B)$ to recover C_B. She listens to C_B and believes that it is in Bob's voice.

Alice speaks a response statement R_A, computes $K_{AC} = h(g^{xz})$, sends g^x and the ciphertext $e(K_{AC}, R_A)$ to Bob. Clark intercepts the message from Alice and decrypts the ciphertext to obtain R_A!

4.3.3 Checking the elapsed time

After obtaining R_A, Clark sends the ciphertext $e(K_{BC}, R_A)$ to Bob. Bob receives $e(K_{BC}, R_A)$, stops the clock and calculates the elapsed time t_B. Without checking the elapsed time t_B, Bob would have been fooled into believing that he is talking to Alice since R_A is Alice's reply to C_B. However, within interval t_B, Alice has to speak C_A, listen to C_B and speak R_A, thus,

$$\delta_B = t_B - (|C_B| + |R_A| + \Delta_A) \geq |C_A| \geq T > \delta.$$

Therefore, by checking the elapsed time t_B, Bob detects the man-in-the-middle attack and hence stops the session.

5. CONCLUSIONS

The present paper describes a scheme for mutual authentication and key establishment between two remote human users. Unlike most of the existing authenticated key establishment protocols where remote authentication is based on sharing a secret/password or knowing remote party's public key, our scheme is based on exchanging of signals representing remote user's biometrics information. Although clock timing plays an important role in our protocols, only relative time is used so that synchronization between two parties is not required. Our technique is especially useful for securing telephony or videoconference communications over open networks. We illustrated our scheme with protocols using audio signals to represent users' biometrics information. It should be noted that security of the protocols can be improved with additional biometrics information such as facial image and mouth movement. Such additional information adds few burdens to the human users, but greatly increases the difficulty of attacking the protocols.

REFERENCE

1. D. Otway and O. Rees, "Efficient and timely authentication", Operating Systems Review, Vol. 21, No. 1, pp. 8-10, 1987.
2. C. Kaufman, R. Perlman, and M. Speciner, Network Security - Private Communication in A Public World, PTR Prentice Hall, Englewoor Cliffs, NJ, 1995.
3. R. H. Deng, J. Zhou and F. Bao, "Defending against redirect attacks in mobile IP," ACM Conference on Computer and Communications Security, pp. 59-67, 2002.
4. P. Zimmermann, PGPfone Owner's Manual, Version 1.0 beta 5, 5 January 1996, `http://web.mit.edu/network/pgpfone/manual`.
5. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using large speech database," IEEE International Conf. on Acoustics, Speech, and Signal Processing (ICASSP), pp.373-376, 1996.

6. Matthias Jilka, Ann K. Syrdal, Alistair D. Conkie and David A. Kapilow, "Effects on TTS Quality of Methods of Realizing Natural Prosodic Variations," 15th International Congress of Phonics Science (ICPhS) 2003.
7. TTS, https://research.microsoft.com/speech/tts/TTS.dll?TTS
8. AT&T, TTS: Synthesis of Audible Speech from Text, http://www.research.att.com/projects/tts/