Credential Wrapping: From Anonymous Password Authentication to Anonymous Biometric Authentication

Yanjiang Yang
Huawei Singapore Research Center
Singapore
yang.yanjiang@huawei.com

Haibing Lu
The Leavey School of Business
Santa Clara University, USA
hlu@scu.edu

Joseph K. Liu
Faculty of Information Technology
Monash University, Australia
joseph.liu@monash.edu

Jian Weng
Jinan University
China
cryptjweng@gmail.com

Youcheng Zhang
Nanjing Unary Information Technology Co., Ltd
China
zyc@unary.com.cn

Jianying Zhou
Institute for infocomm research
Singapore
jyzhou@i2r.a-star.edu.sg

ABSTRACT

The anonymous password authentication scheme proposed in ACSAC'10 under an unorthodox approach of *password wrapped credentials* advanced anonymous password authentication to be a practically ready primitive, and it is being standardized. In this paper, we improve on that scheme by proposing a new method of "public key suppression" for achieving server-designated credential verifiability, a core technicality in materializing the concept of password wrapped credential. Besides better performance, our new method simplifies the configuration of the authentication server, rendering the resulting scheme even more practical. Further, we extend the idea of password wrapped credential to *biometric wrapped credential*, to achieve anonymous biometric authentication. As expected, biometric wrapped credentials help break the linear server-side computation barrier intrinsic in the standard setting of biometric authentication. Experimental results validate the feasibility of realizing efficient anonymous biometric authentication.

Keywords

Anonymous password authentication; Biometric authentication/identification; Anonymous biometric authentication; Public key suppression; Standardization

1. INTRODUCTION

1.1 Anonymous Password Authentication

Anonymous password authentication (APA) strengthens regular password authentication with the protection of user privacy, in such a way that login sessions made by the same user cannot be linked even by the authentication server, not to mention the outsiders who eavesdrop on the communications. APA is a quite useful entity authentication primitive, having many practical applications, e.g., it can enable the employees of an organization to provide critical feedbacks on the management, without the fear of retaliatory actions; another example is that it helps users engage anonymously in the online medical consultation services, with no worry of embarrassment. Recognizing its importance and practicality, ISO/IEC is now in the process of standardizing anonymous password authentication [19].

As per the taxonomy in [19], APA schemes are categorized into two classes: password only APA and storage extra APA. In particular, password only APA works in the standard setting of password authentication, where the authentication server keeps a password database containing all enrolled users' passwords (or password-derived quantities), and the password between a user and the server is the only secret that is used by the two parties to perform entity authentication. An inherent limitation of the password only schemes is the linear server-side computation (linear to the total number of enrolled users), which is clearly problematic when the number of enrolled users is not small. To get around this issue, in ACSAC'09 the storage extra approach was proposed [41].

Central to the storage extra approach is the concept of *password wrapped credential*. More specifically, a user is issued a credential to be used for entity authentication, and she protects the credential with her password (e.g., encrypting the credential with a key derived from password) by generating a password wrapped credential; subsequently when authenticating to the server, the user first recovers the authentication credential from her password wrapped credential using her password, and then engages in the authentication process with the use of the credential. It can be seen that the server-side computation in authenticating the user is the cost to verify the validity of the user's authentication credential, thus independent of the total number of enrolled users. The approach requires a storage facility to store a user' password wrapped credential (so it is called storage extra approach), but the facility needs not be secure, and it can be any device, even a public directory, to retain *portability* - the defining feature of passwords.

In ACSAC'10 the same authors [42] further enhanced their

ASIA CCS '16, May 30-June 03, 2016, Xi'an, China
© 2016 ACM. ISBN 978-1-4503-4233-9/16/05…$15.00
DOI: http://dx.doi.org/10.1145/2897845.2897854

ACSAC'09 scheme with a set of more efficient building blocks, coupled with solutions to some important practical issues such as membership withdrawal and online guessing attacks. The resulting scheme basically brought anonymous password authentication to be a practically usable primitive, as it appeared to have solved all major technical issues that may occur in practical deployment. A manifestation to this is that the ACSAC'10 scheme was selected as a candidate to be standardized by ISO/IEC [19], representing the state-of-the-art in the storage extra APA genre.

Regardless, we find that the ACSAC'10 scheme can still be further improved. Our observation is that a core technicality in materializing the concept of password wrapped credential is to prevent offline guessing attacks from recovering the underlying authentication credential from a password wrapped credential - anyone can enumerate every possible password to "unwrap" the password wrapped credential until a valid credential is produced (password wrapped credentials are public by assumption). Both [41, 42] tackled this issue with the same method: to smash the public verifiability of an authentication credential by encrypting certain credential elements under the server's homomorphic encryption, so that only the server can verify credential validity. In this work we propose a new method to attain such server-designated verifiability[1] - the server simply suppresses the public key of the underlying authentication credential scheme, such that no one else than the server can check the authenticity of credentials. Better efficiency aside, this new method simplifies the system configuration, as the authentication server does not need to have homomorphic encryption in its possession.

1.2 From APA to Anonymous Biometric Authentication

We further extend the idea of password wrapped credentials to *biometric wrapped credentials*, to achieve anonymous biometric authentication/identification. Biometric authentication concerns authenticating a user based on the biometrics derived from her physical traits or behavioral patterns. Biometrics are commonly believed to succeed passwords in the long run, because biometrics also feature portability while not suffering low entropy. The motivation for anonymous biometric authentication is derived either by taking anonymous biometric authentication as successor of anonymous password authentication in specific, or from the necessity of provision of user anonymity in entity authentication in general (anyhow, biometric authentication is an entity authentication technique). The typical setting of biometric authentication is quite similar to password authentication: each user enrolls a biometric sample (the enrolled sample is often refereed to as template) to the authentication server, which ends up maintaining a database containing all enrolled users' templates; a subsequent authentication of a user is dependent on the comparison of a fresh biometric reading of the claimant with her enrolled template in the template database. Under such a setting, it is apparent that to achieve user unlinkability towards the server (i.e., two authentication sessions from the same user cannot be recognized as such by the server), by no means can the server perform asymptotically better than linear computation (with respect to the total number of enrolled users). Worse yet, the complexity of template/reading matching further aggravates the server's actual workload.

Motivated by the role password wrapped credentials play in storage extra APA, we expect biometric wrapped credentials to play a similar role in breaking the linear server-side computation barrier inherent in achieving anonymous biometric authentication in the standard setting. The concept of biometric wrapped credential is akin to password wrapped credential - a user wraps her authentication credential with her biometric information, and only herself can do unwrapping as only she can produce biometric samples approximate enough to that used in wrapping. The noisy nature of biometrics dictates that we are in need of a kind of "fuzzy wrapping", a main challenge in substantiating biometric wrapped credential. To our delight, we find out that many template protection techniques in the biometrics field, e.g., [3, 6, 20, 22, 33], can be exploited for "fuzzy wrapping".

1.3 Our Contributions

Specifically, our contributions are summarized below.

- We improve on the ACSAC'10 APA scheme that is being standardized, by innovating in the way to achieve server-designated credential verifiability, with a new method of public key suppression. This new method not only brings about better performance, but more importantly, simplifies system configuration, which in turn makes the resulting anonymous authentication protocol simpler, and easier for security analysis. This is especially important for practical deployment.

- We extend the concept of password wrapped credential to biometric wrapped credential. Biometric wrapped credentials help break the linear bound of server-side computation in the standard setting, thus advancing anonymous biometric authentication towards practicality.

- We empirically tested our proposal for anonymous biometric authentication, and the experimental results demonstrated its feasibility.

Organization. We review related work in Section 2, followed by an improvement to the ACSAC'10 scheme in Section 3. The idea and materialization of biometric wrapped credential is presented in Section 4. Experimental results are reported in Section 5, and Section 6 concludes the work.

2. RELATED WORK

2.1 Anonymous Password Authentication

Password authentication has been widely studied in the literature, e.g., [5, 8, 9, 14, 16, 21, 23, 39], to name a few. For ease of memorization, users tend to choose short passwords from a relatively small space; thus passwords are of

[1] The server-designated verifiability seems similar to Designated Verifier Signature with the designated verifier being the server. In fact, server-designated verifiability is much more stringent than Designated Verifier Signature: (1) first of all, ordinary Designated Verifier Signature does not provide signer anonymity; (2) more importantly, if using so called Anonymous Designated Verifier Signature such as [13] which provides signer anonymity to instantiate password wrapped credential, then both of signatures and the user signing key are required to be verifiable only to the designated verifier. Unfortunately, Anonymous Designated Verifier Signature only achieves designated verifiability of signatures.

low entropy in nature, and susceptible to brute force guessing attacks. As such, the best a password authentication system can achieve is that an attacker can only validate his guesses of a password by interacting with the authentication server and seeing the server's accept/reject responses. Such *online* guessing attacks are unavoidable in password authentication, but can be addressed with system level measures, e.g., suspending one's account once the number of failed login attempts "she" made exceeds a threshold.

In general, password authentication does not consider protection of user privacy, as the authentication server needs to know who the requesting user is and uses the corresponding enrolled password to authenticate her. Anonymous password authentication (APA) was proposed to fill this gap, and the first such a scheme was due to [37], which combines a password authentication protocol with PIR (Private Information Retrieval), with the former performing mutual authentication, and the latter achieving user privacy protection. Afterwards, several new schemes were presented [24, 34, 40], all of which explicitly or implicitly make use of PIR, and they differ in the extent to which the server-side computation can be converted into pre-computation. The method of pre-computation in fact trades communication for (real time) online computation. APA was also studied with regard to a three-party scenario (i.e., user-gateway-server) [1], and the protocol still explicitly utilizes PIR.

It is not a surprise that all of [1, 24, 34, 37, 40] exploit PIR in their schemes, as they consider APA in the standard setting, where a password database is present at the server side with each entry being a pair of $\langle$user ID, password$\rangle$. Achieving user unlinkability amounts to letting the server not know which entry of the database it actually uses to authenticate a user, which in substance is a PIR problem per se. As such, the lower bound for server-side computation is linear to the total number of entries in the password database.

To break this bound, an unorthodox approach of password wrapped credential is proposed [41] - a user uses an authentication credential for anonymous entity authentication, and she protects her credential with a password by generating a password wrapped credential; only the same password can recover the original authentication credential from the password wrapped credential. The server-side computation in this approach is fixed, equal to the cost of verifying the authentication credential. This approach is termed *storage extra* APA [19] (contrasting to the above-mentioned *password only* schemes in the standard setting), as users need to manage their password wrapped credentials in certain (unprotected) storage. [42] later improved the scheme with a set of more efficient building blocks, as well as addressing several practical issues neglected in [41]. Yet another storage extra APA scheme was presented in [30], in which all users essentially share a global authentication credential in order not to be distinguished by the server. However, sharing of authentication credential causes many consequences, e.g., hard for user revocation; impossible to discern online guessing attacks towards individual users; illegal dissemination of the authentication credential.

The scheme in [42] turns out to be the most promising and practical for achieving APA. But as will be shown later, it can still be improved, and in particular, we innovate in a critical ingredient of the methodology adopted in [41, 42] to achieve server-designated verifiability of authentication credentials - our new method is that the server keeps the public key of the underlying authentication credential scheme to itself. We realize that the idea of "public key suppression" is not new, and software smartcard [17] adopted a similar one: software smartcard was proposed for the sake of safely managing the private keys in the context of PKI (Public Key Infrastructure), as opposed to the usual practice of using hardware smartcard; it works by encrypting a private key with a password; to prevent extraction of the private key from the "software smartcard", the corresponding public key must be held unpublished; otherwise, extraction can be done based on the relationship between the public and private key pair. As pointed out in [17], software smartcard can only work in a "closed PKI", but no concrete examples of closed PKI are given. In a sense, our APA scheme under public key suppression can be a concrete example of closed PKI, even though APA and software smartcard are drastically divided in both the problem scope and the solution technicality.

2.2 (Anonymous) Biometric Authentication

Biometrics such as one's physical traits (i.e., face, fingerprint, voice, and iris) and behaviorial patterns have long been used for identification/authentication purposes. The baseline issue to solve in biometric authentication is the noisy nature of biometrics, i.e., two readings/samples of one's biometric are rarely identical. The bulk of research on biometrics is to enhance the matching accuracy when comparing biometric readings, discussion on which is beyond the scope of this paper.

The standard practice for biometric authentication is that users enroll their biometric readings (called templates) to the authentication server beforehand; then the server can authenticate a user by comparing a freshly captured biometric reading with her enrolled template. Biometrics are personal in nature and cannot be changed during one's life time, thus the templates placed at the server implicate serious privacy concern, and protection of the templates has been a focal point in biometrics research. The basic rationale for template protection is to transform a biometric sample in certain ways such that the resulting template does not reveal the original sample. There are many template protection techniques in the literature, and of relevance to us are those that support *key release* as reviewed below.

The idea of Fuzzy Commitment [22] can be cast as follows: a template is generated as $\alpha = k - x$, where k is a randomly selected key and x is a biometric sample; given another sample x', k can be recovered as $f(\alpha + x') = f(k + (x' - x))$, where f is the decoding function of an error correction code; clearly the genuine k cannot be reconstructed unless the distance between x' and x is below the correction capacity δ of the error correction code (i.e., the code can correct up to any δ-bit errors). Fuzzy Vault [20, 28] works by secret-sharing a key k in a polynomial induced by a set of points derived from a biometric sample; the set together with a set of dubious points forms the template; via error correction code, a sufficiently approximate biometric reading can re-generate a number of genuine points more than the degree of the polynomial, thus being able to recover k. Fuzzy Extractor [3, 6] is a primitive concerning extracting from a biometric sample a secret k, as well as a public quantity q which is the template; q and another reading together can recover k as long as the two samples are close enough. Another technique providing key release is Biotoken and Bipartite Biotoken [32, 33], which basically works on the fact that a biometric sample can be split into a stable part and a unstable part; and only the stable part needs to be protected in the template. Our substantiation of biometric wrapped credential is independent of particular template protection with key release techniques, and can build upon any of them .

Probably due to the fact that biometric authentication by itself is already demanding due to the fuzzy nature of biometrics, there exists few work in the literature investigating provision of user anonymity in biometric authentication. To the best of our knowledge, the only work that studies anonymous biometric authentication is [7], which proposes biometric based anonymous credentials. Key distinctions between our biometric wrapped credentials and biometric based anonymous credentials include: (1) To obtain a biometric based anonymous credential [7], a user needs to register his/her biometric template to the credential issuing authority; in contrast, users do not register their biometric templates to any party (including the authentication server) in our biometric wrapped credential, minimizing the risk of biometrics revelation. This is important in Today's privacy conscious landscape. (2) In [7], a user must store a biometric based anonymous credential in a secure hardware device, which diminishes the portability of biometrics. In biometric wrapped credential, we actually establish a 'software smartcard' for anonymous credential using one's biometric data, thus requiring no extra secure hardware device.

Privacy-preserving biometric recognition [12, 18, 29, 35] also relates to anonymous biometric authentication. It can be cast as follows: a server holds a biometric database containing a number of biometric templates, and a client who holds a biometric reading wants to learn the index of the database entry that matches her reading the most; the goal is to enable such matching while without revealing the client's biometric reading to the server, and without disclosing the database to the client other than the index. The latest technique [18] involves evaluating *garbled circuits* with inputs from the two sides. Achieving anonymous biometric authentication in this (standard) setting is expected to be more involving, as the output would be a secret generated from a matching, rather than simply a matching index; in addition, privacy-preserving biometric recognition seems to have not taken template protection into account.

3. IMPROVING ON THE SCHEME IN [42]

The storage extra APA scheme in [42] employs BBS+ Signature [2] as the underlying authentication credential scheme, complemented with Nguyen's Dynamic Accumulator [27] for user revocation. We improve on a critical link of the scheme, namely the method to achieve server-designated credential verifiability so as to foil offline brute force guessing attacks against password wrapped credentials. Our improvement affects the use of BBS+ Signature only, without touching on the part of Nguyen's Dynamic Accumulator. Thus for clarity, we restrict ourselves to the BBS+ Signature part, and it should be understood that the mechanisms for user revocation and online guessing attacks remain the same.

3.1 Review of BBS+ Signature

BBS+ Signature [2] is a digital signature scheme with efficient protocols for *blind signing* and *blind verification.* BBS+ Signature is built upon bilinear maps. Let G_1, G_2, G_T be cyclic groups of a prime order q. A bilinear map/pairing $\tilde{e} : G_1 \times G_2 \to G_T$ has the following commonly used properties: (1) Bilinearity - $\forall u \in G_1, v \in G_2$ and $x, y \in_R Z_q$, $\tilde{e}(u^x, v^y) = \tilde{e}(u, v)^{xy}$, where $a \in_R S$ denotes a is randomly chosen from set S; (2) Non-degeneration - let g, h be generators of G_1, G_2, respectively, $\tilde{e}(g, h) \neq 1$. A less stated property of bilinear pairing is one-way-ness, i.e., given g and $\tilde{e}(g, h)$, it is hard to compute h. In fact, this property has been implicit in numerous literature.

The public key of BBS+ signature is ($w = h^{\chi}, h \in G_2, g_0, g_1, g_2 \in G_1$), and the private key is ($\chi \in Z_q$). A signature signed upon message m is defined to be (M, k, s), where $k, s \in_R Z_q$, and $M = (g_0 g_1^m g_2^s)^{\frac{1}{k+\chi}} \in G_1$. The signature can be verified as $\tilde{e}(M, wh^k) = \tilde{e}(g_0, h)\tilde{e}(g_1, h)^m \tilde{e}(g_2, h)^s$. More interestingly, signature verification can be conducted in a blind way, such that the holder of a signature proves the possession of the signature to a verifier, while without revealing any information on the signature. Blind verification works as follows, assuming the prover has a signature (M, m, k, s): the prover selects $r_1, r_2 \in_R Z_q$ and computes $M_1 = Mg_1^{r_1}, M_2 = g_2^{r_1} g_1^{r_2}$, which are then sent to the verifier; next the prover and the verifier conduct a standard zero-knowledge proof of knowledge protocol $PoK\{(r_1, r_2, k, \delta_1, \delta_2, s, m) : M_2 = g_2^{r_1} g_1^{r_2} \wedge 1 = M_2^{-k} g_2^{\delta_1} g_1^{\delta_2} \wedge \frac{\tilde{e}(M_1, w)}{\tilde{e}(g_0, h)} = \tilde{e}(M_1, h)^{-k} \tilde{e}(g_1, h)^m \tilde{e}(g_2, h)^s \tilde{e}(g_1, w)^{r_1} \tilde{e}(g_1, h)^{\delta_1}\}$, where $\delta_1 = r_1 k, \delta_2 = r_2 k$. The protocol is a standard commit-challenge-respond process Π: the prover begins by sending a *commitment* message, denoted $\mathtt{CMT}(\Pi_{BBS+})$, to the verifier; the verifier returns a challenge; the prover then responds with a *response*, denoted $\mathtt{RES}(\Pi_{BBS+})$. We omit the details for its straightforwardness.

3.2 Our Improvement

In [42], a user's authentication credential is a BBS+ signature (M, k, s), satisfying $M = (g_0 g_1^u g_2^s)^{\frac{1}{k+\chi}}$, where u is user identity. The corresponding password wrapped credential takes the form of $\langle [M]_{pw}, \mathtt{HE}(s), k \rangle$, where $[M]_{pw}$ denotes that M is properly protected by a password pw (e.g., encryption with a key derived from pw), $\mathtt{HE}(s)$ represents s is encrypted under the authentication server's homomorphic encryption, and k is left unprotected[2]. In particular, the encryption of s under the server's homomorphic encryption is critical in nullifying the public verifiability of the authentication credential, in order to foil offline guessing attacks on $[M]_{pw}$. The effect is server-designated verifiability, i.e., credential verifiability is restricted to the server only. Unable to decrypt $\mathtt{HE}(s)$, the user cannot directly use the above blind verification protocol for authentication. Hence a specially customized variant was presented and used in [42].

We propose an entirely new method to achieve server-designated verifiability - the authentication server withholds the public key w of BBS+ Signature; since signature verification requires the public key, suppression of w makes no one else can recognize a valid signature. An immediate benefit is that the server is freed from the forced use of homomorphic encryption, greatly simplifying the configuration of the server. This is particularly important for practical deployment of the system. Further, we want our method to be

[2] In the complete scheme, k will be accumulated in Nguyen's Dynamic Accumulator for user revocation purpose. Thus k has to be left unprotected to facilitate witness update in case of user revocation and user joining.

general, directly making use of the original blind verification protocol, instead of a customized variant. This would make security analysis much simpler and easier. To that end, we are actually posed a challenge: for the holder to perform the above PoK protocol, she needs the knowledge of $\tilde{e}(g_1, w)$; however, since w is not known, she cannot compute $\tilde{e}(g_1, w)$. Fortunately, a further observation reveals that the server can directly publish $\tilde{e}(g_1, w)$ as a public parameter, while keeping w to itself. The one-way-ness of the pairing operation guarantees that $\tilde{e}(g_1, w)$ does not reveal w. Our security analysis later further shows that this does not impair the security of password wrapped credentials. Since the original blind verification protocol is better in performance than the variant in [42], our new scheme is more efficient.

Details. Given the above idea of suppression of public key, we give below the detailed construction of anonymous password authentication scheme with mutual authentication and key exchange.

Setup: The authentication server sets up system parameters as follows:

(1) determines a bilinear map $\tilde{e} : G_1 \times G_2 \to G_T$ as defined earlier; chooses $\chi \in_R Z_q, h \in_R G_2$ and computes $w = h^\chi$, and keeps (χ, w) as the private key; chooses $g_0, g_1, g_2 \in_R G_1, t \in_R G_T$ and publishes the public key as $(\tilde{e}(g_1, w), h, g_0, g_1, g_2, t)$.

(2) picks and publishes hash functions $H_1 : G_T^2 \to \{0,1\}^{\kappa_0}$, and $H_2 : G_T^3 \to \{0,1\}^{\kappa_1}$, where κ_0, κ_1 are appropriate security parameters.

User Enrollment: Users enroll to the server in advance, each getting an authentication credential. A credential is a BBS+ signature (M, k, s) signed upon user identity u.

Upon obtaining her authentication credential securely, user u wraps s with a key derived from her password pw_u, denoted as $[s]_{pw_u}$ (e.g., encrypting s with a block cipher); then the password wrapped credential is $\widehat{cred}_u = \langle M, [s]_{pw_u}, k \rangle^3$. Finally, the user puts $\widehat{cred}_u$ to her preferred storage such as mobile phone, USB flash memory, or a public facility.

Anonymous Authentication: Suppose user u already has her password wrapped credential $\widehat{cred}_u = \langle M, [s]_{pw_u}, k \rangle$ available at the point of login. The anonymous authentication protocol offering mutual authentication and key exchange between u and the server works as follows.

Step 1. The user does the following:

(1) unwraps $[s]_{pw_u}$ with her password pw_u to get s;

(2) picks $r, v_1 \in_R Z_q$, and computes $R_1 = g_1^r, R_2 = t^{v_1}\tilde{e}(g_1, w)^r$;

(3) constructs $\texttt{CMT}(\Pi_{BBS+})$; sends $R_1, R_2, \texttt{CMT}(\Pi_{BBS+})$ to the server as a login request:

User $\longrightarrow$ Server: $R_1, R_2, \texttt{CMT}(\Pi_{BBS+})$

Step 2. Upon receipt of the login request, the authentication server does the following:

(1) computes $V_1 = \frac{R_2}{\tilde{e}(R_1, w)} = t^{v_1}$;

(2) picks $v_2 \in_R Z_q$ and computes $V_2 = t^{v_2}$;

(3) computes $\tilde{V} = H_1(V_1, V_2)$, and sends back $V_2, \tilde{V}$ to the user:

Server $\longrightarrow$ User: $V_2, \tilde{V}$

Step 3. Receiving the message, the user does as follows:

(1) checks $H_1(t^{v_1}, V_2) \overset{?}{=} \tilde{V}$, and aborts if not;

(2) taking V_2 as the challenge, constructs and sends $\texttt{RES}(\Pi_{BBS+})$ to the server;

(3) stops the protocol by computing a shared key $sk = H_2(t^{v_1}, V_2, V_2^{v_1})$.

User $\longrightarrow$ Server: $\texttt{RES}(\Pi_{BBS+})$

(4) At the server end, the server computes $sk = H_2(V_1, V_2, V_1^{v_2})$ upon validating $\texttt{RES}(\Pi_{BBS+})$, and stops.

Remarks. It is not necessary to follow the protocol step by step, as it is already self-contained. We only outline the intuitions. First of all, we point out that (R_1, R_2) constitutes an encryption of $V_1 = t^{v_1}$ under public key $\tilde{e}(g_1, w)$, so that only the server knows the private key w. This use of $(\tilde{e}(g_1, w), w)$ as a public-private key pair for encryption represents a novelty of our scheme. As such, authentication of the server by the user is through the encryption of V_1, such that only the server can decrypt. Second, V_1, V_2 serve not only for an apparent Diffie-Hellman key exchange, but also as the respective parties' freshness nonces.

3.3 Performance Comparison and More

The above scheme is comparable to the basic scheme (cf. Section 4.2) of [42]. To demonstrate the performance gain under our new method of public key suppression, we provide an analytical performance comparison between the two. As a rule of thumb, for computation overhead we only count the number of exponentiations (we treat point multiplication in G_1, G_2 as exponentiation) and bilinear paring operations, as they are the operations that dominate the computational overhead; besides, a multi-exponentiation is treated as multiple exponentiations (e.g., $g_0^x g_1^y$ is counted as 2 exponentiations). Let $|G|$ denote the bit length of an element in group G, $\texttt{EXP}_G$ an exponentiation operation in G, and $\texttt{Pair}$ a bilinear pairing operation. The comparison results are reported in Table 1. We remark that we had avoided computing bilinear pairing operations as much as possible in both schemes, e.g., $\tilde{e}(g_0, h), \tilde{e}(g_2, h)$ are treated as fixed bases in G_T.

To be specific on the comparison, let the two schemes achieve the same level of security, e.g., 80 bits, then $|q| = 160$, $|G_1| = 171$, $|G_2| = |G_T| = 1024$; and $1\texttt{EXP}_{G_q} \approx 1\texttt{EXP}_{G_1} \approx 1\texttt{EXP}_{G_T} \approx \frac{1}{4}\texttt{Pair}$ (see, e.g., [4]). As such, user-side and server-side computations in our scheme have been improved, respectively, 20% and 30% compared to the scheme in [42]; while communications are similar. While the gain may seem moderate, it is actually not trivial considering that we obtained the gain by working over exactly the same primitive, i.e., BBS+ Signature.

Besides better performance, more important of public key suppression is *generality*. It enables a direct use of the original blind verification protocol of BBS+ Signature; in addition, possession of homomorphic encryption by the authentication server is not required, which greatly simplifies the

[3] We also leave M unprotected, and the only reason is that it is an element in G_1 and it has certain known structure, e.g., if its representation is a (x, y) pair, then the x-coordinate and y-coordinate must satisfy the underlying elliptic curve. Note that the quantities under protection must be random, without known properties.

Table 1: Performance Comparison

	Computation[a]		Communication
	User	Server	(bits)
Basic scheme of [42]	$4\texttt{EXP}_{G_q} + 7\texttt{EXP}_{G_1}$ $+8\texttt{EXP}_{G_T} + 2\texttt{Pair}$	$1\texttt{EXP}_{G_q} + 6\texttt{EXP}_{G_1}$ $+7\texttt{EXP}_{G_T} + 4\texttt{Pair}$	$9\lvert q\rvert + 7\lvert G_1\rvert + 1\lvert G_T\rvert$
Our scheme	$9\texttt{EXP}_{G_1} + 9\texttt{EXP}_{G_T}$ $+1\texttt{Pair}$	$7\texttt{EXP}_{G_1} + 8\texttt{EXP}_{G_T}$ $+2\texttt{Pair}$	$7\lvert q\rvert + 4\lvert G_1\rvert + 2\lvert G_T\rvert$

[a] Homomorphic encryption in [42] is assumed substantiated as ElGamal encryption in group G_q of a prime order q.

system configuration and makes the anonymous authentication protocol much neater and easier for security analysis. These are important factors when it comes to practical deployment.

3.4 Security Analysis

As specified in [41, 42], a storage extra APA scheme must satisfy *Security of Password, User Unlinkability*, and *Authenticated Key Exchange*. It is not hard to see that our scheme meets User Unlinkability and Authenticated Key Exchange: the former is due to blind verification of BBS+ Signature; for the latter, our anonymous authentication protocol is essentially a combination of signature based authenticator and (public key) encryption based authenticator - see the security arguments in [42].

It remains to analyze Security of Password. The biggest threat to Security of Password comes from a collusion of some enrolled users, each having his own authentication credential; they target a particular user's password wrapped credential by means of offline guessing attacks; to assist in their attacks, they may eavesdrop on the victim performing the anonymous authentication protocol with the server. We argue that overseeing the anonymous authentication protocol does not give them more advantage for offline guessing attacks. To see this, recall that the only place where the protocol could reveal information on the underlying authentication credential is the blind verification protocol of BBS+ Signature, which results in (M_1, M_2) and PoK on the credential elements. (M_1, M_2) information theoretically hides M, while the zero-knowledge proof PoK by definition does not reveal any information on the values to be proved.

It thus suffices to model the threat to Security of Password as an adversary with a set of valid authentication credentials in possession, and to show that the adversary cannot discern a particular authentication credential under attack. To this end, Theorem 1 mandates that any PPT adversary against our method of BBS+ Signature under public key suppression has a negligible advantage in distinguishing a valid BBS+ signature from a random "simulation", even with the knowledge of other BBS+ signatures. This ensures that offline guessing attacks to password wrapped credentials is futile, thus Security of Password is attained. We shall stress that this "indistinguishability" property is irrelevant in the original BBS+ Signature, as it does not suppress public key.

Theorem 1. *Define an adversary's advantage in the following "indistinguishability game" as* $|\Pr(\sigma^* = \sigma) - 1/2|$. *In the general group model, any PPT adversary* $\mathcal{A}$ *has a negligible advantage.*

1. Set up the system parameters $g_0, g_1, g_2, h, w = h^\chi$ for BBS+ Signature; publish $\langle g_0, g_1, g_2, h, \tilde{e}(g_1, w)\rangle$ as public parameters and keep $\langle \chi, w\rangle$ as secret.
2. $\mathcal{A}$ can repeatedly ask for signatures by submitting a message each time, upon which a signature under BBS+ Signature is given to $\mathcal{A}$.
3. Finally, $\mathcal{A}$ submits a message m.
4. Toss a fair coin $\sigma \overset{R}{\leftarrow} \{0,1\}$; if $\sigma = 1$ then a valid BBS+ signature on m is returned; else return (M, k, s), where $M \in_R G_1, k, s \in_R Z_q$.
5. $\mathcal{A}$ outputs a bit σ^*, which is a guess on σ.

Proof. (Sketch) For clarity of illustration, the proof proceeds in several steps. First, we consider a simplified format of BBS+ signatures, i.e., a signature is defined as (M, k) such that $M = g_0^{\frac{1}{k+\chi}}$. We need to show that if the simplified BBS+ signatures are secure (i.e., no adversary has non-negligible advantage in the "indistinguishability game"), then the original signatures are secure (with respect to the "indistinguishability game"). To that end, in fact it is easy to transform an adversary $\mathcal{A}$ against the original signatures to an adversary $\mathcal{B}$ against the simplified signatures, both under the "indistinguishability game". Specifically, $\mathcal{B}$ works by choosing $r_1, r_2 \in_R Z_q$ and setting $g_1 = g_0^{r_1}, g_2 = g_0^{r_2}$; then $\mathcal{B}$ can simulate the "indistinguishability game" to $\mathcal{A}$ by querying its own signature oracle. In particular, when $\mathcal{A}$ asks for a signature upon m, $\mathcal{B}$ first queries its own oracle which will return $(\bar{M}, k)$ satisfying $\bar{M} = g_0^{\frac{1}{k+\chi}}$; then $\mathcal{B}$ selects $s \in_R Z_q$ and computes $M = \bar{M}^{1+r_1 m + r_2 s}$, and returns (M, k, s) to $\mathcal{A}$. Clearly (M, k, s) is a valid original BBS+ signature on m, and the simulation is perfect.

The remainder of the proof will work on the simplified BBS+ signatures. The second step involves proving that the simplified signatures are indeed secure. The proof is based on the SDDHI (Strong DDH Inversion) assumption in G_1 with $\tilde{e} : G_1 \times G_2 \rightarrow G_T$, which has been proven to hold in the generic group model [10]. In particular, the SDDHI assumption states that given public parameters $(g_0 \in G_1, g_0^\chi, h \in G_2)$ and the access to an oracle that returns $g_0^{\frac{1}{k+\chi}}$, upon input $k \in Z_q$, no PPT adversary can tell apart $(g_0^{\frac{1}{k'+\chi}}, k')$ from $(M' \in_R G_1, k')$. It is apparent that (g_0, g_0^χ, h) leak more information than $(g_0, \tilde{e}(g_0, h^\chi), h)$, as the latter can be computed from the former, but not the vice versa. Hence the SDDHI assumption certainly holds with respect to public parameters $(g_0, \tilde{e}(g_0, h^\chi), h)$. What we need is that the SDDHI assumption holds with respect to $(g_0, g_1, \tilde{e}(g_1, h^\chi), h)$, where $g_1 \in_R G_1$. It is rather straightforward to transform an adversary against the SDDHI assumption under $(g_0, g_1, \tilde{e}(g_1, h^\chi), h)$ to an adversary under $(g_0, \tilde{e}(g_0, h^\chi), h)$, and the trick is that the latter adversary sets $g_1 = g_0^r$ with a random $r \in Z_q$. This means that if the SDDHI assumption under $(g_0, \tilde{e}(g_0, h^{\chi}), h)$ holds, then it must also hold under $(g_0, g_1, \tilde{e}(g_1, h^{\chi}), h)$.

The final step is to show that under the SDDHI assumption with respect to $(g_0, g_1, \tilde{e}(g_1, h^{\chi}), h)$, no adversary in the "indistinguishability game" can have non-negligible advantage. This step is trivial and the specifics are omitted. □

4. BIOMETRIC WRAPPED CREDENTIAL: TOWARDS ANONYMOUS BIOMETRIC AUTHENTICATION

The idea of achieving anonymous password authentication with password wrapped credentials is somewhat unorthodox, but helps get around the linear server-side computation barrier inherent in the standard setting, thus enormously advancing the field and making anonymous password authentication a practically usable primitive. Presumably a more reliable alternative to passwords, biometrics also possess portability; but achieving *anonymous biometric authentication* in its standard setting suffers from an even worse linear server-side computation problem (although also asymptotically linear, the actual cost is definitely higher because of the complexity of biometrics matching). We are thus motivated to extend the idea of password wrapped credential to *biometric wrapped credential*, in an attempt to make anonymous biometric authentication a realistic tool.

4.1 Biometric Wrapped Credential

The concept of biometric wrapped credential is similar to that of password wrapped credential, but with one's biometric information (in place of a password) being used for protection of an authentication credential. Specifically, during the Enrollment phase, the authentication server issues each user a credential to be used for anonymous authentication; the user wraps the credential with her biometric information, which yields a biometric wrapped credential. Subsequently, each time to authenticate to the server, the user starts by recovering the authentication credential from the biometric wrapped credential with a fresh biometric reading (we assume that the application implementing the biometric wrapped credential approach must ensure the 'liveness' of biometric readings, as required in regular biometric authentication), and then engages in anonymous authentication with the server using the recovered credential. Figure 1 depicts the conceptual comparison between password wrapped credential and biometric wrapped credential. To maximally retain portability, a biometric wrapped credential can be managed at any storage device so as to guarantee its availability at the point of authentication.

Clearly this is quite different from the standard setting for biometric authentication, where each user enrolls a biometric template to the authentication server who ends up managing a database containing all enrolled biometric templates. In the approach of biometric wrapped credential, no biometric template is enrolled to the server, diminishing the risk of biometrics leakage; more importantly, the workload upon the server for anonymously authenticating a user is the cost to verify the authenticity of the authentication credential, thus independent of the total number of enrolled users, breaking the linear server-side bound intrinsic in the standard setting.

4.2 Materialization

The challenge in substantiating the concept of biometric wrapped credential is the noisy nature of biometrics, i.e., different readings of the same biometric trait of the same person, even obtained using the same sensor, are always distinct. Hence unlike passwords, biometric information cannot be directly used to derive a cryptographic key for wrapping an authentication credential. Fortunately, we discover that many existing template protection techniques in the biometrics field support *key release* (e.g., [3, 6, 20, 22, 33] and see Section 2), and they cater to the need of substantiating biometric wrapped credential.

A Unified Abstraction. To facilitate illustration, we provide a unified abstraction of the template protection with key release techniques, as reviewed in Section 2. Let $\mathbb{X}$ denote the space of a particular biometric trait. Basically, a such technique can be described by the following two algorithms:

- $(\mathtt{k}, \mathtt{tpl}) \leftarrow \text{KeyTPLGen}(x \in \mathbb{X})$: This probabilistic key and template generation algorithm outputs a key $\mathtt{k}$ and a template $\mathtt{tpl}$, taking as input a biometric sample x.
- $\mathtt{k}' \leftarrow \text{KeyRelease}(\mathtt{tpl},\ x' \in \mathbb{X})$: The deterministic key release algorithm takes as input a template $\mathtt{tpl}$ and a biometric reading x', and outputs a key $\mathtt{k}'$.

It stipulates that for $\forall x, x' \in \mathbb{X}$, $(\mathtt{k}, \mathtt{tpl}) \leftarrow \text{KeyTPLGen}(x)$ and $\mathtt{k}' \leftarrow \text{KeyRelease}(\mathtt{tpl},\ x')$: $\mathtt{k} = \mathtt{k}'$ iff $\mathsf{dist}(x, x') < \delta$, where dist is a distance function in terms of a certain metric (e.g., Hamming distance) and δ is a predefined threshold.

We point out that a common way for KeyTPLGen to generate $\mathtt{k}$ in those template protection with key release techniques mentioned in Section 2 is to first select a random $\mathtt{k}$, and then embeds it within x in one way or another which results in $\mathtt{tpl}$. The nice thing about KeyTPLGen is that the template $\mathtt{tpl}$ it generates does not disclose information on x. This is where "template protection" comes into play, and the template is often alluded to as "secure template" in the literature.

Substantiation. Given the template protection with key release techniques, it is a bit direct to materialize biometric wrapped credential, i.e., one uses $\mathtt{k}$ generated from her biometric to wrap her authentication credential. Concretely, let us assume the same setup as in the above anonymous password authentication system - BBS+ Signature is used to issue authentication credentials for anonymous entity authentication: let (M, k, s) be the authentication credential of user u; then her biometric wrapped credential is $\widehat{cred}_u = \langle M, [s]_{\mathtt{k}_u}, k, \mathtt{tpl}_u \rangle$, where $(\mathtt{k}_u, \mathtt{tpl}_u) \leftarrow \text{KeyTPLGen}(x_u)$ with x_u being a biometric sample of u. Note that k remains unprotected, as it will be accumulated in an Accumulator for handling user revocation, the same as in password wrapped credential. To unwrap $\widehat{cred}_u$, it is apparent that the user first captures a live biometric reading x'_u and then computes $\mathtt{k}_u \leftarrow \text{KeyRelease}(\mathtt{tpl}_u, x'_u)$.

The anonymous biometric authentication scheme will be the same as in Section 3, with biometric wrapped credentials substituting for password wrapped credentials. As a matter of fact, public key suppression is not necessary in anonymous biometric authentication, and the original BBS+ Signature can be used as it is. This is because biometrics are of high entropy, and there is no concern of brute force guessing attacks, whether they are offline or online. But it should be

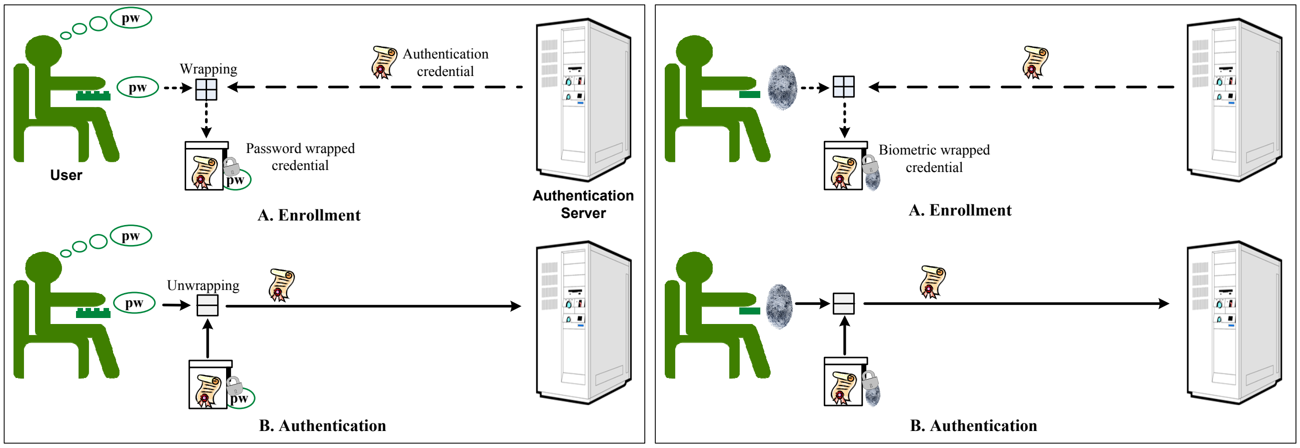

Figure 1: Conceptual Comparison

clear that enforcement of public key suppression would increase the entropy of biometric wrapped credentials, making the system more secure. This would be of help in case the adopted template protection with key release technique is not so strong (see “Caveat” in Section 5 for more details). Another point to note is that the mechanism to tackle online guessing attacks in anonymous password authentication are not necessary herein either, again because of high entropy of biometrics.

In a natural manner, existing literature on template protection with key release, e.g., [3, 6, 20, 22, 33], directly uses `k` for (regular) entity authentication. In comparison, the advantages of our “credential wrapping” strategy include generality and strengthened security. For generality, we can work over any template protection with key release technique and tap on the latest developments in the field. Specially, our approach caters to all the requirements set upon biometric authentication in [11], except multi-modality; but to embrace multiple biometric traits in our approach is a matter of trivial expansion. For strengthened security, there exist attacks [31] to some template protection with key release techniques such as Fuzzy Commitment and Fuzzy Vault, and some attacks assume the accidental disclosure of `k`. In our “credential wrapping”, since `k` is only required to be present in the “bootstrap” step (i.e., unwrapping), and is not involved in the authentication protocol at all, the chance for it to be disclosed is diminished.

Remark. The instantiation of biometric wrapped credentials essentially needs a kind of “biometric encryption”, i.e., only one’ genuine biometrics can do decryption. The fuzzy identity-based encryption [36] can use one’s biometrics as public key or identity, such that encryptions under biometric samples can be decrypted by the private key generated from the biometric template. Apparently, The fuzzy identity-based encryption scheme is not suitable to implement “biometric encryption” we desire.

4.3 A Tailored Alternative

For the sole purpose of anonymous biometric authentication, it can also directly issue BBS+ signature upon `k` generated from a user’s biometric, to be the user’s anonymous credential, e.g., replacing user identity u with `k` in issuing BBS+ signature. The user stores $\langle M, s, k\rangle$ without further protection, while recover `k` at the point of authentication to make the complete credential. This amounts to the aforementioned directly using `k` for entity authentication (anonymous entity authentication in our case), and is orthogonal to the above biometric wrapped credential paradigm, so we stop short of further elaboration.

4.4 Beyond “Closed Systems"

While we concentrate on achieving anonymous entity authentication in “closed” systems (i.e., the authentication credentials are used towards a particular server), biometric wrapped credentials clearly have applicability beyond closed systems and anonymous authentication. As said earlier, guessing attacks are no longer an issue for biometric wrapped credentials, thus nullification of the public verifiability of authentication credentials is no longer required. This suggests that the authentication credentials can be used as usual, to any party in an “open” system. In addition, a credential in wrapping is not restricted to be an anonymous authentication credential, and it can be any other secret quantities, e.g., the private/signing key in standard encryption/digital signature. In such cases, biometrics based wrapping actually acts as a secure “software smartcard” for managing one’s secrets, and has much wider applicability than mere client-server authentication.

4.5 Countering Online Guessing Attacks in A-PA with Biometrics

Recall that online guessing attacks are inevitable in password authentication, and can only be addressed at the system level. Since individual users are not discerned by the authentication server, online guessing attacks are even more troublesome in anonymous password authentication. The scheme in [42] addressed this issue with a virtual TTP (trusted third party) solution, which enlists enrolled users to help the server to scrape the anonymity of the users under attack (in other words, enrolled users and the authentication server together act as a TTP for anonymity scraping). While the idea is interesting, there may be operational difficulties in implementing the virtual TTP solution in practice.

Incorporating the biometrics factor into anonymous password authentication yields a better solution. Specifically, it is not hard to see that credential wrapping will be based on a key derived from both a password and k to harden the password, implementing the so called two-factor authentication. In this case, since the biometric factor is introduced to increase the entropy of passwords, a relatively higher FMR (False Match Rate: the probability of a non-genuine sample is interpreted as match) in biometrics can be tolerated, compared to pure biometric authentication. Moreover, it is preferred that the biometrics factor does not downgrade the portability of passwords. As such, behaviorial biometrics such as keystroke dynamics would suffice, as they arguably do not require the presence of biometric sensors.

5. EXPERIMENTS

To evaluate the feasibility of biometric wrapped credentials, we implemented and tested the anonymous biometric authentication protocol within a BYOD (Bring Your Own Device) prototype. Each user enrolls her smart phone to the system, during which among others, an authentication credential under BBS+ Signature is issued and then wrapped to generate the biometric wrapped credential. When a user anonymously logs in to the system, she is recognized as "Anonymous User" and is granted "read" privilege under mandatory access control to a collection of E-resources such as E-books, E-journals, internal reports and publications.

The biometric trait our experiments adopted is fingerprint, as there are public data sets for testing. In particular, we used the FVC2002-DB1 database [26], which contains images from 100 fingers with 8 impressions per finger. For each figure, we discarded the last 2 impressions in terms of quality, and used 1 for enrollment and the remaining 5 for authentication trials. The NIST NBIS mindtct algorithm [38] was used for minutiae extraction. The number of minutiae extracted per image varies, and some of the minutiae were selected based on quality and quantized to be 1920 bits according to ANSI INCITS 378-2004. For BBS+ Signature, we implemented pairing friendly MNT curves [25] with embedding degree 6, and the order of bilinear groups being 161.

5.1 Implementation Details

The crux of the implementation is the template protection with key release technique, which we chose to implement the idea of Fuzzy Extractor [3, 6]. Specifically, Figure 2 shows the diagram of the KeyTPLGen algorithm: k is computed as HMAC(x, r) by applying HMAC-SHA1 to a biometric sample x and a random number r; tpl includes two elements (d, r) with $d = x \oplus r_\mathsf{c}$, where r_c is a random codeword from the codeword space of a Reed-Solomon code.

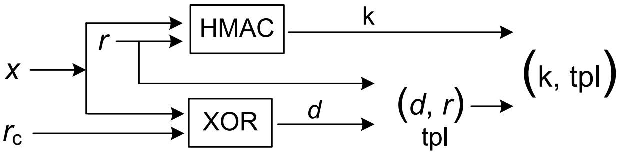

Figure 2: KeyTPLGen

We implemented Reed-Solomon $[1920, 768, 1153]_{2^{11}}$ which can correct up to $\frac{1153-1}{2} = 576$ bits of error (i.e., 30% of 1920). To compute d, each bit of the 1920-bit x is turned into a 11-bit symbol by simply padding with zeros. Such a coding guarantees that *at most* one bit in each symbol of r_c could be corrupted in our case where the corruption comes from the application of different biometric samples (but not from data transmissions as usual). The output of HMAC is 160 bits, so k can be directly used as an AES key for encryption of (M, s) in generating biometric wrapped credential.

The diagram of the KeyRelease is depicted in Figure 3. Given tpl $= (d, r)$ and a biometric sample x', the algorithm starts by computing $r'_\mathsf{c} = d \oplus x' = x \oplus r_\mathsf{c} \oplus x'$, which can be viewed as a corrupted version of r_c. If the number of bits that x and x' differ is less than 576, then the original r_c can be correctly restored, i.e., $r_\mathsf{c} = \text{Encode}(\text{Decode}(r'_\mathsf{c}))$, where Encode and Decode, respectively, are the encoding and decoding algorithms of Reed-Solomon. With r_c in place, x is reconstructed as $x = d \oplus r_\mathsf{c}$, and in turn k can be computed k $= \text{HMAC}(x, r)$.

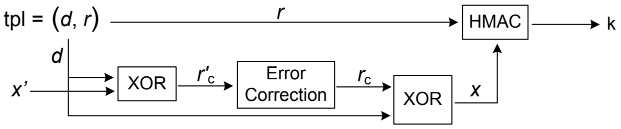

Figure 3: KeyRelease

Caveat. Normally care must be taken in selecting parameters of a Reed-Solomon code, due to the *list decoding* problem [15]. List decoding concerns the possibility to list all the codewords at a Hamming distance larger than the classic error correction capability of the Reed-Solomon code. Casting this to Fuzzy Extractor, it means that an adversary could find r_c used in KeyTPLGen even without going through the KeyRelease procedure, if the parameters of the code are not strong enough to offset the attack. We acknowledge that we did not pay special attention to list decoding in selecting the above Reed-Solomon code: list decoding is in fact not a serious threat in our case of public key suppression. The reason: even an adversary managed to list all codewords, the only way for him to decide the actual r_c is by enumerating every codeword to unwrap the biometric wrapped credential and then testing the recovered "credential" online with the authentication server. Even so, it is prudent to take the list decoding problem into serious consideration when considering practical deployment.

5.2 Experimental Results

The experiments measured two metrics which determine the feasibility of the proposal: one relates to biometric - FMR (False Match Rate) and FNMR (False Non-Match Rate: the probability of a genuine sample is falsely interpreted as non-match), and the other is performance of the anonymous biometric authentication protocol. The client program was coded as an Android App and tested upon a smartphone with a 1.2GHz CPU and 2.0GB RAM, and the server program was run on a PC, Intel 3.1 GHz CPU and 8GB RAM.

Table 2 reports the experimental results on biometric and the client/server computation performance. Specifically, in FMR tests, for each finger we used all other fingers' impressions against it (as per KeyRelease), and obtained 0.8% FMR; for FNMR tests, we tested each finger by using its own other 5 impressions against the one used in KeyTPLGen, and we ended up getting 1.2% FNMR (It should be straightforward in our context that by "matching" or "non-matching" in the FMR and FNMR tests, we mean that the original bio-

Table 2: Experimental Results

Biometric (%)		Client Performance (sec.)		Server Performance (sec.)
FMR	FNMR	Unwrapping	Authentication	0.12
0.8	1.2	1.6	2.58	

metric sample x used in the KeyTPLGen algorithm is recovered or not in the KeyRelease algorithm.). These are rather promising results. While experiments with larger and "lively captured" datasets would be more convincing, this result represents a basic indication on the feasibility of biometric wrapped credentials.

For computation performance, in order not to compound the experimental results we did not take into consideration the communications, and client program and server program were run in isolation. As to client App's performance, we measured separately, the time it takes to unwrap a biometric wrapped credential and the time to complete the client side computation of the anonymous authentication protocol: on average the former is 1.6 seconds and the latter 2.58 seconds. The server application takes about 120 milliseconds to complete the server side computation of the anonymous authentication protocol, averaging over 100 executions. These results suggest that biometric wrapped credentials have demonstrated practically acceptable performance even upon mobile devices.

6. CONCLUSIONS

Our main contributions are two-fold in this work. First, we made an improvement to a state-of-the-art anonymous password authentication scheme (under password wrapped credential) proposed in ACSAC'10, which is being standardized. The resulting scheme is neater, more efficient and deployment friendly. Second, we extended the concept of password wrapped credential to biometric wrapped credential, in a bid to achieve realistic anonymous biometric authentication. As expected, biometric wrapped credentials helped get over the linear server-side computation bound inherent in the typical setting of biometric authentication. We implemented and tested the proposed anonymous biometric authentication protocol, and the experimental results demonstrated the feasibility of anonymous biometric authentication under the auspices of biometric wrapped credentials.

7. ACKNOWLEDGMENTS

Joseph K. Liu is supported by National Natural Science Foundation of China (61472083).

8. REFERENCES

[1] M. Abdalla, M. Izabachene, and D. Pointcheval. Anonymous and transparent gateway-based password-authenticated key exchange. In *Proc. International Conference on Cryptology and Network Security - CANS'08*, pp. 133-148, 2008.

[2] M.H. Au, W. Susilo, and Y. Mu. Constant-size dynamic k-TAA. In *Proc. Security and Cryptography for Networks - SCN'06*, LNCS 4116, pp. 111-125, 2006.

[3] X. Boyen. Reusable cryptographic fuzzy extractors. In *Proc. ACM Conference on Computer and Communications Security - CCS'04*, pp. 82-91, 2004.

[4] X. Boyen. A tapestry of identity-based encryption: practice frameworks compared. *International Journal on Applied Cryptography*, Vol 1(1), pp. 3-21, 2008.

[5] E. Bresson, O. Chevassut, and D. Pointcheval. Security proofs for an efficient password-based key exchange. In *Proc. ACM Conference on Computer and Communication Security - CCS'03*, pp. 241-250, 2003.

[6] X. Boyen, Y. Dodis, J. Katz, R. Ostrovsky, and A. Smith. Secure remote authentication using biometrics. In *Proc. Advances in Cryptology - Eurocrypt'05*, pp. 147-163, 2005.

[7] M. Blanton and M.P. Hudelson. Biometric-Based Non-transferable Anonymous Credentials. In *Proc. Internation Conference on information and Communications Security, ICICS'09*, LNCS 5927, pp. 165-180, 2009.

[8] S. Bellovin and M. Merritt. Encrypted key exchange: password-based protocols secure against dictionary attacks. In *Proc. IEEE Symposium on Research in Security and Privacy - S&P'92*, pp. 72-84, 1992.

[9] V. Boyko, P. Mackenzie, and S. Patel. Provably secure password-authenticated key exchange using Diffie-Hellman. In *Proc. Advances in Cryptology - Eurocrypt'00*, LNCS 1807, pp. 156-171, 2000.

[10] J. Camenisch, etc. How to win the clonewars: efficient periodic n-times anonymous authentication. In *Proc. ACM Conference on Computer and Communication Security - CCS'06*, pp. 201-210, 2006.

[11] S. Cimato, M. Gamassi, V. Piuri, R. Sassi, and F. Scotti. Privacy-aware biometrics: design and implementation of a multimodal verification system. In *Proc. 24th Annual Computer Security Applications Conference - ACSAC'08*, pp. 130-139, 2008.

[12] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft. Privacy-preserving face recognition. In *Proc. International Symposium on Privacy Enhancing Technologies*, 2009.

[13] K. Emura, A. Miyaji, and K. Omote. An Anonymous Designated Verifier Signature Scheme with Revocation: How to Protect a CompanyŠs Reputation. In *Proc. 4th International Conference on Provable Security, ProvSec'10*, pp. 184-198, 2010.

[14] L. Gong, M. Lomas, R. Needham, and J. Saltzer. Protecting poorly chosen secrets from guessing attacks. *IEEE Journal on Seclected Areas in Communications*, Vol. 11(5), pp. 648-656, 1993.

[15] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Transactions on Information Theory*, Vol. 45(6), pp. 1757-1767, 1999.

[16] S. Halevi and H. Krawczyk. Public-key cryptography and password protocols. In *Proc. ACM Conference on Computer and Communication Security - CCS'98*, pp. 122-131, 1998.

[17] D. Hoover and B. Kausik. Software smart cards via cryptographic camouflage. In *Proc. IEEE Symposium on Security and Privacy, S&P'99*, pp. 02-08, 1999.

[18] Y. Huang, L. Malka, D. Evans, and J. Katz. Efficient privacy-preserving biometric identification. In *Proc. Network and Distributed System Security Symposium - NDSS'11*, 2011.

[19] ISO/IEC 20009 (Working Draft): Information technology - Security techniques - Anonymous entity authentication - Part 4: Mechanisms based on weak secrets. http://www.iso.org/iso/home/store/ catalogue_tc/catalogue_detail.htm?csnumber=64288.

[20] A. Juels and M. Sudan. A fuzzy vault scheme. In *Proc. IEEE International Symposium on Information Theory*, 2002.

[21] S. Jeyaraman and U. Topkara. Have the cake and eat it too - infusing usability into text-password based authentication systems. In *Proc. 21st Annual Computer Security Applications Conference - ACSAC'05*, pp. 473-482, 2005.

[22] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *Proc. ACM Conference on Computer and Communications Security - CCS'99*, pp. 28-36, 1999.

[23] M.M. King. Robus passwords. In *Proc. 7th Annual Computer Security Applications Conference - ACSAC'91*, pp. 239-243 , 1991.

[24] H.Y. Lin and W.G. Tzeng. Anonymous password based authenticated key exchange with bub-linear communication. *Journal of Information Science and Engineering*, Vol. 25(3), pp. 907-920, 2009.

[25] A. Miyaji, M. Nakabayashi, and S. Takano. New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 2001.

[26] D. Maio, D. Maltoni, J.L. Wayman, and A.K. Jain. FVC2002: second fingerprint verification competition. In *Proc. Internation Conference on Pattern Recognition*, pp. 811-814, 2002.

[27] L. Nguyen. Accumulators from bilinear pairings and applications. In *Proc. CT-RSA'05*, LNCS 3376, pp. 275-292, 2005.

[28] K. Nandakumar, A.K. Jain, and S. Pankanti. Fingerprint-base fuzzy vault: implementation and performance. *IEEE Transactions on Information Forensics and Security*, Vol. 2(4), pp. 744-757, 2007.

[29] M. Osadchy, B. Pinkas, A. Jarrous, and B. Moskovich. SCiFI: A system for secure face identification. In *Proc. IEEE Symposium on Security and Privacy - S&P'10*, pp. 239-254, 2010.

[30] H.F Qian, J.Q Gong, and Y. Zhou. Anonymous password-based key exchange with low resources consumption and better user-friendliness. *Security and Communication Networks*, Vol. 5(12), pp. 1379-1393, Wiley, 2012.

[31] W.J. Scheirer and T.E. Boult. Cracking fuzzy vaults and biometric encryption. In *Proc. IEEE Biometrics Symposium*, pp. 1-6, 2007.

[32] W.J. Scheirer and T.E. Boult. Bipartite biotokens: definition, implementation, and analysis. In *Proc. 3rd IAPR/IEEE International Conference on Biometrics - ICB'09*, pp. 775-785, 2009.

[33] W.J. Scheirer, W. Bishop, and T.E. Boult. Beyond PKI: the biocryptographic key infrastructure. *Security and Privacy in Biometrics*, pp. 45-68, Springer-Verlag, 2013.

[34] S. Shin, K. Kobara, and H. Imai. A secure construction for threshold anonymous password-authenticated key exchange. *IEICE Transactions on Fundamentals*, E91-A(11): 3312-3323, 2008.

[35] A. Sadeghi, T. Schneider, and I.Wehrenberg. Efficient privacy-preserving face recognition. In *Proc. International Conference on Information Security and Cryptology - ICISC'09*, pp. 229-244, 2009.

[36] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *Proc. Advances in Cryptology - EUROCRYPT'05*, LNCS 3494, pp 457-473, 2005.

[37] D. Q. Viet, A. Yamamura, and T. Hidema. Anonymous password-based authenticated key exchange. In *Proc. Advances in Cryptology - Indocrypt'05*, LNCS 3797, pp. 233-257, 2005.

[38] C.I. Watson *et al. User's Guide to NIST Biometric Image Software (NISB)*, 2007.

[39] X. Wang, M.H. Heydari, and H. Lin. An intrusion-tolerant password authentication system. In *Proc. 19th Annual Computer Security Applications Conference - ACSAC'03*, pp. 110-118, 2003.

[40] J. Yang and Z. Zhang. A new anonymous password-based authenticated key exchange protocol. In *Proc. Advances in Cryptology - Indocrypt'08*, LNCS 5365, pp. 200-212, 2008.

[41] Y.J. Yang, J.Y. Zhou, J. Weng, and F. Bao. A new approach for anonymous password authentication. In *Proc. 25th Annual Computer Security Applications Conference - ACSAC'09*, pp. 199-208, 2009.

[42] Y.J. Yang, J.Y. Zhou, J.W. Wong, and F. Bao. Towards practical anonymous password authentication. In *Proc. 26th Annual Computer Security Applications Conference - ACSAC'10*, pp. 59-68, 2010.