SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2014; **7**:385–396
Published online 22 February 2013 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.729

SPECIAL ISSUE PAPER

Dynamic combination of authentication factors based on quantified risk and benefit

Weili Han[1,2]*, Chen Sun[1], Chenguang Shen[1], Chang Lei[1] and Sean Shen[3]

[1] Software School, Fudan University, Shanghai, China
[2] Hangzhou Key Lab of E-Business and Information Security, Hangzhou Normal University, Hangzhou, China
[3] China Internet Network Information Center, Beijing, China

ABSTRACT

By combining multiple factors during authentication, a service can provide better assurance of security. However, the users are likely to feel inconvenient, or even discard the service. This paper, therefore, addresses this issue and introduces a novel method, referred to as the Quantified riSk and Benefit adaptive Authentication Factors combination (QSBAF). QSBAF balances the requirements for both security and usability in the authentication of an information system and improves the system's ability to respond quickly to emerging risky events. In QSBAF, the authentication factors can be dynamically combined on the basis of quantified risk, benefit measurements, and combination policies. Furthermore, QSBAF provides an adaptive mechanism, which is driven by history data to justify the measurements of risk and benefit. In this paper, we use the online banking system as a typical scenario to demonstrate the usage of QSBAF. We also implement a prototype of QSBAF to evaluate the performance of its feasibility in real application scenarios. Copyright © 2013 John Wiley & Sons, Ltd.

KEYWORDS

quantified risk; quantified benefit; QSBAC; multiple factors authentication; usability

***Correspondence**

Weili Han, Software School, Fudan University, Shanghai, China.
E-mail: wlhan@fudan.edu.cn

1. INTRODUCTION

Nowadays, a service provider of an information system, for example, an online banking system, often encounters the dilemma of whether to strengthen protection first or improve usability first during users' authentication. In an information system providing a sensitive service, the authentication module usually obtains the most security concerns [1], because it is usually the first check point of the service. An authentication process usually uses certain secrets to authenticate the user. Here, we define a single secret, for example, password, Public-Key Infrastructure certificate, even a phone code, during the authentication as an authentication factor. A single authentication factor can only provide limited information about the user's digital identity. And such authentication process is vulnerable under certain types of attacks, for example, the brute force attack, phishing [2]. Molloy *et al*. [3] pointed out that even one-time password (OTP), which is generally regarded safe, can also be accessed by attackers.

Therefore, a secure information system should combine multiple factors to strengthen the protection of the information system. For example, an online banking system requires a two-factor USBKey + password process. However, the enhancement of the system's security is at the expense of its usability. With complex authentication factors, users tend to feel inconvenient, or even discard the service finally. The latter is unacceptable for the service provider.

The current approach to set multiple authentication factors, for example, password + USBKey or password + OTP, of an information system is often based on technical concerns only. The system administrators hold face-to-face discussion to determine the authentication factors based on experience and technical feasibility. Then, static policies of factors combination are made and enforced in the information system during the authentication.

The aforementioned way, however, may lead to the problem of overprotection, if the administrators set multiple authentication factors based on technical concerns. Furthermore, because of the static policies, this way is too rigorous to meet the rapidly changing system context today. These drawbacks will lead administrators to set the most restricted protection for all potential risky scenarios, which will weaken the usability of the system. Moreover, the

system cannot react to the attacks in time. For example, a phishing attack which happened in the online banking system of Bank of China (BOC) in 2011. A victim would be lured to a phishing site and input his or her password and OTPs (two distinct passwords from a SecurID Token) [4]. As a reaction, BOC spent more than 2 months reconstructing its authentication module and later required a three-factor (password, OTPs, and SMS Token) method for transfer and payment of any amount.

Given the aforementioned drawbacks, we leverage the quantified risk and benefit view to set authentication factors and then propose the Quantified riSk and Benefit adaptive Authentication Factor combination (QSBAF) to address the aforementioned issues. We use quantified risk and benefit, because they are two important factors during the decision process [5–7], especially in emergent scenarios.

The main contributions in the paper are as follows:

1) We propose a framework of QSBAF, which can dynamically select the authentication factors based on quantified risk and benefit. In QSBAF, the quantified risk and benefit of access requests are explicitly measured according to the historical data and are used to determine which factors should be combined to authenticate a user. QSBAF can also dynamically update the historical data, and then the measured results will be up to date, especially when a risky event happens.
2) We use online banking system as a typical motivated scenario of QSBAF. We model and define the measurement of quantified risk and benefit within the context of online banking systems. We also propose two approaches for combination policies: the fuzzy inference approach and the risk mitigation approach.

The rest of this paper is organized as follows. Section 2 discusses the background and the motivated scenario of QSBAF, Section 3 introduces the framework of QSBAF. Section 4 describes the quantified risk and benefit measurement and the combination policies by applying QSBAF in an online banking system. Section 5 describes the prototype setup and evaluates the performance of the QSBAF framework. Section 6 discusses other important issues in QSBAF. Section 7 summarizes related work of QSBAF, and Section 8 concludes this paper and introduces the future work of QSBAF.

2. BACKGROUND AND MOTIVATED SCENARIO

2.1. Quantified risk and benefit adaptive access control

Risk and benefit pervasively exist in all accesses to sensitive information [5,6]. In a traditional policy-based access control system, the risk and benefit are implicitly considered. Such a mechanism is rigorous [6,8] and is not a satisfying way to deal with emergent risky access requests.

Therefore, Han *et al.* [6] proposed the framework of Quantified Risk and Benefit Access Control (QSBAC) to strengthen the security of information sharing. The framework of QSBAC mainly consists of the following components: access context, two types of risk (risk of allowing access (RAA) and risk of denying access (RDA)), two types of benefit (benefit of allowing access (BAA) and benefit of denying access), risk mitigation actions (RMA), benefit incentive actions (BIA), an inference engine, decision policies, and a feedback mechanism. The main features of the QSBAC framework are as follows:

(1) In an information system, the quantified risk and benefit of access requests are explicitly measured. The measured quantified risk and benefit give direct supporting evidence for the inference of the access control decision.
(2) Quantified Risk and Benefit Access Control system is policy-driven, where system administrators should preset security policies. The policies are presented in QSBAC-eXtensible Access Control Markup Lanuguage (XACML), an extended version of XACML [9], which represents the quantified risk and benefit features of QSBAC policies.
(3) Quantified Risk and Benefit Access Control uses fuzzy sets instead of Boolean values to represent quantified risk and benefit measurement results. The fuzzy inference method is employed in the policy engine to generate final decision results.

Key steps to enforce QSBAC in an information system are as follows:

(1) Set RMA and BIA. RMA refers to the actions or modules that will reduce the risk of the system by applying them. Similarly, BIA refers to the actions or modules that will increase the benefit of the system by applying them. Both RMA and BIA will allow or deny a request.
(2) Set quantified risk and benefit adaptive policies. Security administrators must set QSBAC policies before the system runs. These QSBAC policies mainly deal with the undefined situations, especially emergent or dynamic application situations.
(3) Measure quantified risk and benefit. The four variables (RAA, RDA, BAA, and benefit of denying access (BDA)) will be measured by preset measurement functions. Meanwhile, applicable RMA and BIA will be considered.
(4) Determine a request according to the policies and measurements. When a request arrives, the policy decision point will evaluate the request and return a result, which includes a decision (to allow or deny the request), necessary RMA, and BIA.
(5) Implement the response with RMA and BIA. A QSBAC-enabled secure system will execute the

DOI: 10.1002/sec

RMA and BIA included in the decision result in addition to denying or allowing the request. The actions could be either on a per-transaction basis or long-term.

(6) Record the runtime context as a data source to optimize the RMA, the BIA, the policies, and the measurement functions of quantified risk and quantified benefit.

As is extended from XACML to express the policies, the following elements are included in XACML [6]. ⟨*PolicySet*⟩ mainly consists of sub policy sets, policies, and obligations; ⟨*Policy*⟩ mainly consists of the following elements: a ⟨*Target*⟩, a set of ⟨*Rule*⟩, a rule-combining algorithm-identifier, an ⟨*ObligationExpressions*⟩, and an ⟨*AdviceExpressions*⟩; and the ⟨*Rule*⟩ element mainly consists of the following elements: a ⟨*mathitTarget*⟩, an ⟨*Effect*⟩, a ⟨*Condition*⟩, an ⟨*ObligationExpressions*⟩, and an ⟨*AdviceExpressions*⟩.

In addition, QSBAC uses fuzzy sets rather than thresholds to define the policies based on measured quantified risk and benefit. The main difference between fuzzy sets and thresholds lies in that a fuzzy method will return a result, as well as a percentage of that membership, for example, a temperature is 70% high, whereas a threshold only returns the result high.

Moreover, after assigning the percentage of membership, the system will infer a result according to the fuzzy logic rather than Boolean logic.

The information systems today are usually complex and must process a large amount of access requests in highly dynamic system contexts. Therefore, we apply the features of QSBAC in QSBAF to address the problem of dynamic authentication of information systems.

2.2. Multiple factors authentication

According to the 2008 guidance proposed by the Federal Financial Institutions Examination Council (FFIEC) [10], the term authentication describes the process of verifying the identity of a person or an entity. According to the FFIEC guidance, authentication methodologies involve three types of basic 'factors'. Something the entity ***knows*** (e.g., password); something the entity ***has*** (e.g., OTP Token, USBKey); and something the entity ***is*** (e.g., fingerprint).

Evolving from the aforementioned three basic categories, currently, there are many prevailing authentication factors, such as username/password, OTP Token, USBKey, SMS Token, and biometric features. Because a single factor only has limited protection strength, a secure information system usually employs two or more authentication factors to enhance the protection strength of the authentication.

2.3. Motivated scenario

Online banking system is a popular service offered by major banks in the world, including those in China. According to Rabkin *et al*. [11], many users feel the convenience of electronic access from personal computers irresistible, despite the possible security risks. Meanwhile, by cracking users' tokens, criminals have found the online banking system an appealing target. From the study by Franklin *et al*. [12], there is a large underground criminal market to trade bank credentials stolen online. These crimes towards online banking systems have forced the use of stronger security protection in these systems. Furthermore, online banking systems should take usability into consideration when setting security policies.

We investigate the authentication factors used by major Chinese online banking systems, as is shown in Table I. From the table, we can see that some of the online banking systems (e.g., CCB) use relatively simple authentication factors. Some banks (e.g., ICBC and ABC) employ more authentication factors, and users must conduct a series of extra steps to finish one transaction request.

A major drawback in the current authentication process is the enforcement of static security policies. As is stated in Section 1, static policies imply the slow response to emerging risky events. Each authentication factor has its own limitation and vulnerability and under certain circumstances, online banking systems might need to change one or more factor(s). A notable example lies with the RSA SecurID [13]. RSA, a U.S. network security provider, announced that they were under an extremely sophisticated cyber attack in March, 2011 [4]. This raised the panic among online banking users whose OTP Tokens were based on RSA's SecurID. Victim banks, such as BOC, however, did not make any rapid response to this risky event, which later on incurred serious criticism from its users about the security of the system.

The rigorousness of current security policies has motivated us to apply QSBAF to online banking systems. The dynamic feature of QSBAF will help the online banks to better deal with emerging risky events and other dynamic

Table I. Authentication factors used by major online banking systems in China.

Platform/bank	Log-in authentication factor	Transaction authentication factor
AliPay	Password (browser plug-in)	Password + browser certificate
BOC	Password + image CAPTCHA	SMS Token + OTP Token
ABC	Password + browser certificate	Browser certificate + OTP Token
ICBC	PIN + password	USBKey + OTP Token
CCB	PIN + password	Password + USBKey

BOC, Bank of China; ABC, Agricultural Bank of China; ICBC, Industrial and Commercial Bank of China;CCB, China Construction Bank.

contexts, in the meantime balancing the protection strength and usability of authentication.

3. FRAMEWORK OF QUANTIFIED RISK AND BENEFIT ADAPTIVE AUTHENTICATION FACTORS COMBINATION

As is shown in Figure 1, the main components of QSBAF are the quantified risk/benefit measurement engine, the history database, the decision engine, the policy repository, and the factor pool.

Before an access request arrives, the system administrator should first of all analyze and model the context of the information system and define the measurement models of quantified risk and benefit in the quantified risk/benefit measurement engine. Such a definition process may involve the discussion among the system managers. Previous risk-based access control mechanisms [8,14–16] are basically risk-aware but fail to consider another main factor benefit. Adding the factor of benefit can allow access requests with high benefit in dynamic contexts. For example, a transaction with income for a bank takes priority over a transaction without income for the bank.

The system administrator should then define the security policies of the system in the policy repository. These policies will give evidence of which authentication factor(s) should be used and combined under certain quantified risk and benefit. We will later illustrate the policies of QSBAF within the scenario of an online banking system.

The system administrator must also set the applicable authentication factors for the system in the factor pool. The selection of authentication factors should take user experience into consideration and should cover different categories of authentication methods.

After the aforementioned settings are ready, QSBAF can work to protect the information system.

When a user launches an access request (e.g., log into an account) to an information system, the quantified risk/benefit measurement engine works to measure the risk and benefit: firstly, it retrieves supporting data from the history database and then measures the quantified risk and benefit according to the preset models.

Later, the measured quantified risk and benefit are transferred to the decision engine. This engine will find the matched applicable policies from the policy repository. Then, the decision engine combines the policies according to the combination method discussed in Section 4.2. The engine finally generates a result, which indicates the authentication strength associated with this access request.

After getting the authentication strength, the decision engine selects one or more authentication factor(s) from the factor pool to enforce them. Higher authentication strength will impose higher security requirements on the system and generally calls for more authentication factors. If the authentication strength is too high, the

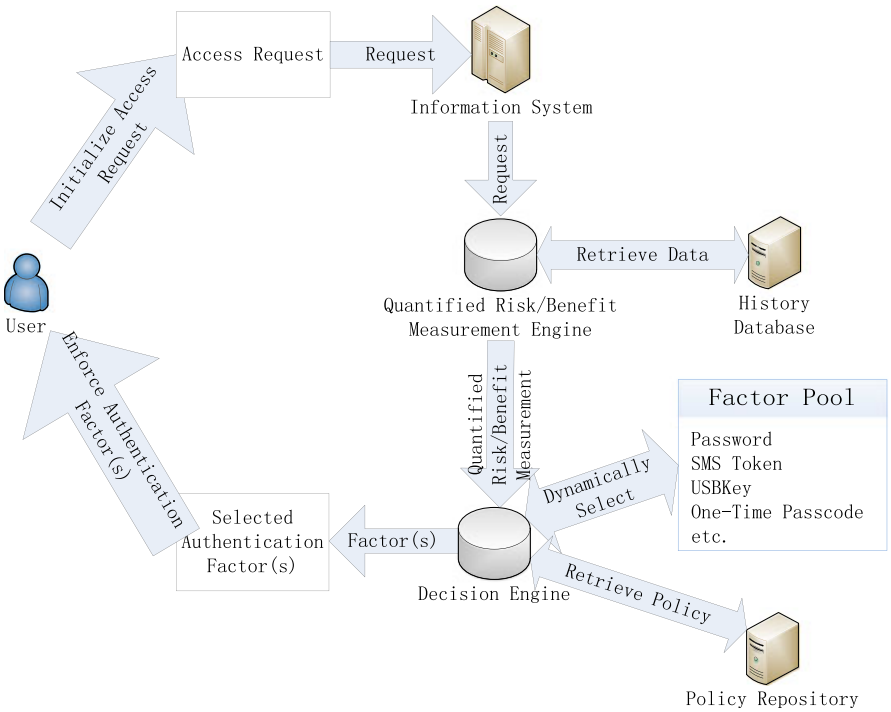

Figure 1. Framework of Quantified riSk and Benefit adaptive Authentication Factors combination.

access request might be denied directly then. Note that, we assume the system's cost will not increase if one authentication factor is added because usually, the relevant devices (e.g., OTP Token, USBKey) have already been delivered to the users with the establishment of bank accounts. If the request is not denied directly, this access request finishes with the enforcement of selected authentication factors.

In QSBAF, the history database stores necessary information to provide evidence for quantified risk and benefit measurement. Each request and the corresponding decision (to allow or deny) are stored in the database. For a single request, this necessary information includes the subject, object, access time, and access context of the request. If the request is allowed, the combination of authentication factor(s) is also stored, as well as the risk and benefit brought by this request, into the history database. To provide useful evidence for the quantified risk and benefit measurement, the database also records a few global variables as follows:

(1) Information about account disclosure. Each event of account disclosure is recorded, and the account loss caused by this disclosure is audited by the system administrator and then stored along with the disclosure. This will provide useful information for the quantified risk/benefit measurement engine to evaluate the quantified risk of log-in behavior.
(2) Information about malicious transaction. Similar to information about account disclosure, each malicious transaction and the corresponding loss are recorded in the database. This will provide useful information for the measurement engine to evaluate the quantified risk of transaction behavior.
(3) Information about financial income. The database stores the total financial income of the online banking system. For instance, with the help of the number of allowed requests, it is possible to calculate the average indirect income related to each log-in request. This helps the measurement engine to evaluate the incoming benefit of a new access request.

In addition, the database records the emerging events and the responses provided by the system administrator. This could assist the system administrator to better deal with new risky events.

The QSBAF system implements adaptive mechanism. The history database provides supporting evidence for the quantified risk/benefit measurement engine and the decision engine. After each decision is made, the history database gets updated with the decision information. Before a user launches a new request, the quantified risk/benefit measurement engine will retrieve the information from the history database and use the updated information to calculate the quantified risk and benefit. In this way, the calculated quantified risk and benefit directly reflect the effect of the newest historical risky events on the current request. And the change of the context will be recorded by the history database. Therefore, the QSBAF mechanism responds to the system context adaptively.

4. APPLY QUANTIFIED RISK AND BENEFIT ADAPTIVE AUTHENTICATION FACTORS COMBINATION IN ONLINE BANKING SYSTEM

4.1. Quantified risk and benefit measurement

According to Cheng *et al.* [8], essentially, risk is about some damage that may occur in the future. In QSBAF, risk can be categorized into the RAA and the RDA. RAA is caused by allowing the subject to access the object. In the context of online banking system, this could be brought by allowing a fake user to log into the online banking system (account disclosure), or by allowing a malicious user to conduct a transaction (malicious transaction). RDA is caused by denying the subject to access the object. This could be brought by denying a normal user to log into the system and therefore it increases the probability of the user to discard this bank account. According to previous researchers, some have studied the RAA [8,14,17,18]. However, few have explicitly used measured RDA. RDA is very important in that high RDA will increase the system's tendency to allow the access.

In addition to risk, QSBAF also considers benefit. In QSBAF, benefit consists of the BAA and the BDA. BAA is caused by allowing the subject to access the object. This could be caused by the increase of market sharing brought by an access request. BDA is caused by denying the subject to access the object.

In this paper, we analyze the online banking system and provide the measurement models of RAA, RDA, BAA, and BDA. We select two typical requests that users launch in online banking systems: log-in and transaction (payment and transfer). We present how to measure the quantified risks and benefits for both cases, respectively.

4.1.1. Log-in

A user usually logs into an established account to perform transactions. This operation is the most common operation in an online banking system.

Risk of allowing access measurement. When a user launches log-in request into an online banking system, RAA mainly comes from the risk associated with account disclosure. If a fake user has logged into an account, he or she is likely to cause account loss. In other words, RAA denotes the expected loss of this log-in request. The raw value of RAA in log-in is calculated as follows:

$$RAA_{raw} = Balance \times DisclosureProb$$

Here, *Balance* is the current balance of the account. *DisclosureProb* denotes the probability of account disclosure

caused by this log-in request. This probability is related to the total disclosure loss T of this bank in the past period, for example, 3 months. T can be measured by

$$T = \Sigma_{i=1}^{n} damage(i)$$

where n is the number of account disclosure in this online banking system in the past period, for example, 3 months. $damage(i)$ is the loss caused by the ith disclosure. The relationship between *DisclosureProb* and T is defined by the system administrator.

Risk of denying access measurement. Risk of denying access mainly occurs when a user (owner of the account) chooses to discard the account after being denied access for a certain number of times. The raw value of RDA in log-in is calculated as follows:

$$RDA_{raw} = Balance \times DiscardProb$$

Here,

$$DiscardProb = \frac{count_{denial}}{bound_{denial}}$$

DiscardProb denotes the probability that the user will discard this account if he or she is denied during log-in. $count_{denial}$ is the total number of log-in denials that a user has encountered since the establishment of the account. $bound_{denial}$ is a constant preset by the system administrator and represents the number of log-in denials that a typical user can tolerate before discarding the account.

Benefit of allowing access measurement. Benefit of allowing access of log-in consists of two parts: the indirect part and the market part. The indirect part comes from the financial income. This income can be viewed as an average transaction benefit brought by the user behaviors (i.e., payment) after logging into this account. The market part comes from the market share increase brought by this log-in request. It refers to that if the user logs in and uses the service provided by a certain bank, this bank will increase its market share directly.

The raw value of BAA is calculated as follows:

$$BAA_{raw} = IndirectIncome + MarketShareIncome$$

Here,

$$IndirectIncome = \frac{totalIncome}{totalAccess}$$

IndirectIncome is the indirect financial income brought by allowing the user to log into the account and can be calculated by the total financial income *totalIncome* coming from the past *totalAccess* times of log-in. *MarketShareIncome* is the income associated with the increase of the bank's market share. It is caused by this log-in request and is preset by the system administrator.

Benefit of denying access measurement

$$BDA = 0$$

In online banking systems, there is no explicit benefit of denying a log-in request. Therefore, we set BDA to 0.

4.1.2. Transaction (payment and transfer)

There are mainly two types of transactions in online banking systems: payment and transfer. Payment is the process of paying a certain amount of money from the bank account to third party organizations. Transfer is the behavior to move a certain amount of money from the owner's bank account to another account, which can be in the same bank or in another bank.

Risk of allowing access measurement. Risk of allowing access of a transaction mainly comes from the possible loss associated with this payment or transfer request. If this request is a malicious one, the amount of this payment or transfer would be lost. Therefore, RAA is related to the amount of money involved in this payment or transfer, and the probability that this request is malicious. In other words, RAA denotes the expected loss associated with this transaction request. The raw value of RAA in transaction is calculated as follows:

$$RAA'_{raw} = Amount \times MaliciousProb$$

Here, *Amount* is the transaction amount of this payment or transfer request. *MaliciousProb* is the probability that this transaction request is malicious. This probability is related to the total loss T' of this bank caused by malicious transactions in the past 3 months. T' can be measured by

$$T' = \Sigma_{j=1}^{m} loss(j)$$

where m is the number of malicious transactions in this online banking system in the past 3 months. $loss(j)$ is the amount of loss caused by the jth loss. The relationship between *MaliciouseProb* and T' is defined by the system administrator.

Risk of denying access measurement. Risk of denying access of a transaction occurs when the user chooses to discard the account after the transaction is being denied or a certain number of times. It is exactly the convenience of online transaction that attracts the user to use online banking systems. However, if the legitimate transaction requests from users are often denied, the users are likely to stop using the account provided by a specific bank. The raw value of RDA in payment or transfer is calculated as follows:

$$RDA'_{raw} = Amount \times DiscardProb'$$

Here,

$$DiscardProb' = \frac{count'_{denial}}{bound'_{denial}}$$

Amount is the transaction amount of this payment or transfer request. $DiscardProb'$ denotes the probability that the user will discard this account if he or she is denied in this transaction this time. $count'_{denial}$ is the total number of transaction denials that a user has encountered during the past year. $bound'_{denial}$ is a constant preset by the system administrator and represents the number of transaction denials that a typical user can tolerate before discarding the account.

Benefit of allowing access measurement. Benefit of allowing access of a transaction consists of the direct part and the market part. The direct part is the obvious fee charged by the bank on this payment or transfer transaction. The market part comes from the market share increase brought by this transaction. The raw value of BAA is calculated as follows:

$$BAA'_{raw} = DirectIncome + MarketShareIncome'$$

Here,

$$DirectIncome = charge$$

DirectIncome is the direct financial income brought by allowing the user to perform this transaction and is equal to the fee charged on this request. The charge value is preset by the system administrator. $MarketShareIncome'$ is the income associated with the increase of the bank's market share. It is caused by this transaction and is also preset by the system administrator.

Benefit of denying access measurement In online banking systems, there is no explicit benefit of denying a transaction request. Therefore, we set BDA' of a transaction request to be 0.

4.1.3. Converting raw value by using Sigmoid function

By using the Sigmoid function, the final measurement function of RAA, RDA, and BAA in log-in and transaction is shown in the following:

$$Obj = \frac{1}{1 + exp^{(-k_{Obj}) \times (Obj_{raw} - mid_{Obj})}}$$

where Obj may be RAA, RAA', RDA, RDA', BAA, and BAA'. *k* and *mid* are two series of constants preset by the system administrator. They decide how the raw values of RAA, RAA', RDA, RDA', BAA, and BAA' can be converted to a final value in the domain (0..1).

4.2. Dynamic combination of authentication factors

In this paper, we propose two approaches to combine factors during authentication: the fuzzy inference approach and the risk mitigation approach.

Fuzzy inference approach. In QSBAF, we employ fuzzy logic to infer the rules to obtain the suitable authentication factors combination. The variables RAA, RDA, BAA, and BDA are represented by fuzzy sets instead of Boolean values. For example, RAA can be 30% high or 70% low instead of simple high or low. Thus, using fuzzy set to represent variables gives more flexibility for inference. The memberships of RAA, RDA, BAA, and BDA include high, mid, and low.

We add a variable authentication strength to denote the necessary authentication factors required by particular RAA, RDA, BAA, and BDA. The higher the authentication strength is, the stronger authentication factor(s) is(are) required. There are six memberships for authentication strength: extremely safe, safe, normal, suspicious, dangerous, and highly dangerous in QSBAF.

An example policy is shown in the succeeding text.

P1: If RAA is low, RDA is mid, BAA is high, and BDA is low, the authentication strength is safe.
P2: If RAA is low, RDA is mid, BAA is mid, and BDA is low, the authentication strength is normal.

The decision engine employs fuzzy logic to infer the decision result. The main steps of inference are listed as follows [16]:

(1) Fuzzification. Use pre-defined membership functions to assign membership to each variable (RAA, RDA, BAA, and BDA).
(2) Inference. The decision engine infers the results generated by each applicable policy. MIN inference algorithm is used here.
(3) Composition. The decision engine combines all inference results. MAX composition algorithm is used here.
(4) Defuzzification. The decision engine generates the authentication strength value from the composition result. CENTROID algorithm is used here.

After obtaining the final authentication strength value, the engine will use the value to decide the membership of the authentication strength. That is, which of the six memberships the current strength falls in. If the strength is too high, the system may deny the request instead of finding necessary authentication factors. Each membership corresponds to a set of authentication factors from the factor pool.

Risk mitigation approach. In this approach, we view each authentication factor, except password, as a risk mitigation

action, and we aim to combine authentication factor(s) to mitigate risk. Variables (RAA, RDA, BAA, and BDA) are represented by fuzzy sets too. Each of the four variables has three memberships; (high, mid, and low). The policies in this approach have a variable result, which represents the decision result (to allow or deny) associated with particular RAA, RDA, BAA, and BDA values. The system has a set of policies that define what kind of access request can be allowed. For example,

P3: If RAA is low, RDA is low, BAA is mid, and BDA is low; then result is allowed.

P3 refers that low RAA is acceptable when RDA is low, BAA is mid, and BDA is low. In fact, as each authentication factor can mitigate certain amount of quantified risk (mainly RAA), the system's goal is to enforce authentication factors with enough strength to reduce RAA, bringing the quantified risk under an acceptable level. A system administrator is required to define the risk mitigation effectiveness of each authentication factor. The decision engine will use the definitions to select suitable authentication factor(s), which will then be combined with password. For instance, if RAA is mid in a transfer transaction where only password is used, we can use an OTP Token to mitigate the risk. RAA then becomes low. Or if RDA is low, BAA is mid, and BDA is low in the transaction, then the request will be allowed if password + OTP Token are enforced according to P3.

Dynamic feature. In both approaches, because the values of RAA, RDA, and BAA are measured from the updated history database, the selection of authentication factors is dynamic. According to Section 4.1, *DisclosureProb*, *DiscardProb*, *IndirectIncome*, *MaliciousProb*, and *DiscardProb* are all related to the historical data, which are stored in the history database. As the history database gets updated, some of the aforementioned values may change, which would lead to the change of RAA, RDA, or BAA. Therefore, the decision engine may infer a different decision in response to the dynamic request context.

4.3. Case study

We use a sample case to illustrate how QSBAF can dynamically select authentication factors according to measured quantified risk and benefit.

Define measurement. The system administrator defines the constants in the quantified risk/benefit measurement engine. This mainly includes the definition of the series of k and *mid*. The administrator should also define the function mapping from T to *DisclosureProb* (See in Table II) and from T' to *MaliciousProb* (See in Table III). The first mapping is used in RAA measurement for log-in, and the second mapping is used in RAA measurement for transaction (payment or transfer).

Table II. Total disclosure loss and disclosure factor.

T(*Amount*: *RMB*)	*DisclosureFactor*
<1000	0.1
1000–10,000	0.3
10,000–100,000	0.5
100,000–1,000,000	0.7
≥1,000,000	1.0

Table III. Total transaction loss and malicious factor.

T′(*Amount*: *RMB*)	*MaliciousFactor*
<500	0.1
500–5000	0.3
5000–50,000	0.6
50,000–500,000	0.8
≥500,000	1.0

Define policies In the fuzzy inference approach, the system administrator defines the relationship between authentication strength and enforced authentication factors, as is shown in Table IV.

In the risk mitigation approach, the system administrator defines risk mitigation consequence of each authentication factor as follows: OTP Token can reduce 0.3, SMS Token can reduce 0.3, and USBKey can reduce 0.4. In the table, the effect column refers that the change in RAA is caused by a given authentication factor. For example, the second row means that adding the factor of OTP can reduce RAA by 0.3. Such decrease may result in a different membership of RAA. For example, RAA changes from mid to low so that P3 may be met and the request can thus be allowed.

User initiate a transaction request User Alice is planning to buy a laptop worth RMB 5000 on a third party e-commerce site. She chooses to make a payment through an online banking system. The site directs her to the web site of the selected online banking system. Alice logs into the online banking system, the detail of which is ignored because we only focus on transaction request in this case.

Table IV. Authentication strength and factors.

Authentication strength and range	Selected factors
Extremely safe, <0.1	Password + CAPTCHA
Safe, 0.1–0.3	Password + OTP Token
Normal, 0.3–0.5	Password + SMS Token + OTP Token
Suspicious, 0.5–0.7	Password + SMS Token + USBKey
Dangerous, 0.7–0.8	Password + SMS Token + USBKey + OTP Token
Highly dangerous, ≥0.8	Deny

Security Comm. Networks 2014; **7**:385–396 © 2013 John Wiley & Sons, Ltd.
DOI: 10.1002/sec

Quantified risk and benefit measurement The quantified risk/benefit engine calculates that the total loss caused by malicious transactions in the past 3 months is $T' = 1,500$ (RMB). According to the relationship between T' and *MaliciousProb*, $MaliciousProb = 0.3$. Therefore, $RAA'_{raw} = 300$. The presets are $k'_{RAA} = 1/400$ and $mid'_{RAA} = 850$, and the engine calculates RAA = 0.202.

Since the establishment of the account, Alice has been denied a transaction request for 12 times, so $count'_{rej} = 12$. The presets are $bound'_{rej} = 60$ and $DiscardProb' = 0.2$. Therefore, $RDA'_{raw} = 200$. The presets are $k'_{RDA} = 1/300$ and $mid'_{RDA} = 200$, and the engine calculates RDA = 0.500.

The current charge of each transaction is 5, so $DirectIncome = 5$. In addition, $MarketShareIncome' = 5$, therefore, $BAA'_{raw} = 10$. The presets are $k'_{BAA} = 1/25$ and $mid'_{BAA} = 20$, and the engine calculates BAA = 0.401.

As is stated in Section 4.1, BDA = 0.

Dynamically combine factors For the fuzzy inference approach, the decision engine selects matched policies and combines their results using fuzzy inference. The resulting authentication strength is located in the range of *safe*. Therefore, the engine selects password and OTP Token from the factor pool to enforce and allows Alice's transaction request.

In the risk mitigation approach, the engine categorizes RAA as mid, RDA as high, BAA as mid, and BDA as low. It then finds the corresponding policy.

P4: If RAA is low, RDA is high, BAA is mid, and BDA is low, then the result is allowed.

We can see that the current request has higher RAA for the system to allow. In order to mitigate the extra RAA and make the total risk acceptable, the engine selects OTP Token to reduce RAA from mid to low. Finally, the transaction request of Alice is allowed with the RMA: password and OTP Token.

5. IMPLEMENTATION AND EVALUATION

To better evaluate our model of QSBAF, we set up a prototype of QSBAC to test QSBAF's efficiency with different number of variables and fuzzy sets.

5.1. Prototype setup

We designed a prototype based on Enterprise XACML Implementation and ran it on a computer with the configuration of CPU: Pentium; Memory: 1G; Operating System: Windows XP; JDK: JDK6.

On the basis of the implementation of XACML [6], we add the following new features to complete the QSBAF mechanism.

(1) Risk measure point (RMP) and benefit measure point (BMP). RMP is responsible for measuring risks (RAA and RDA), whereas BMP is for measuring benefit (BAA and BDA).
(2) Two types of obligation services are as follows: RMA service, which enforces the RMA and BIA service, which enforces the BIA in the response.

5.2. Evaluation

We designed four test cases to evaluate the performances of the decision engine. There can be at most three fuzzy sets (high, middle, and low) and four variables (RAA, RDA, BAA, and BDA). Because the number of variables and fuzzy sets can be cut out, we choose two fuzzy sets for each three variables in Test Case 1 ($2^3 = 8$), two fuzzy sets for each four variables in Test Case 2 ($2^4 = 16$), three fuzzy sets for each three variables ($3^3 = 27$) in Test Case 3, and three fuzzy set for each four variables ($3^4 = 81$) in Test Case 4.

As is shown in Table V, MIN time refers to the average response time consumption when only one rule is evaluated in the decision engine. MAX time refers to the average response time consumption when the maximum number of rules of each test case are evaluated in the decision engine. In Test Case 1, the average response time is from 81.9 to 88.8 ms with the increase of rules. Similarly, in Test Case 2, the average response time is from 84.2 to 93.8 ms. In Test Case 3, the average response time is from 81.4 to 92.9 ms. In Test Case 4, the average response time is from 86.8 to 106.7 ms with the maximum time appears when the number of rules reaches 81, the greatest of all the four test cases. The time of the decision engine response has a linear growth and its range is from 82 to 107 ms. Generally, this time is acceptable for online information systems to generate the decisions.

We can observe from the aforementioned experiment that the increase of rules will not have a significant impact on the time of decision. The administrator could set or adjust the number of variables without worrying about the general performance of the systems.

Even if only two variables are considered (Cases 1 and 2), which is the case for usual risk-aware access services, there is no much difference on time consumption.

Furthermore, the BMA and BIA services perform timely in response to the decision, which makes the selected authentication factors applicable in real scenarios.

Table V. Performance of test cases.

Case	MAX number of rules	MIN time (ms)	MAX time (ms)
1	8	81.9	88.8
2	16	84.2	93.8
3	27	81.4	92.9
4	81	86.8	106.7

6. DISCUSSION

6.1. System administration in Quantified riSk and Benefit adaptive Authentication Factors combination

In QSBAF, a system administrator is required to set up models to measure the quantified risks and benefits. Furthermore, the administrator will set several parameters, such as disclosure probability. Moreover, the administrator must also set policies to combine authentication factors. These tasks are quantitative numbers and formulas and therefore are totally different from traditional security policies. Thus, the administrator needs to learn the knowledge to set the parameters. A case-based method may help the administrator [19].

However, the advantages of QSBAF are also obvious. It can dynamically combine the authentication factors without human intervention. The administrator can also adjust the parameters when security environment changes, for example, a severe attack happens. In this adjustment, k and *mid* are two useful variables in that they can decide the mapping function between raw values and quantified risk and benefit measurements. For RAA and BDA measurements, increasing the system security level can be achieved by mapping lower RAA_{raw} (BDA_{raw}) to higher RAA (BDA), or vice versa.

6.2. Hard boundaries in quantified risk and benefit adaptive authentication factors combination

As is mentioned in the literature of Cheng *et al*. [8] and Han *et al*. [16], there are hard boundaries in the risk adaptive mechanisms. These boundaries are usually explicitly defined by using policy languages. For instance, a certain operation cannot be performed by a specific user. This policy is a static policy but efficient for stable and implicitly applicable scenarios.

In QSBAF, the hard boundaries also exist. In spite of the adjustments of policies and parameters, the user cannot perform the online-banking operations and transactions without an authentication factor, for example, password. Otherwise, the authentication and accountability of the system cannot be ensured, which will lead that the system is not a secure one.

Furthermore, an available authentication factor should be an implicit boundary of QSBAF. As we know, the authentication factors must be enforced after the decision is made by the decision engine. Thus, the enforcement infrastructure of the authentication factors must be available before QSBAF works. Then, the administrator must use the available authentication factors in the combination policies.

7. RELATED WORK

Risk analysis is very important for information systems. Hamdi *et al*. [20] pointed out that generally, risk analysis involves threat analysis, business impact analysis, and cost benefit analysis. However, the analysis process does not take the risk into quantitative factors.

Tsai *et al*. [21] proposed dynamic ID authentication scheme to enhance the security in authentication. This dynamic approach is merely on the basis of smart cards and does not change with the change context. It could serve as a risk mitigation factor in our model.

RSA adaptive authentication [22] is a risk-based authentication platform provided by RSA to balance security, usability, and cost. It is a policy-driven security solution to manage the authentication of information systems, which only considers risk but fails to consider benefit. The RSA risk engine is responsible for selecting matched policies from the RSA policy manager for inference. And the RSA eFraudNetwork is a shard data repository where suspicious identities are stored. After obtaining the inference result, the RSA Risk Engine will select appropriate authentication methods to enforce. The engine also has self-learning ability. Different from RSA Adaptive Authentication, QSBAF takes more into consideration (quantified benefit). Furthermore, QSBAF is used to select factors during authentication.

In order to enforce the selected authentication factors, the system should have effective enforcement methods. The Java Authentication and Authorization Service (JAAS) [23] has proposed a way to enforce selected factors. Using JAAS, the system administrator can develop and encapsulate authentication factors in different LoginModule. Different from JAAS, QSBAF provides an adaptive automatic method to choose authentication factors.

Access control is becoming more and more important in information security area because large systems today tend to provide more sensitive services. Role-based access control is a widely deployed access control mechanism. Kumar [24] used a dyadic formal context to explore the role-based access permission. He pointed out that the proposed method follows the role-based access control constraints: static separation of duties and role hierarchy. Moreover, Magkos *et al*. [25] took the trade-off between users' privacy and security access control into consideration. They aimed to enhance privacy by achieving intractability and unlinkability. Also, they provided security by achieving conditional traceability of users' credentials.

Quantified riSk and Benefit adaptive Authentication Factors combination could also achieve the feature of privacy protection if the privacy disclosure could be regarded as a factor in quantified risk measurement.

Risk and benefit are always two important aspects in the decision process. Zhang *et al*. [7] proposed a framework to use risk and benefit vectors in the access control decision and then used a battlefield as a scenario. Altinkemer *et al*. [5] first analyzed the cost and benefit of authentication systems from the economic perspective. They built a static model of the authentication system and categorized users along two dimensions: people who care more about privacy and people who prefer convenience. They proposed that service providers with larger market share are more likely to justify if it is worthwhile to adopt a new authentication system. In addition, clarification of

potential loss could encourage the service providers to adopt a more secure authentication system. Recently, Liu *et al*. [26,27] proposed incentive-based access control (IBAC). It aims to limit the abuse of access and find the potential threats inside the system. The IBAC also provides a reward mechanism to encourage the users to adopt the risk mitigation effort. In this way, the system could reduce its risk with the effort of all the users of the system. The mechanism relies greatly on users' actions and thus needs to balance the interest of both the system provider's and the users' to encourage reasonable incentive actions. In QSBAF, however, the proper risk mitigation factors are chosen by the system according to the decision engine.

As for the fuzzy logic, Ni *et al*. [28] and Han *et al*. [16] investigated how to leverage the theory of fuzzy logic in the quantified risk based access control field. However, the work of Ni *et al*. and Han *et al*. mainly applied the fuzzy logic to infer the estimated risk, whereas this paper uses the theory to infer a decision in access control. In our work, the risk and benefit will be calculated by formulas. Recently, Zhou *et al*. [29] proposed a generalized fuzzy data envelopment model with assurance regions. The algorithm to obtain the lower and higher bounds was also mentioned in the paper.

Finally, XACML is a widely used policy language [30] in access control. It has been proposed to meet different security requirements be extension [31,32]. In our work, we extend XACML to express the risk and benefit variables in access control.

8. CONCLUSION AND FUTURE WORK

In this paper, we propose QSBAF which dynamically selects authentication factors for an information system based on quantified risk and benefit. We apply QSBAF in the context of online banking system and give the quantified risk and benefit measurement, as well as the combination policies. We also design and implement a prototype to prove that our model is applicable in real system scenarios. As a result, QSBAF can balance the security and usability of an information system and can quickly respond to emerging events.

Our future work includes the implementation of a real authentication system using QSBAF under a big volume of history data and the investigation into further adaptive mechanisms.

ACKNOWLEDGEMENTS

This paper is supported by the 863 project (Grant NO: 2011AA100701), the project of Natural Science Foundation of Shanghai (Grant NO: 12ZR1402600), the project of Innovation Foundation of STCSM (Grant NO: 12511504200), and the Opening Project of DNSLAB, China Internet Network Information Center. Weili Han and Sean Shen are both corresponding authors.

REFERENCES

1. Mao Z, Li N, Chen H, Jiang X. Combining discretionary policy with mandatory information flow in operating systems. *ACM Transactions on Information and System Security* 2011; **14**(3):24.
2. Han W, Cao Y, Bertino E, Yong J. Using automated individual white-list to protect web digital identities. *Expert Systems with Applications* 2012; **39**:11 861–11 869.
3. Molloy I, Li N. Attack on the gridcode one-time password. In *ASIACCS*, Cheung BSN, Hui LCK, Sandhu RS, Wong DS (eds). ACM: New York, NY, USA, 2011; 306–315.
4. Coviello AW, Jr. Open Letter to RSA Customers. EMC Corporation, 2011. URL http://www.rsa.com/node.aspx?id=3872.
5. Altinkemer K, Wang T. Cost and benefit analysis of authentication systems. *Decision Support Systems* 2009; **51**:394–404.
6. Han W, Shen C, Yin Y, Gu Y, Chen C. Poster: Using quantified risk and benefit to strengthen the security of information sharing. *ACM CCS* 2011:2011.
7. Zhang L, Brodsky A, Jajodia S. Toward information sharing: Benefit and risk access control (barac). *POLICY'06*, 2006; 45–53.
8. Cheng P, Rohatgi P, Keser C, Karger PA, Wagner GM, Reninger AS. Fuzzy multi.level security: an experiment on quantified risk.adaptive access control. In *S&P 2007*. ACM: Oakland, California, USA, 2007; 222–230.
9. Organization for the advancement of structured information standards (OASIS). OASIS eXtensible Access Control Markup Language (XACML). OASIS. URL
10. Federal financial institutions examination council (FFIEC). Authentication in an internet banking environment, 2005.
11. Rabkin A. Personal knowledge questions for fallback authentication: security questions in the era of facebook. In *SOUPS '08*. ACM: New York, NY, USA, 2008; 13–23.
12. Franklin J, Paxson V, Perrig A, Savage S. An inquiry into the nature and causes of the wealth of internet miscreants. In *ACM CCS 2007*. ACM: New York, NY, USA, 2007; 375–388.
13. EMC Corporation. RSA SecurID. EMC Corporation. URL http://www.rsa.com/node.aspx?id=1156.
14. JASON. Horizontal integration: broader access models for realizing information dominance. *Technical Report JSR-04-132*, MITRE Corporation, 2004. http://www.fas.org/irp/agency/dod/jason/classpol.pdf
15. Teo L, Ahn G, Zheng Y. Dynamic and risk-aware network access management. In *SACMAT'03*. Yorktown Heights: New York, USA, 2003; 156–162.

16. Han W, Ni Q, Chen H. Apply measurable risk to strengthen security of a role-based delegation supporting workflow system. In *POLICY 2009*. IEEE Press: Piscataway, NJ, USA, 2009; 45–52.
17. Chen C, Han W, Yong J. Specify and enforce the policies of quantified risk adaptive access control. *Proceedings of the 14th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2010)*, Shanghai, China, 2010.
18. Molloy I, Cheng PC, Rohatgi P. Trading in risk: Using markets to improve access control. In *Proceedings of New Security Paradigms Workshop (NSPW'08)*. ACM: Lake Tahoe, California, USA, 2008; 1–19.
19. Khanum A, Mufti M, Javed MY, Shafiq MZ. Fuzzy case-based reasoning for facial expression recognition. *Fuzzy Sets and Systems* 2009; **160**:231–250.
20. Hamdi M, Boudriga N. Computer and network security risk management: theory, challenges, and countermeasures. *International Journal on Communication Systems* 2005; **18**(8):763–793.
21. Tsai JL, Wu TC, Tsai KY. New dynamic id authentication scheme using smart cards. *International Journal on Communication Systems* 2010; **23**(12):1449–1462.
22. RSA. RSA Adaptive Authentication. EMC Corporation. URL http://www.rsa.com/node.aspx?id=3018.
23. Oracle Inc. Java Authentication and Authorization Service (JAAS) Reference Guide. Oracle. URL http://tinyurl.com/d6sm6m7
24. Kumar CA. Designing role-based access control using formal concept analysis. *Security and Communication Networks* 2012:n/a–n/a. DOI:10.1002/sec.589.
25. Magkos E, Kotzanikolaou P. Achieving privacy and access control in pervasive computing environments. *Security and Communication Networks* 2011:n/a–n/a. DOI:10.1002/sec.283.
26. Liu D, Li N, Wang X, Camp LJ. Security risk management using incentives. *IEEE Security & Privacy* 2011; **9**(6):20–28.
27. Liu D, Li N, Wang X, Camp LJ. Beyond risk-based access control: towards incentive-based access control. In *Financial Cryptography, Lecture Notes in Computer Science*, Vol. **7035**, Danezis G (ed). Springer: Berlin, Heidelberg, 2011; 102–112.
28. Ni Q, Bertino E, Lobo J. Risk-based access control systems built on fuzzy inferences. In *ASIACCS*, Feng D, Basin DA, Liu P (eds). ACM: New York, NY, USA, 2010; 250–260.
29. Zhou Z, Lui S, Ma C, Liu D, Liu W. Fuzzy data envelopment analysis models with assurance regions: a note. *Expert Systems with Applications* 2012; **39**(2):2227–2231.
30. Han W, Lei C. A survey on policy languages in network and security management. *Computer Networks* 2012; **56**:477–489.
31. Sun Y, Gong B, Meng X, Lin Z, Bertino E. Specification and enforcement of flexible security policy for active B2 cooperation. *Information Sciences* 2009; **179**(15):2629–2642.
32. Luo J, Ni X, Yong J. A trust degree based access control in grid environments. *Information Sciences* 2009; **179**(15):2618–2628.