Flexible authentication framework with bound authentication and authorization

JAN HAJNÝ, TOMÁŠ PELKA, VÁCLAV ZEMAN
Department of Telecommunications
Brno University of Technology
Purkyňova 118, Brno 612 00
CZECH REPUBLIC
jan.hajny@phd.feec.vutbr.cz http://www.utko.feec.vutbr.cz/
tomas.pelka@phd.feec.vutbr.cz http://www.utko.feec.vutbr.cz/
zeman@feec.vutbr.cz http://www.utko.feec.vutbr.cz/

Abstract: - User authentication is an important part of computer network cryptography. It becomes more and more essential for security with the latest progress in the field of computer science. We analyze latest trends in the article. Our goal is to answer the question whether these trends reflect current state in computer networks. We are focused on security and from this point of view we find the progress unsatisfactory. The reason is that most of current work is focused on operational parameters more than on security. That's why we give an example of less known authentication methods which can solve the most important issues of current protocols – security and protocol flexibility. The main advantage of one of these methods is its mathematical model which can prove security. As the flexibility of the method is also vital for modern multi domain networks we provide a modification of this method to improve the process of authentication and authorization.

Key-Words: - Cryptography, Authentication, Authorization, Security, Zero-Knowledge, Password-based authentication

1 User Authentication

Network security is becoming a more and more discussed topic in a present time. This subject contains not only security of data we are sending but also security of users. We need to solve the problem of user authentication. This task is getting more difficult with heavy use of mobile devices like cell phones, sensors, PDAs, notebooks. Nowadays the solution of user authentication must count with a wide variety of devices – not only computers. These devices are very often limited by hardware resources and capabilities. That's why a modern authentication system must count with such diversity and must be able to give a solution usable over the whole network. Of course there is an advance in protocols used for user authentication too [1], [2]. We are asking a question whether these protocols are sufficient for authentication in such a modern network or if there are some drawbacks and so we need a new solution.

2 Current state

We started with the analysis of the current state in the field of user authentication research. We have chosen more than 100 research projects to find out which topics are popular and where the situation is insufficient. The distribution can be illustrated by the graph in the Fig. 1:

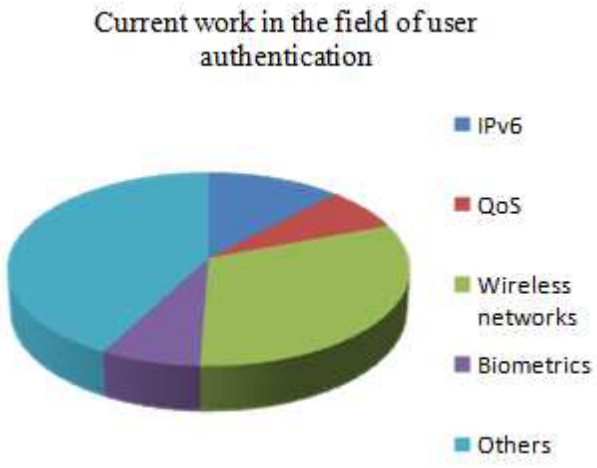

Fig. 1: Research subject distribution

We can identify main trends in current research. Most of the work is focused on wireless networks. This is mainly caused by the increase of mobile devices and the need of cooperation between computer networks (WiFi, WiMAX in case of wireless [3]) and cell phone networks (PLMN – Public Land Mobile Networks). This cooperation must be resolved also in the field of authentication so research is trying to adapt current methods to the new environment. IPv6 is the second area of interest. This is also closely connected to mobile devices because IPv6 is designed to replace IPv4 and has many features for these networks. What is important is the fact that there are not many new authentication methods mentioned. Most of the research projects are focused on the implementation of older methods used in the past [4]. The same situation can be found in projects regarding QoS – mostly adaptation of current methods to a new use.

Biometrics has been a very discussed topic in recent time but not many research projects are focused on authentication only by biometrics. Mostly there is some other conventional authentication method.

3 The need of change

Of course our analysis was not focused only on the distribution of topics. Our main task was authentication. We wanted to find out whether current authentication methods are usable for next generation of networks or if there is a need for a new solution. This is the reason why we took more than 100 of recent papers and searched for trends and new methods. Now we can say that most of research projects are focused on operating parameters and performance. In most of papers the main goal was to create a new signalization protocol and improve parameters like time delay or variance of delay. We can give [5] [6] as examples. This result is connected with a most frequent topic – mobile networks. The need for a new authentication system is driven by the need for mobile networks support. That's why most of research groups focus on operating parameters more than on security. This is the main problem we identified – security in communication systems is not the main goal.

We were looking for a new user authentication solution which can be used in a multi domain network in the beginning of our research project. After the analysis of current state we found out that in majority of new systems there is no new authentication mechanism. The development simply focuses on making old mechanisms better and on adaptation to the new environment. This goes very often against the security. Researchers are moving from 4-way authentication protocols to 3-way and they are preferring challenge - response protocols to eliminate time delay caused by the use of asymmetric cryptography [5]. But this state is not caused by the lack of new authentication mechanisms. Although there are not many we found some solutions. They are not commonly used but they can have very interesting properties for the next generation of networks.

4 New solutions

The main goal of our project is to find an authentication framework which will be flexible enough for use on different kind of appliances. We can see a variety of devices in nowadays networks. We must support servers, user PCs, PDAs, sensors etc.. All of them with different computing power and different requirement for security. There will be a different security level for a sensor and for an administration terminal. Our task is to design a framework which is wide enough to accommodate all of these equipments. Although this is the goal of our project we believe that these networks are more and more frequent and computer only networks are over.

The other goal of our search is security. We tried to find a method which would have some safety advantage over methods used today. If we want to improve the security of a current authentication system we must understand what we do in present systems. The user authentication is based on secret information knowledge. In familiar scenario there is a client who asks for a server's service. The service is granted when the server successfully authenticates the client. Usually a shared secret is involved. The secret information is created by the server (authenticator) and given to the client. We can imagine that as some password. Later, when authentication is needed, the client can send the password to the server and prove that he is the one who claims to be. As the password was given by the server only to the genuine client there is no way how an attacker can have the password so authentication is working fine. But there are still some problems with this scheme. The first one comes when we ask about the way of showing the password to the server. There must be a secure way how to show the password to the server. Usually we need to do this over an insecure environment like the internet. A password encryption is the common solution. But is this a good idea? Isn't it possible to extract the password from the transferred cryptogram? Usually there is no way how to answer

– the security of the protocol is assured by the belief that an attacker cannot decrypt the cryptogram. But there is no proof. Even if we send our password in a trusted cryptogram which cannot be decrypted by attackers we are still giving away some information – at least the length of a password is usually leaked. So we cannot be sure that the password is really sent securely and we have no information about how much of secret information leaked. This is the obvious problem of classical protocols. Fortunately there are solutions.

5 Password-based protocols

The protocols based on knowledge of passwords are a relatively new authentication method (this claim is not entirely true, because the mechanism was published at the end of 80s, more precisely, in 1989) called password-based protocols. The main motivation of this method is solving a limitation of the length of a password to, for example, 8 characters. The password is in this case a secret information that must be kept in mind of a human. Therefore a limitation is caused by the human skills and potential. It is therefore desirable to establish a method that is sufficiently safe but still built on short passwords. It is often advised not to mix this method with the protocols based on asymmetric cryptography. The reason is that asymmetric cryptography keys are incomparably longer than "simple" passwords.

The family of protocols called EKE (Encrypted Key Exchange) [12] was born in the 1989. The main idea of these protocols is that the initiator (client) selects a short-term public key and the shared password for the encryption of a future encryption key. The server can decrypt the public key and use it to secure key relationships, which is opened by the client. Assuming that public keys are random strings, an attacker will not be able to recognize used short-term key by the brute force attack. And even if the attacker found the right key, he would not be able to identify the session key because it is not possible to obtain a secret key from the knowledge of the public key. In the summary we get the following set of features:

- Client owns a password with small entropy. In other words, the attacker is able to reveal the password by a brute force in a real time.
- Offline dictionary attack is not possible. This means that a passive sniffer, which can record one or more sessions, cannot obtain a sufficient number of potential passwords.
- Online Dictionary attack is not possible. This means that an active attacker cannot exploit the protocol to obtain a sufficient number of potential passwords. He may, however, take at least one password from each protocol cycle, in an attempt to discover this password. Ideally, this should be the only thing an attacker can catch.

The protocol EKE has become a basis for the derivation of a number of variants like the Fiat-Shamir Protocol, because the original EKE protocol encryption algorithm doesn't specified the password transformation to the relevant key. Overall, the password-based protocols can be divided into three categories:

- the first case, the principle uses the Diffie-Helman discrete logarithm problem,
- the second group based on the RSA protocol, and therefore on the use of factorization and modular arithmetic.
- the third category gathers protocols that do not fit the previous two.

Some protocols use a variant in which the server possesses a directly shared password hidden in an image that is obtained by a one-way function (similar to hash). In this case the server security break does not have to mean a password disclosure. So we can split password-based protocols by the principle of sharing a secret password. In the case where a client and a server share a secret key, we use symmetric encryption methods and protocols. In this case we talk about the type of EKE (such as DH-EKE, A-EKE, B-SPEKE etc. [13]). It's easier to create such a protocol but a disadvantage is that when an attacker breaks through the server security, he is then able to obtain the secret password directly. In the other case where the server keeps only the image of the password the risk disappears but the cost is that we need to use asymmetric arithmetic. These protocols belong to the group AKE (Asymmetric Key Exchange) and unlike EKE protocols they don't encrypt exchange of messages when the protocol runs. Instead of an encryption, they use redefined mathematical equations and exchange of short-term random values using secret keys. Missing encryption is useful, for example, for the following reasons:

- Simplifies the protocol by removing the need of agreement on the encryption algorithm, in other words, the protocol in such cases becomes independent on one particular encryption algorithm.

- A weak encryption has the effect of undermining the entire authentication protocol. Using more passwords (even if encrypted) makes the protocol susceptible to a number of attacks.
- Software and hardware uses encryption algorithms that may violate laws.

For a more general description of the AKE protocol, its mathematical derivation and generation, see [14]. One possible interpretation based on the AKE protocol is the SRP (Secure Remote Password) protocol proposed by Thomas Wu [15] as a simpler, more efficient and safer alternative to EKE protocols.

5.1 EKE using public keys (AKE)

Notation:

A; *B*	System principals. (Alice and Bob).
P	The password: a shared secret, often used as a key.
R; *S*	Random secret keys (for symmetric cryptosystems).
R(info)	Symmetric (secret-key) encryption of „*info*“ with key *R*.
R^{-1} *(info)*	Symmetric (secret-key) decryption of „*info*“ with key *R*.
Ek (X)	Asymmetric (public-key) encryption of X with (public) key *Ek* .
Dk (X)	Asymmetric (public-key) decryption of X with (private) key *Dk* .
challengeA	A random challenge generated by *A*.
challengeB	A random challenge generated by *B*.
p; *q*	Prime numbers.

Consider the following simple exchange of messages:

1. A generates a pair of keys (public/private), E_A and D_A. The public key is encrypted by a symmetric encryption and a password *P*. *A* sends:
$$P(E_A)$$
2. *P* should be securely shared. *B* is than able to decrypt the message *A*: $P^{-1}(P(E_A)) = E_A$. *B* then generates a random secret key *R*, which is encrypted (asymmetrically) by the key E_A, this value is then encrypted using a password *P*. *B* sends:
$$P(E_A(R))$$
3. *A* knows *P* and D_A. He is able to decrypt the message from *B*:
$$D_A(P^{-1}(P(E_A(R)))) = R$$

Alice and Bob both know all secret information to send encrypted messages after this exchange because *R* will be used to encrypt/decrypt them - $R(message)$. Now let's have a look on a possible eavesdropper. Knowing $P(E_A)$, $P(E_A(R))$, and $R(message)$, a candidate password *P'* can be used to decrypt $P(E_A)$ to produce a candidate public key $E_A = P'^{-1}(P(E_A))$. But learning whether $E_{A'}$ is the public key used in the exchange leads to learning whether there exists a secret key *R'* such that $E_A(R') = E_A(R)$ and $R'^{-1}(R(message))$ makes sense. This finding is the key property of the exchange: a candidate password *P'* cannot be rejected without doing a brute-force attack on *R*[1]. Since E_A and *R* are random numbers from large key spaces, such attacks are expensive, even if the space of passwords is small. So far as brute force off-line attacks are concerned, the relatively small space from which *P* is chosen has been effectively multiplied by the size of the keyspace from which *R* is obtained.

But there are still some drawbacks. For example, an important concern is the possibility of replay attacks. In that case there might be an attacker that inserts old messages to the channel. Protocols must incorporate safeguards, typically in the form of random challenges. Let's consider this version.

1. *A* chooses a random E_A and encrypts it using symmetric encryption and a password *P*. *A* sends:
$$P(E_A) \quad (1)$$
2. *P* should be securely shared. *B* is then able to decrypt the message *A*. *B* then generates a random secret key *R*, which is encrypted (asymmetrically) using the key E_A, this value is then encrypted using the password *P*. *B* sends:
$$P(E_A(R)) \quad (2)$$
3. *A* decrypts the message and obtains *R*, then generates a unique challenge (*challengeA*), which is encrypted using *R*.

[1] Suppose the eavesdropper uses only non-cryptanalytic attacks

$$R(challengeA)$$

4. *B* decrypts the message and obtains the challengeA, and generates his own unique challenge (*challengeB*). Both challenges are encrypted using *R* and sent to *A*.

$$R(challengeA, challengeB)$$

5. *A* compares both challenges and sends Bob's challenge back to *B*.

$$R(challengeB)$$

The challenge-response mechanism, used in steps 3-5, is a standard technique for validating cryptographic keys. In practice, there are already two implementations - AKE based on RSA and ElGamal encryption system. The deployment problems are restrictions on the choice of P for these systems [12], [16].

In this chapter we introduced a protocol family which deals with the problem of short passwords. We can see that there are good methods to provide security even if we are forced to use short passwords. But there is still a need for using these passwords in the protocol and even for storing these passwords on a server side. Although Password-based protocols were very promising for a use in our project we decided to search further for a better solution. What we need is provable security and as we shall see in the next section there are protocols that can also solve the problem of a server side password deposition – the Zero-Knowledge protocols.

6 Zero-Knowledge

There are protocols which can solve the problem of a password transport. We must change our view on authentication. In the Chapter 4 example there was a need for a password transfer. Only if the password is transferred from a client to a server the server can decide whether authentication is successful or not. It's supposed that the server already knows the password but this doesn't have to be the case always. What we really need to transport is not the password but only one bit of information – whether a client knows it or not.

This is the idea of Zero–Knowledge protocols [7] [8]. They don't transfer the whole password which can be abused but they transfer only one bit of information about the client knowledge of password. It's possible to minimize one of the biggest dangers in authentication – the leak of secret information. This leak would be a big problem of course – because if an attacker knows every information like an authorized person then there is no problem for the attacker to confuse an authenticator. The authentication system has no way to distinguish a user from an attacker in this case. Let's see why authentication protocols can have such interesting attributes.

The property of Zero-Knowledge can belong to some kind of systems. One of them is a Interactive Proof System which is suitable for user authentication. The Interactive Proof System consists of two Turing machines. These machines are equipped by communication tapes which can be used for a both-direction communication. One of these machines is polynomially bounded and the other one is unbounded. These machines are called Prover and Verifier in the literature. Prover wants to get some service from Verifier, so Verifier is the authenticator and Prover can be imagined as a client. The situation can be seen in the Fig. 2:

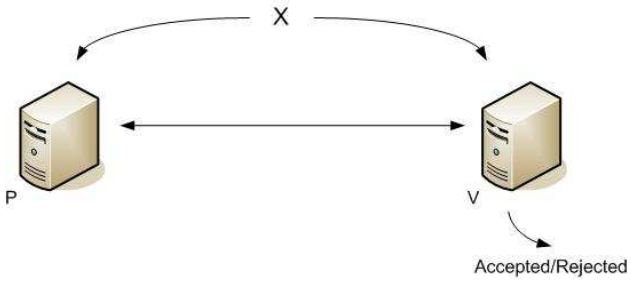

Fig. 2: Interactive Proof System

The Interactive Proof System works by inputting some string X which can be a statement. Then machines run and communicate and at the end Verifier outputs the result Accept/Reject. This can be compared to the process of user authentication – Prover claims that he knows some secret (statement X) and if it is true then he is accepted by the protocol. For this we need two properties of Interactive Proof Systems:

Completeness: If the statement X is true then the probability that the pair (P, V) rejects is negligible in the length of X.

Soundness: If the statement X is not true then for any Prover P* the probability that the pair (P*, V) accepts is negligible in the length of X.

In other words the genuine user must be allowed in almost always and the attacker must be rejected almost always. This is a good start for an authentication protocol but there is still the problem with safety. We need the property of Zero-

Knowledge for the Interactive Proof System to be secure:

Zero–Knowledge [7]: The Interactive Proof System has the property of Zero-Knowledge if for any polynomial time verifier V* there is a simulator M_{V^*} running in expected polynomial time which's output M_{V^*} is indistinguishable from the output from (P, V) for a true statement X:

$$M_{V^*} \sim (P,V) \qquad (1)$$

This existence of a simulator is essential for Zero-Knowledge protocols. With this simulator we can have the control over the leakage of password information.

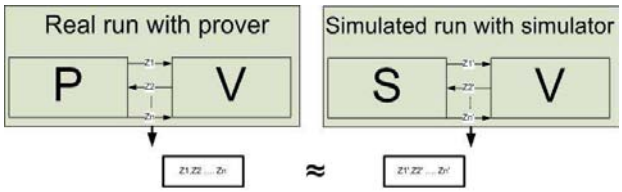

Fig. 3: Zero-Knowledge Simulator

For a Zero-Knowledge Protocol there must be a simulator that can be used instead of Prover and there will be no difference in generated messages. We can see this situation in the Fig. 3. So there is no influence of secret information on how these generated messages look like because simulator does not know the secret information. From this we can see that generated messages does not leak any information about the secret because if they do so then there would be a difference between simulated output and real output which would break the Zero-Knowledge Property. The Zero-Knowledge Property can be divided into three categories:

$M_{V^*} \sim^{C} (P,V)$ - Computational ZK (2),

$M_{V^*} \sim^{S} (P,V)$ - Statistical ZK (3),

$M_{V^*} \sim^{P} (P,V)$ - Perfect ZK (4),

with a raising level of security.

7 Choosing a protocol

There are many examples of Zero-Knowledge protocols. The simplest version was introduced in [8] to illustrate this technique. A Fiat-Shamir is another example which is often considered as one of the first usable protocols of this kind. It was introduced in 1986 [9] but there are many modifications which are used nowadays. One of these modifications is an Ohta-Okamoto protocol [10]. According to [11] there is a big advantage of this protocol – it is not patented and so it can be used in open – source SW. Thanks to the security of the protocol, its efficiency and free nature we have decided to include it to the testing framework of our authentication scheme. Another reason is that this protocol is already implemented in authentication of ssh users [11].

7.1 Fiat-Shamir protocol

We begin the description by introducing the Fiat-Shamir protocol. It works as a basis of further protocols. This protocol gives us a solution for a classical authentication problem of a client and a server. The client wants to get some service from the server. Server provides a service only to authenticated clients. The Fiat-Shamir protocol needs that authenticated clients are given some secret value which works as an input to the protocol. Only clients with proper values are successfully authenticated. The server doesn't know these secrets although they can authenticate these clients. The situation can be illustrated by the Fig. 4: Authentication protocol:

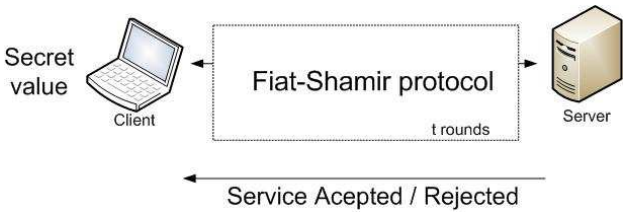

Fig. 4: Authentication protocol

As you can see in the Fig. 4 there are many rounds of a protocol. The number of rounds is *t* and works as a security parameter. Each round has the same structure and in the case of a Fiat-Shamir protocol we rely on the problem of the discrete square root computation. We assume that the computation of a discrete square root is a hard enough problem for high enough numbers. With this assumption we can use the protocol with a Zero-Knowledge proof. If there is an algorithm which can efficiently compute a discrete square root even for high numbers then there would be problems. The actual Fiat-Shamir protocol works as depicted on a Fig. 5: Fiat-Shamir protocol:

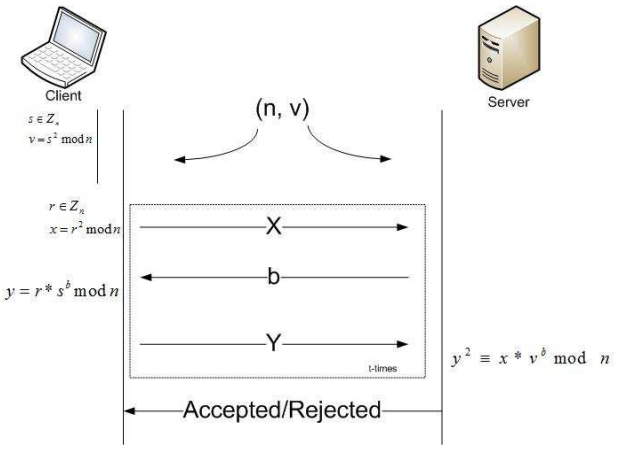

Fig. 5: Fiat-Shamir protocol

There must be a setup before the actual protocol can start. During this phase a modulus *n* is chosen in a form of $n = p * q$ where *p* and *q* are primes. These values can be chosen e.g. by some third party. Then the client must create a secret key and a public key. The secret key *s* is just some randomly chosen number in Z_n. The public key *v* is computed as follows and it is given to the server with a modulus *n*:

$$v = s^2 \bmod n \quad (5)$$

Now we can run the authentication protocol between the client and the server. There must be many rounds as the protocol is iterative. One round consists of three messages – two from the client to the server and one from the server to the client:

- Firstly a random value *r* in Z_n is chosen by the client.
- The first message *x* is computed as $x = r^2 \bmod n$.
- X is sent to the server.
- The server chooses a random bit *b* 0 or 1 and sends it to the client.
- The client computes a response *y* using the formula $y = r * s^b \bmod n$.

One round is accepted if the final equation holds:

$$y^2 \equiv x * v^b \bmod n \qquad (6)$$

All *t* rounds must be accepted for a successful authentication. If only one check of the final equation fails then authentication fails also. We have to prove now that this scheme is a Zero-Knowledge protocol. So we need to check Completeness, Soundness and a Zero Knowledge property.

Completeness:
A user who knows a secret key must be let in almost always. This property is clear from the design of a protocol because if a user knows the secret value *s* then he is able to compute a right answer for a challenge *b*. The equation (6) then holds, because:

$$y^2 = (r * s^b)^2 = r^2 * (s^2)^b = x * v^b \bmod n \quad (7)$$

Soundness:
A user who doesn't know a secret key must be rejected almost always. Let's assume for a contradiction that a client who doesn't know the secret *s* is allowed into the system. Then the final equation (6) must hold. It holds only if a correct answer is sent during the process of an authentication. A client can sent a correct answer for a challenge *i* either if he knows the challenge in advance or if he possess correct answers for both b = 0 and b = 1.

Let's take the first option where the client can guess the challenge. As the challenge is chosen randomly in *t* rounds then the probability that a user can guess all challenges is 2^{-t} which is negligible.
The second option is that the client knows both answers. Then he knows:

$$y_0 = r \bmod n$$
$$y_1 = r * s \bmod n$$

It's easy to compute an inverse value $\overline{y_0}$ with Euclidean algorithm. Then it would be very easy for a client to compute $s = \overline{y_0} * y_1$ which contradicts our assumption that *s* is secret and not computable in a good time. So the user can get inside only if he can guess all *t* challenges or do the discrete square root.

Zero-Knowledge:
There must be a poly-time simulator that generates indistinguishable output from the honest real run without the knowledge of s. We know that we can create an acceptable conversation if we know the challenges in advance. Because the simulator can be slower than the real run (must be poly-time) then we can just guess challenges. The simulator will just discard messages if the guess is not correct. Then we can generate indistinguishable output in the time of 2t which is polynomial. This is because our guess

is in 50% right (bit i = 0 or 1). A more detailed proof can be found here [9].

7.2 Ohta-Okamoto protocol

The Ohta-Okamoto procol is an extension of an original Fiat-Shamir protocol. The main advantage is that values are sent in parallel so the protocol should be more efficient. We have chosen a version used in a ZK-SSH project which is one of few implementations of Zero-Knowledge protocols [11]. The whole process of authentication is also divided into two parts – the setup and the communication between the client and the server.

Setup

The client chooses a modulus n as a product of two primes p and q. Then he randomly generates k integers s_i from Z_n. These work as a secret key. He also chooses a random small integer L and computes $v_i = s_i^L \bmod n$. The public key is made of (v_i, L, n) and is published in this phase. No we can move to the communication between the client and the server where we describe one round of total t rounds. It is shown on a Fig. 6: Ohta-Okamoto protocol:

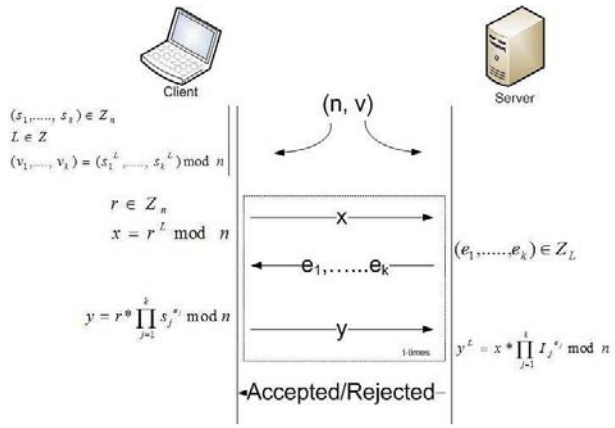

Fig. 6: Ohta-Okamoto protocol

Protocol round

The process is very similar to the original Fiat-Shamir protocol. The main advantage is a variable exponent L (instead of 2 in FS) and parallelization using k challenges e. These improvements lead to better security so we can decrease the number of rounds in comparison with FS. The process is:

- A random value r in Z_n is chosen by the client.
- The first message x is computed as $x = r^L \bmod n$.
- X is sent to the server.
- The server chooses k random challenges (e_1,..,e_k) in Z_L and sends it to the client.
- The client computes a response y using the formula $y = r * \prod_{j=1}^{k} s_j^{e_j} \bmod n$.

One round is accepted if the final equation holds:

$$y^L = x * \prod_{j=1}^{k} I_j^{e_j} \bmod n \quad (8)$$

The client is accepted if all t rounds are accepted. The probability that a cheater is accepted is:

$$p = \left(\frac{1}{L}\right)^{t*k} \quad (9)$$

8 Adding flexibility

As we can see there are protocols that can bring us some extra security over standard systems. With the use of Zero-Knowledge Protocols we can be sure that there is no information leak during the run of the protocol. This gives us the assurance that a user secret will be safe and an attacker won't be able to learn anything. But there are even more benefits of these protocols. The second one goes well with the second demand on modern computer networks. It's the demand for a good flexibility.

Zero-Knowledge Protocols are iterative protocols. This means that they must be run several times – in several rounds. The more rounds we take the safer the authentication process is. Usually the level of security grows superpolynomially with the number of rounds. The number of rounds can be considered as a security parameter t. Now we can go back to the beginning of this paper where we identified the most discussed topics in a user authentication research. The need for a various device support was one of the most problematic areas. Authentication of all types of equipment by one framework is often the desired solution. There is a possibility to use the number of rounds in Zero–Knowledge to support this variety of devices. The main idea is to bind the number of iterations with authorization in the system. Then devices like servers, user PCs and other powerful machines can do a strong authentication with a high security parameter t and small machines like sensors can do less resource consuming authentication with a lower parameter t. If we bind the parameter t with an authorization center we can choose the appropriate level of access of the machine. This leads to a restricted access for weaker equipment like sensors and open access for strong equipment like

computers which can run the authentication protocol with high enough parameter *t*.

The result is a framework which uses one method for authentication (Zero-Knowledge) but is able to reflect the varied spectrum of clients. A general scheme is illustrated in the Fig. 7:

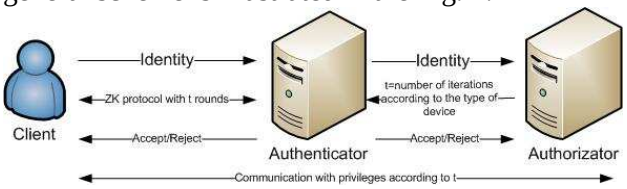

Fig. 7: Binding authorization and authentication

9 Conclusion and future work

The main purpose of this paper is to find the answer for a question whether current authentication protocols are sufficient for nowadays networks or not. We did an analysis of latest trends in user authentication in the first part. After the review of more than 100 papers we are convinced that operation parameters and performance is a more discussed topic than the security of protocols. This leads to solutions where security in not the highest priority and which can have some vulnerabilities. As there are cryptographic protocols which have better properties we do a short example how to use these properties to reflect the latest progress in the area of user authentication.

There are two main challenges we need to resolve. Firstly a higher security and secondly better flexibility – both of them are coming from trends in computer networks. We show that there are available and verified solutions that can be better than conventional protocols. We have chosen the Zero-Knowledge protocols and the next step is an implementation of our modified system.

We have also mentioned a novel protocol relying on the counter-intuitive notion of using a secret key to encrypt a public key as an original candidate for the authentication protocol. We have shortly introduced protocol called AKE.

Acknowledgement:
Sponsored under the National Program of Research II by the Ministry of Education, Youth and Sports of the Czech Republic in 2C08002 Project - KAAPS Research of Universal and Complex Authentication and Authorization for Permanent and Mobile Computer Networks.

References:

[1] SHIEH, Wen Gong, WANG , Mei Tzu. An improvement on Lee et al.\'s nonce-based authentication scheme . In *WSEAS Transactions on Information Science and Applications*. Vol.1, WSEAS Press, 2007. pp. 832-836. ISSN 1790-0832.

[2] ENCHEVA, Sylvia, TUMIN, Sharil. Authentication and Authorization User Management within a Collaborative Community. In *COMPUTER SCIENCE and TECHNOLOGY : Proceedings of the 11th WSEAS International Conference on COMPUTERS* (part of the 2007 CSCC Multiconference). Vol.1, WSEAS Press, 2007. pp. 563-569. ISBN 978-960-8457-92-8. ISSN 1790-5117.

[3] PARK, Wonjoo, KANG, Dongho, KIM, Kiyoung. A Static or Dynamic Reconfiguration Method of Security Functions for Mobile Devices by using Security Profiles. In *Advanced Topics in Information Security and Privacy : PROCEEDING of the 6th WSEAS International Conference on INFORMATION SECURITY and PRIVACY (ISP \'07)*. Vol.1: WSEAS Press, 2007. pp. 146-151. ISBN 978-960-6766-. ISSN 1790-5117.

[4] LÓPEZ, Rafael Marín, PÉREZ, Gregorio Martinez , SKARMETA , Antonio F. Gomez . Implementing RADIUS and Diameter AAA Systems in IPv6-Based Scenarios. *In Advanced Information Networking and Applications conference. 2nd edition.* Washington, DC, USA : IEEE Computer Society, 2005. pp. 851-855. ISBN 0-7695-2249-1. ISSN 1550-445X.

[5] CAO, Xuefei, KOU, Weidong, LI, Huaping. Secure Mobile IP Registration Scheme with AAA from Parings to Reduce Registration Delay. *In Computational Intelligence and Security, International Conference.* Vol.1: IEEE, 2006. pp. 1037-1042. ISBN 1-4244-0605-6.

[6] LEE, Joong-Hee, et al. Moving AAAH Architecture for Mobile IPv6. In Advanced Communication Technology, *The 9th International Conference. 2nd edition.* Gangwon-Do, 2007. pp. 1246-1251. ISBN 978-89-5519-1. ISSN 1738-9445.

[7] GOLDREICH, Oded, MICALI, Silvio, WIGDERSON, Avi. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *In Journal of the ACM*. Vol.1, 1991. pp. 691-729.

[8] QUISQUATER, Jean-Jacques , GUILLOU, Louis C. , BERSON, Thomas A. How to Explain Zero-Knowledge Protocols to Your Children.. *In Advances in Cryptology - CRYPTO '89*. Vol.1, 1990. pp. 628-631.

[9] FIAT, Amos, SHAMIR, Adi. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology - Crypto´86*. Vol.1, 1987. pp. 186-199.

[10] OHTA, Kazuo, OKAMOTO, Tatsuaki. A modification of the Fiat-Shamir Scheme. In *Advances in Cryptology – Crypto'88*. Vol. 403, 1988, pp. 232–243.

[11] GAUPMANN, Andreas, SCHAUSBERGER, Christian, ZEHL, Ulrich. The zk-ssh Project : The Zero-Knowledge Identification Protocol [online]. 2005 [cit. 2008-12-24]. WWW: <http://zk-ssh.cms.ac/docs/zk_protocol.pdf>.

[12] BELLOVIN, Steven M. , MERRITT, Michael. Encrypted key exchange: password-based protocols secure against dictionary attacks. In *Proceedings of the I.E.E.E. : Symposium on Research in Security and Privacy*. Oakland : IEEE, 1992. pp. 72-84. WWW: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.45.3156>.

[13] BOYD, Colin, MATHURIA, Anish. Protocols for authentication and key establishment. Berlin: Springer-Verlag,2003. pp.91-93, pp.247-285. ISBN 3-540-43107-1.

[14] WU, Thomas. The Secure Remote Password Protocol. [online]. 1997 [cit. 2008-12-02], pp. 17.WWW: <http://srp.stanford.edu/ndss.html/>.

[15] WU, Thomas. The Stanford SRP Authentication Project [online]. [cit. 2008-12-02]. WWW: <http://srp.stanford.edu/>.

[16] DENNING, D. E. Cryptography and data security. [s.l.]: Addison-Wesley Publishing Company, Inc. , 1982. 414 s. ISBN 0-201-10150-5.