ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS

Amplifying Anonymity Techniques

Dimitrios Koutoufaris
University of Piraeus

Supervisor : Dr Christoforos Ntantogian

Contents :

Abstract

This project introduces a series of techniques and technologies able to enhance the security of an online transaction, by advancing the anonymity of the parties involved. Usually, the user requests an access to a system, the system asks for some credentials, and instead of a secret value (e.g. a private key) the user can submit an appropriate value, or asset, that is signed as appropriate by a third party trusted authority. In the first part of this project, this aspect is presented in an abstract way. A technique that times the duration of the keystrokes and the latency between them, is then analyzed. This method suggests a smart way to produce a "biometric" salt, which given the proof that it cannot be copied or imitated, it can enhance a password, or even encrypt effectively an entire text or message. The next subject analysed is the Zero Knowledge protocols and the proofs that show that the verifier authority really does not have a clue of any identity of the user that signs in a system. The combination of the last two techniques show the way for a secure scheme. Applications of the Zero Knowledge Protocols are next exposed. Those applications involve internet solutions (e-vote systems, e-commerce) , Internet of Things solutions, and Networks of Sensors. In this paper, both interactive and non interactive techniques are shown. At the end, a “mechanical” implementation of ZKP is exposed, in order to show the importance of this technique.

Prologue

Security incidents like data leakage, eavesdropping of a transaction in a secure channel, analysis of data gained through non-social networks, like enterprise spying, are only some news that refer to the same problem. The fact that data shared through a local network, let alone the whole internet, should flow in a more secure way. It's an everyday habit now, to receive a third party's authentication to login a mailing account, to view your online files in a cloud; the way a third party can authenticate an entity, and give it the right to view its personal data, without revealing anything personal, or any account detail is analyzed in this paper. This technique can apply to any "logging in" instance; from a web session in secure channel (e-pay, e-vote) to authentication of mobile devices and RF pass cards. Different implementations apply to different needs of authentication. When a user uses a personal computer or a handheld smart device, he has computational power to enhance his secure transaction. On the contrary, RF cards authenticate non-interactively.

1. System of Anonymous Authentication [I]

1.0 Introduction

In this first presentation, we illustrate a general method of a three party authentication. This authentication scheme involves the user, the verifier, and the authentication provider, and the access provider. This way of accessing is usually implemented in authentication of a user when using e.g. an email account, and the user is accessing it through various different devices, operating systems, intranets.

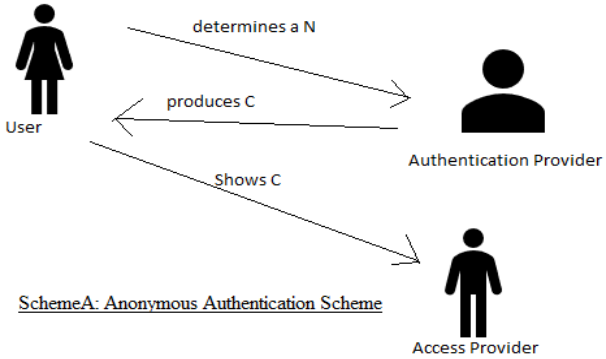
1.1 Basic Requirements

The users of the system ask for and show some credentials, whereas the providers ask and authenticate the credentials. A user U (Issuer) asks one credential from the verifier, Oi . A credential is given always in relation to a nickname N with which user U is identified by the verifier. This credential can also have attributes, which the user can decide to reveal when submitting the credential.

The requests of the nickname, the credential and the verification of the nickname implement with protocols between the user and the appropriate providers. The user U has a unique secret value Su, connected with all his nicknames and credentials. The authentication/identification providers own pairs of public and private keys. This authentication provider uses the secret key to construct the credential. Afterwards, the credential is verified, either by using the authentication provider's public key, or by any other (third party) organization to whom a user shows his credential. When a user shows his credential, he uses the authentication provider's public key, which can be used only along with the protocols private key.

1.2 From an authentication provider, to the "access" provider.

The user U communicates with the authentication provider Oi and determines a nickname N. If N is a candidate to receive a credential with the attribute (attr), the provider Oi produces a credential C signing a "statement" that has the attribute (attr) and the N and sends C to the user U. Now, user U can show his credential C to the access provider. In that way, user U having revealed no information at all, "persuades" the access provider that he owns a signature constructed by the authentication provider, and he knows the unique secret value Su that is related to the credential C. In particular, the user U does not send the authentic credential. This way of (no) revealing a credential along with a zero knowledge attribute, ensures the no-correlation of different past credentials, and the no-correlation of a credential and its relative nickname. Consequently, the user can show the Ov his credential unlimited times without running the risk these connections to be correlated each other, or to a certain nickname. The implementation of this protocol by users and providers makes mandatory for the providers the use of a set of signatures in the typical RSA authentication model. It is clear from above that all the user's credentials are connected to the secret value and that their disclosure may reveal the secret value. In order to avoid this disclosure, countermeasures are implemented, like the connection of the secret value with something (personally) important, like a bank account pin.



SchemeA: Anonymous Authentication Scheme

However, when something so important is not available, the connection of the credentials and the nicknames is proposed, to enhance their mutual protection. When the authentication provider and the access provider is the same, every submit of a credential is accompanied by the production of a new one.

1.4 Showing the credential, de-anonymization

The revelation of a credential, and even of the user's identity can be carried out either by an authentication provider (issuer) Oi, or by an international (third party) supervising organization. The de-anonymization organization is usually a third party organization who intervenes and reveals the identity of user, when the user has committed a crime, or even a minor online violation. This organization usually owns pairs of public and private keys. The user encrypts the nickname N with the public key of the de-anonymization organization. The authentication provider should see the proof that the de-anonymization organization can indeed decrypt and reveal the nickname of a user by submitting a new nickname. The user can choose between de-anonymization organizations and under which circumstances his identity will be revealed.

The user U who has defined a nickname Ni, and has accepted the credential C by the Oi, is known under the nickname Nv to the access provider, proves to Ov that owns C, and that C belongs to the same user as Nv does. The user U practically proves that the secret value Su that is connected with the nickname Ni, also connects to the credential C, and with the nickname Ni. Without this connection, two different users could impersonate another unique user, by using a unique credential.

Finally, Oi cannot correlate the nickname N that gave to the user U with any oher information that Oi gave to the user U. Each time the user U requests a new credential, he submits another secret value or confidential information.

2. Keystrokes password hardening [I I]

2.0 Introduction

We next present a system of password hardening. This system is based on measuring the duration of the keystrokes and the latency between each key pressed when typing. This way of password hardening employs two parties, the user, and the login system / server that does all the calculations, the encryption and the hardening.

2.1 Description

The method is based on the use of a basic password, on the timing of the keystrokes, and the typing of these credentials. At first, a cipher is made out of the user's password, and then another cipher is constructed by the timings of the keystrokes, and the delay between keystrokes. The system then creates a unique encrypted password. When an attacker who knows the legitimate user's password try to submit it, the system will probably reject it, because it can identify each user's typing pattern. This pattern is based on the speed of typing, which is different for every user. This typing speed is that "creates" the second cipher and connected to the first cipher produces a unique enhanced credential. Further down this method is presented and is made clear why only the legitimate owner of a password can only be accepted by the system.

To implement the above model, we used some variables to quantify the typing speed and use it in the user authentication process. To do this, we adopt salting techniques by joining the code with a << random >> biometric quantity. The model predicts that typing the user (account holder α) will exhibit changes in its biometric features (e.g., speed variations), and nevertheless accept the user as true. This is done as follows:

2.1.1 Definition : Distinctive Feature

The concept of a distinctive feature is central to the system. Generally, as a characteristic, we will consider a function Φ: (AxN) -> R_+, where Φ (a, 1) is the measurement of the l successful input into the system. Let Φi be a feature of the system (eg, the minimum of the click time that a key needs to be displayed on the screen). Let also μ_{ai} and σ_{ai} be the mean and the standard deviation of the timed keystrokes $\Phi(a, j_1), ..., \Phi(a, j_h)$ of the last h successful logins into the system. If each new entry has a new timed keystroke, then as distinctive attribute we mean the timing that holds $|\mu_{ai}-t_{ai}|>k\sigma_{ai}$. From this relation, it can be deduced that if Φi is a distinctive feature, then for every new time t_i with $t_i>\mu_{ai}+k\sigma_{ai}$ ή $t_i<\mu_{ai}-k\sigma_{ai}$, the user is not recognized because it is "timed" outside the typing model (which is strictly defined from the values μ and σ).

The main goal is to secure the user from offline attackers who have access to stored data of the pwd and tables of the mean and the standard deviation,which will make them a hard time to reconstruct hpwd even with brutforce. Even if the user chooses a "simple" pwd contained in bruteforce dictionaries, then, given that the user's typing features are not distinctive, the attacker has at least the same difficulty to find hpwd, as to find pwd. On the contrary, if all the user's typing features are distinctive, the difficulty the attacker has to face increases by a multiplicative factor by 2^m, where m is the number of the distinctive features.

2.2 How the system works

When an account is initialised, the program chooses hpwd randomly out of a ring Zq, where q is a sufficiently large prime, let's say with 160 bits. It then constructs a table of pairs (instruction table) with which it can reconstruct the new hpwd. The first column holds the standard deviation of the timed keystrokes that are less than the user's standard deviations, where the second has the standard deviations that are equal or greater than the user's ones. In this way, the system only accepts password changes from the certified user. The pair table has as many rows as the characters that the user types in his password. After the first successful user logins, the standard deviations in the matrix table can be changed, following the user's typing pattern.

2.3 A polynomial instance

In this implementation, hpwd is constructed with the aid of a random polynomial f_a , a polynomial ring Zq[x] of grade m-1, such that $f_a(0)$=hpwd . Points of the polynomial are the pairs of the instruction table. Let G be a pseudorandom family of equations such that for each key K and each character χ, then GK(χ) is a pseudorandom equation, element of the multiplicative ring Z_q. In practice, a possible implementation could be GK(χ)=F(K,x), where F can be an encryption function SHA-1. For each account, two data structures are stored in the system:

- the instruction table showing how to use the measurements to construct hpwd. Specifically, every feature Φi is written in the table in the form <i,ai,bi> where :

$$\alpha_{ai} = y^0_{ai} \cdot G_{\mathrm{pwd}_a}(2i) \bmod q$$
$$\beta_{ai} = y^1_{ai} \cdot G_{\mathrm{pwd}_a}(2i+1) \bmod q$$

The y^0_{ai},y^1_{ai} are elements of Zq* . At first when a user initializes a pwd all the 2m values {y^0_{ai},y^1_{ai}} , 1<=i<=m are chosen in such a way that all the points {(2i,y^0_{ai}),(2i+1,y^1_{ai})} 1<=i<=m be on the polynomial f_a of the Z_q[x] of grade m-1, with $f_a(0)$=hpwd.

- A fixed-size encrypted history file with measurements of all the characteristics of the last h successful entries. Specifically, if j_1, ..., j_ℓ logins tests were successful, then this file contains Φ_i(a,j) for 1<=i<=m and $j_{\ell-h+1}$<=<=j_l. This file initially has zero values, and then it is encrypted with hpwd using a symmetric key. This file should have a fixed size, because in this way it will not show how many successful entries have been made.

2.3.1 Steps of the polynomial implementation

a. If Φi is a characteristic of the i character of pwd, and if we denote pwd' each new input try to login with pwd, then the program uses pwd' to decrypt $\alpha_{\alpha i}$ if and only if $\Phi_i(\alpha,l) < t$, , otherwise it uses pwd' to decrypt the $\beta_{\alpha i}$.

$$(x_i, y_i) = \begin{cases} (2i,\ \alpha_{ai} \cdot G_{\mathsf{pwd}'}(2i)^{-1} \bmod q) & \text{if } \phi_i(a,\ell) < t_i \\ (2i+1,\ \beta_{ai} \cdot G_{\mathsf{pwd}'}(2i+1)^{-1} \bmod q) & \\ & \text{if } \phi_i(a,\ell) \geq t_i \end{cases}$$

The program stores m points $\{(x_i,y_i), \}$ $1<i<m$.

b. The program, composes the hardened password as below: Note that λi is the Lagrange interference factor

$$\mathsf{hpwd}' = \sum_{i=1}^{m} y_i \cdot \lambda_i \bmod q$$

$$\lambda_i = \prod_{1 \leq j \leq m, j \neq i} \frac{x_j}{x_j - x_i}$$

Then the program decrypts the history file with hpwd'. If the plaintext history file that is produced is the same as the original, then the login is successful.

c. The program updates the history file, calculates the new medium $\mu_{\alpha i}$ and the standard deviation $\sigma_{\alpha i}$ for each Φ_i attribute, for the last h successful entries, encrypts the new history file with hpwd' and saves replacing the old one.

d. The program creates a new polynomial f_α of the ring $Z_q[x]$ of grade m-1 such that $f_\alpha(0)$=hpwd' .

e. For each distictive characteristic with $|\mu_{\alpha i}-t_i|>k\sigma_{\alpha i}$, the program chooses random values for the y^0_{ai},y^1_{ai} from the ring Z_q* , with the following restictions :

$$\mu_{ai} < t_i \quad \Rightarrow \quad f_a(2i) = y^0_{ai} \wedge f_a(2i+1) \neq y^1_{ai}$$
$$\mu_{ai} \geq t_i \quad \Rightarrow \quad f_a(2i) \neq y^0_{ai} \wedge f_a(2i+1) = y^1_{ai}$$

For all the rest distinctive characteristics those that $|\mu_{\alpha i}-t_i|<k\sigma_{\alpha i}$, if successful logins are less than h from the creation of the account, the program sets $y^0_{ai}=f_a(2i)$, και $y^1_{ai}=f_\alpha(2i+1)$.

f. The program replaces the instruction table with a new, structured entry $<i,\alpha'_{\alpha i},\beta'_{\alpha i}>$, for each feature Φi :

$$\alpha'_{ai} \quad = \quad y^0_{ai} \cdot G_{\mathsf{pwd}'}(2i) \bmod q$$
$$\beta'_{ai} \quad = \quad y^1_{ai} \cdot G_{\mathsf{pwd}'}(2i+1) \bmod q$$

where y^0_{ai},y^1_{ai} have been created in the previous step (5) .

Let an offline attacker have access to the interaction table and the history file. Since the history file and y^0_{ai},y^1_{ai} are encrypted with pwd, and pwd is selected from significantly less options than hpwd is made, the easiest way to find hpwd would be to find the pwd first . We will show how that the encrypton scheme is safe for two reasons. First, finding pwd is no easier than hacking a hash cipher with a previously stored hash value. Secondly, the cost to compute hpwd is many times greater than finding pwd.

2.4 An exponential implementation

Suppose we choose a large enough prime p (let's say 1024 bits in the binary) so that it is impossible to attempt to calculate the corresponding discrete logarithm with requirement q/(p-1) . Let us also g a class element q in Zp *. The fundamental difference of this implementation is that $hpwd_\alpha$ is set to equal to $g^{fa(0)}$mod p, and instead of $\alpha_{\alpha i}$, $\beta_{\alpha i}$ to be stored in the instruction table,, the following are stored:

$$\begin{aligned} \gamma_{ai} &= g^{\alpha_{ai}} \bmod p \\ \delta_{ai} &= g^{\beta_{ai}} \bmod p \end{aligned}$$

Obviously, even if the attacker knows the pwd, the above method hides the y^0_{ai},y^1_{ai} , values from it, since he cannot calculate discrete mod p values. In this implementation, the way $hpwd_\alpha$ is created by pwd_α is a bit different since fa(0) is hidden by the program. Let Pwd' be the password entered by the user.

a. The program creates a table of pairs for each distinctive feature :

$$(x_i, z_i) = \begin{cases} (2i,\ (\gamma_{ai})^{G_{\mathsf{pwd}'}(2i)^{-1} \bmod q} \bmod p) \\ \hfill \text{if } \phi_i(a,\ell) < t_i \\ (2i+1,\ (\delta_{ai})^{G_{\mathsf{pwd}'}(2i+1)^{-1} \bmod q} \bmod p) \\ \hfill \text{if } \phi_i(a,\ell) \geq t_i \end{cases}$$

and then stores m pairs.

b. The program now sets

$$\mathsf{hpwd}' = \prod_{i=1}^{m} (z_i)^{\lambda_i} \bmod p$$

where λi is the Lagrange interference factor. Then, it decrypts the history file with hpwd' . If the decryption produces a history file the same as the original, the login to the account is successful. If the entry is not valid, the program stops here.

c. The program updates the history file, calculates the average $\mu_{\alpha i}$ and the standard deviation $\sigma_{\alpha i}$ for each Φ_i attribute for all successful entries in the account, encrypts the new history file with hpwd' , and saves it to the previous one.

d. The program constructs a new random polynomial f of the ring $Z_q[x]$ of grade m-1, such that f(0)=0 .

e. For each distinctive feature Φi with $|\mu_{\alpha i}-t_i|>k\sigma_{\alpha i}$, the program selects new random values y^0_{ai},y^1_{ai} from the set Zq* such that the following are valid :

$$\mu_{ai} < t_i \quad \Rightarrow \quad f(2i) = y^0_{ai} \wedge f(2i+1) \neq y^1_{ai}$$
$$\mu_{ai} \geq t_i \quad \Rightarrow \quad f(2i) \neq y^0_{ai} \wedge f(2i+1) = y^1_{ai}$$

For all the other distinctive features, that is $|\mu_{\alpha i}-t_i|<k\sigma_{\alpha i}$, if the successful entries are less than h from the creation of the account, the program sets y^0_{ai}=f(2i), και y^1_{ai}=f(2i+1) .

f. The program replaces the instruction table with a new, with logs of the form $<i,\alpha'_{\alpha i},\beta'_{\alpha i}>$ for each feature Φi.

$$\gamma'_{ai} \quad = \quad (\mathsf{hpwd}' \cdot g^{y^0_{ai}})^{G_{\mathsf{pwd}'}(2i)} \bmod p$$
$$\delta'_{ai} \quad = \quad (\mathsf{hpwd}' \cdot g^{y^1_{ai}})^{G_{\mathsf{pwd}'}(2i+1)} \bmod p$$

Also here, y^0_{ai},y^1_{ai} have been created in the previous step (5) .

2.5 An implementation with vector spaces

There is also a third implementation that uses vector spaces. This is prefered when some of the table attributes are not dinstictive, let's say the first m / 2, and the system will be able to make an instruction table of with the rest (dinstictive) elements. Here we present an implementation that constructs the instruction table using only one of each line. Here, hpwd$_\alpha$ is defined as the determinant of a table with elements from Z_q, where q is selected as before (quite large prime, eg with 160 bits in the binary system). In particular, when an account is created, m random vectors with data from the Z_q^m are constructed. Then hpwd$_\alpha$ = det($\underline{\upsilon}_{\alpha 1}$, $\underline{\upsilon}_{\alpha 2}$, ... , $\underline{\upsilon}_{\alpha m}$) mod q . So, the construction table holds a log for each feature Φi with the form $<i,\underline{\alpha}_{\alpha i},\underline{\beta}_{\alpha i}>$, where :

$$\underline{\alpha}_{ai} \quad = \quad \underline{\upsilon}_{ai} \cdot G_{\mathsf{pwd}_a}(2i) \bmod q$$
$$\underline{\beta}_{ai} \quad = \quad \underline{\upsilon}_{ai} \cdot G_{\mathsf{pwd}_a}(2i+1) \bmod q$$

Note that when creating pwd, and generally when there are no distinctive features, the elements of $\underline{\alpha}_{\alpha i}$,$\underline{\beta}_{\alpha i}$ are the same. The entry process is discussed below. Let pwd' be the string of characters that a user enters when logging in.

a. The program computes for each feature Φi :

$$\underline{\upsilon}_i = \begin{cases} \underline{\alpha}_{ai} \cdot G_{\mathsf{pwd}'}(2i)^{-1} \bmod q & \text{if } \phi_i(a,\ell) < t_i \\ \underline{\beta}_{ai} \cdot G_{\mathsf{pwd}'}(2i+1)^{-1} \bmod q & \text{if } \phi_i(a,\ell) \geq t_i \end{cases}$$

and stores m vectors $\underline{\upsilon}_i$} , 1<i<m .

b. The program sets hpwd'=det($\underline{\upsilon}_1$, $\underline{\upsilon}_2$, ... , $\underline{\upsilon}_m$) mod q . Then, it decrypts the history file with hpwd'. If decryption gives an 'appropriate' history file, then the entry is successful. Otherwise, the entry process stops here

c. The program updates the entries in the history file, calculates the standard deviation of the $\sigma_{\alpha i}$ and the average $\mu_{\alpha i}$ for each feature that is timed, and for all h successful entries, encrypts the new history file with hpwd ', and saves it to the old history file.

d. The program pick at random vectors <w_1,w_2 , ... , w_m> from Z_q^m such that det($\underline{w}_1$,$\underline{w}_2$,... , $\underline{w}_m$) mod q = hpwd' .

e. The program then does one of the following steps, depending on whether there are distinctive features.

a) If there is no distinctive feature, the program sets: $\underline{\upsilon}^0_{\alpha i} = \underline{\upsilon}^1_{\alpha i} = \underline{w}_i$, 1<i<m

b) In other case, the program sets new random vectors $\underline{\upsilon}_1$, $\underline{\upsilon}_2$, ... , $\underline{\upsilon}_m$ from Z_q^m , such that :

$$\forall b \in \{0,1\}^m : \quad \det(\underline{u}_1^{b(1)}, \ldots, \underline{u}_m^{b(m)}) \bmod q = 1$$

$$\underline{u}_i^{b(i)} = \begin{cases} \underline{e}_i & \text{if } b(i) = 0 \\ \underline{u}_i & \text{if } b(i) = 1 \end{cases}$$

and $\underline{e}_i$ is the unitary vector.

Then, the program chooses new random vectors $\underline{\upsilon}^0_{\alpha i}$, $\underline{\upsilon}^1_{\alpha i}$ of Z_q^m for each distictive feature Φ_i with the following restrictions :

$$\mu_{a\,i} < t_i \quad \Rightarrow \quad \underline{v}^0_{a\,i} = \underline{w}_i \wedge \underline{v}^1_{a\,i} \neq W \cdot \underline{u}_i$$

$$\mu_{a\,i} \geq t_i \quad \Rightarrow \quad \underline{v}^0_{a\,i} \neq \underline{w}_i \wedge \underline{v}^1_{a\,i} = W \cdot \underline{u}_i$$

For the rest features, those with $|\mu_{\alpha i}-t_i|<k\sigma_{\alpha i}$, the program sets $\underline{\upsilon}^0_{\alpha i} = \underline{w}_i$, και $\underline{\upsilon}^1_{\alpha i}$ = W*$\underline{\upsilon}_i$.

f. Finally, the program replaces the instruction table with a new table with logs of the form <i,$\underline{\alpha}'_{\alpha i}$,$\underline{\beta}'_{\alpha i}$> for each feature Φi :

$$\begin{aligned} \underline{\alpha}'_{a\,i} &= \underline{v}^0_{a\,i} \cdot G_{\mathsf{pwd}'}(2i) \bmod q \\ \underline{\beta}'_{a\,i} &= \underline{v}^1_{a\,i} \cdot G_{\mathsf{pwd}'}(2i+1) \bmod q \end{aligned}$$

where $\underline{\upsilon}^0_{\alpha i}$, $\underline{\upsilon}^1_{\alpha i}$ are the new vectors that were constructed during the previous step (5).

3. Zero Knowledge Proofs and Protocols [III, IV]

3.0 Introduction

Modern day authentication systems utilize cryptography and the concept of secret keys by allowing participators to communicate only if they have the proper identification: their own public and private keys. This system relies on both parties having previously agreed upon the secret key system, and requires each user to maintain secure possession of their own keys. Though this method may seem cryptographically secure, there are a variety of vulnerabilities. Attackers could obtain a users private key, or eavesdrop on the conversation and obtain the transmitted data. A zero-knowledge protocol is secure alternative to modern authentication systems because it allows for user authentication without transmitting vital data. With the current degree of online privacy at an alarmingly low rate, there is a need now more than ever for cryptographically sound protocols that allow for safe and secure transactions.

The use of web platforms and intranet systems led to the implementation of protocols where the server should not be aware of the exact identity of the verified user, but only accept him on the ground of success on some “challenges”. This implementation is what we name Zero Knowledge Protocols. However, it goes without saying that the verifying organization should prove somehow that it indeed does not have a clue about the identity / secret value of a user. This is how we moved to the Proofs of Zero Knowledge.

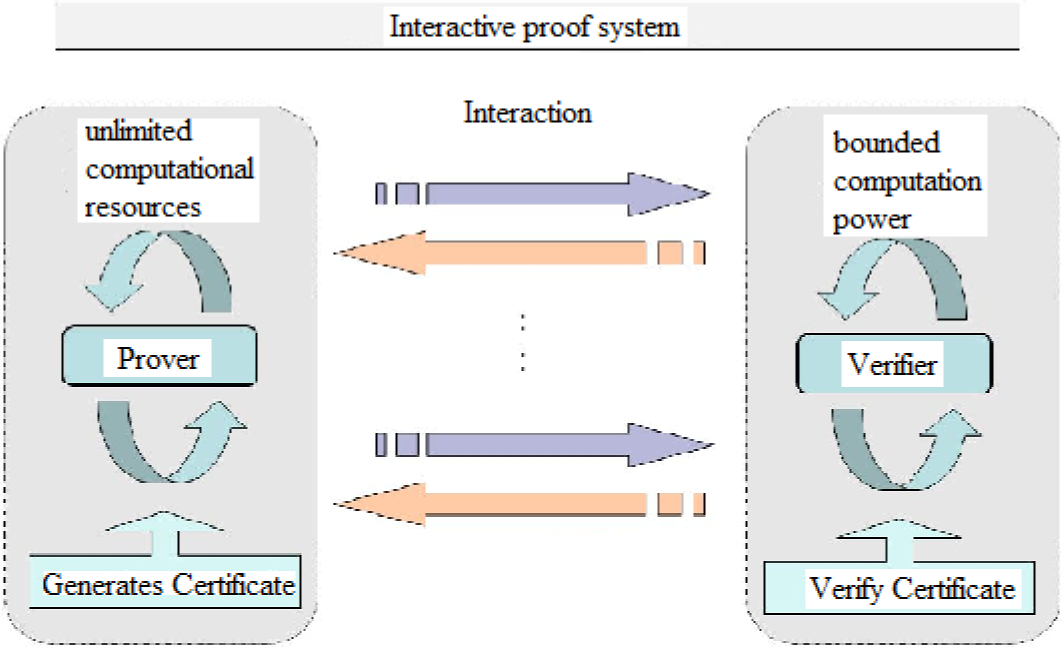
3.1 Scheme Overview

The exact implementation consists of the repetition of a challenge or a series of challenges, where the correct responses of the applicant grants him the access which he asks for. It goes without saying that the verifier can learn nothing about the applicant, but only that he can somehow succeed the challenges. The applicant should not cheat the verifier. The verifier can be sure about this by choosing the number of the challenges / the repetitions of a challenge. Given that the applicant is authentic, he should pass all challenges; it is the verifier's decision to accept a minimum probability of a fake applicant. The verifier cannot impersonate the applicant to any third party. Although some ZKP protocols are prone to man-in-the-middle-attacks, relevant cryptographic applications can make the scheme avoid this threat. (e.g. hash functions that can preserve the integrity of an applicant's identity).

The trust of the system lies on the type of proofs that are used. Probabilistic proofs here are used in the context that the verifier decides at which extend he can minimise the existence of a fake applicant. For some intranet systems, less than 10% could be enough, whereas in other web servers 1% of an impersonation threat would seem grave.

3.2 Necessary Definitions

Zero Knowledge Protocols embed the properties of Completeness and Soundness. Completeness is met when verifier and prover always both fulfill the correct function of the protocol, whereas Soundness means that a user can extract some knowledge in polynomial time (polynomial executions of the protocol) when impersonating an authentic user. In addition, the Zero Knowledge Property should be met, meaning that the all the information the verifier conveys is only what the prover gives him. Some more definitions follow in order to make clear the aspect of the proof we are demonstrating here.



3.2.1 Definition : Simulator

"A simulator is a method or procedure that generates fake (generated without the prover) views that are indistinguishable from a genuine (generated with the prover) view of a proof (Mikucki, 1999)"

When a simulator exists for a proof, that proof is a Zero Knowledge Proof. A simulator for a proof exists when you can recreate this proof, lets say faking a legitimate user, and this recreation cannot be distinguished from the legitimate user's view of proof.

3.2.2 Definition : Perfect Zero

"A protocol is said to be perfect zero knowledge if real and simulated transcripts are completely indistinguishable from one another"

3.2.3 Definition : Computationally Zero

"A protocol is computationally zero knowledge if an observer restricted to probabilistic polynomial time tests cannot distinguish real from simulated transcripts”

3.2.4 Definition : Statistical Zero

“A protocol is statistical zero knowledge if there is a negligible difference between the probability distributions of real and simulated transcripts”

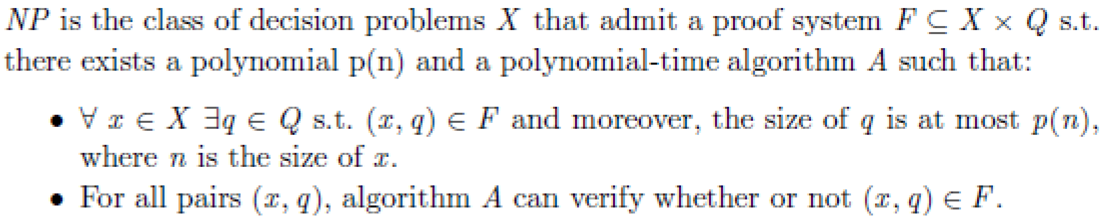
It is clear that the properties of Zero Knowledge and Soundness cannot guarantee the security of a system. The system's security should lie on computationally difficult problems. The system where the authentication lies on computationally difficult problems is referred as "provably secure system". Comparing the techniques of Zero Knowledge and Public Key authentication, we can comment that :

- ZK protocols are resistant to exhaustive tries and choosen text attacks and its security is not degraded; however it's not provably secure, where the PKP is.
- “ZK protocols are usually less efficient than PK protocols” . Certain environments that require real time computations, PK protocols are prefered.
- Computational functioning : most ZK and PK protocols function depended on the same unproven assumpions: quadratic residuosity, factoring, discrete logarithm are examples of implementations of both the protocols.

Therefore, the unbreakable scheme of the ZK protocols is not based on the difficulty to pass the challenges, but on this single capability: a correct answer can only be verified by the verifier/authority. There is no way to find an algorithm that could produce or foresee “correct answers to the verifier's challenges. This property makes a Zero Knowledge protocol be a Nondeterministic Polynomial problem.

3.2.5 Polynomial vs non Polynomial time algorithm problems

If P is the class of decision problems which can be solved by a polynomial time algorithm, then NP is the class of those decision problems that cannot be solved by a polynomial algorithm, or there not exist any solution at all. In particular, the following definition enlighten the matter:



NP is the class of decision problems X that admit a proof system $F \subseteq X \times Q$ s.t. there exists a polynomial p(n) and a polynomial-time algorithm A such that:

- $\forall\, x \in X\ \exists q \in Q$ s.t. $(x, q) \in F$ and moreover, the size of q is at most $p(n)$, where n is the size of x.
- For all pairs (x, q), algorithm A can verify whether or not $(x, q) \in F$.

The definitions of NP Complete problems and Polynomial reduction of a problem follow:

3.2.6 Definition: Polynomial Reduction

“Let *A* and *B* be two problems. We say that *A* is polynomially Turing reducible to *B* if there exists an algorithm for solving *A* in a time that would be polynomial if we could solve arbitrary instances of problem *B* at unit cost." This is denoted

$A \leq_T^p B$

3.2.7 Definition: Nondeterministic Polynomial

A decision problem X is NP-complete if

- $X \in NP$; and
- $Y \leq_T^p X$ for every problem $Y \in NP$.

3.3 Implementations

There are several approaches to ZKP. To provide a web-applicable solution, we can focus on graph isomorphism since all the necessary calculations are performed on integers within predictable computations time. In case of an approach that would rely on large prime numbers and discrete logarithm factorisation, predictability would not guaranteed as they appear in a random manner. Since the "solutions" of this challenge rely on the keys a third party trusted authority usually offers, this approach is usually implemented in authentication using personal computers, networks and servers. The graph isomorphism can be used even with rf pass cards, e-tickets and generally systems where the user does not need to interact with the system to gain access.

We now provide two implementations, where the prover needs to show the knowledge of a situation.

3.3.1 Non-isomorphic graph permutation

Let be a graph, for which the prover knows a coloring. In order to give a response to the verifier's challenge, the prover shall not reveil all the coloring of the graph, but only show the colors of a couple of adjacent lines () to an edge. If |E| is the number of the edges, the zero knowledge verification can continue up to $|E^2|$ times. During each challenge, the following steps take place:

"The prover permutes the three colors at random. This allows him to conceal the true coloring throughout the repetition of the steps.

• The prover "hides" the coloring from Vani (perhaps using encryption schemes).

• The verifier selects an edge of the graph at random.

• The prover reveals the colors of the two nodes for which the selected edge is incident.

• The verifier confirms that the two colors are valid, i.e., different."

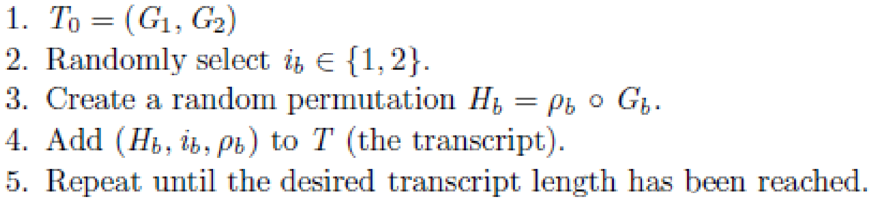
If everything is legit, (the graph is three-colorable, the prover knows a coloring) the verifier can never challenge an edge, on which the prover will give (less than three) color. If the graph is not three-colorable, or the prover tries to fool the verifier, there are 1/|E| odds that the prover may be caught. This probability exponentially decreases after $|E^2|$ challenges. A similar implementation includes a discrete logarithm test and sharing a set of public and private keys.

3.3.2 Another graph coloring implementation

This is the classic problem of Graph Isomorphism: Given two graphs, let them be G_1 and G_2, can we transpose the nodes of G_1 and the result be the graph G_2? The key pint here is that a prover knows of a certain isomorphism and he does not reveal it. The implementation goes as follows: Let the two graphs G_1 and G_2 have 5 nodes

1. The prover knows a permutation σ of the graph G_1, and this will be the secret asset. This σ permutation maps the {5,4,3,2,1} G_1's nodes to the {1,2,3,4,5} G_2's nodes. He selects another permutation π at random. He then takes the resulting graph Γ=π(G_1) and sends the graph Γ to the verifier.
2. The verifier sends the prover a random integer i,which can be equal to 1 or 2.
3. The prover computes a new permutation ρ such that Γ = ρ(G_i) . If i=1, the prover uses ρ=π. If i=2, the prover uses r=σ(π) , where σ is a fixed (and secret) permutation such that G_1 = σ(G_2) .
4. At last, the verifier checks that graphs Γ and G_i are indentical.

We should now show that this protocol is zero knowledge. We have already stated that if there exists a simulator for a protocol, then this protocol is zero knowledge. The following forgery algorithm generates false views of proofs, that never appear. These views shall be indistinguishable from the real ones.



1. $T_0 = (G_1, G_2)$
2. Randomly select $i_b \in \{1, 2\}$.
3. Create a random permutation $H_b = \rho_b \circ G_b$.
4. Add (H_b, i_b, ρ_b) to T (the transcript).
5. Repeat until the desired transcript length has been reached.

This algorithm creates perfectly possible transcripts, probably identical to the real ones. Indeed, it does not reveal any information of the graphs or the secret permutation, as a consequence any third party cannot learn anything, and an impersonator has practically very little chances to respond correctly to the relative challenges.

3.3.3 Discrete logarithm factorization

At first, a third party trusted authority chooses a positive integer n to be the product of two large primes, let's say n=p*q, where p and q should be primes large enough, with 512 or 1024 bits. The trusted authority then generates the set of a public and a private key for the prover.   The public key is a υ such that: $x^{2} = \upsilon$ modn and also $\upsilon^{(-1)}$ belongs to the ring (or multiplicative group) Z_{n} .

1. The private key is s such that: s=min{s=sqrt($\upsilon^{(-1)}$)modn}
2. The prover now chooses r such that gcd(r,n)=1 and $x=r^{2}$ modn ; he then sends x to the verifier.
3. The verifier sends the prover a random bit b, 0 or 1.
4. If b=0, the prover sends the verifier r and he lastly concludes that $x=r^{2}$ modn, and of course that the prover knows that r = sqrt(x)
5. If b=1, the prover sends the verifier y = (r*s) modn. The verifier then finds out that $x=(y^{2}*\upsilon)$ modn and concludes that the prover knows sqrt(x/υ) .

Since an impersonator could try to deceive the verifier in a single try attempt, the verifier can test the prover enough times until he is convinced that the prover really knows the secret s. It is strictly recommended that the prover should change r in each round.

3.4 Paradoxical ID Systems that use ZKP [V]

Zero Knowledge Protocols and Anonymous Authentication have been combined in the past to construct secure and flexible authentication schemes. During the Conference of Eurocrypt 1988 an instance of a ZKP was introduced. The technique suggests an easy way to enhance security, even between devices, such as the modern cell phones, laptops and smart cars. The instance involves a user/a device, and a verifier/a service provider. The device first sends the verifier a random test number and its identity. The verifier then asks the device a "random large" question, and then the device responds with a witness number. The access is granted when the verifier can reconstruct the test number from the witness number, the question and the identity, according to numbers published by the authority, and standard rules that regulate the interaction. This protocol enables interaction between users leaving in shadow their identity and the identity of their devices. This interaction can also be used to check messages that the device sends. "This method of proof is non-transitive" . A signature is a non-interactive proccess. This method also offers the verifier the capability to convince a third party that a genuine device signs a message. However, the third party does not learn any secret value at all that the device transmits to the verifier.

3.4.1 The actual implementation

The interactive protocol described above is based on the computation of discrete logarithm.

The following formula describes the way the protocol functions:

$B^{\upsilon}{}_{*}J$ mod(n) = 1, J=Red(I), where:

B: is a unique authentication number the device holds, which is related to its identity I

n: is a large enough composite number

υ: is the exponent published by the authority and known to any verifier

J: is the shadowed identity of the device, it's a number as large as n, it includes the indentity I, which is half shorter than n, wrapped with redundancy dependent on I. This

redundancy appears as a function Red(I). An ISO certification that was valid 30 years ago ISO-DP 9796 was based on a technique similar to the previously described. The authentication proccess takes place in the following four steps:
a. The device sends its identity I and a test nubmer $T = r^{\upsilon}$ mod n , where n is (postive) integer, r is a random (non zero) integer from the (multiplicative group) Z_n

b. The verifier asks a question d, which is a random integer between 0 and υ-1.

c. The device responds with a witness number t = $(r□B^d)$mod n, where B is the authentication number.

d. The verifier needs to check the witness number t, so he makes the following computation:
$(J^d * t^{\upsilon})\text{mod } n = J^d\ (r * B^d)^{\upsilon} \text{ mod } n = (J * B^{\upsilon})^d{}_* r^{\upsilon} \text{ mod } n = T$

3.5 Different Instances

3.5.2 Discrete identities sharing the same exponent [VI]

There is an implementation where two discrete entities (with different identities) get involved in an authentication scheme using the same secret exponent. Each entity (or device) has a unique authentication number B, which depends on its identity, I_1 and I_2 ; this appears in the following equations :

$$B_1^{v} \cdot J_1 \bmod n = 1, \text{ with } J_1 = Red(I_1),$$
$$B_2^{v} \cdot J_2 \bmod n = 1, \text{ with } J_2 = Red(I_2).$$

The two entities can cooperate on the same computer (or network) and participate in an authentication scheme following the steps bellow:

- Both entities transmit their identities I_1 and I_2 and their test numbers T_1 and T_2 .

In particular, the following equations apply :
$T_1 = r_1^{\upsilon}$ mod n and $T_2 = r_2^{\upsilon}$ mod n , where r_1,r_2 are random numbers from Z_n^*.
The personal computer (or newtwork server) sends to a third trusted party (verifier) the two identities I_1 and I_2 and the common test number computed from:

$$\begin{aligned} T &= T_1 \cdot T_2 \bmod n \\ &= (r_1 \cdot r_2)^{v} \bmod n \\ &= r^{v} \bmod n \end{aligned}$$

It goes without saying that $r = r_1 * r_2$ mod n.

- The verifier asks a question d, which an random integer from 0 to υ-1.

- Both entities send each a witness number t_1 and t_2 , and it applies :

$t_1 = r_1 * B_1^d$ mod n , and $t_2 = r_2 * B_2^d$ mod n .
Then, the server sends the verifier the common witness number t:

$$\begin{aligned} t &= t_1 \cdot t_2 \bmod n \\ &= (r_1 \cdot B_1^d) \cdot (r_2 \cdot B_2^d) \bmod n \\ &= (r_1 \cdot r_2) \cdot (B_1 \cdot B_2)^d \bmod n \\ &= r \cdot (B_1 \cdot B_2)^d \bmod n. \end{aligned}$$

- "In order to check such a witness number t, the verifier computes the product of the d power of the shadowed identities J_1 and J_2 by the υ power of witness number t, that is "

$$\begin{aligned} J_1^d \cdot J_2^d \cdot t^v \bmod n &= J_1^d \cdot J_2^d \cdot (r_1 \cdot B_1^d \cdot r_2 \cdot B_2^d)^v \bmod n \\ &= (J_1 \cdot B_1^v)^d \cdot (J_2 \cdot B_2^v)^d \cdot r^v \bmod n \\ &= T. \end{aligned}$$

This protocol of cooperation between entities with discrete identities and the same exponent, also applies to any number of entities and suggests a new multiple signature scheme.

3.5.3 Two entities with different exponent, sharing the same identity

The scenario of this case includes two devices who have stored their unique authentication numbers B_1,B_2 related to the same identity I, and use different exponents υ_1 and υ_2, as follows : mcd(υ_1,υ_2)=1

$$B_1^{v_1} \cdot J \bmod n = 1 \text{ and } B_2^{v_2} \cdot J \bmod n = 1, \text{ with } J = Red(I).$$

The cooperation between the two devices may simulate an entity with the identity I, and the exponent $\upsilon=\upsilon_1*\upsilon_2$, $(B^{\upsilon} *J)$mod n = 1 , $B_1 = B^{\upsilon}{}_2$ mod n and $B_2 = B^{\upsilon}{}_1$ mod n .
The two devices (or entities) function on the same network (or a shared personal computer) , negotiate an authentication procedure with a verifier following the steps of the next protocol :

- The two devices send the identity I, and the test numbers T_1, T_2 which are the υ power in Z_n of the corresponding random integers r_1 and r_2 from the Z_n * .

$T_1 = r_1^{\upsilon 1}$ modn , $T_2 = r_2^{\upsilon 2}$ modn .
The network server sends the verifier the common identity I, and the common test T :
$T = (T_1*T_2)$ modn $=(r_1*r_2)^{(\upsilon 1*\upsilon 2)}$ modn $= (r_1*r_2)^{\upsilon}$ modn $= r^{\upsilon}$ modn , $r = (r_1*r_2)$modn

- The verifier asks a question d, which is a random integer ranging (1,υ-1). The personal computer now translates this question in $d_1 =(d/\upsilon_1)$ modn , $d_2 = (d/\upsilon_2)$modn for each entity.

- Each entity sends a witness number

$t_1 = r_1B_1^{d1}$ modn
$t_2 = r_2B_2^{d2}$ modn
The personal computer sends the verifier the common witness number
$t = (t_1*t_2)$modn $= (r_1B_1^{d1} *r_2B_2^{d2}$) modn $= r*(B^{d1\upsilon 1+d2\upsilon 2})$modn .

- Let's call $d' = d_1\upsilon_1+d_2\upsilon_2$. The verifier now checks witness number t, by computing $J^{d'}*t^{\upsilon}$ modn and testing weither this quantity is equal to T .

$$\begin{aligned} J^{d'} \cdot t^{v} \bmod n &= J^{d_1 \cdot v_2 + d_2 \cdot v_1} \cdot (r_1 \cdot B_1^{d_1} \cdot r_2 \cdot B_2^{d_2})^{v_1 \cdot v_2} \bmod n \\ &= (J \cdot B_1^{v_1})^{d_1 \cdot v_2} \cdot (J \cdot B_2^{v_2})^{d_2 \cdot v_1} \cdot (r_1 \cdot r_2)^{v} \bmod n \\ &= T. \end{aligned}$$

This protocol offers a unique method for anonymous authentication between a client, a server, and a third party authority. Usually, the third party authority provides the other parties with a secret value (exponent)

Similarly, to sequential protocol, interrupted sessions are the main threat for interactive ZKP protocols in which all challenge questions are sent in parallel. In this scenario, an attacker prepares a set of graphs and corresponding permutations to G1 or G2; consequently, the attacker is able to respond to only one a priori chosen challenge for each graph. As long as the expected response does not come, the attacker interrupts and restarts the authentication procedure with the same graphs, expecting to obtain the different responses. In this scenario, the attacker’s chances drop exponentially, with the linear growth of the number of challenges requested by the verifier. We can calculate these chances using the Bernoulli scheme: the probability of at least one success in N trials of nested session impersonation attacks can be expressed as

$$P(N,p) = \sum_{k=1}^{N} \binom{N}{k} (p)^k (1-p)^{N-k}$$

Where p = ($1/2^t$) and t is the number of challenges.

Taking advantage of the fact that this probability is supplemented by the probability of zero successes in N trials (k = 0), we are able to simplify the formula to

$P(N,p) = 1-(1-p)^N$

To authenticate securely, the number of necessary attempts should be high enough to make such an attack impractical and infeasible. In the case of 30 challenges the attacker would need at least half-a-year to perform an interruption attack with only 10 % success probability.

For 40 challenges, a successful attack is completely infeasible even assuming a smaller success probability.

3.6 An Elliptic curve implementation[VI]

A wide variety of zero-knowledge protocols based on the Discrete Logarithm Problem (DLP) has been proposed so far, e.g. in [29], [33]. The Discrete Logarithm Problem is defined over arbitrary cyclic groups. A common example of cyclic group is the multiplicative group Z^*_n of order n, where n is a prime number and the group operation is multiplication modulo n. In such a group the Discrete Logarithm Problem (DLP) can be defined as follows: Given a prime n, a generator g of Z^*_n and an element b from Z^*_n, find the integer x, $0 \leq x \leq n-2$ such that $g^x = b (\bmod n)$. Another common example of cyclic groups are elliptic curve groups which are defined over an additive group F of order n (note that n is no longer necessarily a prime number). The analogous problem to DLP over elliptic curve groups is called ECDLP (Elliptic Curve Discrete Logarithm Problem) and can be defined as follows: Given an elliptic curve E over a field F of order n (refered to as F_n from now on), a generator point G from E/F_n (quotient group) and a point B from quotient it is computationally hard to find x such that B = x*G. In this section, we show how well established zero-knowledge protocols based on the DLP can be adapted under the Elliptic Curve Discrete Logarithm Problem (ECDLP). This adaptation is a key step for porting such protocols to low constrained devices because of the Elliptic Curve Cryptography (ECC) advantages. As one can see in the Appendix, ECC can offer the same level of security as other public key cryptosystems, using smaller key sizes. This fact makes it suitable for implementations that concern constrained environments as it saves computational time and memory space and consequently reduces energy requirements. Such restrictions consist the real challenges when considering implementations on embedded devices.

3.7 Zero Knowledge Proof of Discrete Logarithm with Coin Flip[VII]

One of the first zero-knowledge protocols of discrete logarithm that was already presented. Its elliptic curve analogous is as follows: Given an elliptic curve E over a field F_n, a generator point G from $E=F_n$ and B from E/F_n (quotient group) Prover wants to prove that he knows x such that B = x*G, without revealing x.
Protocol Steps:

- Prover generates random r from F_n and computes the point A = r _ G
- Prover sends the point A to Verifier
- Verifier flips a coin and informs the Prover about the outcome
- In case of HEADS Prover sends r to Verifier who checks that r*G = A
- In case of TAILS Prover sends m = x + r(mod n) to Verifier who checks that

m*G = (x + r)*G = x *G +r* G = A + B
The above steps are repeated until Verifier is convinced that Prover knows x with probability $1-2^{-k}$ for k iterations.

3.7.1 Why it works:

The protocol works as expected because in each iteration the steps to be executed depend on the outcome of the coin that the Verifier flips and the Prover cannot affect this. It needs to be executed for many iterations in order for the Prover's cheating probability to become very small. A dishonest Prover in each iteration can be prepared for only one of the coin outcomes and thus his cheating probability is 1/2. For example, if he prepares for TAILS he can generate a random m, compute A = m*G-B and send this point A to Verifier. But if HEADS come up this attack will not work. That is because he will need to compute a value r from F_n that generates A and that is an instance of the ECDLP. Thus, after k iterations, the Verifier is convinced with high probability ($1-2^{-k}$) that the Prover is honest.

3.8 Schnorr’s Protocol

The elliptic curve version of Schnorr's protocol, slightly modified, is the following: Prover and Verifier agree on an elliptic curve E over a field F_n a genereator G from E/F_n . They both know that B is from E/F_n and Prover claims he knows x such that B = x*G. He wants to prove this fact to Verifier without revealing x.

3.8.1 Protocol Steps:
Prover generates random r from F_n and computes the point A = r*G
Prover sends the point A to Verifier
Verifier computes random c = HASH(G;B;A) and sends c to Prover
Prover computes m = r +c*x(mod n) and sends m to Verifier
Verifier checks that P = m*G-c*B = (r+c*x) *G-c*B = r*G + c*x*G-c*x*G = r*G = A

3.8.2 Why it works: This protocol is superior to the previous one as it needs to be executed for one round. Verifier's coin flips (in correspondence with the Coin Flip protocol) are simulated using a hash function known only to him. A dishonest Prover has a tiny chance of cheating as he would have to fix the value of P = m*G-c*B before receiving Verifier's hash value c. Under the assumption that the hash function used by the Verifier is secure, a Prover who does not know x, the discrete logarithm of B, cannot cheat.

3.9 From Schnorr's Protocol to Digital Signature

We assume that with the use of a hash function and an agreement on an initial message m we can remove the interactivity from such protocols. The Verifier's random choices can be replaced with bits produced by a secure hash function. Thus, the next protocol is proposed.

3.9.1 Protocol Steps:

- Prover generates random r from F_n and computes the point A = r*G
- Prover computes c = HASH(x*P; r*P; r*G)
- Prover computes s = r + c*x(mod n)
- Prover sends to Verifier the message: "s||x*P||r*P||r*G"
- Verifier computes c = HASH(x*P; r*P; r*G)
- Verifier checks that s*G = (r+c*x)*G = r*G+c*x*G =r*G + c*B = A + c*B
- Verifier checks that s*P = (r+c*x) *P = r *P +c*x*P

3.9.2 Why it works

This protocol is the application of the non interactive instance of the protocol that is proposed further above. The Prover simulates both the Prover and the Verifier with the use of a hash function and publishes the transcript of this whole dialogue. This way the Prover sends only one message and the Verifier either accepts or rejects. The Prover generates a random number as
in previous protocols but the Verifier's random choices are simulated by hashing the input along with a value calculated from the Prover's choice of r. Thus, the Verifier's random choice depends on Prover's random choice and it is made hard to fake the outcome. The value c is really a challenge for the Prover as it is computed from the hash function and it is out of his control. If the Prover does not know x, in order to cheat he would try to find s satisfying s*G = r*G+c*x*G
which is an instance of the discrete logarithm problem. He could not cheat by enumerating random r values, as it would be too hard to find a matching value for c.

4. Several attack attempts [VII]

4.1 Offline Dictionary Attacks

In this scenario we assume that an attacker has a pair of the public graphs G1 and G2 and attempts to calculate if a given password candidate satisfies the relation G1 = Πcand(G2), where Πcand is computed from the password candidate. Password dictionaries are different from language dictionaries. Therefore, we ran tests using dictionaries specially created for this purpose. The efficiency of dictionary attacks is, however, questionable. On the one hand, according to Florencio and Herley, users tend to pick low quality passwords that are easy to guess. These passwords are also often reused among many web-sites of different security measures . On the other hand, Schneier claims that only 3.8%of users' passwords can be found in passwords dictionaries. Therefore, even if a single password can be verified with a dictionary in a reasonable time, the probability of success is really low. Moreover, a commercially deployed application can make this process even more difficult by taking advantage of a tool that imposes strong users passwords. Nyang proposed a generic framework for the whole family of interactive zero-knowledge proof authentication protocols. It is based on secret coin tossing and it introduces additional complexity as it requires additional communication and computations to perform simplified EKE to proof that the server has the knowledge of the client's verifier.

4.2 Brute-force Attacks

To perform a naive brute force attack, we created a program that enumerates and tests passwords of a given length. We limited the number of characters to the keyboard accessible characters only. The speed of such an attack was comparable to the one obtained for dictionary attacks; and on average we were able to generate and test 226 passwords per second. For the computer used for evaluation, testing a password of size 6 characters or more was infeasible due to long computation time, which was estimated to longer than one year.

The users are usually not able to conduct offline dictionary, or brute force attacks. Adversaries will rather try to hijack other accounts by
1) attempting to install a keylogger to monitor user's input;
2) by using social engineering attacks such as phishing;
3) by compromising the server database;
4) by applying some well-known web specific attack such as request forge or cross-site scripting; or
5) in case if the adversary is on the same LAN network, by compromising the network connection for the user by applying tools such as SSL Strip.
A more sophisticated attack could rely on influencing the random numbers generator in a way that the graphs would be predictable or that they would belong to a class that it is known to be breakable, e.g., graphs with fixed genus or trees.

5. Application in Sensor Networks[VIII]

We implemented the new proposed zero-knowledge protocols based on the ECDLP using the Wiselib platform which is a generic algorithm library. Then, we evaluated the implemented protocols on two popular hardware platforms equipped with popular low-end microcontrollers (Jennic JN5139, TI MSP430) as well as 802.15.4 RF transceivers in terms of execution time, code size, message size and energy consumption.

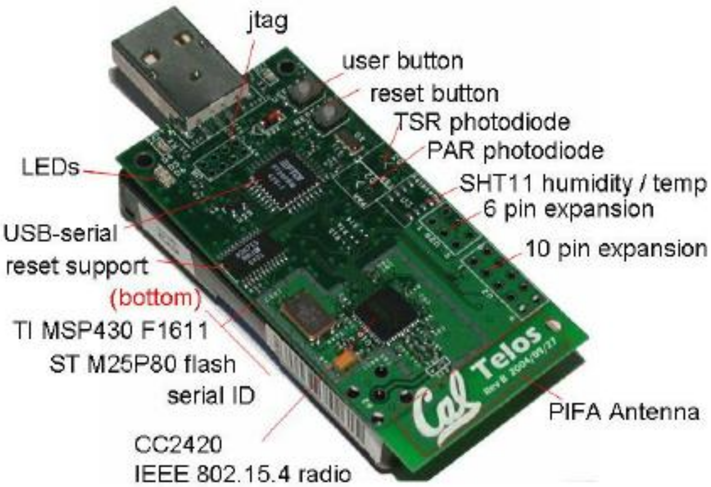
5.1 Wiselib: A Generic Algorithm Library for Sensor Networks

We decided to implement our algorithms using Wiselib [5]: a code library, that allows implementations to be OSindependent. It is implemented based on C++ and templates, but without virtual inheritance and exceptions. Algorithm implementations can be recompiled for several platforms and rmwares, without the need to change the code. Wiselib can interface with systems implemented using C (Contiki), C++ (iSense), and nesC (TinyOS). A future plan for this library is to be adapted for C-based mobile phone operating systems like Android and iPhone OS. Furthermore, an important feature of Wiselib are the already implemented algorithms and data structures. Since different kind of hardware uses different ways to store data (due to memory alignment, inability to support dynamic memory, etc.), it is important to use these safe types as much as possible since they have been tested before on most hardware platforms. As of mid 2010, the Wiselib includes about 40 Open Source implementations of standard algorithms, and is scheduled to grow to 150-200 algorithms by the end of 2011. Additionally, Wiselib runs on the simulators Shawn and Tossim, hereby easing the transition from simulation to actual devices. Tossim is a popular tool in the TinyOS community as it allows to simulate the exact source code that will run on the hardware and by using Power-Tossim it can provide accurate estimates on the power consumption of an application. Shawn allows repeatability of simulations in an easy way by using only a single configuration file. It provides many options such as packet loss, radius of communication, ways of communicating and even mobility in an abstract way, without needing to provide specific code for every range. This Wiselib feature allows us to validate the faithfulness of our implementation and also get results concerning the quality of our algorithms without time consuming deployment procedures and harsh debugging
environments. Finally, an advantage of Wiselib is that with the aid of template specializations the algorithm code can be optimized and adapted for certain platforms. Depending on the compilation process, the compiler can select the code that fits best for the current platform (e.g. if there is a 32-bit processor) and exploit the presence of special platform hardware (e.g. the Jennic AES hardware for speedup of crypto routines).

5.2 Hardware

As mentioned earlier we used two diferent low-end devices for evaluating the implemented protocols. The first device is the Coalesenses iSense and the second is the Crossbow
TelosB. These devices are quite popular for their application in the area of wireless

sensor networks. The first device (iSense) consists of a Jennic JN5139 32-bit RISC controller running at 16MHz. The ROM of this controller is 192Kb and its RAM is 96Kb that can be shared among program and data. It is equipped with 2.4Ghz IEEE 802.15.4 compliant RF transceiver (CC2420 chip) that can achieve bandwith up to 250 Kbps. Finally, this device runs the iSense firmware. The second device (TelosB) consists of a Texas Instruments MSP430 16-bit microcontroller running at 8MHz. Its RAM is 10Kb and the program flash memory is 48Kb. This device is also equipped with 2.4GHz IEEE 802.15.4 compliant RF transceiver (CC2420 chip) able to achieve data rates up to 250Kbps. Finally, the TelosB device can run TinyOs 1.1.10 [34] (or higher) or the Contiky operating system



5.3 What is TelosB

TelosB is of the methods of wireless network which has sensor devices, solar radiation sensor, humidity temperature sensor, USB, Antenna, deformation meter and microcontroller. It has total 8 sensor channels with high compassion which is mostly used for research or experimental purpose in an organization. The main advantages of it are:

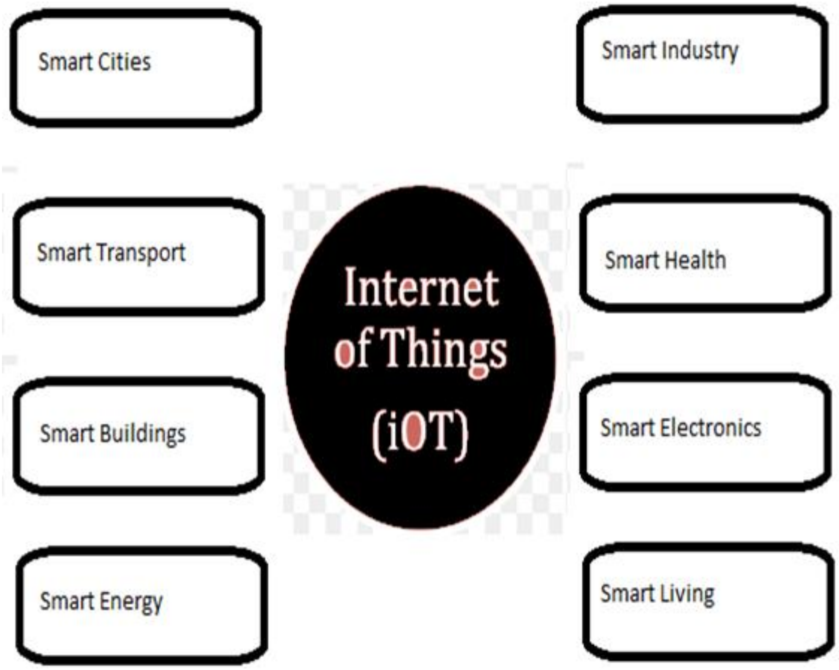
- We can get this device at a very affordable cost which attracts the customers to buy the product.
- This device can be used without any hassle or does not require any technical skills to operate. It is very compatible to use this product.
- It can be used in large organizations for research or for production purposes.
- We can add as many sensors as we want as per the requirement of the customers or clients for their business purposes.
- We have radio communication in this device which transmits the data via radio within a stipulated time.
- We have to always keep in mind that the TelosB device has to be kept away from heat. We should not connect on the right side of the laptop as there hot air is generated which makes the device loose its connectivity.

6. Authentication in Smart Mobile Devices [VIII]

6.1 Introduction

IoT a reality over the Internet in which things including people, objects, information and places to be connected through wireless or wired network at any time, at any place. With this new revolution, Internet is expanded from communication devices to the enterprise assets and consumer goods. IoT creates an intelligent environment and unique addressing is implemented to enable communication1. So every object connected can be tracked. Each participant autonomously interacting and communicating via internet and no centralized authority is there to control the objects. Figure 1 depicts the Internet of Things in Smart Environment.

IoT facilitated the interaction of human with anyone over the world with a smart sensor device. IoT include technologies to acquire and process contextual information like sensors, Near Field Communicators, Global Positioning Systems etc. The IoT brings many opportunities to the society but these technologies penetrate all the aspects related to the communicator and require solution to improve security and privacy. Now billions of internet connected devices found and creates open global network connectivity for people to improve people's lives. As a result trillions of things connected via internet and more IoT applications have been implemented. IoT brings many opportunities in business, industry, and technology to increase its performance, at the same time adding complexities to information technology. From the technology perspective the data in IoT is generated by machines and increase the density by Moore's law. A smart object with enough memory is capable to recognize and store information about people and other object in the network. Hence a major functional requirement of IoT is the preservation of security improvements and privacy.



Protection of data is a serious issue, when devices are connected to outside world. In IoT a person is always traceable and smart devices collect the data and information without their knowledge hence violate the security service let alone. Almost all information collected by these smart sensors is private and confidential. So this security related challenges need to be addressed by the research community.

In IoT communication enabled between smart object and social medias and is vulnerable to trudy involvements. The mobility, dynamic nature and weak physical security of Mobile devices also made it a surface for attack. IoT connected devices lead to the privacy leakage and exposure of authentication credentials to the hackers. So a more secure authentication mechanism from the client device itself is required for secure browsing6,7. This research presents a literature review and a promising prototype for authentication in IoT environment.

6.2 Authentication

For implementing trust in IoT communications and ensuring the goals of information security, we are required to take necessary care for server authentication and user authentication. Authentication is the process of validating one’s identity in communication and ensures the reliability of origin of communication. It is one of the primary goals of security and acts as a gateway in front of a secure system to prevent the malfunctions. When more devices are connected, then a new mechanism need to be developed to authenticate the users and devices. The authentication mechanisms used in commercial applications categorized into four – something you know, something you have, something you are and some place where you are. Among these most common authentication scheme used is user ID and password submission mechanism over a Secure Socket Layer connection. Sometimes the systems calculate the cryptographic hashes and avoid the transmission of plain text password. But the credentials are sent via the internet and the availability of wireless hotspots are growing so vulnerable to access by the intruders even if it is hashed. Also 3G GSM connection is unsafe and crackable within 2 hours. Hence we require a solution without revealing our secret for authentication. Conventional authentication to a system always results overhead to the server and time consuming procedure at end user machine. So to overcome this issue, current researches focus on a solution for Memory and Battery constrained Smart devices. This research extends a light weight solution in small footprint with high performance and low cost for IoT environment.

Also when we analyzed the IoT devices, it is found that majority of devices lack password and authentication mechanisms. The use of weak passwords and traffic encryption is a major issue in IoT. Now a day people increasingly used their hand held mobile devices for banking, payment, shopping etc. hence it will be beneficial to protect their identity and ensure authentication.

In IoT the possible communications are device to device, device to human, human to human and hence support heterogeneous entities and networks. As devices have no prior knowledge about other entities and no SSL communication is enabled, evesdropping is possible. Moreover IoT smart devices with sensors and actuators exchange and collect the personal data for authentication and chance to have unauthorized revelation of identity. So for personal data protection and anonymity we require an entirely different access control, authorization and attack detection mechanisms. The discrimination from sensor output is a big problem and the privacy law is still unprepared for IoT.

In traditional authentication process client submits its user id and password, client machine creates the hash of the password then transmit the user id and password hash via network. The reply packets from the server are also transmitted via network. A public Wi-Fi or 3G mobile broadband is used to transmit these credentials and is vulnerable to attacks. A hacker can sniff the credentials and can use it later to avail services from the server or he can use some software's to recover the password from the hashes.

The authentication mechanisms are mainly classified into private key based, public key based and one time signature based. Public key based systems require high computation, communication and storage overhead. Also existing private key mechanisms are not feasible for resource constrained devices and an internet security standard like TLS does not support small embedded units.

Due to portable nature, wireless connections and devices connected together in network access layer, IoT require a specific security concern. Hence Zero Knowledge proof is a best choice for such devices.

Slawomir et al extends web applications with Zero Knowledge Proof (ZKP) algorithm based on isomorphic graphs. There experimental evaluation shows ZKP is feasible with existing web standards with advantages of asymmetric key cryptography. This solution allows server to verify the authenticity of web client without directly checking the secret credential of client.

In Implementing Zero Knowledge Authentication with Zero Knowledge (ZKA_wzk), Lum Jia Jun and Brandon provide a practical web/python implementation of Zero Knowledge authentication protocol. This implementation is used to prove that it is able to prove the password is correct without revealing the password. The simplicity and ease of their implementation prove that Zero Knowledge Protocol is suitable choice for IoT authentication13.

In 2012 Manish P Gangawane finds the importance of Zero Knowledge Proof in wireless sensor network for identification of attacks. IoT devices attach with a variety of sensors and connected to wireless networked environment. These sensors are automatically controlled and there is an issue of security. In this he implemented Zero Knowledge Proof for the verification of sender sensor nodes14.

Parikshit N Mahalle et al. in Idenity Authentication and Capability based Access Control (IACAC) for the Internet of Things tried to implement authentication and access control in Internet of Things. Paper presents a secure ECC based integrated approach for authentication and access control and claimed that method is efficient in terms of computational time. The protocol is suitable to defy Denial of Service attack, Man in the Middle attack and reply attack15.

In 2013 Xuanxia Yao et al proposed a lightweight multicast authentication mechanism for small scale IoT applications. The authors analyzed the importance of Nyberg's fast one way accumulation in security and revised the algorithm to make it suitable for lightweight environment. Also they present an evaluation of the model based on probability theory and evaluate their design for the performance aspects. In the paper they claimed that multicast authentication algorithm meets the requirements of resource constrained applications16.

Tuhin Borgohain et al. analyzed various authentication systems implemented to preserve the privacy of user credentials in Internet of Things. In first part of their paper they proved that Multi Factor Authentication systems are not applicable to the field of Internet of Things even though it provides greater security to user credentials. The paper suggested the importance of OAuth for IoT based security. The method results a secure experience for login to the resource server. They point out the relevance of an access token to access the resources from server.

In September 2014 Padraig Flood et al presented a graph theory based ZKP approach for securing the Internet of Things. The purpose of their research is to determine a security infrastructure for embedded processors in Internet of Things and a resource efficient alternative for existing standards. They summarized the study by pointing the need of future researches required in IoT.

Most recently in January 2015 Jitendra Kurumi and Ankur Sodhi conducted a survey of Zero Knowledge Proof for authentication, identification, key exchange and other cryptographic operations. The surveys proved that ZKP implementations solved the problems in cryptography and provide lightweight solutions within small footprint.

In Real time authentication system for RFID applications, Swathi Kumari introduced a new security layer for authentication. The application captures location information then matches it with predefined authorized location for granting access to the system. In Real. This method is suitable for RFID devices and offer secure authentication using back end servers when compared with previous methods for RFID authentication.

Jae-Kyung Park et al. proposed authentication service to resolve the existing certificate problems and presents a certification device. The system is based on Public key cryptography and the operators need to prepare separate certification method22.

Traditional Authentication and Access Control solutions are not suitable for resource and battery constrained smart environment. The lack of implementation of lightweight authentication mechanism is concentrated on this research and proposes a new light weight method for trust management specifically for smart mobile devices.

Smart Mobile devices are manufactured by consumer goods makers and lack the data security in many cases. At the same time intelligent objects in these devices are prawn to security flaws. So an Authentication module with at most care is a requirement for these devices.

6.3 Proposed System and Methodology

A strong authentication and access control module suitable for available footprint is designed based on Zero Knowledge Protocol. ZKP is a concept which allows a communication party to prove that he knows a secret without revealing the secret. The verifier only knows that information is true19. The properties of ZKP include completeness, soundness and zero knowledge.

A device wish to connect to a resource owner must require registering with the resource owner. Resource owner select a group G and select a random number g0 belongs to the group G. The clients who wish to communicate with the owner must agree with these global public elements. In registration process the client inputs user ID and password. An authentication application at client side generates the hash of the password X and compute Y= g0X and sends user ID and Y to the resource owner, server stores these information in its SQLite database.

In authentication module when client initiates communication then resource owner generate a onetime token OTP by applying Pseudo Random number Generation algorithm and save in data base with Clients user ID. The server then encrypts the OTP with a 4 digit key generating from system clock by combining current hour and minute. Resource owner send this encrypted OTP via Short Message Service. The authentication module installed in client device decrypt it by current time and retrieves the OTP for authentication. The client is only able to decrypt it within 60 seconds, now all devices used the standard time from satellites so no synchronization is required.

After decrypting the OTP user select a random key r which is also an element of group G and calculates g0r and concatenates this with Y and token. Next procedure is to apply hashing algorithm to prepare the digest C from the concatenated result and compute
Z=r-C*X. Finally the client sends C and Z to the resource owner.

When C and Z from the client received, server calculate Y^C, $g_0^Z = g_0^{X*C}$, $g_0^{r-CX} = g_0^r$. Now the server has g0r, Y and OTP concatenate all these apply same hashing procedure and verify the received hash. For hash preparation we propose a revised Fast – One way Accumulation suitable for IoT environment. The hash preparation algorithm is now explained below.

Step 1: Read and separate the plain text password/ characters from the text from which the system need to produce the hash.
Hence password P is treated as $P = P_1P_2P_3P_4\ldots\ldots P_n$

Step 2: Map each character in the plaintext to another set of values by applying a simple mathematical function and we can designate them as
$Y_1=H(P_1)$, $Y_2=H(P_2)$, $Y_3=H(P_3)$,………. $Y_n=H(P_n)$

Step 3: Use encrypted OTP received from the resource owner as the initial key value for hashing.

Step 4: Prepare an HMAC with OTP by applying cumulative hashing
$H\ (\ldots\ldots H\ (H\ (H\ (OTP,\ Y_1),\ Y_2)\ Y_3)\ \ldots..\ Y_n)$

For preparing and verifying the hashes both resource client and resource owner need to agree with a hash function and seed value. Here we use the same OTP received in encrypted format from the server for ZKP implementation. Also a secure way is identified for symmetric key exchange. Finally we evaluated our algorithm by a prototype implementation in mobile operating system. The main functionalities included in the prototype summarized in Table 1

Table 1	Functionalities of prototype model
Agreement of Global Public Elements Registration	
Registration with Resource Owner	
Token Generation and Encryption	
Retrieval of Token by Decryption	
Hash Preparation and Verification	

6.4 Results and Discussion

Based on the initial prototype model, we proposed a light weight power efficient authentication and access control algorithm for smart mobile devices in IoT. Table 2 depicts the computational and memory requirements of the cryptographic protocols as per the theoretical considerations.

User authentication is very crucial requirement for accessing sensitive information from IoT enabled environment. For secure banking and online shopping applications, now we trust HTTPS based on asymmetric key cryptography but not suitable for IoT.Asymmetric key algorithms are more secure than private key algorithms but additional cost and power will be required. So for IoT environment we choose symmetric key system.

The computation overhead of proposed scheme is very low because we use simple mathematical functions to prepare the hashes and require less memory and clock cycles when compared with existing MD5 and SHA algorithms.

Table 2	Requirements of cryptographic protocols			
Protocol	**Message size supported**	**No of Iterations**	**Amount of Calculation**	**Memory Requirements**
ZKP	Large	Many	Large	Large
Public Key	Large	One	Very Large	Large
Private Key	Large	One	Small	Small

`

Table 2	Requirements of cryptographic protocols	
Function	**Time Requirement (ms)**	**Memory Requirement (bytes)**
Key Agreement	1.07	64 bytes
Registration with Resource Owner	2.03	320 bytes
Token Generation and Encryption	25 KB encryption 3 ms	12 bytes
Retrieval of Token by Decryption	3 ms	---
Hash Preparation and Verification	119	16 bytes

Proposed authentication method use an algorithm based on Zero Knowledge Proof so an entity can authenticate without reveling the secrets to the resource servers and all computations carried out at user's browser. Zero Knowledge Protocols require small computations and are light weight hence less memory is required for its operations, suitable for memory and power constrained smart mobile devices. We measured the computation time required for a secure authentication and calculate the memory requirement. The performance matrix for the proposed scheme in terms of computational time and memory requirement is summarized in Table 3.

Low communication overhead is required by the proposed scheme because the length of the message exchanged between user and server is too short. The proposed method fulfils the properties of Zero Knowledge proof and provides solutions against various threats in network.

6.5 Future Enhancement

We extended an approach for Zero Knowledge Proof for authentication on mobile devices in IoT to reduce computation and communication overhead. In this work we use same OTP for verification and hash preparation, and plan to develop an algorithm for key exchange in IoT.

With the introduction of GPS, NFC and RFID, location of the devices are traceable but some times the user need to hide their location from services. We propose a context based filter to preserve privacy based on situation. The future work also concentrates on the design of a small server to act like a firewall in between server and requester and hence develop a complete attack resistant and resilient solution for mobile devices in IoT.

7. Industrial IoT and ZKP[IX]

7.1 Introduction

Industrial Internet of Things (IOT) is a distributed network of smart sensors that enables precise control and monitoring of complex processes over arbitrary distances. The great advantage of the industrial IoT is counterbalanced by a security weakness. The insertion of a smart device capable of extracting protected data or malicious actions can infect the whole network with relative ease. Thus it becomes imperative to discover whether or not new devices have the right capabilities and compatibilities with other sensors. This article presents a zero knowledge protocol that achieves precisely that objective while keeping the sensor data private.

We are awash with sensors and devices with more processing power than many of the standalone computers that reside on our desks. Smart phones, micro-computers, ambient light control systems, thermostats that can adjust to and report minute changes in our daily life, and the ubiquitous fitness gadgets constitute a whole technological species that is starting to coexist with us through the same Internet environment we populate with our communication devices. And this is the simple side of what has now become fashionable to call the "Internet of Things" (IoT). The real revolution is taking place in a different setting, an industrial one, where in each industry—from manufacturing to refineries and transportation—myriad smart sensors are connected through shared API's. This new form of networked computing power—the so called "Industrial Internet of Things"—will likely dwarf what we conceive of as the present day Internet. The industrial IoT has many characteristics that make it different from the consumer smart devices that most people are familiar with. First, the pervasiveness and interconnectivity of smart sensors, coupled with the unpredictability of their inputs, make response times autonomous of human intervention. Whereas a fitness tracker running out of power does not necessitate an urgent response, the failure or delayed emergency signal from a smart sensor controlling several valves in a refinery can trigger an undesirable reaction chain from other sensors and actuators leading to overall system failures. Second, the Industrial IoT has all the characteristics of an open, distributed system dealing as it does with a large, diverse quantity of information while exhibiting massive concurrency. It is also asynchronous since the behavior of the environment is not necessarily predictable by the system itself, which leads to the need for autonomous reactions. This points to a necessarily decentralized system, since it would be hard for a central unit to have up-to-date information on the state of the whole system. Third, the distributed nature of the Industrial IoT makes it open to a host of security threats, since a single break into a component of the distributed fabric can compromise the entire the system.

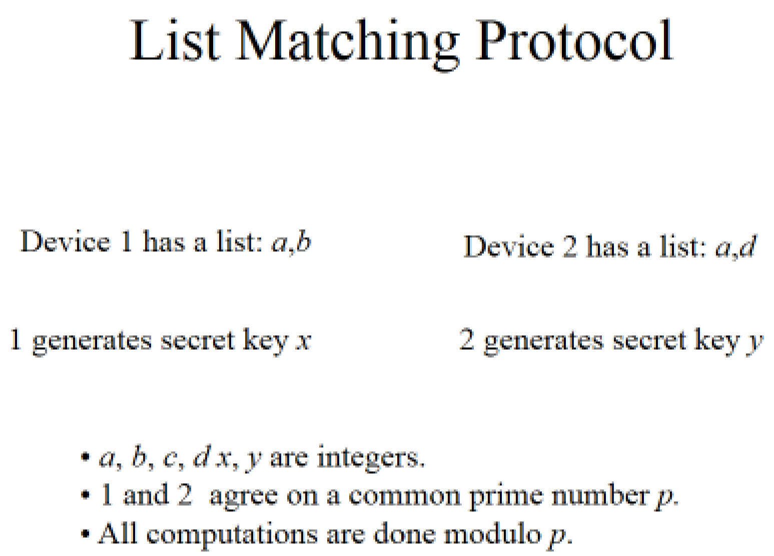
Since the behavior of such open systems has been analyzed in quite detail, in this contribution about the IoT wel focus on the last point—the security aspects of the Industrial IoT. Specifically, we describe a mechanism that we invented to deal with a pervasive problem with smart sensor networks: Discovering whether or not new devices have the right capabilities and compatibilities with other sensors, while keeping their data private. While there are some existing proposals to address the curity problems in IoT, most of them do not offer mechanisms for protecting the privacy of device apabilities or their characteristics. Recently, a group at IBM proposed the use of block chains—the query basis of Bitcoin—for all devices in the world of IoT. While in principle it could offer a robust solution to the security problem, it is unrealistic to imagine the implementation of a blockchain for all deices on a global scale in a way that allows the blockchain to scale as the size of the device network increases. Since the mechanism that we describe for solving security and trust issues in the Industrial IoT involves zero knowledge cryptographic techniques, we first provide a lightning survey of such techniques, to be followed by a simple exposition of the mechanism that solves these problems. The interested reader can consult the reference to our original patent and a paper for a more detailed explanation of what is involved in setting up these mechanisms.

7.2 A Secure Protocol

The mechanism exploits the use of two fundamental cryptographic primitives: hash functions and public key systems. In general, cryptographic functions operate on inputs such as ``messages" and ``keys,” and produce outputs such as ``cipher texts" and ``signatures.” It is common to treat all of these inputs and outputs as large integers according to some standardized encoding. Throughout this exposition I
assume any value involved in a cryptographic function is a large integer, no matter what it may be called.

A cryptographic hash function,H, is a mathematical transformation that takes a message m of any length, and computes from it a short fixed---length message, which we will call H(m). This fixed length output has the important property that there is no way to find what message produced it short of trying all possible messages by trial and error. Equally important, even though there may exist many messages that hash to the same value, it is computationally infeasible to find even two values that ``collide.” This practically guarantees the hash of a message can ``represent” the message in a way that makes it very difficult to cheat. An even stronger property we will require is the output of a cryptographic hash function cannot be easily influenced or predicted ahead of time. Thus someone who wanted to find a hash with a particular pattern (beginning with a particular prefix, say) could do no better than trial and error. In practice, hash functions such as MD-5 and SHA (secure hash algorithm) are often assumed to have these properties. Public key encryption relies on a pair of related keys, one secret and one public, associated with each individual participating in a communication. The secret key is needed to decrypt (or sign), while only the public key is needed to encrypt a message (or verify a signature). A public key is generated by those wishing to receive encrypted messages, and broadcasted so the message’s sender can use the key to encode the message. The recipient of this message then uses their own private key in combination with their public key to decrypt the message. While slower than secret key cryptography, public key systems are preferable when dealing with networks of devices that need to be reconfigured fairly often. Popular public key systems are based on the properties of modular arithmetic. Now onto the mechanism that removes the disincentive inherent in having to disclose the private content of the data and capabilities of a new sensor, which is inserted into an industrial network. Conversely, we don't want the device to learn any of the data and capabilities of the installed base of sensors in the enterprise. And yet we want it to be able to interact with those devices that share some similar capabilities and contain signatures that certify them as belonging to the network. For the sake of clarity I'll describe such a mechanism in pictorial fashion. Consider two devices, device 1 and device 2, each of which has a list of attributes. We want to find out if any of those attributes (it could be ID numbers, capabilities, memory, etc.) are common to both of them. We assume the lists contain two items each, device 1 with items a and b and device 2 with items a and d.

The procedure for discovering if any of the elements in the lists are common to both devices without revealing works as follows:



List Matching Protocol

Device 1 has a list: *a*,*b*

Device 2 has a list: *a*,*d*

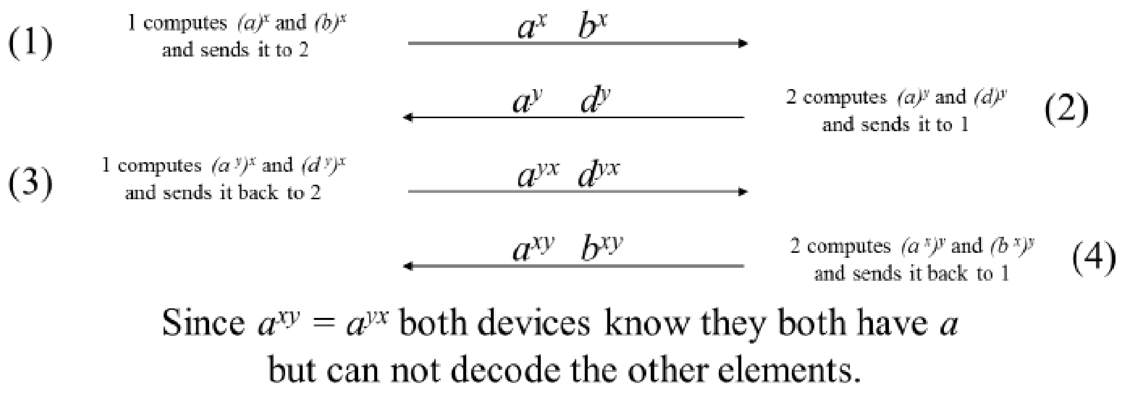
1 generates secret key *x*

2 generates secret key *y*

- *a*, *b*, *c*, *d* *x*, *y* are integers.
- 1 and 2 agree on a common prime number *p*.
- All computations are done modulo *p*.

Scheme B: The Authentication Procedure

The procedure works as follows:



(1) 1 computes $(a)^x$ and $(b)^x$ and sends it to 2 — $a^x \quad b^x$

(2) 2 computes $(a)^y$ and $(d)^y$ and sends it to 1 — $a^y \quad d^y$

(3) 1 computes $(a^y)^x$ and $(d^y)^x$ and sends it back to 2 — $a^{yx} \quad d^{yx}$

(4) 2 computes $(a^x)^y$ and $(b^x)^y$ and sends it back to 1 — $a^{xy} \quad b^{xy}$

Since $a^{xy} = a^{yx}$ both devices know they both have *a* but can not decode the other elements.

7.3 The Authentication Procedure

Notice the entire security of the operation would be compromised if device 1 or device 2 were able to compute either x or y from the data they initially sent to each other. But that is almost impossible because of the intractability of the discrete logarithm problem: Given integers a and b and prime p, it is computationally hard to find and integer n such that $b^n = a \pmod p$

This method, which I illustrated using only two inputs from both devices, generalizes to any large set of data and thus allows for either device to find whether or not they have a set in common without revealing what the data.

7.4 Conclusion

The ease of communication among devices and the ensuing exchanges of data mediated by the Internet have raised interesting problems concerning both the security of the data exchanged, and the need to keep it private in the context of the Industrial IoT. In particular given the ease with which smart sensors can be inserted into an industrial setting raises the issue of its certification and ability to interact with other sensors in the network in a trusted fashion without revealing the content of its data or capabilities. While the standard answer would resort to a trusted third party or the implementation of a blockchain, or to desist in having the data exposed to manipulations that could actually reveal its nature or the identity of the target. We have shown, that it is possible to use zero knowledge techniques to solve these problems in ways that ensure privacy without having to resort to trusted third parties. Moreover, this mechanism can be implemented and deployed on most smart sensors. The adoption of these techniques will accelerate the adoption of a distributed network of smart sensors and machines in the service of enterprise, thus leading to an Industrial IoT that will coexist with the one we are all familiar with as of today.

8. A “Mechanical” Analog of ZKP[X]

8.0 Introduction

This section describes a method that imitates the digital authentication Zero Knowledge Protocols. It is applied on the nuclear warhead identification, where the United Nation inspectors has to identify the class, the type and a few other weapon properties, but are not allowed to reveal the exact distracting power of the items examined.

Existing nuclear arms-control agreements between the United States and Russia place limits on the number of deployed strategic nuclear weapons. Verification of these agreements takes advantage of the fact that deployed weapons are associated with unique and easily accountable delivery platforms, that is, missile silos, submarines and strategic bombers, to which agreed numbers of warheads are attributed. The next round of nuclear arms-control agreements, however, may place limits on the total number of nuclear weapons and warheads in the arsenals. This would include tactical weapons as well as deployed and non-deployed weapons. Such agreements would require new verification approaches, including inspections of individual nuclear warheads in storage and warheads entering the dismantlement queue. This is a qualitatively new challenge because the design of nuclear weapons is highly classified information that cannot be exposed to international inspectors. A viable verification approach therefore has to resolve the tension between reliably verifying that the inspected warhead is authentic while avoiding disclosure of information about its design1–4. Practitioners and policy makers have been well aware of this conundrum, and prior work by national laboratories in the United States, Russia and the United Kingdom addressed it by using ‘information barriers’2,4. These barriers consist of sophisticated automated systems that process highly classified information measured during an inspection, but only display results in a yes/no manner. Such systems are inherently complex, and require both parties to trust that they have no ‘trapdoors’ hidden from the inspector, which could be used to cause a system to declare invalid objects as authentic, nor side channels unknown to the host, which could leak classified information to the inspector or others. These concerns are serious obstacles to adopting such systems. In this work we consider a fundamentally different approach to this problem. Rather than trying to acquire and analysing classified data behind an engineered information barrier, we use the cryptographic notion of zero-knowledge proofs to ensure that sensitive data are never measured in the first place.

8.1 Zero-knowledge proofs (explanation with marbles)

These proofs, invented in the 1980s by Goldwasser, Micali and Rackoff5, have become an important tool of modern cryptography. They achieve the paradoxical goal of allowing one to prove that a statement is true without revealing why it is true. Such proofs are extremely useful for many digital applications, including privacy-preserving data mining, electronic voting and online auctions6. To achieve zero knowledge, Goldwasser et al. extended the traditional notion of a proof from a static text to a protocol, which involves randomization and interaction between the prover and verifier. At the end of the protocol, the verifier has a high degree of confidence that the statement is correct, while the prover is guaranteed that the verifier did not learn anything about the data underlying the truth of the statement. For our application, the host submitting warheads for inspection takes the role of the prover and the inspector the role of the verifier. Whereas classical zero-knowledge proofs are digital protocols, proving statements about mathematical objects, we illustrate the concept using a physical zero-knowledge protocol that is closely related to our proposed verification approach (Fig. 1): Alice (the host) has two small cups both containing X marbles, where X is some number between 1 and 100. She wants to prove to Bob (the inspector) that both cups contain the same number of marbles, without revealing to him what this number X is. To do so, Alice prepares two buckets, which she claims each contain (1002X) marbles. Bob now randomly chooses into which bucket which cup is poured. Once this is done, Bob verifies that both buckets contain 100 marbles. The protocol reveals no information on X because, regardless of the value of X, Bob always sees 100 marbles in both buckets. However, if the cups did not have the same number of marbles, then no matter how Alice prepares the buckets, with a probability of 50% after the pouring at least one of the buckets will not contain 100marbles. If Alice and Bob repeat this game, say, five times, then if Alice consistently cheats she will be caught with probability (1 – 225).95%.

8.2 From marbles to neutrons

The relevance of the above protocol to our setting is that we want to show that two or more putative warheads have identical neutron transmission and emission counts under irradiationby high-energy neutrons. We follow the template approach for warhead verification2, inwhich a radiation measurement generates a complex and unique fingerprint of an inspected item. This fingerprint is then compared against the fingerprint of one or more templates to confirm that all items are materially identical. Template selection is a critical and challenging step. In practice, templates could be directly selected from deployed weapons so that the inspecting party has high confidence in their authenticity. Strong chain of-custody measures would have to be in place to assure the inspectors that the templates have not been swapped out between visits. In the case of weapon systems that are not currently deployed, and in other cases where a trusted reference item may not be available, differential measurements could still be valuable. In these cases,measurements of a large number of warheads in a batch could confirm that they are allmaterially identical.Combined with some other supporting evidence (for example, records confirming the ‘pathway’ or ‘provenance’ of at least some of these items), this could provide confidence in the authenticity of all items in the inspected batch7. We compare the submitted items by recording the transmission pattern of 14-MeV neutrons, as well as recording the intensity of neutrons emitted at large angles from the items. Active interrogation of nuclear warheads or arhead components with high-energy neutrons and other types of radiation has been successfully demonstrated, but does require detailed safety analyses. Note that, even when exposed to strong neutron sources, the induced fission events in a nuclear warhead produce less than a milliwatt of heat,which is far less than the heat already generated by a-decay and spontaneous fission in items containing, for example, kilogram quantities of plutonium. Neutron radiographic images of warheads contain highly classified information, but in our case they are actually never measured. Rather, by analogy to the marbles example, they are recorded using detectors that are preloaded with the negative of the radiograph. Preloaded values are not revealed to the inspector. As in the marbles example, after the measurement and if the host is telling the truth, the inspector always sees the same number of counts in every detector. Furthermore, as in the marbles example, preloads supplied with the submitted items are shuffled at random, so if the items actually differ, then no matter how the preloads are chosen, with significant probability the image will not be uniform, and a mismatch will be present on both items. Unlike the marbles example, neutron measurements are inherently statistically noisy. To avoid conveying information through the noise distribution we use preloaded values that are noisy as well. In particular,

since the signal added during interrogation will have a Poisson distribution, we also use a Poisson distribution for the preloads. Using the fact that the sum of two Poisson distributions is also Poisson in character, our protocol achieves the following: the neutron count obtained by any measurement on the template or on any valid submitted item is distributed according to the Poisson distribution with mean equal to a previously agreed-on value, Nmax, and a standard deviation equal to (Nmax) 0.5. Since Nmax is known in advance by both sides, neither the measurement nor its noise reveals any new information. Nmax for transmission could reasonably correspond to the maximum number of

counts that is expected in the absence of a test item. If a submitted item varies from the true warhead (or the submitted preloads are not identical) an image may be seen that could contain sensitive information. This will be an additional strong incentive for the host not to cheat. For simplicity of operation, we envision that the host places the

detectors for each measurement in a removable board that forms part of the measurement system. Crucially, the inspector chooses which board to use with which test item. As in the marbles example, this means that if the host uses non-identical boards to try to mask invalid items, then with 50% probability the invalidity will be made more evident by the measurement with the mismatched boards. Since we expect that this ‘game’ will be repeated many times, even a risk-tolerant host would not accept the resulting low chance of success. We note that testing multiple warheads in parallel is an attractive option, because it makes the probability of detecting the use of non-identical preloads significantly higher.

Once the measurements have been completed and the detectors read out, the inspecting party can verify the functionality of the detectors by exposing them to additional neutrons. This is an important advantage of the proposed method. For inspection systems proposed so far, specialists from the inspecting country would not be allowed to examine any equipment once it has seen classified information. Although we examine here neutron measurements using preloaded non-electronic detectors, there may be other non-electronic zero-knowledge protocols for warhead verification that can avoid the use of engineered information barriers. Indeed, such systems could be complementary to the neutron measurements discussed here. Monte Carlo analysis We now show how our approach can be implemented in practice, and that small differences between two objects can be reliably detected. We have analysed the approach with a series of simulations using the general Monte Carlo N-Particle (MCNP) transport code8. Construction of a physical experimental set-up is under way.

We propose to use 14-MeVneutrons from a deuterium-tritium neutron generator9 to interrogate test items, allowing detailed transmission profile measurements and also measurements of neutron intensities at large angles due to elastic and inelastic scattering, fission and (n,2n) reactions. The neutrons from the generator are collimated by 60cm of polyethylene and illuminate the inspected item (Fig. 2). An array of neutron detectors placed at a distance of 50 cm behind the centre of the item provides the transmission measurements. Additional detectors (not shown) can be positioned with additional shielding at large angles to the beam, that is, in the shadow of the collimator, to measure neutrons emitted from the test item.

Test item The test item used for this analysis is the unclassified 'British Test Object' (BTO), which consists of concentric rings of polystyrene, tungsten (two rings with a combined mass of 7.74 kg), aluminium, graphite, and steel. The BTO has an outer diameter of 18.9cm and a height of 5 cm. This test object does not contain special or other nuclear materials, but is used to develop and calibrate imaging systems for diagnostic analysis of nuclear weapons10. (n,2n) reactions in the tungsten used in this test object provide a reasonable approximation of induced fission events expected for a nuclear warhead or warhead component. Neutron multiplication in a real item would increase net neutron production rate by some finite amount, but the effect is extremely small for transmission measurements. Furthermore, the energy threshold (,10MeV) used for the transmission detectors renders them insensitive to fission and (n,2n) neutrons. The BTO is placed in a container in order to avoid revealing to the inspector the appearance or orientation of the inspected item inside the container. Detector array To assess the viability of our proposed protocol, we work with a board holding a hexagonal array of 367 detectors consisting of 21 rows of 17 or 18 detectors within an area of about 42 cm342 cm. The assumed area of each detector (pixel) is 2 cm2. By rotating the BTO, the board can image it in any orientation. In the analysis below, detectors are assumed to be sensitive to neutron energies.10 MeV. Neutrons scattered from the walls of the room are not included in the calculations, but preliminary studies indicate that room return has at most a small impact when using 10-MeVthreshold detectors, particularly if the room is specially prepared for
the inspection—for example, with borated polyethylene in front of borated concrete walls.

8.3 Diversion scenarios

To examine diversion scenarios, in which material is removed or replaced, we need to define a decision rule to distinguish passed from failed tests. For our present purposes, we use a very simple rule looking for statistical outliers on predefined groups of pixels. If we denote individual detector counts by the numbers X_1,…,X_n, then we can define new numbers Y1,…,Yk, where every Yj is the sum of a small number of the X_{is}, divided by the expected standard deviation of Y for a match case (that is, inspected item identical to template). We define the test to be positive (that is, diversion detected) if there is at least one j with jYjj.T, where T is a threshold chosen such that in the match case for every
j the probability that jYjj.T is at most pfp/k where pfp is our allowed false positive rate. Concretely, in our setting, we examine k5295 nondisjoint seven-pixel windows defined

by a central detector and its six nearest neighbours. In this case, to achieve a false positive rate pfp#0.05, the threshold can be computed numerically to be T53.76 standard deviations. Sensitivity of the measurements to diversion scenarios increases with Nmax and the associated improvements of counting statistics. We therefore examine in the following a series of different diversion scenarios and a range of values for Nmax to determine system requirements (Table 1). In the full-removal scenario, both tungsten rings are removed from the BTO,which is easily detected even for very lowdetector counts. Similarly, if lead is used to substitute both tungsten rings, the diversion is clearly distinguishable even by simple visual inspection of the detector board (Fig. 4, top). Our proposed statistical test identifies the diversion in the full-substitution scenario with a probability of true positives, ptp, of .0.99, even for Nmax as low as 1,000 detector counts. The local-removal and local-substitution scenarios are more challenging. In these cases, a 36u sector of the outer tungsten ring is removed or replaced, which corresponds to a diversion of 543 g of tungsten contained in the BTO. To achieve a detection probability of 95%, an Nmax of 5,000 is required in the case of the localized tungsten removal. Whenlead is used to substitute for tungsten in the 36u sector, Nmax increases to 32,000 for the same detection probability. Note that in these studies no use has been made of the emission detectors at large angles. The more realistic case of substitution of 238U for 235U in a nuclear weapon component results in a reduction by a factor of about two in the induced fission rate due to 14-MeV neutrons. Substitution of reactor-grade for weapon-grade plutonium has a small effect on the directly induced fission rate, but a large effect on the spontaneous fission rate, which could be detected passively by operating the side detectors in the absence of the neutron source. Thus the calculations presented here are conservative. We note that 5% of the items will be flagged as invalid by our proposed test procedure due to the set 5%false positive rate, even in the case where all items are valid. Retesting flagged items will rapidly determine their validity. If a detection probability for invalid items of 95% is deemed too low, either routine retesting or a greater Nmax can be implemented to increase this value. The optimization of any retesting scenario, and study of a wider range of host strategies for cheating, as well as inspector strategies for analysing signal patterns to find such cheating, will be the subject of future research. Preloadable non-electronic detectors Perhaps the most critical aspect of a viable implementation of the proposed verification approach is the choice of the detector technology. The detectors must have the capability to be preloaded with a desired neutron count before the inspection. At a minimum, this preload has to persist for hours or days and its decay rate, if present, be well characterized. Preloaded counts must be indistinguishable from counts accumulated during irradiation of the test items. Detectors should be energy selective so that the effect of low energy neutrons returning from room walls can be minimized. They should be insensitive to c-rays, have high efficiency, and permit total counts in the range discussed above. Finally, relying ona non-electronic detection mechanism is highly advantageous given that complex electronic components and circuits are potentially vulnerable to tampering and snooping. We find that at least two detector technologies can meet these criteria: superheated emulsions ('bubble detectors') and neutron activation analysis detectors. In superheated emulsions, neutron recoil particles trigger the formation of macroscopically observable bubbles from microscopic droplets that are dispersed in an inertmatrix11. These detectors can be configured to have essentially any desired energy threshold from10 keVto 10MeV.

Commercially available, polymer-based bubble detectors are limited to a maximum bubble count of the order of a few hundred bubbles, beyond which camera-based imaging techniques cannot resolve bubbles individually. Superheated drop detectors produced with an aqueous gel can be used up to much higher bubble counts. Either optical tomography or magnetic resonance imaging allow the counting of bubbles hidden in the depth of the fluid12,13. If the highest Nmax is desired, multiple detectors can be exposed in series. By the proper choice of a compliant matrix, detectable ageing (growth) of bubbles can be eliminated. Net detection efficiency of the order of 1% can be easily achieved. The emulsions can be contained in opaque containers so that a preload is not visible to the inspector. For neutron activation analysis, imaging can be undertaken using, for example, an array of hexagonal prisms made of zirconium14. 90Zr has a neutron activation threshold of 12 MeV through an (n,2n) reaction. The resulting 89Zr has a half-life of 3.3 days, which must be taken into account to determine the required level of preloading. Counting the c-rays from 89Zr decay in high-purity germanium well detectors should give a net detection efficiency of about 0.25%. For 3-cm-long prisms, with a cross-sectional area of 2 cm2, this would provide an Nmax in the range of 20,000 within one hour, for a commercially available DT neutron generator producing 33108 neutrons s21 (ref. 15). Indium has an appropriate activation response to fission neutrons, with reduced sensitivity in the range of 14MeV, for use in the side detectors. It will be important to ensure that unshielded, preloaded activation samples are not in the presence of c-ray detectors before their final exposure in order to avoid providing the inspector with any information about the preloads. Detectors can be preloaded with counts with the appropriate statistical properties by exposing them to energetic neutrons for a pre-calculated period of time and/or through a pre-calculated depth of shielding. As discussed above, statistical noise in the measurement will not reveal any information However any systematic measurement errors must be well understood, such that while one detector may be characterized by a different efficiency from another, which can be calibrated out, this efficiency must not vary significantly between the preload and the measurement processes. For example, it is important to maintain control over the temperature of bubble detectors during irradiation. The DTneutron generator must also be well controlled and measurable, so that there is no significant variation in the shape of the neutron field produced nor in the total number of neutrons emitted when irradiating items. An accurate neutron flux monitor can be used to set the irradiation time, so perfect reproducibility is not required in the rate of neutron production. We anticipate that these requirements can be met, but the techniques to achieve the necessary degree of control need to be demonstrated and validated. The steps following a measurement should be relatively straightforward. Since the information contained in the detectors is in principle unclassified, protocols can be devised that permit using both host-provided and inspector-provided measurement tools. We anticipate that, depending on the strength of the neutron source, the measurements themselves could be completed within hours. Readout would be very quick in the case of bubble detectors, but could take days in the case of some activation detectors. This is not a major constraint, since readout could take place in parallel with other measurements and activities at the site. The authentication process could be accelerated dramatically if N warheads are processed simultaneously (including, for example, M,N reference items). Typically, authentication of one warhead per day can be considered a reasonable target, especially since dismantlement of the warhead itself (including recovery of fissile material and removal of classified features) would take much longer. Authentication is therefore not a significant bottleneck in the process.

Developing a practical inspection system for nuclear warhead verification will be a major undertaking. Ideally, such a system should be jointly developed by partners from weapon and perhaps also nonweapon states. The successful UK–Norway Initiative has shown that such collaborations are possible16. Similar efforts could be undertaken for the system proposed here. They would help refine and demonstrate to the satisfaction of all parties the robustness of the method in practical situations, where systematic errors in measurements, small misalignments, or variations in environmental conditions may pose additional challenges that are difficult to anticipate with computer simulations.

More generally, we believe that our approach and techniques could have other applications beyond the area of nuclear disarmament. Once data are measured and converted to digital form, secure comparisons and computations can be performed using many cryptographic tools17,18. However, if the measurement device itself cannot be trusted, it is best to ensure the data are never measured in the first place. This could be the case not just for state secrets, but also for personal data, such as biometric data or results of medical tests. For example, following earlier versions of the present work, it was proposed19 that similar ideas could be used to compare DNA found in a crime scene with a suspect’s DNA without actually measuring the latter and creating a DNA profile. Exploring other such applications is an exciting future direction of research on zero-knowledge proofs.

8.4 Conclusion

Authenticating nuclear warheads without revealing classified information represents a qualitatively new challenge for international arms control inspection. Here we have shown an example of a zero-knowledge protocol based on non-electronic differential measurements of transmitted and emitted neutrons that can detect small diversions of heavy metal from a representative test object. This technique will reveal no information about the composition or design of nuclear weapons when only true warheads are submitted for authentication, and so does not require an engineered information barrier. The zero-knowledge approach has the potential to remove a major technical obstacle for verifying deep cuts in the nuclear arsenals, which will probably require verification of individual warheads, rather than whole delivery systems. Timely demonstration of the viability of such an approach could be critical for future rounds of arms-control negotiations.

Closing Paragraph

It is generally known that most transactions on the web are signed by a third party authority. A new idea on the secure schemes, is to give the user the opportunity to choose the method of authentication. Some online secure transactions usually change the user’s private key after every successful login; this change though is silent, the user has no clue of this. A future study could research how the elliptic curve method and the graph isomorphism may combine to produce a more secure Zero Knowledge Protocol. We would suggest, in that future research, that the graphs could have vertices on some of the ellipsis’s points.

References and Sources

I.A Quick Introduction to Anonymous Credentials
Gregory Neven IBM Zürich Research Laboratory

II.Password Hardening Based on Keystroke Dynamics
Fabian Monrose, Michael K. Reiter, Susanne Wetzel
Bell Labs, Lucent Technologies Murray Hill, NJ, USA

III.A Primer on Zero Knowledge Protocols
Gerardo I. Simari
Department of Science and Computer Engineering, National University of the South, Buenos Aires

IV.The Feasibility and Application of using a Zero-knowledge Protocol Authentication Systems
Becky Cutler, Tufts University

V.A "Paradoxical" Identity-Based Signature Scheme Resulting from Zero-Knowledge
Louis Claude Guillou, Jean-Jacques Quisquarter

VI.Elliptic Curve Based Zero Knowledge Proofs and Their Applicability on Resource Constrained Devices
Ioannis Chatzigiannakis, Apostolos Pyrgelis, Paul G. Spirakis, Yannis C. Stamatiou
Research Academic Computer Technology Institute and Computer Engineering and Informatics Department, University of Patras

VII.A Practical Zero-Knowledge Proof Protocol for Web Applications
Slawomir Grzonkowski, National University of Ireland
Peter Corcoran, National University of Ireland

VIII.A Secure Authentication Infrastructure for IoT Enabled Smart Mobile Devices – An Initial Prototype
K. A. Rafidha Rehiman and S. Veni

IX.Ensuring Trust and Security in the Industrial IoT
Bernardo A. Huberman (Hewlett Packard Labs HPE-2016-21)

X. A zero-knowledge protocol for nuclear warhead verification
Alexander Glaser, Boaz Barak, Robert J. Goldston