

Applications

Pascal Lafourcade

Université Joseph Fourier, Verimag

24th November 2008

Outline

- 1 Secret Sharing
- 2 Interactive Zero Knowledge Proofs
 - Principle
 - Funny example: Rubik's Cube
 - Example: Graph Coloring
 - Fiat-Shamir
- 3 Blind Signature
- 4 E-Voting
- 5 E-Cash
- 6 Side Channel
- 7 Conclusion

Outline

- 1 Secret Sharing
- 2 Interactive Zero Knowledge Proofs
 - Principle
 - Funny example: Rubik's Cube
 - Example: Graph Coloring
 - Fiat-Shamir
- 3 Blind Signature
- 4 E-Voting
- 5 E-Cash
- 6 Side Channel
- 7 Conclusion

Shamir 1979

Initial Problem

Eleven scientists are working on a secret project. They wish to lock up the documents in a cabinet so that the cabinet can be opened if and only if six or more of the scientists are present.

- What is the smallest number of locks needed?
- What is the smallest number of keys to the locks each scientist must carry?

Shamir 1979

The minimal solution uses 462 locks and 252 keys per scientist.

Integrity

The lock can be opened with $m = 6$ parts over $n = 11$.

Confidentiality

No way to open the lock with less than $m = 6$ parts over $n = 11$.

Shamir 1979

What is the smallest number of locks needed?

Shamir 1979

What is the smallest number of locks needed?

Idea : For each group of five scientists, there must be at least one lock for which they do not have the key, but for which every other scientist does have the key.

There are $\binom{11}{5} = 462$ groups of five scientists, there must be at least 462 locks.

Shamir 1979

What is the smallest number of locks needed?

Idea : For each group of five scientists, there must be at least one lock for which they do not have the key, but for which every other scientist does have the key.

There are $\binom{11}{5} = 462$ groups of five scientists, there must be at least 462 locks.

What is the smallest number of keys to the locks each scientist must carry?

Shamir 1979

What is the smallest number of locks needed?

Idea : For each group of five scientists, there must be at least one lock for which they do not have the key, but for which every other scientist does have the key.

There are $\binom{11}{5} = 462$ groups of five scientists, there must be at least 462 locks.

What is the smallest number of keys to the locks each scientist must carry?

Similarly, each scientist must hold at least one key for every group of five scientists of which they are not a member, and there are $\binom{10}{5} = 252$ such group

Shamir 1979

What is the smallest number of locks needed?

Idea : For each group of five scientists, there must be at least one lock for which they do not have the key, but for which every other scientist does have the key.

There are $\binom{11}{5} = 462$ groups of five scientists, there must be at least 462 locks.

What is the smallest number of keys to the locks each scientist must carry?

Similarly, each scientist must hold at least one key for every group of five scientists of which they are not a member, and there are $\binom{10}{5} = 252$ such group

If we generalize we get $\binom{n}{m-1}$ and $\binom{n-1}{m-1}$.

Secret Sharing

- How keep nuclear code secret in British Army?

Secret Sharing

- How keep nuclear code secret in British Army?
- Burn it, but do not preseve integrity

How to Share a Secret Code I



1234567



How to Share a Secret Code I

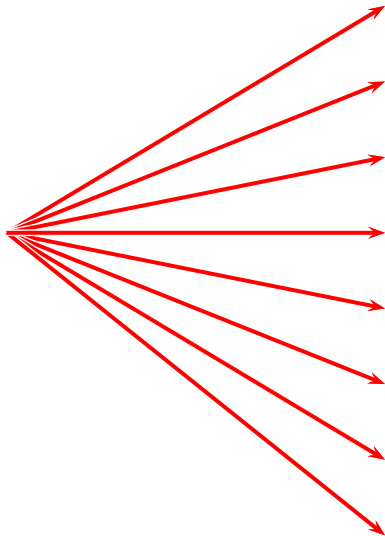


1234567



Problem of Integrity and Confidentiality

How to Share a Secret Code II



1234567



1234567



1234567

1234567



1234567



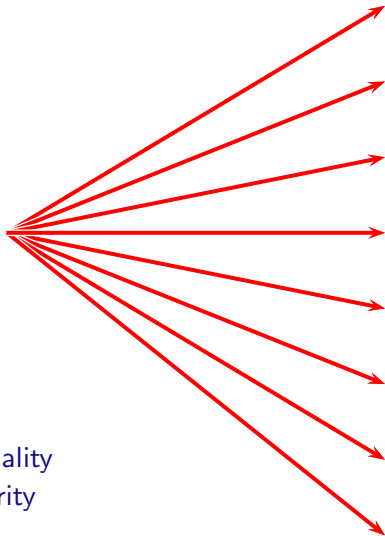
1234567

1234567



1234⁸567⁷⁸

How to Share a Secret Code II



1234567



1234567



1234567

1234567



1234567



1234567

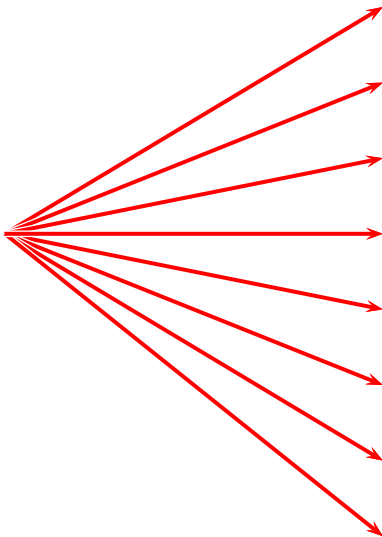
1234567



1234⁸567⁸

Problem of Confidentiality
No problem of Integrity

How to Share a Secret Code II



23572



11567



734567

534567



934567



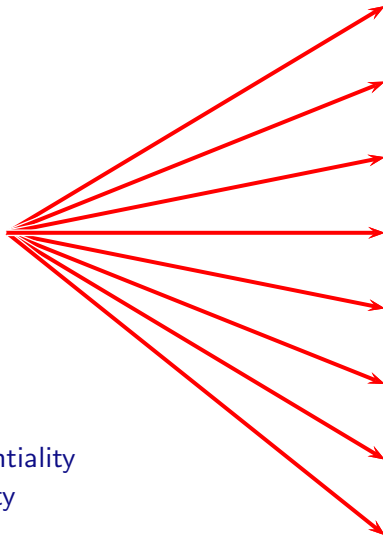
563317

114567



455967⁸

How to Share a Secret Code II



23572



11567



734567

534567



934567



563317

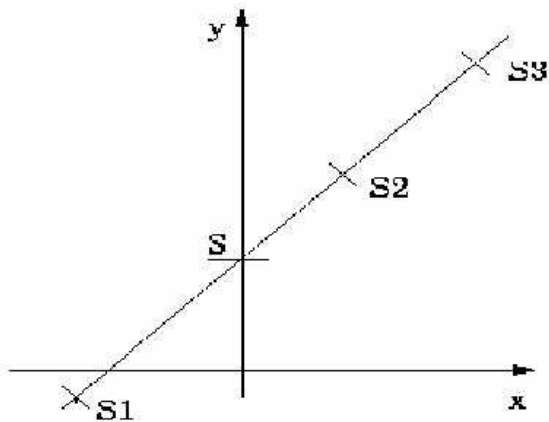
114567



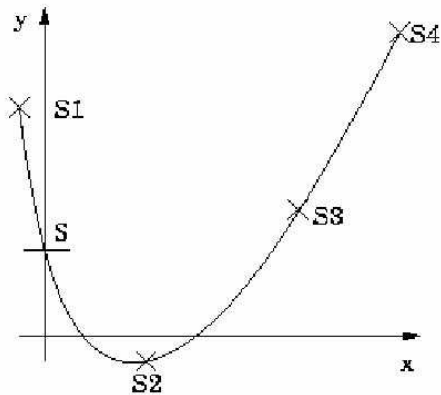
455~~9~~67⁸

No Problem of Confidentiality
Problem of Integrity

(2,5)



(3,5)



(m,n)

It takes $n + 1$, points to define a polynomial of degree n .
Using Lagrange interpolation to recover the secret

Lagrange Interpolation

The Lagrange interpolating polynomial is the polynomial $L(x)$ of degree $\leq (n - 1)$ that passes through the n points $(x_1, y_1 = f(x_1)), (x_2, y_2 = f(x_2)), \dots, (x_n, y_n = f(x_n))$, and is

$$L(x) := \sum_{j=0}^k y_j \ell_j(x)$$

$$\ell_j(x) := \prod_{i=0, i \neq j}^k \frac{x - x_i}{x_j - x_i} = \frac{(x - x_0)}{(x_j - x_0)} \dots \frac{(x - x_{j-1})}{(x_j - x_{j-1})} \frac{(x - x_{j+1})}{(x_j - x_{j+1})} \dots \frac{(x - x_k)}{(x_j - x_k)}.$$

Shamir 1979

Initialization

Dealer chooses his secret $k \in Z_p$

Distribution

Dealer generates

$$P(X) = k + \sum_{i=1}^{m-1} a_i X^i$$

Send $s_i = (i, P(i))$ to each participant i

Reconstruction

Using Lagrange interpolation, with m distinct parts we compute $P(x)$

Weakness of Shamir

- In Shamir Secret Sharing there is no mechanism to identify if a share is valid or not.
- A total confidence is done to the dealer.

Weakness of Shamir

- In Shamir Secret Sharing there is no mechanism to identify if a share is valid or not.
- A total confidence is done to the dealer.

Verifiable Secret Sharing introduced by Feldman 1987, based on Discret Logarithm.

Verifiable Secret Sharing

Distribution like Shamir

Dealer generates

$$P(X) = k + \sum_{i=1}^{m-1} a_i X^i$$

Send $s_i = (i, P(i))$ to each participant i

Plus

Each server received : $g^k, g^{a_1}, \dots, g^{a_{m-1}}$

Verification

$$g^k \prod_{j=1}^{m-1} ((g^{a_j})^i)^j = g^{s_i}$$

Outline

- 1 Secret Sharing
- 2 Interactive Zero Knowledge Proofs
 - Principle
 - Funny example: Rubik's Cube
 - Example: Graph Coloring
 - Fiat-Shamir
- 3 Blind Signature
- 4 E-Voting
- 5 E-Cash
- 6 Side Channel
- 7 Conclusion

Interactive Zero Knowledge Proofs

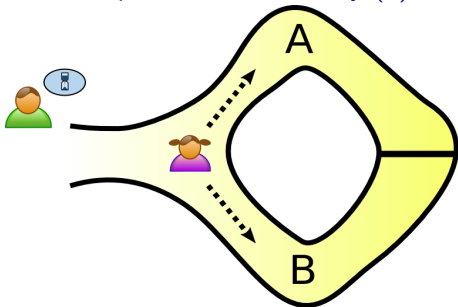
In an interactive zero knowledge proof, a prover P interacts with a verifier V to demonstrate the validity of an assertion without revealing anything about the assertion to the verifier.

An Example: The Cave Story (1)

- a cave shaped a circle
- a magic door at one side
- an entrance at the other side
- Victor will pay Peggy only if she knows the secret
- Peggy won't tell the secret until she would have been paid.

Interactive Zero Knowledge Proofs

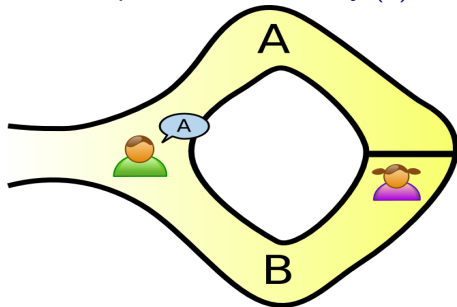
An Example: The Cave Story (2)



First, Victor waits outside while Peggy chooses a path.

Interactive Zero Knowledge Proofs

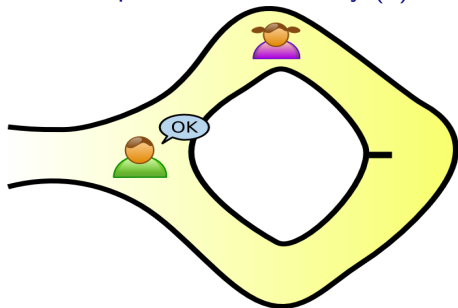
An Example: The Cave Story (3)



Then Victor enters and shouts the name of a path.

Interactive Zero Knowledge Proofs

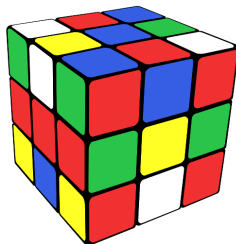
An Example: The Cave Story (4)



At last, Peggy returns along the desired path (using the secret if necessary).

Rubik's Cube

Assume Alice knows how to solve the Rubik's Cube.

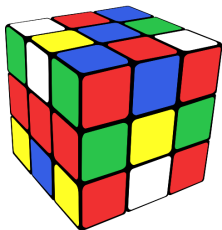


Question ?

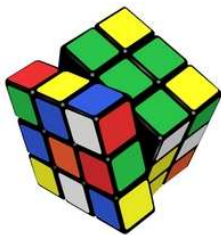
She wants to prove to Bob she has the skill to solve a given scrambled Rubik's cube without revealing it. How can she do it?

Rubik's Cube

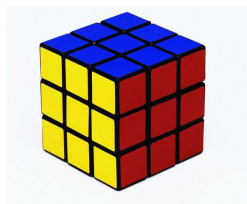
Alice scrambles the cube and proposed a new cube to Bob



1



2



3

Bob asks Alice from the current position (2) to go back to the initial position (1) or to solve it (3).

Repeating this process k times Bob is convinced that Alice knows the secret with a probability $1/2^k$.

Graph 3-coloring

Definition

3-coloring A 3-coloring of a graph is an assignment of 3 colors to vertices such that no pair of adjacent vertices are assigned to the same color.

3-coloring Problem

Given a graph G , the problem of deciding if the graph G is 3-colorable is an NP problem. (cf Garey and Johnson Book)

Problem: Alice wants to prove to Bob she knows a 3-coloring c of a given graph G .

Graph 3-coloring

- 1 Alice chooses a permutation π of the 3 colors ($\pi \circ c$ is still a 3-coloring of the graph G). And she transmits to Bob $e_u = H(\pi(c(u)) || r_u)$ for $u \in V$ and r_u random value.
- 2 Bob asks colors for u and v in V .
- 3 Alice answers $r_u, r_v, \pi(c(u)), \pi(c(u)), \pi(c(v))$ which allows Bob to confirm messages send by Alice.

Given a permutation π

$$A \rightarrow B : \forall u \in V, e_u = H(\pi(c(u)) || r_u)$$

$$B \rightarrow A : u, v$$

$$A \rightarrow B : r_u, r_v, \pi(c(u)), \pi(c(u)), \pi(c(v))$$

Graph 3-coloring

Playing several time this procedure Bob is convinced that Alice has a 3-coloring of G .

Completeness

If Alice knows the coloring then Bob will accept her proof.

Soundness

If Alice does not know the coloring then Bob will detect it with probability $\frac{1}{\#edges}$

Zero-knowledge

Bob just sees two random colors. Hence he learns nothing about the 3-coloring of G .

RK : ZKP are sensible to Man in the middle attack.

Fiat-Shamir

- A trusted center chooses $n = pq$, and publishes n but keeps p and q secret.
- Each prover A chooses a secret s with $\gcd(s, n) = 1$, and publishes $v = s^2 \bmod n$.

Fiat-Shamir

Procedure

A proves knowledge of s to B by repeating:

- 1 A chooses random r and sends $r^2 \bmod n$ to B.
- 2 B chooses random $e \in \{0, 1\}$, and sends it to A.
- 3 A responds with $a = rs^e \bmod n$
- 4 B checks if $a^2 = v^e r^2 \bmod n$.

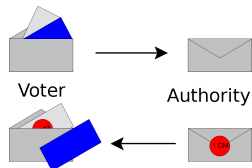
Fiat-Shamir

- if A follows the protocol and knows s , then B's check will always work
- if A does not know s , then they can only answer the question with probability $1/2$.

Outline

- 1 Secret Sharing
- 2 Interactive Zero Knowledge Proofs
 - Principle
 - Funny example: Rubik's Cube
 - Example: Graph Coloring
 - Fiat-Shamir
- 3 Blind Signature**
- 4 E-Voting
- 5 E-Cash
- 6 Side Channel
- 7 Conclusion

Blind Signature



Use specific encryption scheme (e.g.: RSA).

3 steps:

- the voter encrypt the message
- the authority sign the encrypted message
- the voter decrypt the message.

RSA

- Key Generation
 - $p, q \xleftarrow{\$} \mathcal{P}$
 - $n = pq, \phi = (p - 1)(q - 1)$
 - $e \mid 1 < e < \phi, \gcd(e, \phi)$
 - $n, e \rightarrow$ public key
- $m \rightarrow c = m^e \pmod n$
- $c^d = m \pmod n \rightarrow m$

Blind Signature with RSA

Bob can obtain a blind signature of m from Alice by this way:

- Bob hide is message: $m' \stackrel{\text{def}}{=} mr^e$, with $r \stackrel{\$}{\leftarrow} \mathbb{Z}_n$, and sends m' to Alice.
- Alice signs m' : she sends $s' \stackrel{\text{def}}{=} m'^d \pmod n$ to Bob.
- Bob retrieves the signature s of m by computing:

$$s = \frac{s'}{r} = \frac{m'^d}{r} = \frac{m^d r^{ed}}{r} = \frac{m^d r}{r} = m^d \pmod n$$

Scheme using Blind Signature

Usually used by voters to get a token. e.g.:

- Voter send his encrypted pseudonym
- Authority signs it
- Voter send his vote with his token (anonymously).

Used along with anonymous channels.

Outline

- 1 Secret Sharing
- 2 Interactive Zero Knowledge Proofs
 - Principle
 - Funny example: Rubik's Cube
 - Example: Graph Coloring
 - Fiat-Shamir
- 3 Blind Signature
- 4 E-Voting**
- 5 E-Cash
- 6 Side Channel
- 7 Conclusion

Security Properties

- Privacy
- Receipt-Freeness
- n-Robustness
- Correctness
- Universal Verifiability
- Individual Verifiability
- Democracy:
 - Eligibility
 - Prevent Multiple Voting
- Fairness

Privacy

No participant other than a voter should be able to determine the value of the vote cast by that voter.

Receipt-Freeness

Voters must neither be able to obtain nor construct a receipt which can prove the content of their vote.

This property prevents coercion and vote buying.

n-Robustness

Faulty behavior of a n -coalition of authorities can be tolerated. No coalition of voters can disrupt the election and any cheating voter will be detected.

Verifiability

Correct voting processes must be verifiable to prevent incorrect voting results.

There are two kinds of verifiability :

- Universal (Public) Verifiability
Any participant or passive observer can convince himself of the validity of individual votes and of the final tally of election.
- Individual Verifiability
Every eligible voter can verify that his vote was counted.

Democracy

There are two requirements to satisfy in democracy :

- Eligibility : Only authorized voters are allowed to vote.
- Prevention of Multiple Voting : All eligible voters are allowed to cast the scheduled vote's number (function of the election system and his part in it) and not more, such that each voter has his intended power in deciding the outcome of the voting.

Fairness

No participant can gain any knowledge, except his vote, about the (partial) tally before the counting stage.

- To enforce privacy.
- To not influence the intentions of the voters who has not yet voted.

Example

A secure and Optimally Efficient Multi-Authority Election Scheme
by R. Cramer, R. Gennaro and B. Schoenmakers.

Properties achieved

- Eligibility
- Universal Verifiability
- Privacy
- Robustness
- No vote duplication
- No interaction between voters
- Receipt-freeness

Using homomorphic properties of encryption.

Advantage

Time and communication complexity is minimal for authorities and individual.

Complexity for a voter is independent of the number of authorities.
Voter only posts a ballot and a prove a his validity.

Main Idea of the Scheme

Consider l voters: V_1, \dots, V_l and n authorities: A_1, \dots, A_n , using a *bulletin board*.

- 1 Voters post ballots to the bulletin board, with a proof of validity.
- 2 Tallers compute sum of all votes using homomorphic property of encryption.

$$\Pi\{m_i\}_k = \{\Sigma m_i\}_k$$

this ensures universal verifiability, with encryption based on ElGamal Scheme.

Bulletin Board

It is viewed as a public broadcast channel with memory. Nobody can erase any information from the bulletin board and all informations are public.

Recall Elgamal

- $G = (\langle g \rangle, *)$ finite cyclic group of prime order q .
- x : **private** key.
- $y = g^x$: **public** key.

$$E(m; r) = (g^r, y^r m) \rightarrow (c, d) \text{ and } D(c, d) = \frac{d}{c^x}$$

Robust threshold ElGamal cryptosystem

Idea is to be resistant against a coalition, i.e. messages can only be decrypted when a substantial set of receivers cooperate.

(Pedersen 91)

Key Generation

Each authorities A_j have a share key s_j , commit publicly $h_j = g^{s_j}$ and Λ is a set of t shares.

$$s = \sum_{j \in \Lambda} s_j \lambda_{j, \Lambda} \quad \text{where} \quad \lambda_{j, \Lambda} = \prod_{l \in \Lambda \setminus \{j\}} \frac{l}{l - j}$$

It is the Shamir's (t, n) -threshold [Sha79] with public key $h = g^s$

Robust threshold ElGamal cryptosystem

Decryption of $(x, y) = (g^\alpha, h^\alpha m)$ without reconstructing s

- 1 Each Authorities A_j broadcast $w_j = x^{s_j}$ and proves in zero-knowledge that

$$\log_g h_j = \log_x w_j$$

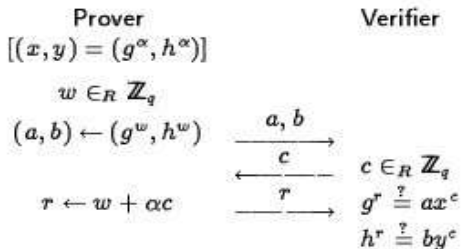
- 2

$$m = y / \prod_{j \in \Lambda} w_j^{\lambda_{j,\Lambda}}$$

Assure that $n - t$ authorities can be malicious.

Proof of knowledge

Using Chaum-Pederson protocol.



Homomorphic encryption

Probabilistic encryption with the following property:

$$c_1 * c_2 = E_r(m_1 + m_2)$$

where $c_1 = E_{r_1}$ and $c_2 = E_{r_2}$

Property verified by ElGamal.

Efficient proofs of validity

Consider Elgamal encryption $(x, y) = (g^\alpha, h^\alpha m)$ with $m \in \{m_0, m_1\}$ and the prover knows the value of m .

To show that (x, y) is indeed in this form without revealing the value of m , by given:

$$\log_g x = \log_h(y/m_0) \quad \text{or} \quad \log_g x = \log_h(y/m_1)$$

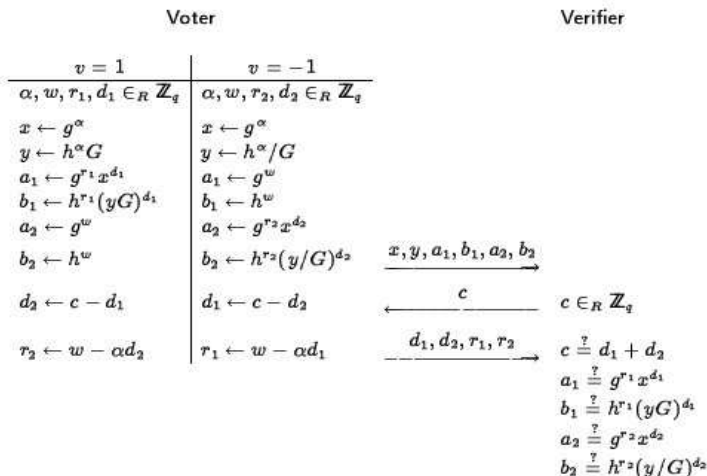
Prover either knows a witness for left part or right part depending of the choice of m .

Multi-authority election Scheme

- 1 V_i voters post ballots (x_i, y_i) to the bulletin board, with a proof of validity.
- 2 Proof of validity are checked and authorities product $(X, Y) = (\prod_{i=1}^l x_i, \prod_{i=1}^l y_i)$
- 3 Finally authorities execute decryption to get $W = Y/X^s = G^T$

if yes is G and no is G^{-1} then T represents the difference between yes and no.

Multi-authority election Scheme



“Proofs” of properties

- Eligibility: due to the bulletin board and PKZ
- Universal Verifiability: proofs are checkable
- Privacy: due to ElGamal encryption.
- Robustness: against t malicious authorities
- No vote duplication
- **Attack** for receipt-freeness, if voter gives the used secret key.

Outline

- 1 Secret Sharing
- 2 Interactive Zero Knowledge Proofs
 - Principle
 - Funny example: Rubik's Cube
 - Example: Graph Coloring
 - Fiat-Shamir
- 3 Blind Signature
- 4 E-Voting
- 5 E-Cash**
- 6 Side Channel
- 7 Conclusion

Notions

Payment ONLINE: Need confirmation of the bank

Payment OFFLINE: Confirmation not needed

Warnings

- Bank code
- integrity
- non-repudiation
- authentication
- privacy

Historic

- Credit cards over SSL (Paypal mostly used in US)
- e-cheques (Netcash 1993)
- Virtual credit cards (First Virtual 1994)
- Encrypted credit cards (Cybercash 1994)
- Mondex/SET -; Chip-SET (C-SET)
- EMV: Europay, Mastercard, Visa 1994
- Many, many others ... (Digicash 1994, Geldkarte)

SET or CSET

Used for Visa and MAsterCard 1997 with 3 principals:

- Customer (C)
- Merchandizer (M)
- Bank (B)

Properties Aimed

- Mutual authentication between M and C
- Authentication of B by C and M
- Secrecy of the command between C and B
- Secrecy of modalities of the payment between C and M
- Non repudiation of the transaction of the 3 principals
- Freshness of the transaction

Anonymity: B should not know that C bought something to M

Fairness: C should sign before M

SET: Secure Electronic Transaction

Uses RSA, DES and SHA1.

Card number known by C and B

Number and price of the command known by C, M and B

Command known by C and M

Notations

- N_M and N_C nonces
- Od = Oder details
- Pd = Payment details
- $Trans = C, M, (N_C, N_M), Price, hash(Od), hash(Pd)$

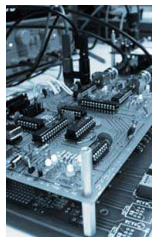
SET

 $C \rightarrow B : C, M$ $M \rightarrow C : N_M$ $C \rightarrow M : \{Trans\}_{K_C^{-1}}, \{Od\}_{K_M}, \{Pd\}_{K_B}$ $M \rightarrow B : \{Trans\}_{K_C^{-1}}, \{Trans\}_{K_M^{-1}}, \{Pd\}_{K_B}$ $B \rightarrow M : \{Results, hash(Trans)\}_{K_B^{-1}}$ $M \rightarrow C : \{Results, hash(Trans)\}_{K_B^{-1}}$

Outline

- 1 Secret Sharing
- 2 Interactive Zero Knowledge Proofs
 - Principle
 - Funny example: Rubik's Cube
 - Example: Graph Coloring
 - Fiat-Shamir
- 3 Blind Signature
- 4 E-Voting
- 5 E-Cash
- 6 Side Channel**
- 7 Conclusion

Different Kind of Side Channel



How to determine a secret or a key by observing:

- Time : it is linked to the secret
- Power Analysis Attack: measure the power used by the cryptosystem
- SPA (Simple), DPA (differential)
- Cache Attack: analysing the cache default can leak information
- FaultAttack: attack by injecting some faults
- Electromagnetic attack ...

First paper

Timing Attacks on Implementations of Diffie–Hellman, RSA, DSS,
and Other System... Paul Kocher - CRYPTO - 1996

Naive Example Side Channel

- Access Control with 10 digit (0..9)
- Code composed of 4 digits
- At each mistake a red light is turn on, otherwise it is the green one

Naive Example Side Channel

- Access Control with 10 digit (0..9)
- Code composed of 4 digits
- At each mistake a red light is turn on, otherwise it is the green one

With at most 40 tries we can deduce the secret code.

Timing attack on Pin Code

For an 8 bytes pin code, we have $(2^8)^8 = 256^8$ possibilities for Brute Force attack.

Timing attack on Pin Code

For an 8 bytes pin code, we have $(2^8)^8 = 256^8$ possibilities for Brute Force attack.

Program

```
for ( i = 0 ; i <= 7; i++)  
    if ( pinCarte[i] != pinPresente[i] ) return false;  
return true ;
```

- Present $n : 0, \dots, 256$ for the first byte $(n, 0, 0, 0, 0, 0, 0, 0)$
- Measure the execution time, the maximum give the first part of the key.
- Repeat it

We have only $8 * 256 = 2048$ possibilities.

Timing attack on Pin Code: Correction

Program

```
boolean test = true ;  
for ( i = 0 ; i <= 7; i++)  
    test = test && ( pinCarte[i] == pinPresente[i]);  
return test ;
```

Acoustic cryptanalysis I

In his book *Spycatcher*, former MI5 operative Peter Wright discusses use of an acoustic attack against Egyptian Hagelin cipher machines in 1956. The attack was codenamed "ENGULF".



HAGELIN M-209 CIPHER MACHINE (GVG / PD)



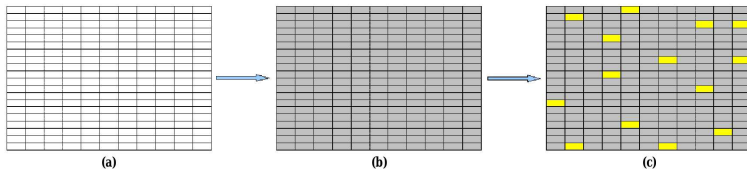
Acoustic cryptanalysis II

In 2004, Dmitri Asonov and Rakesh Agrawal of the IBM Almaden Research Center announced that computer keyboards and keypads are vulnerable to attacks based on differentiating the sound produced by different keys.



Cache Miss Attack Principle

For a text p and a key k , we will put data in the cache, encrypt p with k and look if our data are still in the cache or not.



1st round of Optimized AES

The system (E) shows us the equations of the first round :

$$(E) = \begin{cases} (x_0^{(1)}, x_1^{(1)}, x_2^{(1)}, x_3^{(1)}) \leftarrow T_0[x_0^{(0)}] \oplus T_1[x_5^{(0)}] \oplus T_2[x_{10}^{(0)}] \oplus T_3[x_{15}^{(0)}] \oplus K_0^{(1)}, \\ (x_4^{(1)}, x_5^{(1)}, x_6^{(1)}, x_7^{(1)}) \leftarrow T_0[x_4^{(0)}] \oplus T_1[x_9^{(0)}] \oplus T_2[x_{14}^{(0)}] \oplus T_3[x_3^{(0)}] \oplus K_1^{(1)}, \\ (x_8^{(1)}, x_9^{(1)}, x_{10}^{(1)}, x_{11}^{(1)}) \leftarrow T_0[x_8^{(0)}] \oplus T_1[x_{13}^{(0)}] \oplus T_2[x_2^{(0)}] \oplus T_3[x_7^{(0)}] \oplus K_2^{(1)}, \\ (x_{12}^{(1)}, x_{13}^{(1)}, x_{14}^{(1)}, x_{15}^{(1)}) \leftarrow T_0[x_{12}^{(0)}] \oplus T_1[x_1^{(0)}] \oplus T_2[x_6^{(0)}] \oplus T_3[x_{11}^{(0)}] \oplus K_3^{(1)}. \end{cases}$$

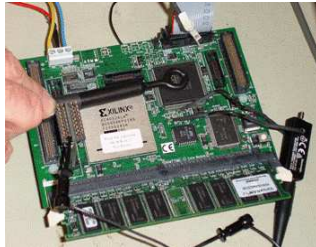
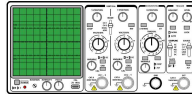
The initial intermediate state $x^{(0)}$ is calculated according to the key k and the plain text p :

$$\forall i \in [0, 15], x_i^{(0)} = p_i \oplus k_i$$

Countermeasures

- Avoiding or deactivating the cache.
- hardware implementations
- AES standard algorithm:
You can modify the encryption software to make it use the AES standard algorithm which does not look up the tables. Thus our attack does not work. However the standard algorithm is less powerful than the optimized algorithm, which we are attacking, on computers 32 bits or much.
- Playing with the look up tables:
You can play with the look up tables. For that you can put randomly the tables in memory, put each tables more than one time in memory or you can cache all the 4 tables before (or after) each encryption. By that our attack can't work any more but the encryption process is slower.

Setup for Power Analysis Attack

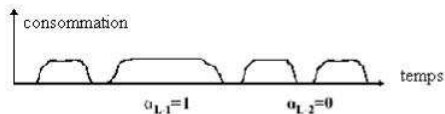
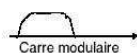
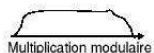


Simple Power Attack on RSA Signature

Signature si $y^a \bmod n$, where y is the message, n public and is the secret key.

Program

```
s = 1 ;  
for ( i = L-1 ; i >= 0; i --) {  
    s = s*s mod n ;  
    if ( a [ i ] == 1)  
        s = s*y mod n ;  
}
```



Outline

- 1 Secret Sharing
- 2 Interactive Zero Knowledge Proofs
 - Principle
 - Funny example: Rubik's Cube
 - Example: Graph Coloring
 - Fiat-Shamir
- 3 Blind Signature
- 4 E-Voting
- 5 E-Cash
- 6 Side Channel
- 7 Conclusion**

Today

- Secret Sharing
- Zero-knowledge
- Blind Signature
- E-Voting
- E-Cash
- Side Channels Attacks