A matinee of cryptographic topics

3 and 4 November 2014

Questions

- How can you prove yourself?
- How can you shuffle a deck of cards in public?
- Is it possible to generate a ElGamal key pair such that nobody has any information about the public key, but still be able to find it without solving DLOG?
- How can Danish sugar beet farmers and buyers place bids in an auction and decide the answer but without revealing any info?
- Is it possible to flip a coin over the electric telephone?
- How many citations can you get from a two-page paper?

Oversikt

Recap: Schnorr

Public input: $G = (g), |G| = q$ and h.
Private input to $\mathcal{P}$: a such that $h = g^a$.

Prover		Verifier
$u \xleftarrow{r} \mathbb{Z}_q$		
$\alpha \leftarrow g^u$	$\alpha \longrightarrow$	
	$\longleftarrow e$	$e \xleftarrow{r} \mathbb{Z}_q$
$r \leftarrow ae + u$	$r \longrightarrow$	$g^r \stackrel{?}{=} \alpha h^e$

Question

Can we use this for identification? Can we do it in a single round?

Fiat-Shamir heuristic

Enter the random oracle model. Then any zero knowledge proof where the verifier only asks random questions can be turned into a non-interactive proof.

Technique: Whenever the verifier should send a random value

- In theory: Ask the oracle for a value based on all previous data in the protocol.
- In practice: Use a hash function on all previous data

Fiat-Shamir(Schnorr)

Let H be a cryptographic hash function.

1. $u \xleftarrow{r} \mathbb{Z}_q$
2. $\alpha \leftarrow g^u$
3. $e \leftarrow H(\alpha)$
4. $r \leftarrow ae + u$
5. Output (α, r)

Verification: Check $g^r = \alpha h^{H(\alpha)}$

Idea

If H does just as good a job of selection something random, then it could just as well have been chosen by the verifier. Hence, the prover must know a such that $g^a = h$.

Key difference: The transcript cannot be simulated, hence it can be verified at any time.

Let's get sidetracked for a moment ...

Prover		Verifier
$\gamma \xleftarrow{r} \mathbb{Z}_q$	$\Gamma \leftarrow g^{\gamma}$	
$u_i \xleftarrow{r} \mathbb{Z}_q$	$U_i \leftarrow g^{u_i}$	
$w_i \xleftarrow{r} \mathbb{Z}_q$	$W_i \leftarrow \Gamma^{w_i}$	
$a_i \xleftarrow{r} \mathbb{Z}_q$	$A_i \leftarrow g^{a_i}$	
	$C_i \leftarrow \Gamma^{a_{\pi(i)}}$	
$\Lambda_1 \leftarrow \prod_{i=1}^{k} x_i^{u_i}$		
$\Lambda_2 \leftarrow \prod_{i=1}^{k} z_i^{-w_i} y_i^{u_i}$		
	$\xrightarrow{\Lambda_1, \Lambda_2, \{A_i\}, \{C_i\}, \{U_i\}, \{W_i\}, \Gamma}$	$\rho_i \xleftarrow{r} \{0,1\}^{\ell}$
		$B_i \leftarrow g^{\rho_i} U_i^{-1}$
	$\xleftarrow{\{\rho_i\}}$	
$b_i \leftarrow \rho_i - u_i$		
$\sigma_i \leftarrow w_i + b_{\pi(i)}$		
$D_i \leftarrow \Gamma^{b_{\pi(i)}}$		
	$\xrightarrow{\{D_i\}}$	$\lambda \xleftarrow{r} \{0,1\}^{\ell}$
		$R_i \leftarrow A_i B_i^{\lambda}$
	$\xleftarrow{\lambda}$	$S_i \leftarrow C_i D_i^{\lambda}$
$r_i \leftarrow a_i + \lambda b_i$		
$s_i \leftarrow a_{\pi(i)} + \lambda b_{\pi(i)}$		
	$\xrightarrow{\{\sigma_i\}}$	$\Phi_1 \leftarrow \prod_{i=1}^{k} x_i^{-\rho_i}$
		$\Phi_2 \leftarrow \prod_{i=1}^{k} z_i^{\sigma_i} y_i^{-\rho_i}$
		$\Gamma^{\sigma_i} \stackrel{?}{=} W_i D_i$
	$\mathsf{CP}(\log_g h = \log_{\Phi_1 \Lambda_1} \Phi_2 \Lambda_2)$	
	$\xleftarrow{\mathsf{SS}(S_i = R_{\pi(i)}^{\gamma})}$	

M. Blum, 1986

Theorem

Any theorem can be proven in zero knowledge.

`http://www.mathunion.org/ICM/ICM1986.2/Main/icm1986.2.1444.1451.ocr.pdf`

How to prove yourself

Authentication scheme Alice can prove to Bob that she is Alice, but someone else (Eve) cannot prove to Bob that she is Alice.

Identification scheme Alice can prove to Bob that she is Alice, but Bob cannot prove to someone else that he is Alice.

Signature scheme Alice can prove to Bob that she is Alice, but Bob cannot prove even to himself that he is Alice.

(Further reading: Fiat, Shamir: How to prove yourself: Practical solutions to identification and signature problems)

Section 4

Commitments

Who gets the car?

Alice chooses large primes p and q, both congruent to 3 modulo 4.
b is chosen at random from $\{0, 1\}$

Alice		Bob
$p, q, n = pq$		n, x
		$y \leftarrow x^2 \pmod{n}$
$a_0^2 \equiv a_1^2 \equiv y$	$\xleftarrow{\quad y \quad}$	
	$\xrightarrow{\quad a_b \quad}$	
	$\xleftarrow{\quad (p, q) \text{ or } \perp \quad}$	

If $a_b \equiv \pm x$, Alice wins. If $a_b \not\equiv \pm x$, Bob wins.
Idea: $\gcd(x - a_b, n)$ will compute a nontrivial factor of n.

More elegant solution

Principles

- Bob must make a choice, and commit to it
- Alice flips the coin, announcing the result without knowing Bob's choice.
- Bob reveals his choice

Examples

We want to commit to x.

g^x Computationally hiding, unconditionally binding

$g^x h^r$ Unconditionally hiding, computationally binding (Pedersen)

(g^{x+r}, h^r) Computationally hiding, unconditionally binding

Question

Can a scheme be both unconditionally hiding and binding?

Section 5

Secret sharing

http://cs.jhu.edu/~sdoshi/crypto/papers/shamirturing.pdf